D1.1 State of the art

Deliverable ID:	**D1.1**
Project acronym:	**ASTAIR**
Grant:	**101114684**
Call:	**HORIZON-SESAR-2022-DES-ER-01**
Topic:	**HORIZON-SESAR-2022-DES-ER-01-WA1-1**
Consortium coordinator:	**Ecole Nationale de L'Aviation Civile (ENAC)**
Edition date:	**11 March 2024**
Edition:	**01.01**
Status:	**Official**
Classification:	**PU**

Abstract

This deliverable discusses the state-of-the-art related to interaction between human actors and automated systems for the management of ground operations in potentially highly automated airports. Furthermore, in this deliverable, existing support algorithms and tools for fleet management and path planning will be reviewed, which could be used to enable collaboration between human actors and support automation. The literature review serves as a starting point for the activities in WP2 (Support algorithms) and WP3 (Automation Supervision & Control HMI design and development).

Authoring & approval

Author(s) of the document

Organisation name	Date

TUD	26/2/2024
ENAC	6/12/2023
ADP	21/2/2024

Reviewed by

Organisation name	Date
ADP	23/2/2024
ENAC	26/2/2024

Approved for submission to the SESAR 3 JU by[1]

Organisation name	Date
TUD	26/2/2024
ADP	26/2/2024
ENAC	26/2/2024
ECTL	26/2/2024
DBL	26/2/2024

Rejected by[2]

Organisation name	Date

Document history

Edition	Date	Status	Company Author	Justification
00.01	1/10/2023	Initial version	Sharpanskykh / TUD	The table of content and the overall setup
00.05	10/01/2023	Intermediate version	Sharpanskykh / TUD	Draft of the deliverable is written and provided for internal review
01.00	26/02/2024	Release	Sharpanskykh / TUD	The final version with all comments of the project participants addressed
01.01	11/03	Release	ENAC	Minor corrections

[1] Representatives of all the beneficiaries involved in the project

[2] Representatives of the beneficiaries involved in the project

ASTAIR

AUTO-STEER TAXI AT AIRPORT

ASTAIR

This document is part of a project that has received funding from the SESAR 3 Joint Undertaking under grant agreement No 101114684 under European Union's Horizon Europe research and innovation programme.

Table of contents

List of figures

List of tables

Executive summary

The goal of the ASTAIR project is to design a seamless partnership between Human and Artificial Intelligence (AI) to manage and perform engine-off and conventional airport surface movement operations at major European airports. ASTAIR original approach to automation is to consider an integrated airport system instead of many separate sub-systems, analyse the level of autonomy an AI system could take on tasks and to make the automation controllable by humans at different levels.

With the introduction of high-level automation for airport surface movement operations, the role of operators and airport operation procedures will significantly change. The key to optimize the overall performance of the collaboration between humans and AI is to adapt intelligent systems to the operators' modus operandi. This will ensure logical consistency across manual and automated control and reduce the cognitive distance between levels of automation by mapping system functions to goals and mental model of operators. In ASTAIR, we will propose interactive tools and adaptative AI algorithms that take advantage of operators' expertise for controlling and engaging with the automation at diverse levels.

As a first step towards developing the ASTAIR solutions, this report identifies the state-of-the-art on airport ground operations, Human-AI Interaction, and path and motion planning and fleet management algorithms.

Related work on airport ground operations sets the operational context and identifies important procedural elements of conventional and engine-off taxiing, which will be elaborated further in the project's concept of operation and use cases.

Related work focusing on Human-AI Interaction has been surveyed. The EASA Artificial Intelligence Roadmap 2.0 Report is first described to define levels of automations and identified research gaps for such levels of automation. We then describe related work covering Human-Automation Teaming (HAT), studies of human behaviors when using automated systems, mixed-initiative interaction and AI-Explainability (XAI) that are topics related to the design and evaluation of highly automated interactive systems. From this state of the art, methods and design guidelines that should be applicable during the project are identified, to understand users' needs and to design new technologies including AI.

Fleet management is crucial for airports to prevent congestion. We have reviewed the literature on fleet management for aircraft, tug fleet and ground support equipment. The fleet management for aircraft mainly focuses on assignment of aircraft fleet to flight legs. Tug fleet management is related to allocation of tugs or Taxibots to aircraft so that taxiing operations can be handled more fluently. This involves not only the allocation of tugs or Taxibots but also the planning their conflict-free paths.

In ASTAIR, the path and motion planning algorithms are intended to be used for controlling the movement of tugs or taxi-bots and aircraft on airport surface layouts. To this end, the literature survey is presented in this document related to existing path and motion planning algorithms, including their comparison with respect to requirements relevant for ASTAIR.

Furthermore, we reviewed literature on methods combining path and motion planning with target or task assignment, which usually have better computational properties in comparison with when these problems are considered separately.

Based on this state-of-the-art, several research directions have been identified to be explored during the ASTAIR project that integrate the different aspects discussed in this document.

Regarding the Human-AI Interaction in ASTAIR, we will be targeting high levels of automation (Levels 2B and 3A according to the EASA's classification). Previous work on how to design efficient interactions for such high levels of automation is scarce and often studied within very narrow and controlled settings. To address this challenge, the following research directions will be considered in ASTAIR:

- Clearly defining the roles and tasks allocation between AI and Humans.
- Identifying shared goals, constraints and representations to enable an efficient partnership between humans and AI with adequate situation awareness and control.
- Investigating interactions that enable fluid transition from different levels of automation according to user preferences or AI performances.

Based on the comparison of solvers for path and motion planning, it is concluded that compared to optimal solvers, bounded sub-optimal solvers perform better in terms of computational time, while slightly decreasing the solution quality. On the other hand, unbounded suboptimal solvers generate solutions much faster than optimal and sub-optimal solvers, however the completeness of the obtained solutions are not always guaranteed even though a feasible solution exists. Thus, the trade-offs between solution quality, completeness and computational complexity should be considered while selecting the best solver.

Co-funded by the European Union

1 Introduction

1.1 Purpose of this document

This document provides background information and state-of-the-art on solution approaches related to the electrification and automation of airport surface movement operations, human-machine interactions in automated systems, management of the vehicle fleet, path and motion planning for automated vehicles, and explainable AI solutions to improve the understanding of the human-in-the-loop.

Another goal is to determine the research directions based on comprehensive evaluation and comparison of existing approaches and the requirements of the cases that will be dealt with in ASTAIR.

In particular, we want this document to feed the design of novel interactions for supervising and controlling a highly automated airport that will be carried out in WP3. Furthermore, based on a comprehensive review of existing path and motion planning and task assignment approaches, and a set of ASTAIR-related requirements, we identified candidate solution techniques to be further elaborated in WP2.

The ASTAIR solution is aimed to be capable of controlling the movements of airplanes and tugs (Taxibots and conventional) on a platform between parking stands and runway entries, between parking aprons, and managing the fleet of available tractors on service roads. To achieve this, AI solutions including the AEON based multi-agent routing solution that compute 4D trajectories with speed profiles for aircraft and tugs, fleet management solution that assigns Taxibots to aircraft taking the contract-based constraints and availability of Taxibots into consideration, and relevant interactive tools for operators. Regarding these concepts, this document presents the state-of-the-art solutions including fleet management and path and motion planning for airports and other environments, human-machine interactions, and explainable AI.

1.2 Scope

In this document, we review 3 interrelated domains relevant to the ASTAIR project, which are mainly the (i) airport surface operations regarding conventional and engine off taxiing, (ii) Human-AI interactions, (iii) fleet assignment and path and motion algorithms for airport surface movement operations

A review of airport ground movement operations defines the main context that is considered within the project including conventional and engine off taxiing techniques and constraints. Some of the described aspects of airport surface movement operations will be further elaborated in the ASTAIR concept of operation and use cases considered in the project.

The review of the state-of-the-art regarding Human-AI interactions includes Human-Automation Teaming, controlling automated systems, mixed-initiatives with high automation levels, and understanding AI behaviors. Introducing higher levels of automation in airport operation will likely introduce changes for the involved persons with redefined roles and tasks among the stakeholders with some possibly transferred to AI. However, AI alone will not be able to handle the complex and constantly evolving situation involved in airport operations. Humans will need not only to supervise AI systems but also likely to work with AI systems to optimize their performances or to cope with failures. This results in the need to study related work covering Human-Automation Teaming (HAT), studies of human behaviors when using automated systems, mixed-initiative interaction and AI-Explainability (XAI) that topics related to the design and evaluation of highly automated interactive systems. From this state of the art, methods and design guidelines that should be applicable during the project are identified, to understand users' needs and to design new technologies including AI.

An overview of existing research on fleet management for the assignment of aircraft fleet to flight legs, allocation of tug fleet to aircraft, and assignment and routing of ground handling vehicles to complete ground handling tasks are presented. To reduce congestion at airports flight scheduling and aircraft fleet management are critical and a considerable amount of previous research is relevant to this area. To reduce emissions and improve the capacity usage, electrification and automation of taxiing and ground handling operations have also gained importance in recent years. Thus, the recent literature includes the studies focusing on tug fleet management for electric taxiing and the fleet management regarding ground support equipment.

Algorithms for path and motion planning are explained focusing on the solution approaches applied for airport surface movement on airport surface area and the state-of-the-art path and motion planning algorithms that are used in various environments. Furthermore, the recent research directions that combine path planning with target assignment are summarized. A comprehensive evaluation and comparison of the state-of-the-art solvers regarding the computational complexity and solution quality is presented. Also, the explainable AI frameworks proposed for path and motion planning are briefly mentioned.

1.3 Structure of the document

The remainder of this deliverable is organized as follows.

In **Section 2**, to provide the context, the concepts of both conventional and engine-off airport surface movement operations are introduced.

In **Section 3**, related work focusing on Human-AI interaction is described. Concepts and directions from the EASA Artificial Intelligence Roadmap 2.0 Report as well as several research fields are presented. Several guidelines and methodologies and research directions to be investigated within the ASTAIR project are identified.

In **Section 4**, the state-of-the-art solution methodologies for fleet management and path and motion planning are presented. This section is split into two subsections. Fleet management is described in **Section 4.1** and the content related to path and motion planning algorithms is presented in **Section 4.2.**

Section 4.1 includes three subsections. Existing research on fleet management for the assignment of aircraft fleet to flight legs is presented in **Section 4.1.1**. The literature on allocation of tug fleet to aircraft is given in **Section 4.1.2**. Recent studies on management of ground support equipment fleet are summarized in **Section 4.1.3**.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

In **Section 4.2**, algorithms for path and motion planning are explained in four subsections. **Section 4.2.1** focus path and motion planning algorithms for airport surface movement, **Section 4.2.2** explains the state-of-the-art path and motion planning algorithms that are used in various environments, **Section 4.2.3** summarizes the recent research directions heading towards the solution of path planning combined with target assignment**, Section 4.2.4** presents a comparison of solvers in terms of complexity and solution quality, and **Section 4.2.5** provides with a brief summary of explainable AI methods for path and motion planning.

Lastly, in **Section 5**, conclusions and research directions are provided.

2 Airport ground movement operations

In this section, both conventional and engine-off airport ground movement operations are considered, which are within the scope of ASTAIR. Furthermore, several SESAR projects relevant to ASTAIR are reviewed.

2.1 Conventional aircraft taxiing

During the landing and take-off (LTO) cycle, on average the aircraft spend most of the time on the ground, as they have to manoeuvre different aerodrome layouts to take-off or land. Conventional departure procedures include pushback (with engines-off) from the parking stand and taxi (with engines-on) till they lift-off from the runway, while the arrivals follow an engine-on schedule till the parking stand (see Figure 1).

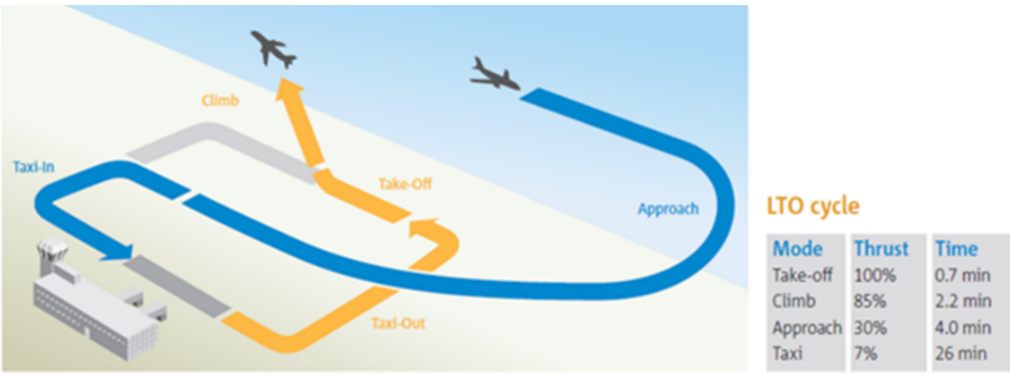

Figure 1: Landing and take-off cycle.

Conventional operating methods on ground involve keeping main engines-on or use single engine technique to taxi aircraft from gate to runway or vice versa as shown in Figure 2.

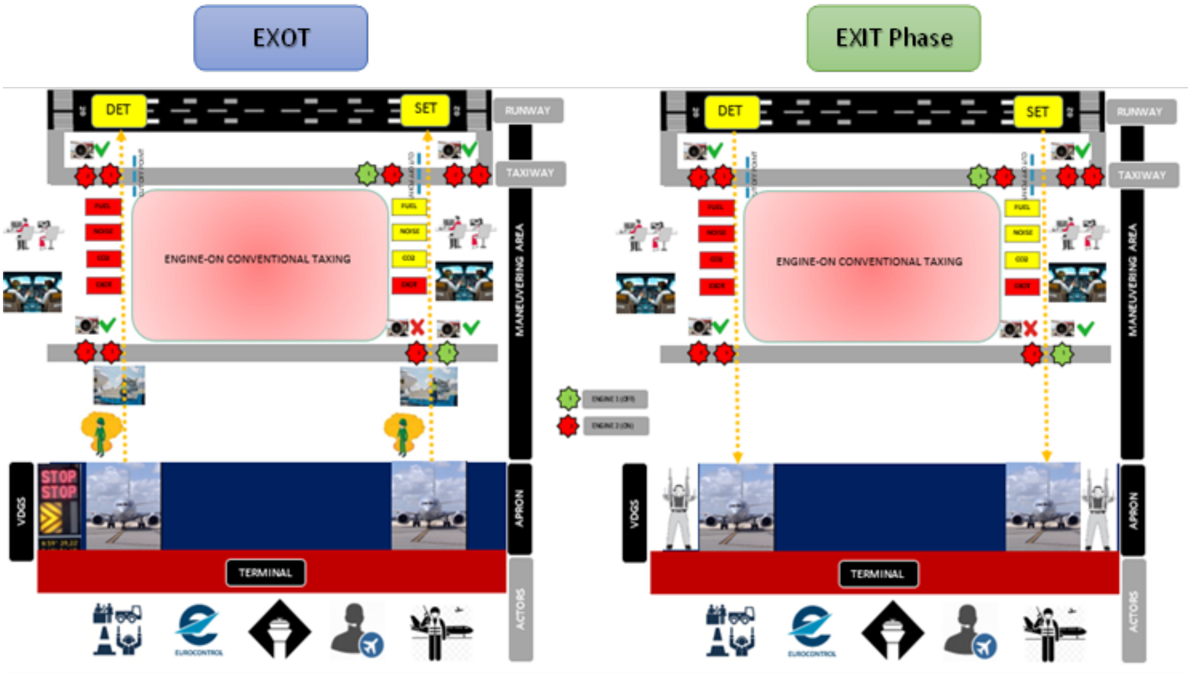

Figure 2: Current operating method – DET and SET

The Dual Engine Taxi (DET) is normally adopted during both Taxi-Out (EXOT) and Taxi-In (EXIT) phases of aircraft ground operations. Single Engine Taxi (SET) method is usually used by airlines during the EXIT phase more than the EXOT phase, to save fuel during longer taxiing times at the airports. It can also be seen that these methods add procedures/personnel on the ground that eventually increase the turnaround time for the AO. Figure 2 also shows that engines are kept on right from the aircraft off-block time till the on-block time, even though there are stop and go situations that arise, like de-icing of an aircraft or airside delays causing aircrafts to hold for longer duration.

In Figure 2, the double red polygon symbols indicate both engines are ON while the red and green polygon symbols indicate one engine is OFF and the other one is ON. The other parameters that are compared for the conventional technique with new AEON solution(s) are fuel saving, noise, CO_2 emission and EXOT or EXIT times. Also, the actors and stakeholders who are involved at various stages of operation, from pushback on apron, taxiing in the manoeuvring area to take-off on the runway, are illustrated.

In the following we review the main ground phases of the flight.

The pushback is the movement of an aircraft from a nose-in parking stand using the power of a specialized ground vehicle attached to or supporting the nose landing gear. It is commonly the second part of a taxi in push out procedure at airport terminal gates and will be necessary to depart from all except self-manoeuvring parking stands, unless the aircraft type is capable of power back and local procedures allow this. Once the Pilot in Command (PIC), has given the confirmation of 'brakes released' to the person in charge of the ground crew who are to carry out the "Pushback", the ground crew becomes temporarily responsible for the safe manoeuvring of the aircraft in accordance with either promulgated standard procedures or as specifically agreed beforehand.

The traditional pushback method involves attaching a ground vehicle to the aircraft nose landing gear by means of a towbar. An alternative method, which is more common for pushback, is the use of a specialized vehicle called a 'towbarless tug'. This tug positions two low level 'arms' either side of the aircraft nose landing gear and these are used to engage with the aircraft gear leg and raise it slightly off the ground. These specialized vehicles can also be used to tow aircraft forward.

Effective communication between the person in charge in the flight deck and the person in charge of the ground crew, and between the members of the ground crew team is critical. If the aircraft is pushed back prior to the intended flight and the person in charge of the flight deck is therefore an aircraft commander, the procedures of the aircraft operator may require that the designated pilot flying, who may be the co-pilot, should oversee the pushback and in this case all communications with the ground crew will be undertaken by that person rather than necessarily by the aircraft commander.

Taxi-out (EXOT) is defined by the time taken by the aircraft to move from a gate or a parking stand to a runway take-off point.

After being cleared by the ground crew and with the warmed-up engines, the pilot in control is able to taxi according to the instructions/clearances received from the ground controller (or ATC) to designated hold points near the proposed take-off runway or de-icing pads or along with the taxiways etc. The choice of speed to drive through taxiways would depend largely on the human factor, airport speed limitations and airline internal policies. It is often seen at many airports that where speed limitations are not set the pilots operate the aircraft at a highest speed in order to achieve the allotted Calculated Take-Off Time (CTOT).

Normally, based on the respective airport operating plan, the routing of the departing aircraft are planned in a way to avoid intersection conflicts, jet blasts or any other safety concerns and aid in quicker and seamless exits.

The Flight Crew and the ATCO/Ground Controller are in constant contact to exchange any real time updates and guidance. Today, most ANSP at airports update real time data through D-ATIS (Datalink Automatic Terminal Information Service) that enhances the safety for the Flight Crew (FC) and reduces interaction time with the ATCOs.

Once taxied to the designated hold point close to the runway, the Flight Crew prepares the aircraft for take-off after taking into consideration the entire pre-departure checklist. On FC's confirmation for aircraft readiness to take-off, necessary communications are exchanged between FC and ATCo, who in turn provide line-up and take-off clearances to the FC. This way a smooth transition for take-off is achieved.

Taxi-in (EXIT) is defined by the time taken by the aircraft to move from the runway touch down point to a parking stand or a gate.

Once the pilot in command touches down the aircraft on the runway and exits to a taxiway, the Flight Crew is instructed by the ATCO to contact the Ground Crew to be guided to the parking stand. Upon entering the taxiway, the pilot in command either operates all engines or taxi using single engine.

Single engine taxiing (SET) procedures are nowadays more frequently adopted by the airlines upon arrivals rather than departures. The Flight Crew maximises the speed while taxiing (both during EXIT and EXOT phase) in order to reduce the round-trip time on ground. The aircraft is guided into the apron

Co-funded by the European Union

stands using Visual Docking Guidance System (VDGS) or hand signals by trained marshallers. After stopping at the designated nose wheel position with chocks on, brake released and upon engines being switched off and set into the cooling mode and once confirmed by the pilot in control, the Ground Staff begins the ground handling operation.

2.2 Engine-off taxiing

Among the solutions for engine-off taxiing developed in the past years, two are particularly useful to be applied in the airport environment, which are currently considered by major airports:

- Non-autonomous taxiing techniques based on Dispatch Towing Vehicle Electric Taxi System (DTVETS) (see Figure 3)
- Autonomous taxiing techniques, referring to Nose & Main Landing Gear Electric Taxi System (NLGETS & MLGETS) and categorised under Landing Gear Electric Taxi System (LGETS)

Both these solutions are reviewed in this section.

Dispatch Towing Vehicle Electric Taxi System (DTVETS)

The DTVETS System is a dispatch towing system that allows aircraft to taxi for departure to the runway end with engines off. It may also be used for arrival aircraft with some procedure change after the aircraft has left the rapid exit track. It was specially designed to tow aircraft safely, efficiently and without causing fatigue damage to the nose landing gear and does not have speed or distance limitations of normal tow trucks.

During The EXOT phase of DTVETS-based taxiing the pushback is performed in the same way as normal operations, however DTVETS is always in line with the aircraft. When the pushback is completed, control is handed over to the pilot. Pilot control of the DTVETS is performed in the same way as normal taxi operations, steering via tiller and nose gear and braking via the aircraft brakes. No thrust needs to be applied, as DTVETS operates like a car with automatic transmission, accelerating when brakes are not applied. The DTVETS tug also functions as an aircraft push-back tug.

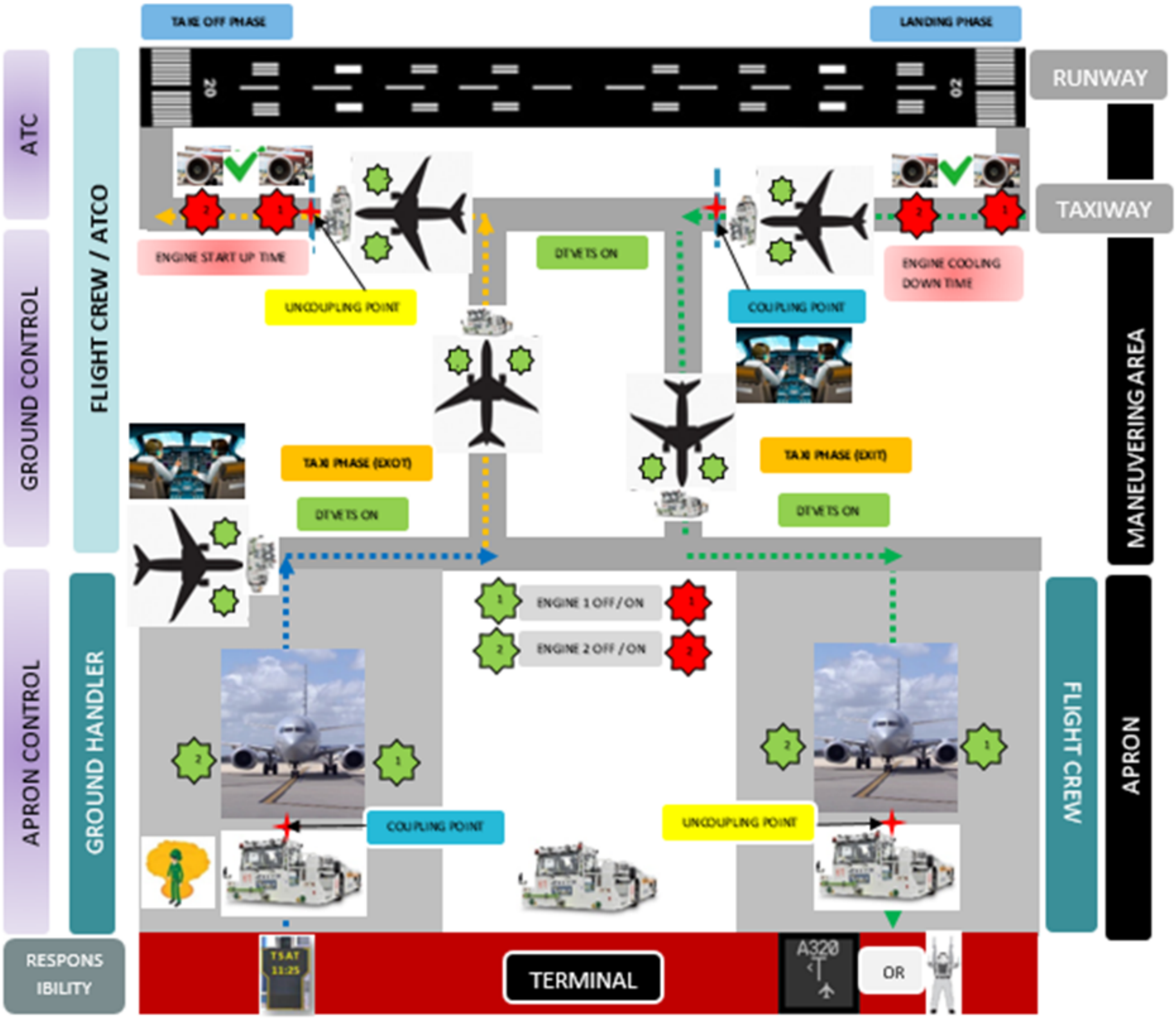

Figure 3: CONOPS diagram flow for DTVETS

Most aircraft require no modifications to use of DTVETS. As illustrated in Figure 3, once the DTVETS is attached at the "coupling point" on the apron and cleared for "delivery" by the Ground Controller, the DTVETS will push the aircraft back from the gate/stand in the same manner as it is done today under the direct responsibility of the trained DTVETS driver. Once pushback is completed, the control would then be transferred/switched to the pilot in command and the flight crew can begin DTV taxi movement (DTVETS ON) till the uncoupling point/area where the aircraft engines can be started while being connected to DTVETS, closer to the assigned runway end. Once decoupled and the control switches to the DTVETS driver, the tug is driven back in the manoeuvring areas of the airport to the next operation.

On arrival, the aircraft can use the DTVETS technology as shown in Figure 3. The assigned DTVETS tug will be stationed at an area close to the runway/taxiway designated point and once the aircraft reaches the designated coupling point, the DTVETS driver would attach the tug to the aircraft nose wheel and transfer or switch control to the flight crew for DTV taxi movement. During this phase of coupling the pilot can decide to switch off the engines and allow them to cool down during the DTVETS coupling

EUROPEAN PARTNERSHIP
Co-funded by the European Union

and control handover process is taking place. Once the DTVETS is attached, the flight crew can steer the aircraft to the gate / parking stand with engines off. Upon reaching the parking stand the pilot can decouple and transfer back the control of the tug to the DTVETS driver, who in turn can position the tug for its next assignment.

The Landing Gear Electric Taxi System (LGETS)

The Landing Gear Electric Taxi System (LGETS) is an on-board innovative in-wheel electric taxi system with electric motors integrated in the nose wheel – termed as Nose landing gear electric taxi system (NLGETS) or in the main landing gear – termed as Main landing gear electric taxi system (MLGETS). It enables pilot-controlled forward and reverse movement in gate and terminal areas without tractors or jet engines. The technology also comes with optional camera/sensor systems that will provide pilots with improved situational awareness for all manoeuvres. The LGETS is designed to reserve the use of the aircraft engines for take-off and flight. It practically eliminates engine usage during ground movement except during engine start-up, warm-up and taxi onto the runway.

With LGETS, the pilot in control is responsible for the pushback. The pilot control of the LGETS is performed in the same way as normal taxi operations, steering via tiller and nose gear and braking via the aircraft brakes till the aircraft reaches the designated cut – off point. As per the airside operational constraints, the pilot in control can decide to start the engines during the taxiing phase of the aircraft or after reaching the designated cut off point.

LGETS-, DTVETS-based, as well as single engine taxi can be used independently or in combination, as was explored in SESAR AEON project.

2.3 Related SESAR projects

Some SESAR project dealing with problematics close to ASTAIR's scope have been identified. Total Airport Management is terminated but ADP, who was coordinating the project, is a partner of ASTAIR consortium. CODA and TRUSTY are funded on the same program as ASTAIR and are on-going, ENAC participates in both of them.

2.3.1 TAM - Total Airport Management

The Total Airport Management project (PJ04 TAM, grant 733121) is interesting for ASTAIR development at several levels. First because centralization and automation of ground movement promoted in ASTAIR follows the same philosophy as PJ04 TAM, but also because PJ04 looked into the usage of AI for routing.

In the context of the PJ04-W2-Solution 29.3 Environmental performance management, the Level 1AI has been experimented to help decision making in order to manage environmental performance.

Two use cases have been considered:

- ENV friendly time slot for conducting runways inspections.
- Fuel-savings taxing routes (studies of taxiing speed to propose alternative taxiing routes with recommended speeds profile for specific flights).

Co-funded by the European Union

The developed models proposed to the operational solutions and decision-makers decided whether the proposed solution will be applied. This corresponds to the level 1B (in reference to EASA level of automation). ASTAIR will go further into looking for conflict-free routing.

2.3.2 CODA – Controller adaptative Digital Assistant

The CODA project aims at developing a system in which hybrid human-machine teams collaboratively perform tasks. To do so, the system put together state of art from different fields: i) Prediction models to foresee future situations and have the system know which activities will be carried out by the operators and their impact on the same human performance; ii) Neurophysiological assessment of mental states to enable the system to know operators’ real current level of workload, attention, stress, fatigue, and vigilance by validating the predicted cognitive models and maximizing the effectiveness of the interaction between the human and the machine by developing an HMPE; iii) AI-based adaptable and explainable systems, to have the system act to prevent future performance or safety issues. Specifically, the project will show how a system could adapt to specific situations and react accordingly by using advanced adaptable and adaptive automation principles that will dynamically guide the allocation of tasks. The system will assess the operator's cognitive status, use current traffic data to foresee the future tasks that the operator will need to perform in the future, and calculate the impact of those tasks in terms of cognitive complexity. With this information, the system will predict the future mental state of the operator and will act accordingly by developing an adaptive automation strategy. For example, imagine an ATCO managing a complex traffic situation and experiencing a medium workload. The system is aware of this (thanks to the neurophysiological assessment). It predicts that the additional upcoming tasks the ATCO will need to take care of will increase their workload, exceeding the maximum an operator can handle. To avoid this, the system decides how to act, following an adaptation strategy: it may, for instance, increment the level of automation, enable additional AI-based tools, or request a sector splitting.

ASTAIR and CODA do not share the same approach on Human Automation Teaming, especially in the use on neurophysiological measures, nevertheless some questions on delegation strategies may be addressed similarly.

2.3.3 TRUSTY – TRUStworthy inTellingent sYstem for remote digital tower

Remote digital towers (RDT) are taking place around the world to ensure efficiency and safety. TRUSTY harnesses the power of artificial intelligence (AI) to enhance resilience, capacity, and efficiency in making significant advancements in the deployment of digital towers. The overall goal of TRUSTY is to provide adaptation in the level of transparency and explanation to enhance the trustworthiness of AI-powered decisions in the context of RDT. Through the video transmission data from RDT, TRUSTY considers the following major tasks:

1. Taxiway monitoring (i.e., bird hazard, presence of a drone, autonomous vehicle monitoring, human intrusion, etc.).
2. Runway monitoring (approach and landing) misalignment warning and the corresponding explanation.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

To deliver trustworthiness in an AI-powered intelligent system several approaches are considered:

- 'Self-explainable and Self-learning' system for critical decision-making
- 'Transparent ML' models incorporating interpretability, fairness, and accountability
- 'Interactive data visualization and HMI dashboard' for smart and efficient decision support
- 'Adaptive level of explanation' regarding the user's cognitive state.
- "Human-centric AI" enhances the trustworthiness of AI-powered systems.
- "Human-AI teaming" to consider users' feedback to insure some computation flexibility and the users' acceptability.

To achieve the goal, TRUSTY will rely on the SotA developments in interactive data visualization, and user-centric explanation and on recent technological improvements in accuracy, robustness, interpretability, fairness, and accountability. We will apply information visualization techniques like visual analytics, data-driven storytelling, and immersive analytics in human-machine interactions (HMI). Thus, this project is at the crossroad of trustworthy AI, multi-model machine learning, active learning, and UX for human and AI model interaction.

TRUSTY and ASTAIR will most probably share some problematics concerning human centric AI and humain AI teaming, thus staying closely in touch will be fruitful for the project.

3 State of the art on Human-AI Interaction

In this section, we describe the state-of-the-art regarding interactions in mixed initiatives automated systems. We start by reviewing the **EASA's artificial Intelligence Roadmap 2.0 report** to clarify levels of automations and specific challenges identified for the ASTAIR project. We then define and discuss **Human-Automation Teaming** research. We then articulate related work relevant for understanding and controlling highly automated systems among several dimensions including **studies of human behaviors when using automated systems, mixed-initiative interaction** and **AI-Explainability** that we will build upon to design shared representations between human and AI to enable efficient collaboration. We conclude with a review of **methodologies** adequate for designing highly automated systems.

3.1 EASA – Artificial Intelligence Roadmap 2.0

The ARTIFICIAL INTELLIGENCE ROADMAP 2.0 report [1] from EASA defines three levels of Automation according to the roles of Humans and AI. Figure 4 describes the roles of Humans and AI for these three levels.

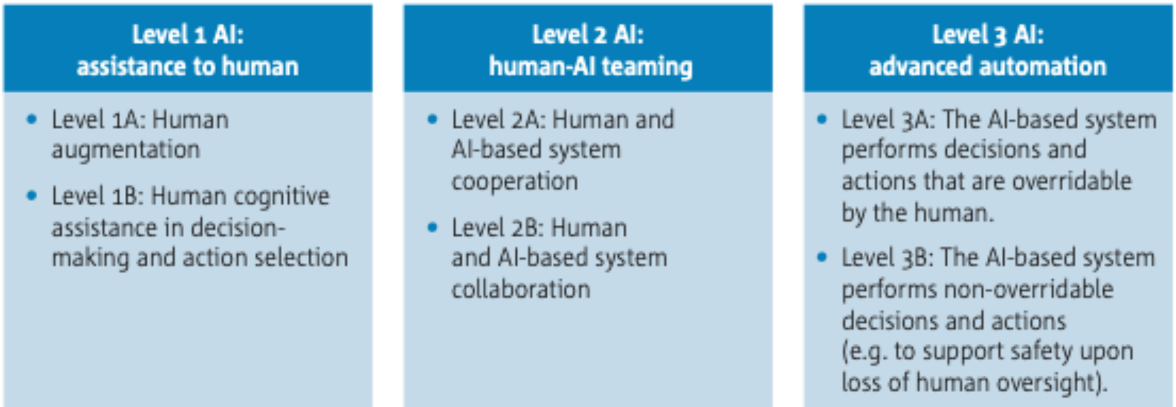

Figure 4: EASA levels of automation defined in [1].

The levels 2A and 2B are different because of the two terms cooperation and collaboration that are defined as follows:

Cooperation *is a process in which the AI-based system works to help the end user accomplish their own objective and goal. The AI-based system will work according to a predefined task allocation pattern with informative feedback on the decision and/or action implementation. Cooperation does not imply a shared vision between the end user and the AI-based system. Communication is not a paramount capability for cooperation.*

Collaboration *is a process in which the human and the AI-based system work together and jointly to achieve a common goal (or work individually on a defined goal) and solve a problem through co-constructive approach. Collaboration implies the capability to share situational awareness and to readjust strategies and task allocation in real time. Communication is paramount to share valuable information needed to achieve the goal, to share ideas and expectations.*

For the first two levels (1 and 2), EASA also proposed several guidelines regarding human-factors among several dimensions including AI operational explainability, Human-AI teaming, Modality of interaction and style of interface, Error management, Workload management, Failure management and alerting systems, Integration or Customization of Human-AI interface [2].

The report also specifies that: *"For Level 1A, existing guidelines and requirements for interface design should be used. For Level 1B, an initial set of design principles are proposed for the concept of operational explainability. For Level 2A and Level 2B, new objectives have been developed and others from existing human factors certification requirements and associated guidance have been adapted to account for the specific end-user needs linked to the introduction of AI-based systems."*

The report from EASA also provides an impact assessment of different levels of automations on end-users regarding Human-AI Interaction, Explainability and Guidance. Figure 5 and Figure 6 summarize the impact assessment.

	OVERALL IMPACT ASSESSMENT	**HAII** Expected level of evolution in the human-AI interaction (HAII) compared to existing interactions	**EXPLAINABILITY** Expected level of explainability needed during operation	**GUIDANCE** Need for specific human factors certification guidance linked with the introduction of AI-based systems
Level 1A Human augmentation	The implementation of an AI-based system is not expected to have an impact on the current operation of the end user. e.g. Enhanced visual traffic detection/indication system in flight-deck. e.g. The analysis of aircraft climb profiles by an AI-enhanced conflict probe when checking the intermediate levels of an aircraft climb instruction.	No change compared to existing systems.	No change compared to existing systems as the implementation of an AI-based system at Level 1A is impacting neither the operation, nor the interaction that the end user has with the systems.	No need for dedicated guidance. Existing guidelines and requirements for interface design should be used. e.g. CS/AMC 25.1302
Level 1B Human assistance	The implementation of an AI-based system is expected to impact the current operation of the end user with the introduction of, for example, a cognitive assistant. e.g. Cognitive assistant that provides the optimised diversion option or optimised route selection. e.g. An enhanced final approach sequence within an AMAN	Medium change: There is a need for explainability so that the end user is in a position to use the AI outcomes to take decisions/actions.	Explainability is there to support and facilitate end-user decisions. At this level, decision still requires human judgement or some agreement on the solution method.	Specific guidance needed. Need for operationalising the explainability concept in the frame of future design and certification. → Definition of attributes of explainability with design principles.
Level 2A Human-AI teaming: Cooperation	Level 2A corresponds to the implementation of an AI-based system capable of teaming with an end user. The operation is expected to change by moving from human-human teams to human-AI-based system teams . More specifically, Level 2A is introducing the notion of cooperation as a process in which the AI-based system works to help the end user accomplish their own objective and goal. The operation evolves by taking into account the work from the AI-based system based on a predefined task allocation pattern. e.g. AI advanced assistant supporting landing phases (automatic approach configuration) e.g. conflict detection and resolution in ATM.	Medium change: Communication is not a paramount capability for cooperation. However, informative feedback on the decision and/or action implementation taken by the AI-based system is expected. HAII evolution is foreseen to account for the introduction of the cooperation process.	With the expected introduction of new ways of working with an AI-based system, the end user will require explanations in order to cooperate to help the end user accomplish their own goal. A trade-off is expected at design level between the operational needs, the level of detail given in an explanation and the end-user cognitive cost to process the information received.	Specific guidance needed Existing human factors certification requirement and associated guidance will have to be adapted for the specific needs linked with the introduction of AI. → Development of future design criteria for novel modality of interaction and style of interface as well as criteria for HAT, and criteria to define roles and tasks allocation at design level.

Figure 5: First part of the table from the EASA report [2] presenting the anticipated human factors guidance modulation.

Level 2B HAT; Collaboration	Level 2B corresponds to the implementation of an AI-based system capable of collaboration. On top of the evolution linked to the notion of HAT, the collaboration will make the operation evolve towards a more flexible approach where the human and the AI-based system will both communicate and share strategies/ideas to achieve a common goal. e.g.: Virtual co-pilot in single-pilot operations	High change: Existing human factors certification requirements and associated guidance are adapted to the specific needs linked with the introduction of AI. → Development of design criteria for novel modality of interaction and style of interface as well as criteria for HAT, and criteria to define roles and tasks allocation at design level.	With the expected introduction of new ways of working with an AI-based system, the end user will require explanations in order to collaborate, negotiate or argument towards a common goals. A trade-off is expected at design level between the operational needs, the level of detail given in an explanation and the end-user cognitive cost to process the information received.	Specific guidance needed Existing human factors certification requirements and associated guidance will have to be adapted to the specific needs linked with the introduction of AI. → Development of future design criteria for novel modality of interaction and style of interface, criteria for HAT, and criteria to define roles and tasks allocation at design level.
Level 3A More autonomous AI	The AI-based system is operating independently with the possibility from the end user to override an action/decision only when needed. No permanent oversight from the end user. A significant modification in the current operation is expected. e.g. UAS ground end user managing several aircraft	Very high change: Expected change in the job design with evolution in HAII to support the end user being in a position to override the decision and action of the AI-based system when needed.	In order for the end user to override the AI/ML systems' decision, the appropriate level of explanation or information is going to be needed for the good operation of the system.	Specific guidance needed. On top of the specific guidance needed for Level 2, EASA anticipates additional guidance development.
Level 3B Fully autonomous AI	There is no more end user. The AI-based system is fully autonomous. e.g. Fully autonomous flights e.g. Fully autonomous sector control.	N/A: The end user is effectively removed from the process. There is no requirement for end-user interaction.	There is no need for explainability at the level of the end user. There is no end user.	N/A in operation.

Figure 6 : Second part of the table from the EASA report [2] presenting the anticipated human factors guidance modulation.

The impact of levels 2B and 3A are important [2]. For level 2B, the impact on Human AI Interaction will require the *"Development of design criteria for novel modality of interaction and style of interface as well as criteria for Human-Automation Teaming, and criteria to define roles and tasks allocation at design level."* For level 3A there are *"expected change in the job design with evolution in HAII to support the end user being in a position to override the decision and action of the AI-based system when needed."*

In summary, based on the data from the EASA report [2] we identified several research directions for the ASTAIR project. We will focus mostly on 2B and 3A automation levels according to this classification since the need for additional work is explicitly identified. For such levels of automation, we will need to investigate the roles and tasks allocation between AI and Humans as well as to identify relevant criteria to validate such allocation. We will also need to formulate guidelines that will serve not only as themes or concerns for designing for high level of automation (2B-3A) but also as practical tools that can be leveraged by designers of such systems. In particular, providing guidelines and recommendations on how to design interactions for overriding AI decisions and how to collaborate effectively with the AI needs to be explored in the project.

3.2 Human-Automation Teaming

Human Automating Teaming (HAT) can be defined as a group of human and autonomous agents, performing activities and achieving outcomes together towards a common goal [3]. In particular in HAT, the autonomous agents work alongside humans performing essential tasks and teamwork functions that a human would [4]. They now perform complex tasks with no or little intervention of humans, which require to engage with other teammates to achieve team objectives [5].

Humans and machines have different capabilities. While autonomous agents are able to manage workload better, human operators can adapt to new situations better [6], thanks to the way human agents communicate to each other. For instance, machines are usually better than humans at solving problems involving a high number of variables but are almost unable to take into account new variables that were unknown during their design [7]. In addition, most of the safety-critical automated systems still rely on humans in a range of non-nominal or critical situations [8]. The unpredictable nature of airport ground traffic and operation makes full automation difficult, hence requiring human knowledge to assess situations and specify relevant strategies to the algorithm.

Artificial intelligence agents are expected to perform in certain ways before they can be considered as teammates. In particular, research has found that humans expect AI teammates to have at least instrumental skills for completing collaborative tasks, shared understanding of human teammates, sophisticated communication abilities for information exchange and human-like performance [9]. Furthermore, humans expect agents to perform like humans while collaborating with them to complete tasks [9]. In other words, humans are more likely to collaborate and coordinate effectively with high-performance AI, which behaves like humans, which can be directed and whose actions can be anticipated.

Although designing human substitutes is unrealistic, novel research can build upon computer-supported cooperative work (CSCW) research to improve interactions with AI by identifying where AI can outperform human performance for relevant roles and tasks allocations and facilitate mutual understanding between AIs and humans. We have extensively covered CSCW fundamentals in AEON deliverable 1.3 and we redirect readers to the document for an understanding and an overview of CSCW [10]. We focus the remainder of the section on the relevant research that will support the design of optimal collaborations between human and AI teammates in ASTAIR.

3.3 Understanding and controlling automated systems

In this section, we describe previous work related to the design of Human-AI interaction. We start by reviewing work from organizational psychology and Human Computer Interaction (HCI) related to the design of interactive systems able to facilitate collaboration and delegation of tasks between human and AI. We also cover interaction styles and existing approaches for mixed initiative systems as well as present recent work in AI explainability related to automated airports.

3.3.1 Delegation aspects from organizational psychology

While there is little agreement on exactly what constitutes an intelligent agent, many definitions embody a user-interface model that differs from the traditional one where users perform tasks with the help of computer-based “tools”. In contrast, the “delegation” model associated with agents is based on entrusting tasks to an autonomous, sometimes anthropomorphized system, whose performance is monitored and evaluated. This change in user-interface model is a dramatic one since delegation can be a difficult and often-avoided behavior in humans. Agent-interface designs need to overcome well-established drawbacks in delegation. For this purpose, designers should find the management sciences and organizational psychology literatures to be as relevant as that of traditional human factors. This section describes issues regarding task delegation between humans as they pertain to the design of intelligent-agent–user interfaces [11].

Nearly all definitions of "agent" contain some combination of the following traits: Ability to work asynchronously and autonomously, Ability to change behavior according to accumulated knowledge, Ability to take initiative, Inferential capability (i.e. capable of abstracting), Prior knowledge of general goals and preferred methods, Natural language, Personality.

"Delegation" is the process of passing on responsibility for a task to a subordinate by giving him/her authority to act on your behalf, but without giving up control, or ultimate accountability [12]. For many reasons, delegation is often an unnatural and taxing activity: Managers often feel that they can perform a task better than a subordinate; Managers often enjoy doing certain tasks even if it may be more efficient to delegate them; For urgent tasks, needing to be done immediately, managers often believe that explaining the task to a delegate will be a waste of time; Managers fear that the subordinate will fail at the task; not only may the task not be accomplished, but the subordinate may feel bad.

For delegating to succeed, the following design guidelines should be followed [11]:

- The benefits of delegation need to exceed the cost. Agents are more appropriate for some tasks than for others. Users must have the option of delegating vs. self-performing tasks.

- Delegation requires sophisticated, interactive communication: Users should be encouraged to convey the intentions and goals of a task to the agent; Natural language interfaces can be used for tasks that are easy to describe.; For many complex task environments, interface dialogues could employ speech—act structure; Agents must be designed to indicate clearly when instructions are not understood; Anthropomorphizing agents with the use of facial displays and vocal intonation may be useful in conveying comprehension and lack of comprehension; For some tasks, it may be most efficient for the user to convey what is desired by demonstration.

- Delegation requires trust: Build agents to be reliable and use them in stable environments; Create specialized agents capable of a small, circumscribed set of capabilities, and emphasize their expertise. Increase the observability of the agent's behaviors; Provide the user with data about the predictability of the agent's behaviors; Design the agent to evolve, with the user, through stages of trust; Train users to understand how the agent works; Use anthropomorphized agent interfaces.

- Performance controls are key part of delegation: Designers need to emphasize sub-tasking, scheduling and deadlines; Users need to be able to solicit and receive status reports at any time; For some tasks, users need an independent way of checking on agents' performance while the task is being carried out; Users need a way of evaluating agents' performance in such a way that the agents' subsequent performance will be improved or strengthened.

- Delegation depends on personality and culture: Experience with delegation may make agent-based interfaces easier to use; Managers who already are effective at delegating tasks to humans may prefer and excel at using intelligent agents, while ineffective delegators may be so with both human and computer-based delegates; Applications for international use need to be designed with careful considerations for cultural differences in leadership and delegation.

While these guidelines focus on human-to-human delegation, we will use these guidelines while designing interaction between human operators and AI within the ASTAIR project.

3.3.2 Studies on human behavior with automation

This section presents the latest results on various studies on human behavior with automation. Even if their application domains are unrelated to air traffic management and the subjects are not always experts of their domain, their results and design implications might prove useful for the ASTAIR project.

3.3.2.1 Delegation with knowledge about AI

When collaborating with artificial intelligence (AI), humans can often delegate tasks to leverage complementary AI competencies. However, humans may delegate inefficiently. Enabling humans with *knowledge about* AI can potentially improve inefficient AI delegation. A between-subjects experiment (two groups, n = 111) has examined how enabling humans with AI knowledge can improve AI delegation in human-AI collaboration [13]. The task consisted of classifying images. One group was informed of the capabilities of AI vs Humans at classifying (e.g. "Humans are superior with images which necessitate social intelligence or highly complex perception" or "AI is superior with images with distinctive patterns or objects"). The findings suggest that AI knowledge-enabled humans align their delegation decisions more closely with their assessment of how suitable a task is for humans or AI (i.e., task appraisal). Delegation decisions closely aligned with task appraisal increase task performance. However, AI knowledge lowers future intentions to use AI, suggesting that AI knowledge is not strictly positive for human-AI collaboration.

The significance of human appraisal for AI delegation decisions indicates that we must consider human attributes in designing AI for human-AI collaboration. As a complement to the AI-attribute- focused principles, a new design principle might state, "Make clear what humans can do." An AI could provide information on average (or even individualized) human performance to promote efficient delegation. On the other hand, AI might also state what humans cannot do. For example, it could make interacting humans aware of their biases ("Create awareness of human biases").

Besides including educational features in AI-based tools, practitioners must also invest in AI training and upskilling programs for humans that promote basic AI literacy, which might be difficult to learn while using an AI-based tool.

The user's intention to continue using AI is arguably one of the most critical metrics for HCI researchers and practitioners when designing AI for human-AI collaboration. A lack of human intention to use AI dooms human-AI collaboration from the outset. Information overload might be a potential reason for lowered AI usage continuance intention (AUCI) AUCI through AI knowledge and is worth exploring. HCI researchers need to better understand how to balance AI knowledge's positive and negative effects, or which specific components of AI knowledge are decisive for the positive and negative effects of AI knowledge.

In summary AI knowledge-enabled humans align their delegation decisions more closely with their assessment of how suitable a task is for humans or AI.

3.3.2.2 Human-machine interaction, environment, and performance factors

In the manufacturing field, a semi-automated system that entails human intervention in the middle of the process is a representative collaborative system that requires active interaction between humans and machines. User behavior induced by the operator's decision-making process greatly impacts system operation and performance in such an environment that requires human-machine collaboration. There has been room for utilizing machine-generated data for a fine-grained understanding of the relationship between the behavior and performance of operators in the industrial domain, while multiple streams of data have been collected from manufacturing machines. A study has been conducted with a large-scale data-analysis methodology that comprises data contextualization and performance modelling to understand the relationship between operator behavior and performance [14]. For a case study, machine-generated data were collected over 6-months periods from a highly automated machine in a large tire manufacturing facility. The authors devised a set of metrics consisting of six human-machine interaction factors and four work environment factors as independent variables, and three performance factors as dependent variables. The modelling results reveal that the performance variations can be explained by the interaction and work environment factors. Even if conducted in a factory, the analyzed system shares some aspects with Air Traffic Control (the use of alarms, human intervention or proactiveness). This research may thus inform us on how to assess the performance of the whole system.

3.3.2.3 Appropriate reliance on AI systems

The promises of AI systems to augment humans in various tasks hinge on whether humans can appropriately rely on them. Recent research has shown that appropriate reliance is the key to achieving complementary team performance in AI-assisted decision making. The problem of whether the Dunning-Kruger Effect (DKE) among people can hinder their appropriate reliance on AI systems has been explored [15]. DKE is a metacognitive bias due to which less-competent individuals overestimate their own skill and performance. Through an empirical study (N = 249), the authors explored the impact of DKE on human reliance on an AI system, and whether such effects can be mitigated using a tutorial intervention that reveals the fallibility of AI advice and exploiting logic units-based explanations to improve user understanding of AI advice. The tasks consist of presenting a text and letting the subject choose another text among 4 others that would best match the semantics of the presented text. The AI condition would highlight those parts of the text that would best help the subjects choose the correct answer. The tutorial condition would provide more explanation on why the answer is correct or not.

They found that participants who overestimate their performance tend to exhibit under-reliance on AI systems, which hinders optimal team performance. Logic unit-based explanations did not help users in either improving the calibration of their competence or facilitating appropriate reliance. While the tutorial intervention was highly effective in helping users calibrate their self-assessment and facilitating appropriate reliance among participants with overestimated self-assessment, the authors found that it can potentially hurt the appropriate reliance of participants with underestimated self-assessment.

An implication for the design of tutorials designed for promoting appropriate reliance should not only reveal the shortcomings of users or AI systems (i.e., when they are less capable of making the right decision), but also their strengths (i.e., when they are capable or more capable).

In summary, humans who overestimate their performance tend to exhibit under-reliance on AI systems, which hinders optimal team performance.

3.3.2.4 Human attitude toward Human or Automation leadership

It remains unclear how power functions in interactions with both humans and robots, especially when they directly compete for influence. An experiment where every participant was matched with one human and one robot to perform decision-making tasks has been conducted [16]. By manipulating who has power, the authors created three conditions: human as leader, robot as leader, and a no-power-difference control. The results showed that the participants were significantly more influenced by the leader, regardless of whether the leader was a human or a robot. However, they generally held a more positive attitude toward the human than the robot, although they considered whichever was in power as more competent.

The authors believe this suggests a new way that we can design for an agent's influence by designing for its power. For example, if an AI agent is known to perform well yet users are reluctant to adopt its suggestions for reasons such as algorithm aversion, we might consider increasing its power by giving the AI an expert framing, a higher organizational position, or the power to reward users. By doing so, we might be able to increase its perceived competence and thus its influence on users.

In summary, humans tend to be significantly more influenced by the leader, regardless of whether the leader was a human or a robot. However, they generally held a more positive attitude toward the human than the robot, although they considered whichever was in power as more competent.

3.3.2.5 Performances and delegation in hybrid teams

A study on how humans make decisions when they collaborate with an artificial intelligence (AI) in a setting where humans and the AI perform classification tasks has been performed [17]. The experimental results suggest that humans and AI who work together can outperform the AI that outperforms humans when it works on its own. However, the combined performance improves only when the AI delegates work to humans but not when humans delegate work to the AI. The AI's delegation performance improved even when it delegated to low-performing subjects; by contrast, humans did not delegate well and did not benefit from delegation to the AI. This bad delegation performance cannot be explained with some kind of algorithm aversion. On the contrary, subjects acted rationally in an internally consistent manner by trying to follow a proven delegation strategy and appeared to appreciate the AI support. However, human performance suffered as a result of a lack of metaknowledge—that is, humans were not able to assess their own capabilities correctly, which in turn led to poor delegation decisions. Lacking metaknowledge, in contrast to reluctance to use AI, is an unconscious trait. It fundamentally limits how well human decision makers can collaborate with AI and other algorithms.

With inversion, humans still contribute to the superior result; without them, the AI would not reach it. Inversion might also improve human work perspectives. Humans are more motivated when working in a stimulating environment [18]. For example, classifying easily identifiable images is perhaps routine and boring, whereas the classification of difficult images could be an interesting challenge. Inversion might enable humans to spend less time on mundane tasks and more time on challenging tasks, thereby creating a more fulfilling workplace. Thus, receiving assignments from a machine could be interpreted not only as a delegation to humans but also as freeing humans from tedious tasks. The AI would not be the humans' boss but rather an assistant who swipes away distractions from the real work.

In summary, Humans and AI who work together can outperform the AI that outperforms humans when it works on its own. However, the combined performance improves only when the AI delegates work to humans but not when humans delegate work to the AI.

3.3.2.6 Characteristics and dynamics of human-AI teams

There are many unknowns regarding the characteristics and dynamics of human-AI teams, including a lack of understanding of how certain human-human teaming concepts may or may not apply to human-AI teams and how this composition affects team performance. An article outlines an experimental research study that investigates essential aspects of human-AI teaming such as team performance, team situation awareness, and perceived team cognition in various mixed composition teams (human-only, human-human-AI, human-AI-AI, and AI-only) through a simulated emergency response management scenario [19]. Results indicate dichotomous outcomes regarding perceived team cognition and performance metrics, as perceived team cognition was not predictive of performance. Performance metrics like team situational awareness and team score showed that teams composed of all human participants performed at a lower level than mixed human-AI teams, with the AI-only teams attaining the highest performance. Perceived team cognition was highest in human-only teams, with mixed composition teams reporting perceived team cognition 58% below the all-human teams. These results inform future mixed teams of the potential performance gains in utilizing mixed teams over human-only teams in certain applications, while also highlighting mixed teams' adverse effects on perceived team cognition.

In summary, for such scenario, the results tend to show that teams composed of all human participants may perform at a lower level than mixed human-AI teams, with the AI-only teams attaining the highest performance.

3.3.2.7 Human perception and acceptance of imperfect AI

AI technologies have been incorporated into many end-user applications. However, expectations of the capabilities of such systems vary among people. Furthermore, bloated expectations have been identified as negatively affecting perception and acceptance of such systems. Although the intelligibility of ML algorithms has been well studied, there has been little work on methods for setting appropriate expectations before the initial use of an AI-based system. Some authors used a Scheduling Assistant - an AI system for automated meeting request detection in free-text email - to study the impact of several methods of expectation setting [20]. They explore two versions of this system with the same 50% level of accuracy of the AI component, but each designed with a different focus on the types of errors to avoid (avoiding False Positives vs. False Negatives). They show that such different focus can lead to vastly different subjective perceptions of accuracy and acceptance. They also design an Accuracy Indicator (Figure 7) and an Example-based Explanation (Figure 8) to depict the expected level of performances of the AI to help prepare users at coping with imperfection, as well as a slider to control the performance of the AI (Figure 9). They show that user satisfaction and acceptance can be improved through these simple expectation adjustment techniques. They also show that focus on High Precision rather than High Recall of a system performing at the same level of accuracy can lead to much lower perceptions of accuracy and decreased acceptance.

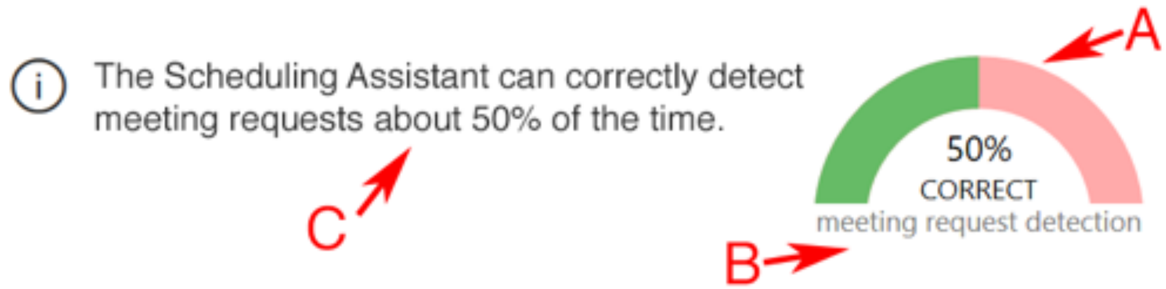

Figure 7: Accuracy Indicator [20].

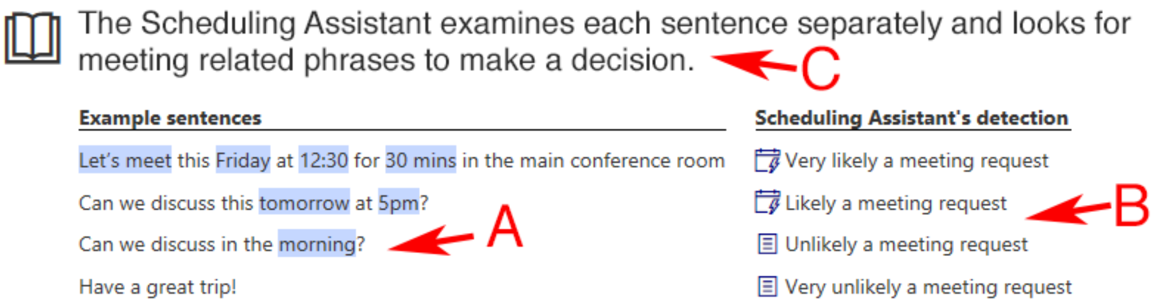

Figure 8: Example-based explanation [20].

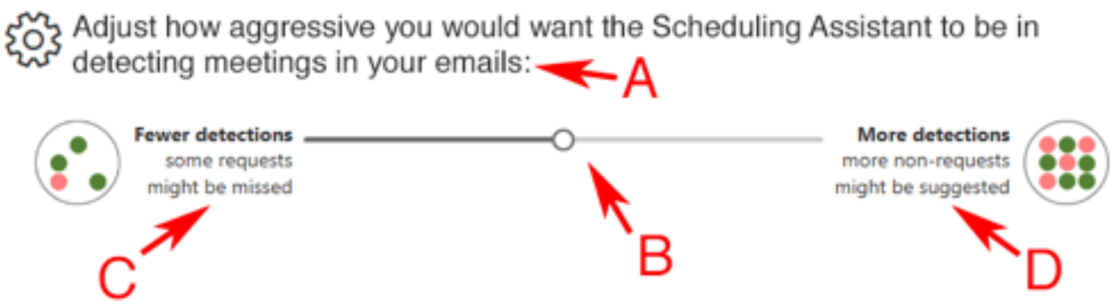

Figure 9: Performance control [20].

In summary, different focus on the types of errors to avoid (avoiding False Positives vs. False Negatives) can lead to vastly different subjective perceptions of accuracy and acceptance.

3.3.2.8 Dealing with AI mishaps

Many subfields of machine learning share a common stumbling block: evaluation. Advances in machine learning often evaporate under closer scrutiny or turn out to be less widely applicable than originally hoped. Some researchers conducted a meta-review of 107 survey papers from natural language processing, recommender systems, computer vision, reinforcement learning, computational biology, graph learning, and more, organizing the wide range of surprisingly consistent critique into a concrete taxonomy of observed failure modes [21]. Inspired by measurement and evaluation theory, they divided failure modes into two categories: internal and external validity (see Figure 10). Internal validity issues pertain to evaluation on a learning problem in isolation, such as improper comparisons to baselines or overfitting from test set re-use. External validity relies on relationships between different learning problems, for instance, whether progress on a learning problem translates to progress on seemingly related tasks. This work might help the way we assess the performance of our solutions.

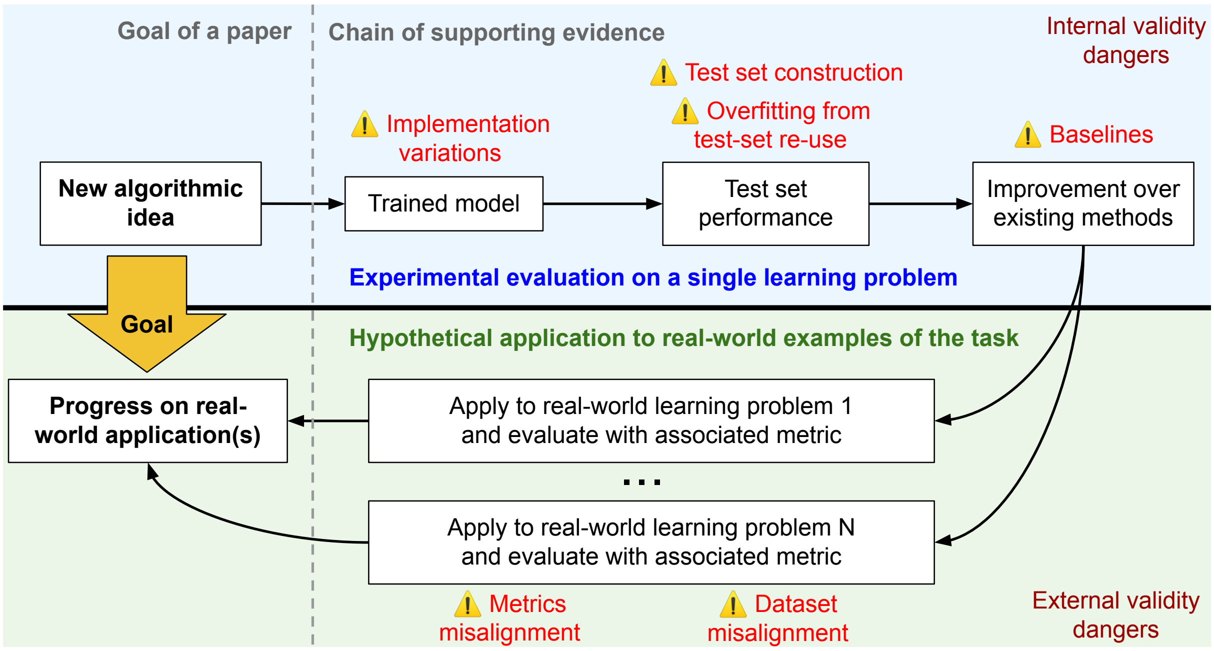

Figure 10: Framework for AI failure modes [21].

Inappropriate design and deployment of machine learning (ML) systems lead to negative downstream social and ethical impacts – described here as social and ethical risks – for users, society, and the environment. Despite the growing need to regulate ML systems, current processes for assessing and mitigating risks are disjointed and inconsistent. A group of researchers interviewed 30 industry practitioners on their current social and ethical risk management practices and collected their first reactions on adapting safety engineering frameworks into their practice – namely, System Theoretic Process Analysis (STPA) and Failure Mode and Effects @Analysis (FMEA) [22]. Their findings suggest STPA/FMEA can provide an appropriate structure for social and ethical risk assessment and mitigation processes. However, they also find nontrivial challenges in integrating such frameworks in the fast-paced culture of the ML industry. Even though our project does not primarily address social and ethical risks, such analyses may inform on some related aspects such as balance between AI and humans as well as issues regarding responsibilities.

Traditionally, AI research has been more concerned with improving accuracy rates of the algorithms than putting humans in the loop. However, recent work found that while accuracy is good, controllability may be better for specific tasks [23] and could prevent some risks due to misunderstanding of the AI. However, while this holds true for simple tasks as discussed in the study, for high level of automation the results might be different.

3.3.2.9 Summary

Most of the studies mentioned above rely on very specific experiments designed to be controllable and to lead to statistically significant results. Their concerns, results and implications are useful for ASTAIR as such. However, most of them admit that generalization is at stake. ASTAIR is meant to be a more complex ecosystem than simple image or text recognition applications. An important research question is thus the evaluation of the impact of ASTAIR interaction and automation on the ATC activity. The controllability of our future experiments will likely not reach the level of controllability of the above research. Still, we will have to design the experiments in such a way that they will inform us as well as the community on the usability of our approach and enable us to formulate new guidelines for designing Human-AI interaction.

3.3.3 Styles of interaction with automation

In a suggestive interface, the user gives hints about a desired operation to the system by highlighting related components in a graphical scene, thus improving the usability of gestural interfaces and augments typical command-based modelling systems [24]. Chateau is an instance of a suggestive interface for 3D drawings [24]. Chateau infers possible operations based on the hints and presents the results of these operations as small thumbnails. The user completes the editing operation simply by clicking on the desired thumbnail. The hinting mechanism lets the user specify geometric relations among graphical components in the scene, and the multiple thumbnail suggestions make it possible to define many operations with relatively few distinct hint patterns. The suggestive interface system is implemented as a set of suggestion engines working in parallel and is easily extended by adding customized engines.

Schmidt & Herrmann [25] have proposed the concept of intervention interfaces to enable joint control "where the majority of decisions are automated but where users can intervene ". Given the expertise needed to supervise airport ground traffic and the uncertainty inherent to airport ground movements, full automation alone will not ensure an optimal operation of the system. Gradual control of the automation decision power by the end-users is therefore required.

Calhoun et al. [26] surveyed the literature to compare adaptable and adaptive automation in application with different levels of automation. They define adaptable automation as user-initiated change in the level of automation and adaptive automation as system initiated. They found that adaptable automation (the human operator assigns how automation is applied) has been found to aid human's situation awareness and provide more perceived control versus adaptive automation (the system assigns automation level) that may impose less workload and attentional demands by automatically adjusting levels in response to changes in one or more states of the human, task, or the

Co-funded by the European Union

environment. In their survey, they found a very limited number of studies comparing the two conditions but for the one that did, their result show that adaptable automation was not only preferred over adaptive automation, but it also resulted in improved task performance and, notably, less perceived workload.

In ASTAIR, we want to explore different levels of automation and possibly consider transition between such levels according to user preferences but also AI performances. This might prove successful in improving the overall system performance.

3.3.4 Authoring and programming automation

Interacting with automation can be seen as a way to control a program that automatically controls entities. As such, interacting may also be considered as a form of programming. Usable programming of usable automation is thus an important stake, especially if end-users are to be involved in the design process [27]. Few works exist that tackle this topic. Automated machine learning (AutoML) is envisioned to make ML techniques accessible to ordinary users [28].

For automation to be safe and usable, it needs to be suitable to the activity it supports, both when authoring it and when operating it. Vizir is a Domain-Specific Graphical Language and an Environment for authoring and operating airport automations [27]. Vizir combines visual interaction-oriented programming constructs with activity-related geographic areas and events. Vizir offers explicit human-control constructs, graphical substrates and means to scale-up with multiple automations.

The authors devised a set of guidelines for such programming tools: Foster a continuum of usage between authoring, controlling and supervising; Provide space-based and event-based constructs; Make current state and future behavior visible; Foster both seamless and “seamful” hybrid control; Foster interaction-oriented programming.

Decision-making is a key software engineering skill. Developers constantly make choices throughout the software development process, from requirements to implementation. While prior work has studied developer decision-making, the choices made while choosing what solution to write in code remain understudied. In a mixed-methods study, researchers examine the phenomenon where developers select one specific way to implement a behavior in code, given many potential alternatives [29]. They call these decisions implementation design decisions. The mixed-methods study includes 46 survey responses and 14 semi-structured interviews with professional developers about their decision types, considerations, processes, and expertise for implementation design decisions. They find that implementation design decisions, rather than being a natural outcome from higher levels of design, require constant monitoring of higher-level design choices, such as requirements and architecture. They also show that developers have a consistent general structure to their implementation decision-making process, but no single process is exactly the same. They discuss the implications of their findings on research, education, and practice, including insights on teaching developers how to make implementation design decisions. This research might be related with the way automation is designed, especially if we consider real-time decisions by controllers to delegate a task to AI as a programming activity.

Co-funded by the European Union

While this is not expected, if during the ASTAIR project using Machine Learning (ML) becomes a relevant approach for AI, providing usable means to program Machine Learning might be useful to adapt to particularities in each airport. Automated machine learning (AutoML), a novel concept for automating the whole ML pipeline without (or as little as possible) human intervention, is envisioned to make ML techniques accessible to ordinary users. Recent work has investigated the role of humans in enhancing AutoML functionality throughout a standard ML workflow. However, it is also critical to understand how users adopt existing AutoML solutions in complex, real-world settings from a holistic perspective. To fill this gap, a study has conducted semi-structured interviews of AutoML users (N = 19) focusing on understanding (1) the limitations of AutoML encountered by users in their real-world practices, (2) the strategies users adopt to cope with such limitations, and (3) how the limitations and workarounds impact their use of AutoML [28]. The findings reveal that users actively exercise user agency to overcome three major challenges arising from customizability, transparency, and privacy. Furthermore, users make cautious decisions about whether and how to apply AutoML on a case-by-case basis. The authors suggest to: Foster User Agency in Developing Workarounds, Foster User Agency in (Non-)Use of AutoML, Support Domain-Specific Customizability, Provide Multifaceted Transparency, Enhance Data Privacy, Support Collaborative Work behind AutoML.

3.3.5 Initiative in Human Automation Teaming

Mixed-initiative interaction is concerned with interaction strategies where each agent (human or machine) takes turn at the most appropriate time to contribute to a task where it performs best [30]. The goal of mixed initiative interaction is to create collaboration between humans and artificial intelligence, leveraging the strengths and capabilities of both parties. By combining human expertise, creativity, and contextual understanding with artificial intelligence's analysis capacities and automation, mixed initiative systems can tackle complex tasks more effectively and efficiently than either humans or AI alone. Contrary to fixed initiative interaction where either a single human or system has always control of the interaction flow, in mixed-initiative interaction any agent can take the control of the interaction at any time.

Mixed-initiative interaction raises many challenges. For instance, intelligent agents are not good at guessing about goals, needs and intents of users, at considering the costs and benefits of automated actions, at performing timely and to advise users when they can perform better using automation [31]. To address these issues, Horvitz has proposed factors to be considered when designing mixed-initiative interaction [31]. These factors emphasize the need of identifying the added value of automation, understanding and predicting user's intents and goals, providing users with timely and non-invasive automation actions with means to control them, and appropriate communication between human and AI agents to clarify users' intentions. The author also suggests that mixed-initiative systems should allow users to make "efficient references to object and services" by maintaining a memory of recent interactions with users. Furthermore, a mixed-initiative system should provide means to "gracefully degrade the precision of services" to offer potential solutions even in unpredictable situations where correcting automation mistakes would be too costly.

With today's AI advances, intelligent systems are now capable of making their own decisions without the need of human input. Collaboration between humans and autonomous intelligent systems poses newest challenges. For instance, defining roles and tasks is not trivial anymore. Linked to task design and interpersonal dynamics, several strategies to negotiate who will do what have been proposed.

These strategies include static division of labor, adjustable automation, mixed-initiative collaboration and adaptive automation [32]. Although mixed-initiative collaboration has been investigated significantly, static division of labor, adjustable automation and adaptive automation has received little attention [33]. In static division of labor, the allocation of tasks to humans and agents is permanent. While in adjustable automation, the level of autonomy can be regulated by humans, in adaptive automation, the agent can alter its level of automation in response to the performance and human behaviors.

To fluidify the interaction between humans and intelligent agents, Luciani et al. have proposed fined grain interaction for continuous adjustment and immediate experienced results [34]. Not only designing fine-grained interaction enables the design for fast turn-taking and short response time, but also for closer and simultaneous interaction with a partially agentive system. This results in both parties being active and getting continuous feedback allowing adjustments to be made without interruption. Based on air traffic control activity observations and a co-design approach with air traffic controllers (ATCOs), Luciani et al. have built an assisted sketching tool that allows ATCOs to take better decision in guiding aircraft and managing the traffic [34]. Using visual cues and interaction techniques, the authors have managed to materialize uncertainty which encouraged participants to find alternative strategies to perform their tasks. In addition, they have found that sketching routes was a good way to feed the system, give directions to pilots, and share their intentions and plans between air traffic controllers for internal collaboration. Not only designing fine-grained interaction made the authors consider the whole interaction design rather than the interface alone, but it also allowed them to create displays with less visual clutter. Further research towards adaptability could be carried out to change the system behavior while it is being used.

When autonomous systems cannot regulate their own behavior appropriately, external bounding of autonomous behavior is required to assure their ongoing safety and effectiveness. Bradshaw et al. have proposed policies to support the implementation of adjustable autonomy in mixed-initiative interaction [35]. Their approach consists in constraining the autonomy of the system rather than generating plans for what an agent should do. They argue that human coordination mechanisms are required to assure effective teamwork among humans and agents. As an example of such mechanisms, they introduce agreements, a set of policies and information required for coordination, that can be represented within the system to govern specific aspects of joint activity among the parties. The policies can affect different aspects of coordination such as initiative, delegation, notifications, supervision, or human action constraints. This also allows for artificial agents to be adaptive and self-adjust their autonomy consistently with the policies, providing hands-off control among team members at any time, and renegotiation of roles and tasks when new opportunities arise or when breakdown occurs. Moreover, the policies can also support agents to anticipate adjustments.

Intelligent systems can support humans take decisions under risks. Risky decisions can be defined as taking a decision without knowing the exact consequences. Although intelligent systems are not affected by cognitive biases, fatigue, recent experience and environmental factors, humans still outperform AI in unknown and complex situations. This highlights the need for humans and intelligent systems to partnership. Xiong et al. argue that when the decision task is associated with higher uncertainty, human-centered research should be carried out [36]. When uncertainty is high, research on transparent AI, explainable AI and trustable AI can support human decision-makers valuing their own output. In complex systems with high automation such as aviation, a proper level of transparency is required to enable operators to understand the system strategy and the internal working conditions.

Representing the machine intent, the perception of the environment, and system status can help build a mental model of the system and reduce decision-making conflicts [36]. In addition, the authors suggest that human-machine collaboration for making decisions under risk should allow dynamic task allocation based on task requirements and the capability and characteristics of humans and machines. Furthermore, the machine should be able to adapt to human's cognitive limitations and rapid behavioral changes. Finally, both humans and machines as a team should be able to identify, understand and align with each other's goals, values, and intention to take timely and unintrusive initiatives.

In ASTAIR, we envision an environment where humans and artificial intelligence collaborate as a team. In our vision, some of the intelligent systems can be autonomous and make decisions on their own. To ensure the safety of passengers on the airport ground, uncertainty in risky decisions must be limited. Therefore, transparency is key. Human operators should be able to understand the intent, the knowledge and the status of intelligent systems to build a mental model of their operating and anticipate their and teammates' actions. Literature shows that maintaining a memory of recent actions should help build a mental representation of the AI teammates' operating. Human operators should also be provided with fine-grained interaction for closer and simultaneous interaction with AI and encouraging fast adjustments without interruption. Finally, operators should be able to constrain the AI with appropriate interactive tools by either limiting its autonomy so the operators can engage themselves into the task or degrade the AI performance with a simpler model that human operators can understand and manipulate easily.

Finally, the ethics guidelines for trustworthy AI delivered by High-Level Expert Group on Artificial Intelligence (setup by the European commission)[3] would be useful during ASTAIR project implementation with the guidelines to realise a trustworthy AI. In particular, we will build upon the assessment list for trustworthy artificial Intelligence [37] to provide inputs for the validation of the solution.

3.3.6 Explainable AI (XAI): definition and application to ATM and aviation

As for any interactive system, the adoption rate of AI algorithms is not only dependent on the performance of the algorithms, but also on the way the algorithms are perceived and understood by the users. Moreover, laws are enforcing the "Right to Explanation" [38].

While early AI systems were quite easy to understand for humans, the recent rise of Deep Learning (DL) models, including Deep Neural Networks (DNNs), has increased complexity of the algorithms [39]. DNNs are even described as "black boxes". Especially in critical domains, the ability to explain the model is now considered crucial for building trust and deployment of artificial intelligence systems [40]. The danger lies in using decisions that are not justifiable, or that miss detailed explanations of their behavior [39].

A concrete example of critical incidents related to misunderstanding AI systems is the occurrence of mode confusion. Mode confusion is defined as the user's incorrect understanding of the current and future status and behavior of the automation [41]. On the other hand, mode awareness is the person's

[3] https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai

Co-funded by the European Union

awareness of the current automation mode [42]. The automotive industry has explored a variety of mode awareness interfaces and interface elements designed to enhance the driver's awareness of current automation modes or overall automation capabilities [43]. The industry of commercial aircraft is more resistant to changing user interfaces due to the high financial costs, the need for pilot retraining and the complex certification procedures. This comes with an increased safety risk, as several critical incidents in aviation have been linked with misunderstanding autopilot modes [44], [45], [46].

The field of explainable artificial intelligence (XAI) aims at enabling users to understand the inner workings of AI systems and to get insights into the results of the algorithms [40]. The goal of XAI is to create machine learning techniques that 1) produce more explainable models while maintaining a high level of learning performance, and 2) enable users to understand, trust, and manage the emerging generation of AI systems [39]. Zhu et al. defined *explainability* in the context of XAI as "as being clear of obscurity and understandable in all aspects" and being able "to answer why questions" [47]. Kaadoud et al. defined *explanation* as "information in a semantically complete format, which is self-sufficient and chosen according to the target audience regarding its knowledge, its expectations and the context" [48]. Degas et al. [49] differentiate between "Understandability / Intelligibility" and "Comprehensibility", where the first explain the functioning of the model without explaining the internal algorithm, and the latter include the explanation of learned knowledge. Moreover, XAI is distinguished from "Observable AI", which allows to understand black-box systems from observation of all potential combinations of input and their related outputs [47].

Current XAI systems exhibit a diverse set of dimensions and functionalities for simple exploratory data analysis to understanding complex AI models [40]. Two main XAI techniques exist: (i) Ante-hoc techniques aim at optimizing an already transparent AI model by adding constraints or features to increase transparency through metrics, data visualization, etc. Explanation is considered from the very beginning of the training. (ii) Post-hoc techniques aim at explaining black-box AI models (e.g., DNNs). An external model mimics a base model's behavior to generate an explanation to the user [48], [49].

XAI systems have been applied in a variety of domains, including machine learning, robotics, multi-agent systems, computer vision, Knowledge Representation and Reasoning, etc. [48]. It is important to take the context (users, goals, environmental context) into account in XAI, although this is challenging [48].

In the context of air traffic management (ATM), Degas et al. [49] proposed a Design Space on XAI use in ATM, including "explanation" as one dimension. They observed that mainly four types of explanations have been used (numeric, rules, textual and visual explanations) or a combination of those explanation types. Their analysis also shows that most existing solutions are post-hoc solutions. Most of the methods aimed at improving explainability of prediction tasks (e.g. landing time prediction), while modelling / simulation were explored least. The authors suggest that optimization / automation would present an important use case, but which has so far rarely been studied in the literature. They argue that "fully understanding the underlining reasons of conflict avoidance procedures (e.g., explaining why one aircraft is moved away from its planned trajectory and not another), sequencing, or any other optimization result, is more than required to be accepted and used by human operators such as ATCO" (p.19). Finally, based on their findings, the authors propose to distinguish three types of XAI (see Figure 11): (1) Descriptive XAI which describes an AI model or output and is the basis for the following steps; (2) Predictive XAI which predicts the behavior of an AI model to a specific input or modification and allows to ask "what if" and "why not" questions; and (3)

Prescriptive XAI which detects errors or unwanted behavior of an AI model, suggests solutions to overcome these and allows to ask “how to” questions.

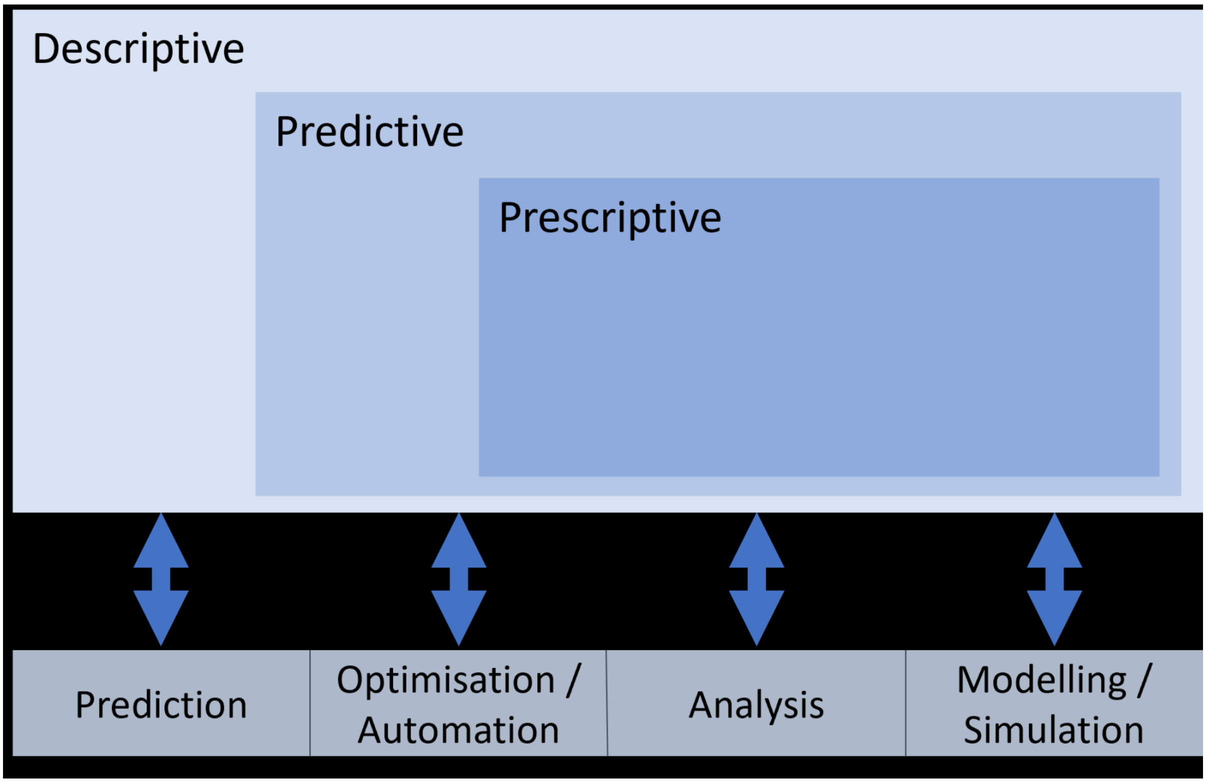

Figure 11: Synthesis of XAI framework for ATM [49]

Prior work in the cockpit demonstrated that bi-directional communication with a shared language between the autonomous agents and the pilot could enhance the teaming efficiency [50]. Such a language between automation of surface movements and airport ground operation stakeholders is required. We believe that a human-centric approach is necessary to maximize the capacity of humans to share their knowledge with the system using constructs that they are familiar with. However, we are still lacking knowledge on shared representations between human and AI agents to create an effective partnership for airport ground operations. Furthermore, interaction is necessary to ensure that each agent has an adequate situation awareness on its tasks and the behaviors of other agents [51].

The AI tools that will be used in the ASTAIR project are optimization based. This implies that we will have to investigate what inputs and outputs the AI is using to model and compute solutions to solve users’ problems. Based on the identification of such parameters, we will need to align end-users' goals and expectations so that we can create shared representations for humans and AI to work efficiently together. As demonstrated by Degas et al. [49] few prior works have explored optimization as a use case for XAI. The Astair project will allow contributing to this field of research.

3.4 Designing Human-AI systems with high levels of automation

EUROPEAN PARTNERSHIP
Co-funded by the European Union

The promises of advances in AI created opportunities for creating new interactions and new user experiences that would otherwise not be possible. This trend has led to the idea of AI as a design material in the research community, with the hope that HCI researchers and designers can effectively envision and refine new uses for AI that have yet to be imagined [52].

From the Human-Computer Interaction (HCI) community, researchers have proposed principles, frameworks, and guidelines to help understand the characteristics of such interaction for over decades [20], [31], [53], [54], [55]. For example, Amershi et al. [53] proposed 18 applicable guidelines for human-AI interaction in mass market products as text editors or calendars, which are categorized into four groups, including initially, during interaction, when wrong, and over time. Cimolino and Graham [55] reviewed prior work about human-AI shared controls and contributed a four-dimensional framework as an analysis tool, including AI role, supervision, influence, and mediation. Unfortunately, most of these guidelines are either very generic or focusing on mass-market products and not necessarily adequate for critical systems such as airports' ground operations.

In ASTAIR we will focus on designing interactions between humans and AI systems for automation levels categorized as 2B and 3A according to EASA's classification. Recent work suggests that in order to do so, designers need to understand both how the system-side AI works, but also how people think about, understand, and use AI tools and systems [56]. Conducting user-centered design activities such as interviews and stakeholders' workshops to understand the contexts, needs and goals of our end-users remains important [57]. It is particularly important to conduct activities ensuring that both users' and AI goals and constraints are well aligned together. As identified by Xu et al., it is very important to clarify the envisioned roles of humans and AI [58]. This is a main challenge in the ASTAIR project and part of our work will be dedicated to the identification of automation opportunities and then to the exploration of several alternative alternatives that corresponds to the level 2B and 3A of the EASA classification.

Another important aspect identified by Feng et al. [59] is related to the use of prototypes to communicate with end-users and technical teams. They conducted a study with 27 user experience practitioners in which they prototyped and created a design presentation for an AI-enabled interface while having access to a simple AI model training tool. Their results suggest that communicating AI concepts to end-users could be very challenging but that iteratively using prototypes was very helpful. The authors suggest that starting with even incomplete and not very efficient AI models could help elicit new requirements and improve both the interaction design and the AI for the task.

Designing for highly automated systems is also more difficult and more challenging than designing from traditional systems [56], [60], [61], [62]. HCI researchers have discussed challenges that persist in designing human-AI interaction encountered by designers [56] such as failing to recognize the appropriate situations where AI might help or envisioning novel features that exceed AI's current capabilities. Yang et al. [61], proposed a mapping of the challenges faced by designers according to several phases of a typical user centered design process [63]. Fortunately, the type of AI that will be used in ASTAIR is based on optimization techniques and not Machine Learning approaches which poses many of the identified challenges. Moreover, using ML algorithms would have required additional user-centered work to collect relevant data, validate that both the training and the results are adequate to end-users [64] as well as to cope with AI outcome uncertainty [61].

Co-funded by the European Union

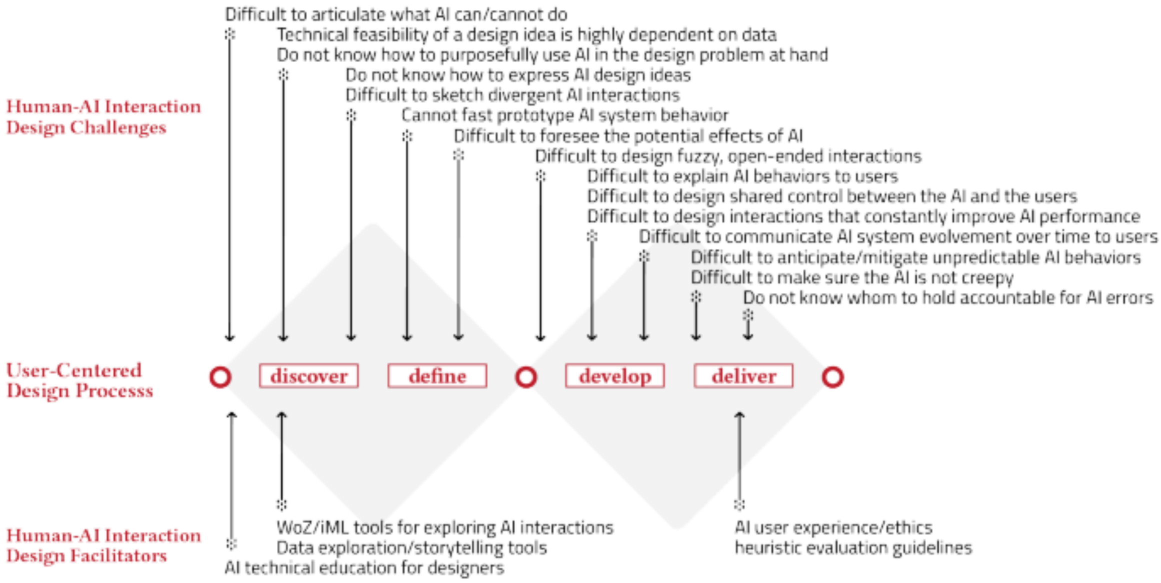

Figure 12: Mapping the human-AI interaction design onto a user-centered design process [61]

In ASTAIR we will face several design challenges induced by the use of AI. To overcome these challenges and mitigate the risks, we will follow recommendations from the literature and use a user-centered approach to carefully identify needs and requirements from the user perspective. We will also involve the AI team so that we get a mutual understanding of AI possibilities and be able to align users and AI constraints and goals.

4 State of the art on support algorithms for fleet management and path planning

To manage and perform engine-off and conventional airport surface movement operations in ASTAIR, the support algorithms previously developed in AEON will be further elaborated and extended.

These algorithms address two broad categories of tasks: (1) algorithms for tug fleet management; (2) algorithms for conflict-free and efficient path planning of all aircraft, tugs, and other ground vehicles.

This section reviews literature on both to identify research directions for algorithmic extensions related to ASTAIR goals and requirements to be further elaborated in WP2.

In particular, algorithms for tug fleet management should be able to adapt the assignment of tugs in real time during operation based on changes of schedules, as well as be able to take diverse temporal constraints into account.

Algorithms for path planning should be computationally efficient, should consider spatiotemporal constraints reflecting the airport's traffic rules. Furthermore, these algorithms should be able to dynamically adapt motion trajectories of aircraft and other ground vehicles, taking into account spatial and temporal constraints provided by ATCos, change of runway mode of operation, weather conditions, wake turbulence categories of aircraft. Furthermore, motion trajectories of aircraft and tugs should be optimized taking into account energy use and manoeuvrability.

The related literature was reviewed taking these considerations into account.

4.1 Algorithms for fleet management

Airport congestion is a major cause for the large delays that affect the air transport industry. Flight scheduling and fleet assignment are fundamental stages of the airline planning process. The problems faced by airlines when making their flight scheduling and fleet assignment decisions are highly complex, particularly when the airlines operate in congested, slot-constrained airports. In many airports, particularly in Europe, airlines are limited in the number of slots they can use because the declared capacity of airports is insufficient to accommodate peak period demand, constraining the choices of airlines in terms of time and frequency of flights [65]. A vast amount of previous research has focused on aircraft fleet management for assigning fleet types to flights.

An average of 4%–7% of fuel is burnt during ground activities at airports (taxing, waiting, and extra fuel carried to complete the journey at the destination airport). The greenhouse gas emissions released by airports is not only contributing to global warming, but also impacting the health of local communities living next to airports [66]. Thus, recent studies include electrification of taxiing operations to reduce emissions and create a positive environmental impact. Tug fleet management plays a critical role in improving the efficiency of taxiing operations.

In addition to aircraft fleet assignment and tug fleet allocation, fleet management is also required for allocating and routing autonomous vehicles to complete ground handling operations at the airport.

We present an overview of existing research on fleet management for the assignment of aircraft fleet to flight legs in **Section 4.1.1**, allocation of tug fleet to aircraft in **Section 4.1.2**, and assignment and routing of ground handling vehicles to complete ground handling tasks in **Section 4.1.3**.

4.1.1 Fleet assignment: aircraft fleet to flight legs

Flight scheduling determines the set of legs that the airline flies. A station is an airport serviced by the airline. A leg consists of an origin station, a destination station, a departure time, and an arrival time. After airline planners determine the flight schedule, each leg must be assigned a type of aircraft, or fleet type, which is called fleet assignment [67].

The fleet assignments must satisfy certain operational constraints, such as coverage, maximum overnight stays, and airport compatibility. Fleet assignments are tactical decisions, and changes in demand and maintenance requirements require an intermediate decision-making process to capture these changes before a flight's day of departure [68]. The factors considered in assigning a fleet to a flight leg are passenger demand, revenue, seating capacity, fuel costs, crew size, availability of maintenance at arrival and departure stations, gate availability, and aircraft noise [69].

[70] formulate and solve the fleet assignment problem as an integer linear programming model, permitting assignment of two or more fleets to a flight schedule simultaneously. The objective function can take a variety of forms including profit maximization, cost minimization, and the optimal utilization of a particular fleet type.

[65] propose a mixed-integer linear optimization model for integrated flight scheduling and fleet assignment. The objective is to maximize the expected profits of an airline that operates in congested, slot-constrained airports. Both airline competition and airline cooperation are dealt with in the model, though in a simplified manner. The model was applied to a case study involving the main network of TAP Portugal, which comprises 31 airports and 100 daily flight legs. [71] develop a modeling and optimization environment to identify the optimum fleet composition and the network of routes that best serve the predicted demand and demonstrate the ability of this environment to solve large fleet assignment and scheduling problems to near optimality by applying it to the United States Northeast Corridor using a fleet of electric and hybrid-electric regional aircraft.

[72] present a time-space network model and mixed integer programming formulations for the integrated flight scheduling and fleet assignment problem. A time–space network for a single aircraft type consists of a set of activity nodes and arcs. An activity node represents the occurrence of certain event. In the context of flight scheduling and fleet assignment, there are two types of events, departure event and arrival event. Each event uniquely corresponds to one activity node. The time and location of the node is exactly the time and location of its correlated event. Two events with the same time and location generate only one node in the network. Based on this definition, each flight corresponds to one departure event and one arrival event. An arc in the time–space network is a directed arc connecting two activity nodes. There are three types of arcs in this network, which are flight arc, ground arc and wrap-around arc. Each flight corresponds to one flight arc starting from its departure event and ending at its arrival event. Each ground arc connects two subsequent nodes at the same airport. Wrap-around arc is a special type of ground arc. It connects the last node and the first node at certain airport. The wrap-around arc represents circulation of aircrafts between two consecutive days. To keep the flow balance in the network, there is a need to guarantee the flow balance at each activity node. Each activity node has input flows (along input flight arcs, input ground arcs and input wrap-

around arcs) and output flows (along output flight arcs, output ground arcs and output holding arcs). The proposed mixed integer programming models are based on the well-known multi-commodity flow problem which generates relatively small optimality gaps and multi-commodity flow problem with side constraints which is NP-hard. The main constraints ensure that exactly one aircraft type is assigned to the mandatory flight legs (1), at most one aircraft type is assigned to the optional flight legs (2), the flow balance is maintained at each node in the network (3), maximum number of the available aircrafts is not exceeded for each aircraft type (4), flow values of the beginning wrap-around arc and ending wrap-around arc at the same airport are the same which guarantees that the schedule is repeated daily (5), the number of passengers choosing one flight leg is smaller than passenger capacity assigned to it (6), the market share of each itinerary follows the trend that they are proportional to utility value (7), the number of the flight leg copies assigned to each airport resource slot must be smaller than capacity of the slot (8).

Similarly, [69] solve the large-scale integer program of the basic daily fleet assignment problem. The mathematical model of the problem is a large multi-commodity flow problem with side constraints defined on a time-expanded network. These problems are often severely degenerate, which leads to poor performance of standard linear programming techniques. The large number of integer variables can make finding optimal integer solutions difficult and time-consuming. The methods used to attack this problem include an interior-point algorithm, dual steepest edge simplex, cost perturbation, model aggregation, branching on set-partitioning constraints and prioritizing the order of branching. The algorithm finds solutions with a maximum optimality gap of 0.02% and faster than using default options of a standard LP-based branch-and-bound code. The integer programming formulation of the basic fleet assignment does not consider the maintenance and crew planning constraints.

Let F is the set of available fleets, S_f is the number of aircraft in each fleet $f \in F$, C is the set of cities in the schedule. The set of flights in the schedule is denoted by L. Each flight $i \in L$ is alternatively represented by the elements (o, d, t) where $o, d \in C$ are respectively the origin and destination and t is the time. t^- and t^+ denote the times preceding and following the time t. The set of nodes N include the elements (f, o, t) where $f \in F$, $o \in C$, and t is the takeoff or landing time at o. The mathematical model of the basic fleet assignment problem [69] is given in the equations (i), (ii),..., (vii).

$$z = \min \sum_{i \in L} \sum_{f \in F} c_{fi} x_{fi} \qquad (i)$$

$$\sum_{f} x_{fi} = 1, \quad \forall i \in L\,, \quad (ii)$$

$$\sum_{d} x_{fdot} + Y_{fot^-t} = \sum_{d} x_{fodt} + Y_{fott^+}, \quad \forall\, (f, o, t) \in N\,, \quad (iii)$$

$$x_{fi} - x_{fj} = 0, \quad \forall (i, j) \in H, \quad (iv)$$

$$\sum_{i \in O(f)} x_{fi} + \sum_{o \in C} Y_{fot^n t^1} \quad \leq S(f), \quad \forall\, f \in F\,, \quad (v)$$

$$Y_{fott^+} \geq 0, \qquad \forall\, (f, o, t) \in N, \quad (vi)$$

$$x_{fi} \; \in \{0,1\}, \quad \forall\, f \in F, \forall\, i \in L \quad (vii)$$

where c_{fi} denotes the cost of assigning fleet type *f* to flight *I* and x_{fi} is the binary decision variable which takes 1 when fleet type *f* is assigned to flight *i* and 0 otherwise. The objective (i) is to minimize the sum of all assignments. x_{fi} variables are alternatively represented as x_{fodt} where (o, d, t) corresponds to the flight $i \in L$, $o, d \in C$ and t are respectively the cities and time in the schedule. The constraint set (ii) ensures that each flight leg is flown by exactly one fleet. The constraint set (iii) include balance constraints. The fleet assignment solution must satisfy balance constraints that force the aircraft to circulate through the network of flights. The decision variable x_{fodt} which is also written as x_{fi} is equal to 1 if fleet *f* flies the flight leg from *o* to *d* departing at time t, and 0 otherwise. Y_{fott^+} where $f \in F$, $c \in C$, and $[t, t^+]$ is a time interval, are called ground arc variables that count the number of aircraft on the ground at each station at every point in time for each fleet. The balance constraints are enforced by modeling the activity at each station with a timeline for each fleet. This timeline has entries designating the arrivals and departures from the station for each fleet. Each departure (arrival) from the station splits an edge and adds a node to the timeline at the departure (arrival + refueling/baggage handling) time. In constraint set (iv) the flight legs of each *required through* are enforced to be flown by aircraft of the same fleet. Certain pairs of flights are required to be connected. These connections are called *required throughs,* and the set of *required throughs* is denoted by *H*, with elements (*i, j*), $i, j \in L$. The schedule may need to violate the minimum ready times for some flights because of fleet size restrictions. These special short ready times are also modeled as *required throughs*. The constraints in set (v) are the fleet size constraints which count the number of aircraft of each fleet used in the solution. Each fleet network is sliced at 3am EST and the flow across this cut set is counted. The set of *O*(*f*) denotes the flight arcs whose time span contains 3am EST. (*f, o*, t^n) is the last node in a timeline which is the node that precedes 3am EST. The successor of the node (*f, o*, t^n) is the node (f, o, t^1). (vi) and (vii) show the continuous and binary decision variables.

The scale of the flight legs, the equipment types, complex operational constraints, maintenance requirements, and other complex criteria specified by the route planners necessitates the development of a sophisticated optimization suite to generate swaps of flight legs among the different equipment types for the allotted fleet assignments. [68] propose a swapper optimization suite (SOS) which uses optimization models to generate the optimal swaps, for one of the largest airlines in Japan.

Even though the assignment of aircraft fleet to flight legs is less relevant regarding the application areas of ASTAIR, the mathematical models for assigning the aircraft to timeslots can be used as a guideline in ASTAIR for assigning taxibots to aircraft within specific time windows.

4.1.2 Tug fleet management

The introduction of towing techniques involves a considerable increase in the number of vehicles running on taxiways and service roads. The safe and efficient use of these vehicles implicitly requests the redefinition of the procedures previously in force and, when needed, the introduction of new ones. The AEON project [73] designed and assessed interconnected solutions to enable an optimized allocation of a fleet of tugs to aircraft, predefined routing providing speed profiles to avoid conflicts, dedicated HMI for Air Traffic Controllers as well as a new role, the Tug Fleet Manager. In the long/medium planning phase, the AEON fleet management algorithm supports the operator in the estimation of the adequate number of tugs, considering the needs of a given airport (and its stakeholders) in each period considering its specific traffic conditions. In addition, considering the

Co-funded by the European Union

arrival and departure sequences and the operational constraints of the tugs fleet, this algorithm sizes the fleet of tugs needed and at the executory level can reallocate the fleet if needed.

The AEON tug scheduling algorithm takes the following as input: (1) the airport road networks, their types and the time it takes to traverse the airport using the different networks, (2) flights schedule during a day of operation to estimate the drop-off time and corresponding energy based on a single-agent version of the path planning algorithm, (3) the list of tugs present at the airport that has to start and end their day of operations at the depot with a full battery. Using these input parameters, the tug fleet management algorithm creates a tug schedule. This schedule includes the town aircrafts, the associated tugs, and when (and where) the tugs are going to recharge.

The literature also includes methods for optimal allocation of tug fleet considering collision free taxiing and finding the optimal tug fleet size.

[66] propose a mixed Integer linear programming (MILP) model which aims at assigning electric powered tow-tractors for airplanes to complete taxiing operations with minimum jet-fuel usage. The flight schedule which includes aircraft type, arrival time, departure time and the gate number, is known in advance. Each aircraft completes its taxiing operations by following physical lines which are available in most airports as taxiways. A mesh network is generated to enable surface movements. Each intersection is a node, and nodes are connected to each other by arcs (links). Airplanes can follow each other on the same link by respecting the minimum allowed safety distance. No two airplanes can travel from opposite directions on the same link at the same time. All parallel links are assumed to be separated from each other by a sufficient distance to ensure collision free taxiing. Travelling times between two nodes is bounded by a fastest travelling time. Fuel consumption rate is assumed to be constant per minute of operation, although fuel consumption rate changes when aircraft speed is changed. When they are not serving an aircraft, tow-tractors would not conflict with other moving aircraft. The objectives are to minimize airport ground operations cost, fuel cost and delay cost. The performance of the proposed model is shown for a case at Montreal’s Pierre Elliott Trudeau International Airport (YUL) that has three runways which can be used in both directions and handles an average of 730 flights daily through its 89 gates. The network of the YUL taxiways includes 125 nodes and 282 arcs. An airplane may enter (or exit) the network through gate or runway nodes. In the case study, 60 gates, 16 entry/exit points on runways, and 49 intersections between taxiways were considered. In addition to the tow-tractor assignments, minimizing taxiing collisions and determining the optimum number of tow-tractors were also the part of the proposed model.

[74] propose an end-to-end optimization framework for electric towing vehicles (ETVs) dispatchment at large airports. They integrate the routing of the ETVs in the taxiway system where minimum separation distances are ensured, with the assignment of these ETVs to aircraft towing tasks and scheduling ETV battery recharging. The results show that the 913 arriving and departing flights can be towed with 38 ETVs, with battery charging distributed throughout the day. The fleet size is shown to increase approximately linear with the number of flights in the schedule.

[75] propose strategic and disrupted models to create an adaptive vehicle-to-aircraft assignment, using Mixed Integer Linear Programming. The objectives are to maximize the number of towed aircraft and minimize the schedule changes for vehicle operators. Vehicle and aircraft routing, conflict avoidance, and energy usage are also considered in the models. Authors investigate also the impact of fleet size and general on-time performance on the assignments. [76] and [77] also study the vehicle-to-aircraft assignments for ETVs. They propose a Linear Programming model for selecting the aircraft to be towed,

to maximize fuel reduction. [76] perform sensitivity analysis on ETV fleet size. [77] include also the collision avoidance in the model. [78] combine vehicle-to-aircraft assignment with the vehicle and aircraft routing, by simulating all ground movement.

[79] present a receding horizon genetic algorithm (RHGA) for dynamic resource allocation. They consider a fleet of tugs operating along a coastline with the purpose of preventing oil tankers from drift grounding. The main role of these tugs is that if an oil tanker loses manoeuvrability through steering or propulsion failure, there will be a tug sufficiently close that it can intercept the drifting oil tanker before it runs ashore. The tugs must dynamically be assigned moving target positions for tracking such that the overall risk of any oil tankers drifting aground is minimised. A simulated case study on optimal positioning of a fleet of tugs along the northern Norwegian coast serves as a means of evaluating the algorithm. The proposed algorithm plans iteratively the movement trajectories for each individual tug such that the net collective behaviour of the tugs outperforms that of stand-by tugs stationed at bases located uniformly along the coast. An improved version of this algorithm is later presented by [80] to solve the same tug fleet optimization problem. A receding horizon mixed integer programming (RHMIP) model for optimal dynamic allocation of tug vessels to oil tankers was proposed by [81].

Despite being the most relevant concept for ASTAIR, fleet assignment models focusing on tug allocation to aircraft is rather new and existing research on this area is comparatively less. The main contributions are the outcomes of the projects such as AEON and the recent research on assigning electric towing vehicles to aircraft. The remaining research focus on tug allocation in maritime. The innovative solutions we aim to design and develop for assigning taxibots to aircraft in ASTAIR will provide a significant contribution to state-of-the-art in this area.

4.1.3 Fleet management for ground handling

The aircraft Ground Handling (GH) operations represent the airside activities at airports in charge of processing passengers, cargo, facilities, and supplies at and around parked aircraft. Most of these operations are performed by different service providers, using specialized vehicles and equipment known as Ground Support Equipment (GSE) whose management is core to GH [81].

Automation of ground handling processes using electric vehicles plays an important role in improving efficiency and reducing carbon emissions. Automation of tasks requires strategic allocation and scheduling of tasks given a limited size of heterogeneous GSE fleet, as well as creating the conflict free routes for the GSE vehicles traveling on aircraft stands.

Assignment of GSE fleet to a heterogeneous set of ground handling tasks and generating task sequences that minimize both the turnaround time between consecutive flights and the makespan for all vehicles becomes a challenging problem when the available GSE fleet is limited, and the flights are frequent.

[82] present a framework that combines task allocation and path planning for automation of ground handling operations, using a multi-agent perspective. In this study, the task allocation problem is handled using an integrated solver that combines an auction algorithm with a mixed integer programming model which is used to generate bids at each round of the auction. For each candidate task, possible assignment to a potential position of existing schedule of each GSE vehicle is evaluated by reoptimizing the schedule of the vehicle including the candidate. The candidate is assigned to a

vehicle if only that vehicle is the winner of the auction in that round. By this way, partial schedules are created by taking the interests of different agents into account at each decision phase until all tasks are allocated. To develop the mixed integer programming model, the task scheduling problem for a single vehicle is converted into a single vehicle pick-up and delivery problem with time windows, which also considers the movements of GSE vehicles on the paths in addition to processing times of tasks. The considered GSE fleet includes refuelling, catering, baggage handling, water and lavatory service vehicles.

To lower the ramp risk and improve the aircraft ground handling efficiency, [83] propose solutions for accurate tracking, collision detection, and optimal scheduling of airport Ground Support Equipment which includes vehicles with one carriage, such as tractors and shutters, as well as the baggage transit trains that contain one tug plus multiple dollies. For optimal scheduling of GSE, a mixed-integer linear programming model that aims to minimize the total rental cost and travel time of the equipment while respecting the constraints that include flight timetables, speed limits, size of available GSE fleet, maximum number of dollies that can be attached to baggage transit trains. An efficient heuristic algorithm is proposed to solve the model.

[84] develop a mathematical model for determining the number of airport equipment dedicated for the baggage loading and unloading. The demand for the carts and loaders is predicted and based on the prediction, the optimal number of equipment that can handle all flights is obtained. [85] design a model for scheduling aircraft ground handling operations with uncertain durations which might be due to breakdowns, weather conditions, cargo loading and unloading incidents. Critical Path Analysis and Monte Carlo Simulation are used to improve the aircraft ground handling operations during the turnaround.

[86] study the GSE scheduling problem with mixed fleet of fuel vehicles and electric vehicles with time windows and the objective of minimizing the sum of time, energy and emission costs and propose an optimal fleet configuration model. Scenarios with different characteristics of road network scale, terminal configuration and flight are tested and results show that scenario characteristics affect the optimal fleet allocation strategy.

[87] consider the problem of scheduling de-icing vehicles. The objective is to minimise the delay of flights due to de-icing, and the travel distance of the de-icing vehicles. They propose a greedy randomised adaptive search algorithm. A case study of real-life data from Stockholm Arlanda Airport shows that proposed method performs significantly better compared to simple scheduling strategies.

[88] propose a ground handling management structure which allows the automation of operations to face the growing demand for this service. It is shown how at operations level, information exchange with the airport collaborative decision-making system turns possible on-line fleet assignment to ground handling tasks. This is done by designing different heuristics for assignment of fully automated or semi-automated vehicles to ground handling tasks. Numerical results for an actual airport are presented to illustrate the potential performance of automated ground handling operations.

4.2 Algorithms for path and motion planning

Algorithms for path and motion planning are explained in four subsections. **Section 4.2.1** focus path and motion planning algorithms for airport surface movement, **Section 4.2.2** explains the state-of-the-art path and motion planning algorithms that are used in various environments, **Section 4.2.3** summarizes the recent research directions heading towards the solution of path planning combined with target assignment**, Section 4.2.4** presents a comparison of solvers in terms of complexity and solution quality, and **Section 4.2.5** provides with a brief summary of explainable AI methods for path and motion planning.

4.2.1 Path and motion planning for airport surface movement

Research in airport ground movement include the path and motion planning of aircraft taxiing on airport surface layouts.

[89] develop a mixed-integer linear programming formulation to optimize the timed taxiing routes of all aircraft on an airport surface. The constraints of the model include boundary constraints which enforce initial location, initial time based on pushback ready time, runway exit time, routing constraints, timing constraints, conflict constraints, and time windows constraints. The objective is weighted function of emissions, taxiing times, deviations from intended departure times. A minimum and a maximum taxiing speed exist for each aircraft type. The taxiway grid is represented by a directed graph. The aircraft can hold at any node of the graph. Only the conflicts between taxiing aircraft are resolved, Deviations from departure times are allowed only if they do not affect the departure slot of other flights. Amsterdam Schiphol Airport is used as a case study.

[90] integrate speed profiles into conventional routing and scheduling problem. Speed profile optimization problem is defined as a multi-objective optimization problem where the objectives are to minimize total taxi time and fuel consumption. The routing and scheduling problem is to route aircraft from source to destination locations in a time and fuel-efficient manner, respecting routes and schedules of other aircraft while preventing conflicts between them. The airport surface is represented as a directed graph, where the edges represent the taxiways and the vertices represent the taxiway crossings, intermediate points and sources/destinations such as gates, stands and runway exit points. All edges of taxiway network are assumed to be bidirectional. Only one aircraft can travel along one edge at a time so that a minimum safety distance from all other aircraft is ensured. The period when the edge is not used by any other aircraft is called a time window. The k-Quickest Path Problem with Time Windows (k-QPPTW) is used to solve the routing and scheduling problem. The k-QPPTW algorithm, which was proposed by [91], sequentially routes aircraft considering their pushback/landing time, while respecting time windows corresponding to edges, and generates a set of k-best solutions regarding minimum taxi time and maximum allowed speed. These potential routes are used as input for the speed profile optimization problem. A major European hub, Zurich Airport (ZRH), is used as case study. A similar case was analyzed for Manchester Airport in the study of [92], where the Pareto front for taxiing time and fuel consumption is found by applying an immune inspired multi objective optimization algorithm (PAIA).

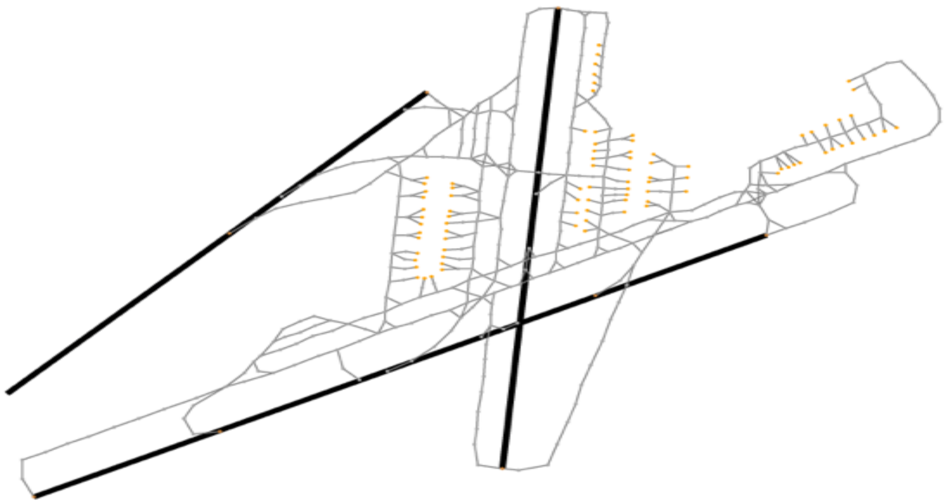

Figure 13 : Layout of ZRH with taxiways [54]

[93] minimize the taxiing time considering the runway exit availability and modeling realistic flight holding patterns at intersections. The presented integer programming model includes flow conservation constraints, taxi-out constraints, taxi safety constraints, runway occupancy constraints, taxiway occupancy constraints, re-taxiing avoidance constraints, parking position and runway taxiing avoidance constraints, route uniqueness constraints, boundary constraints, runway exit availability constraints and existing plan constraints.

[94] present a mixed-integer linear programming optimization method for the coupled problems of airport taxiway routing and runway scheduling, that is validated at Heathrow Airport. The mathematical model involves taxi timing constraints for speed and conflicts. Heathrow layout is represented by the 126-node graph structure. The setup included 240 aircraft, 122 of which were arrivals.

[95] propose optimization-based solution approaches for simultaneous aircraft scheduling and routing in terminal area, to minimize delays. The disturbed traffic situations are generated by simulating multiple delayed arriving/departing aircraft and a temporarily disrupted runway. Timing and routing decisions are proposed for Milan Malpensa Airport (MXP).

[96] proposed ground taxiing route optimization model that avoids hotspots on the surface and minimizes total taxiing time. Hotspots are the areas where taxi conflicts are most likely to occur.

In the study of [97] the Airport Surface Petri Nets (ASPN) is modeled with Colored Timed Petri nets (CTPN). Optimal paths are obtained based on the evolving states of Petri nets. The path finding problem for several aircraft is solved by finding the optimal path of each aircraft separately considering dynamic obstacles. The use of petri nets in modeling airport surface movement is also observed in the research of [98], [99], [100], [101], [102]. [100] propose a colored taxiway-oriented Petri net model. Access priorities of aircraft for a road section are adjusted by decreasing the priority of delayed aircraft. Sun and Hua [101] use fuzzy Petri net for aircraft trajectory segment sequencing,

[103] present an autonomous dispatch motion control framework for multiple carrier aircraft taxiing on the deck. The problem of finding the optimal coordinated taxiing trajectory is defined as a

centralized optimal control problem, where the constraints are based on safety limit for taxiing velocity, physical limit on maximum aircraft front wheel steering angle, acceleration limit, collision-free conditions between aircraft and other obstacles, and boundary conditions (parking position, heading angle, …, etc.). Feasible taxiing trajectories for several active aircraft and corresponding control inputs are found by solving the optimal control problem. Optimal control technique is also used by [104] to consider motion and control constraints in path planning for unmanned ground systems including vehicles and robots, which are widely used in aerospace, military, civil and other fields.

[105] use genetic algorithm to solve the taxiing problem on a complex taxiway network at Chengdu Shuangliu airport, considering the taxiway operation rules and conflict avoidance. The results are compared to the results of Dijikstra algorithm.

[106] model the collaborative path planning for multiple carrier-based aircraft as a multi-agent reinforcement learning problem.

[107] simulate aircraft ground movements at Lisbon International Airport, to predict taxi times for TaxiBots, semi-robotic towbarless tractors suitable for dispatch towing at medium to large airports.

Path and motion planning algorithms focusing on airport surface area usually requires defining the environment as complex graphs including intersections of taxiways and edges. Thus, they are highly adaptable for solving the problems in ASTAIR, in which case the layouts of different airports are stored in the databases in the form of graphs.

4.2.2 State-of-the-art algorithms for path and motion planning

The problem of path planning for multiple robots ranks among the most challenging problems of artificial intelligence and particularly of theoretical robotics ([108], [109], [110]). A group of robots in a certain environment need to move from their initial positions to the given goal positions. The robots are required to avoid obstacles and must not collide with each other during their movements. Thus, the task is to find spatial-temporal paths from the initial to the goal position for each robot such that these paths do not intersect at the same time point ([111]). Application domains of Multi Agent Path Finding (MAPF) include robotics, robotics, video games and logistics ([112], [113]). MAPF is also applied for autonomous aircraft towing vehicles ([114]).

Motion planning is the extension of path planning. Motion planning aims at generating interactive trajectories in workspace when robots interact with dynamic environment, therefore motion planning needs to consider kinetics features, velocities and poses of robots and dynamic objects nearby [115].

MAPF solvers include optimal and bounded sub-optimal solvers ([116], [117], [118], [112], [119], [120]), fast prioritized planners without any completeness/optimality guarantees ([121], [122]), and complete, non-optimal algorithms ([123], [124], [111]).

[111] defines the environment of robots as bi-connected graphs with at least two unoccupied vertices, where robots are placed in its vertices, which is equivalent to the problem of pebble motion on graphs. To solve this class of the MAPF, [111] proposes a polynomial time algorithm, BIBOX, which scales well in highly connected 2D and 3D spaces. Another complete, non-optimal algorithm for solving the cooperative multi-agent path planning algorithm is PUSH AND ROTATE ([123], [124]). The algorithm is complete for the class of instances with two unoccupied locations in a connected graph.

A group of optimal and sub-optimal MAPF solvers are based on conflict-based search. Conflict Based Search (CBS) is an optimal multi-agent pathfinding algorithm, which is presented by [113]. At the high level, a search is performed on a Conflict Tree (CT) which is a tree based on conflicts between individual agents. Each node in the CT represents a set of constraints on the motion of the agents. At the low level, fast single-agent searches are performed to satisfy the constraints imposed by the high-level CT node. In many cases the two-level formulation enables CBS to examine fewer states than A*. The classical A* algorithm ([125]) can route a single agent to its destination. In a coupled approach, a simple MAPF solver can be implemented by concatenating all the single agent states into a joint state and then using a generic search algorithm like A* for traversing the joint space to find the joint state solution. Coupled approaches tend to provide stronger guarantees on feasible paths and minimum cost by exploring the joint space. However, they have a high computational cost as the dimensionality of the joint space increases with the number of robots ([126]).

[119] formalize the problem of optimal pathfinding for multiple agents using a search tree called the increasing cost tree (ICT) and present a search algorithm, called the increasing cost tree search (ICTS) that finds optimal solutions. ICTS is a two-level search algorithm. The high-level phase of ICTS searches the increasing cost tree for a set of costs (cost per agent). The low-level phase of ICTS searches for a valid path for every agent that is constrained to have the same cost as given by the high-level phase. The search strategy of the proposed algorithm is compared to A* search and outline the benefits and limitations. It is also claimed that the proposed formalization allows further pruning of state space and the pruning techniques for ICTS are studied further by [127].

Meta-agent CBS (MA-CBS) ([120]) generalizes CBS by merging groups of agents into meta-agents when beneficial. Improved CBS (ICBS) ([117], [118]) improves MA-CBS. ICBS guarantees finding optimal solutions for cooperative pathfinding problems ([128]). Enhanced CBS (ECBS) ([116]) is the modification of CBS which trades off optimality for speed. [116] develop several suboptimal variants of CBS, relaxing the high- and low-level searches to allow them to return suboptimal solutions. These are Greedy-CBS (GCBS), a fast suboptimal solver, Bounded CBS (BCBS) that uses a focal-list in low- and high-levels and ensures that the returned solution is within a given suboptimality bound, and Enhanced CBS (ECBS) in which the high- and low-levels share a joint suboptimality bound. Nested ECBS (NECBS), which is proposed by [127] is a nested architecture based on ECBS, where collisions within meta-agents are resolved with ECBS. The merging technique from CBS is extended to ECBS, which results in Meta-Agent ECBS (MA-ECBS) and using ECBS to resolve the collisions between agents within the same meta-agent, results in Nested ECBS (NECBS). NECBS preserves the completeness and bounded-suboptimality of ECBC and has a higher success rate than ECBS and its state-of-the-art variants for a runtime limit of 5 minutes.

[130] proposes a set of efficient decoupled approaches that break down the multi-agent path finding problem into a series of single-agent searches, which are named as Cooperative A* (CA*), Hierarchical Cooperative A* (HCA*), and Windowed Hierarchical Cooperative A* (WHCA*). The algorithms are performed on maze-like environments and compared to Local Repair A*. Local Repair A* (LRA*) (Stout [131]) describes a family of algorithms widely used in the video-games industry. Each agent searches for a route to the destination using the A*, ignoring all other agents except for its current neighbours. The agents follow their routes, until a collision is imminent, and the remaining route is replanned. In Cooperative A*(CA*) the task is decoupled into a series of single agent searches. The individual searches are performed in three-dimensional space-time and consider the planned routes of other agents. A wait move is included in the agent's action set. After each agent's route is calculated, the

Co-funded by the European Union

states along the route are marked into a reservation table. Entries in the reservation table are avoided during searches by subsequent agents. The reservation table represents the agents' shared knowledge about each other's planned routes. Agents may be different in speed or size, however the reservation table must be capable of marking off any occupied region. The order of agents might affect the solution quality, which might be dealt using Prioritized Planning. Hierarchical Cooperative A* (HCA*) improves performance using a heuristic, which ignores both the time dimension and the reservation table. Abstract distances are perfect estimations of the distance on a 2-dimensional map ignoring the agents' interactions. Windowed Hierarchical Cooperative A* (WHCA*) limits the space-time search depth to a dynamic window, spreading computation over the duration of the route. M* ([132]) is an A*-based algorithm that dynamically changes the branching factor based on conflicts.

[133] propose safe interval path planning (SIPP) and compare SIPP against HCA* ([130]). Safe intervals represent time using the indices of contiguous periods, instead of using timesteps. This idea greatly decreases the number of states that need to be searched, without sacrificing the theoretical guarantees on optimality. The maximum number of safe intervals for any given configuration is at most the number of dynamic obstacles whose trajectories intersect in that configuration. SIPP allows for very fast planning in dynamic environments when planning time-minimal trajectories. Generalized Safe Interval Path Planning (GSIPP) ([134]) extends the results from planning with safe intervals to derive a state dominance relationship for dynamic environments that can be applied to continuous cost domains. To deal with the uncertainty in the predicted trajectories of moving obstacles, they propose Generalized Probabilistic Planning with Clear Preferences (PPCP) ([128]).

Bounded suboptimal SIPP algorithms include weighted SIPP (WSIPP), Weighted SIPP with Duplicate States ($WSIPP_d$), Weighted SIPP with Re-expansions ($WSIPP_r$), Focal SIPP (FocalSIPP) ([135]).

[128] develop Any Angle Pathfinding Algorithm based on SIPP for multiple agents (AA-SIPP(m)). This is a decoupled prioritized planner that applies Any Angle SIPP to multiple agents. Typically, in 2D grid pathfinding an agent is presumed to move from one traversable (unblocked) cell to one of its eight adjacent neighbours. Sometimes diagonal moves are prohibited, restricting an agent's moves to the four cardinal directions only. The limitations of 8 (or 4) connected grids increased the popularity of any-angle pathfinding. In any-angle pathfinding, an agent is allowed to move into arbitrary directions and a valid move is represented by a line segment, whose endpoints are tied to the distinct grid elements (either the center or the corner of the cells) and which does not intersect any blocked cell. Single agent any-angle pathfinding algorithms, Theta* ([136], [137]), optimal any-angle path finder Anya ([138]), fast near-optimal any-angle path finder with 2k neighbourhoods ([139]), find shorter and realistic paths. When multiple agents follow any-angle paths the conflicts can occur at any point. Using AA-SIPP(m), cooperative pathfinding problems are solved under any angle assumption. The proposed multi-agent planner AA-SIPP(m) is compared with the grid-based planners including SIPP for multiple agents (SIPP(m)) and coupled CBS based solvers ICBS and ECBS. The proposed method is complete under well-defined conditions, as well as highly efficient in practice. The success rate of AA-SIPP(m) is extremely high (>97%) and the average solution cost is significantly better (up to 20%) than the one achieved by both coupled and decoupled planners, that rely on cardinal-only moves ([128]).

[140] extend SIPP by developing Any Time SIPP (ASIPP) planner, which works well in dynamic environments, since any time planners find an initial solution quickly. They demonstrate the real-time capabilities of the Any Time SIPP planner in UAV domain, planning paths on large maps with 50 dynamic obstacles in a short time. Each obstacle is treated as a sphere with a radius and a trajectory.

A trajectory is a list of points, where each point has state variables, specifying its configuration and time. The points in the trajectory list are ordered from earliest time to latest time. Thus, the trajectory shows how the obstacle is predicted to move. The proposed algorithm extends SIPP to anytime planning by combining it with ARA* (Anytime Repairing A*). ARA* performs anytime planning by running a series of weighted A* searches with decreasing values of ε ([141]). Weighted A* does not guarantee optimality, however it has been shown that the obtained solution is not larger than the optimal solution times ε. ARA* initiates a weighted A* search with a high ε, to find an initial solution quickly and decreases the value of ε. Given enough time, ARA* will reach ε = 1 and return the optimal solution. The results of ASIPP and SIPP are compared.

[142] combine SIPP and Constrained Path Following Control. First, they plan the reference trajectory by the safe interval path planning algorithm that is capable of handling any-angle translation and rotations. Second, the path following problem is treated as the constrained control problem. They use an extension of Any Angle Safe Interval Path Planning Algorithm (AA-SIPP). AA-SIPP allows following not only edges that were initially present in the graph but also the newly build ones that represent the shortcuts. AA-SIPP is extended to AAt-SIPP to handle not only the translation moves but also the rotation (turn-in place) moves. In the studied problem, the robot is modelled as an open disk of radius r =0.5l, where l is the size of the grid cell, and the robot's action space includes wait in place, rotate in place, translate from one un-blocked cell to the other. Trajectory of a robot is a sequence of such actions. The dynamic obstacles are translating-and-rotating open disks of radii r and move in the same way as the robot, and the static obstacles are a set of blocked cells. The path planning problem is to find a collision free trajectory that is at each moment of time robot is at least r units away from the closest static obstacle(s) and at least 2r units away from the closest dynamic obstacle(s). For path planning, it is assumed that the robot accelerates/decelerates instantaneously. After the trajectory is planned, a path following problem is solved, constructing a control that will follow the prescribed trajectory. Supposing that a robot model is differentially flat, authors use a model based on Brunovsky normal form, which has constraints on maximum linear velocity and acceleration although these constraints were ignored at the path planning stage. Thus, the trajectory is refined considering these constraints. To make the refined trajectory close to the original one, it is assumed that the spatial movement on each segment of occurs in three stages: highest possible acceleration to required velocity, a uniform motion with constant speed and highest possible deceleration to a full stop. A 46 × 70 grid representing a warehouse-like environment was used in experiments. The size of each cell was 1m2 and the size of the robot and the dynamic obstacles was 0.5. Translation speed and rotation speed were 1 m/s and 180 degrees per second, respectively. 128 dynamic obstacles were moving on a grid. 100 different path finding instances were generated randomly. For the path-following algorithm the parameters such as the maximum velocity and maximum acceleration were set. Maximum velocity was set to 1 m/s as the same value was used for the path -planning algorithm. For the acceleration rate, three different values were tested.

Existing research also includes reduction-based solvers where multi-robot path finding problems are reduced to network flow models and combinatorial optimization problems and solved using the network flow algorithms from graph theory, mixed integer linear programming (MILP), answer set programming (ASP) and boolean satisfiability problem (SAT) solvers.

[143], [144], and [145] study the optimal multirobot path planning on graphs, using a special type of multi-flow network and integer linear programming. In these studies, they show how the problem of multi-agent path planning on collision-free unit-distance graphs (CUGs) can be reduced to network

flow problems and exploit the results from graph theory. They focus on a specific case of multi-agent path planning problem where the goals of agents are not pre-determined and obtain the paths while assigning each agent to a different goal, using an adapted version of maximum flow algorithm. Yu and LaValle [146] present near optimal solutions to multi-robot path planning problem on graphs.

[147] uses a time-space network (TSN) and mixed integer linear programming (MILP) to model the problem of dispatching and routing automated guided vehicles (AGVs) with vehicle and machine buffer capacities while avoiding conflicts.

[148] exploits the SAT solver to optimize the makespan of a sub-optimal solution for relatively small instances. [149] use Answer Set Programming (ASP) approach to solve multi agent path finding problems.

[149] present a study on improving the performance of reduction-based solvers for the problem of multi-agent pathfinding, using graph pruning strategies.

LA-MAPF generalizes MAPF to agents with different shapes and sizes. Each agent has a fixed shape around a reference point and can occupy multiple vertices at the same time. A vertex conflict happens when the shapes of two agents overlap at some timestep, and an edge conflict happens when the shapes of two agents overlap at some time when they move to their respective next vertices. Multi constraint CBS (MC-CBS) ([150]) is a state-of-the-art optimal solver for LA-MAPF. Multi-Constraint CBS (MC-CBS) adds multiple constraints (instead of one constraint) for an agent when it generates a high-level search node. [151] improves both the success rate and runtime of MC-CBS by generalizing the mutex based symmetry breaking techniques to LA-MAPF and proposing a new a mutex-based conflict selection strategy (MC-CBS-MS). [150] embed a procedure to the well-known MAPF algorithm PUSH and ROTATE enabling it to solve MAPF considering large ages, which is able to find solutions for non-trivial instances. The proposed procedure is called P&R-LA. [152] show how the problem of MAPF for large agents can be reduced to pebble motion on (general) graph. The procedure moves away the agents away from the edge which is needed to perform a move action of the current agent. More MAPF instances with large agents on arbitrary non-planar graphs (roadmaps) were solved compared to the state-of-the-art MAPF solver–Continuous Conflict-Based Search (CCBS) [153].

A metaheuristic that has wide applications for path and motion planning is particle swarm optimization (PSO). Particle swarm optimization (PSO) is used by [154], for robot path planning in dynamic environments. Obstacles of different shapes (convex, concave and curved) with varying velocities are considered. [155] combine Particle Swarm Optimization with Tabu Search for autonomous mobile robot path planning. Other studies include path planning of mobile robots based on specialized genetic algorithm and improved particle swarm optimization ([156]), hybrid multi-objective bare bones particle swarm optimization for solving the three-objective robot path optimization model where the objectives are path length, smoothness and safety of path ([157]), second-order oscillating particle swarm optimization algorithm for mobile robot path planning with complex constraints ([158]), motion planners inspired by particle swarm optimization to generate conflict free paths ([159]), local and global path planner using particle swarm optimization to find conflict-free paths ([160]), path planning using PSO based on grid network ([161]). [162] use particle swarm optimization for path planning of UAVs in three-dimensional space, where UAV flight must consider multiple factors such as altitude, terrain, and obstacles.

Multi-Agent Motion Planning (MAMP) is the task of finding conflict-free kino-dynamically feasible plans for agents from start to goal states.

Robots use motion planning algorithms to plan their trajectories both at global and local level. One of the widely used robot architectures for autonomous robots is the hybrid deliberative/reactive architecture ([163], [164]), which uses the deliberative layer and the reactive layer to realize high-level long-term planning and local reactive planning, respectively. A typical example is where the maps of the environment are constructed using information from sensors like the light detection and ranging (LIDAR), high-level paths are planned by using the algorithms such as A*, and reactive strategies for speed control or local planning are used to cope with dynamic and uncertain scenarios. High-level planning, local planning or instant reactions are evaluated by the behaviour manager to generate a better combined planning ([115]).

[165] define the problem of multi-agent cooperative motion planning using Signal Temporal Logic (STL) specifications, where robots can have nonlinear and nonholonomic dynamics. Authors claim that existing methods that are based on discrete abstractions and model predictive control (MPC) for motion planning are not scalable. [165] suggest timed waypoints to abstract nonlinear behaviours of the system as safety envelopes around the reference path defined by those waypoints. They encode the search for the waypoints which satisfy the STL requirements as a mixed integer linear program (MILP). The automatic task and motion planning according to high-level specifications is expected in an intelligent and autonomous robotic system. It is not straightforward to directly derive a specific sequence of locations to visit for each agent from these high-level specifications. Temporal Logic (TL), especially Signal Temporal Logic (STL) provides a mathematically precise language for specifying tasks and rules over continuous signals with explicit time semantics. Two approaches for motion planning from TL specifications are discrete abstractions and MPC. Abstraction-based methods discretize the state space and generate an abstract graph to perform the motion planning. MPC methods discretize the trajectory with a fixed timestep, and the states at each timestep are viewed as the decision variables of an optimization problem. The disadvantage of abstraction methods is that the number of abstracted states can grow exponentially. Also, the graph generation requires domain knowledge. Similarly, for MPC-based methods, the number of required timesteps might be too large for long-horizon planning. [165] use piece-wise linear (PWL) reference paths, which are sequences of timestamped waypoints, to handle more expressive STL specifications. The constraints are recursively encoded over the timestamped waypoints. Also, to determine the tasks that are to be completed by a group of cooperative agents, the multi-agent STL is defined, and subtasks are automatically assigned to each agent such that they cooperate without colliding. The encoded constraints are linear because of the PWL structure. Thus, optimal solutions can be found using MILP.

[166] present a scalable and effective multi-agent safe motion planner (S2M2) that enables a group of agents to move to their desired locations while avoiding collisions with obstacles and other agents, with the presence of rich obstacles, high-dimensional, nonlinear, nonholonomic dynamics, actuation limits, and disturbances. They address this problem by finding a piecewise linear path for each agent such that the actual trajectories following these paths are guaranteed to satisfy the reach-and-avoid requirement. The spatial tracking error of the actual trajectories of the controlled agents can be precomputed for any qualified path that considers the minimum duration of each path segment due to actuation limits. Using these bounds, a collision-free path for each agent is found by solving Mixed Integer-Linear Programs and agents are coordinated using the priority-based search. They demonstrate the method by benchmarking in 2D and 3D scenarios with ground vehicles and

quadrotors, respectively, and show improvements over the solving time and the solution quality compared to two state-of-the-art multi-agent motion planners, ECBS-CT ([167]) and MAPF/C+POST ([168]).

ECBS-CT is a generalization of ECBS for the MAMP problem. In the high-level search, it takes a problem instance and a suboptimality bound w ≥ 1 as input and it generates a solution with a cost which is not higher than w times the optimal cost. Thus, it generates optimal or bounded suboptimal solutions. The low-level search uses SCIPP, which is developed by [167] and is a generalization of SIPP that is suitable for focal search. ECBS-CT solve the MAMP problem in the state lattice world representation. State lattices ([169]) are extensions of grids that are able to model motion constraints and suitable for planning non-holonomic and highly constrained agents with limited manoeuvrability. A state lattice is constructed by discretizing the configuration space into a high-dimensional grid and connecting the cells of grid with motion primitives. A motion primitive models kino-dynamically feasible actions of the agent. A state in a lattice is a tuple of the form (x, y, z, θ, v, …), where x, y, z are the coordinates of the agent's centre, θ is the orientation, v is the velocity, … etc. An edge in a state lattice is associated with the duration and a list of cells swept by the agent to execute a motion ([167]).

MAPF/C+POST is a method which is used by ([168]) for multirobot trajectory planning in known, obstacle-rich environments. They perform this solution approach on a quadrotor swarm navigating in a warehouse setting. First a roadmap generation procedure, which generates sparse roadmaps annotated with possible interrobot collisions, is used. Later, valid execution schedules are found in discrete time and space, using discrete planning. Finally, smooth trajectories are created using continuous refinement. Safe and smooth trajectories for a high number of quadrotors in dense environments with obstacles are computed in a short time.

Multi-Agent Motion Planning (MAMP) is the problem of computing feasible paths for a set of agents each with individual start and goal states within a continuous state space. By extending the optimal MAPF technique, Conflict-Based Search (CBS), to continuous state spaces, [126] propose an efficient and scalable MAMP solver, CBS-MP. They compare the suggested solver with standard coupled and decoupled Probabilistic Roadmap (PRM) variants and ECBS-MP, another CBS extension to solve MAMP problems.

Multi-agent motion planning (MAMP) is a critical challenge in applications such as connected autonomous vehicles and multi-robot systems. [170] model the problem of coordination of connected self-driving vehicles as MAMP and formulate the problem using a novel, flexible sphere-based discretization for trajectories and propose a space-time conflict resolution approach adhering to kinematic constraints. They use a depth-first conflict search strategy to improve scalability and compare the results with state-of-the-art solvers.

[126] presents an overview of some of the state-of-the-art MAPF and MAMP solvers in Table 1. MAMP is a superset of MAPF. in coupled approaches, all agent paths are computed in unison. These approaches work in the joint space of all agent states. They tend to provide stronger guarantees on feasible paths and minimum cost by exploring the joint space. However, they have a high computational cost. Decoupled approaches work in single-agent spaces allowing to rapidly compute feasible paths for problems with many agents. However, individual agent state spaces are explored in isolation, and later solutions are combined. This prevents ensuring completeness and optimality. Due to the trade-off between faster computation times and finding optimal cost solutions, hybrid approaches are used to leverage the strengths of both coupled and decoupled techniques. For

example, a hybrid MAPF method, M*, solves the MAPF problem by initially planning a set of individual policies in a fully decoupled manner. These policies are then used to guide a coupled search over the joint state space. When an inter-agent conflict arises, the coupled search is backtracked until the last collision-free joint state, and the conflicting agents are merged into a coupled meta-agent. New collision-free paths are computed using a coupled planner for the meta-agent. If all agents are in collision at the same place and time, M* may become a fully coupled planner as long as the inter-robot conflict remains unresolved. In motion planning, the state space is the set of all possible agent configurations known as the configuration space. In response to the complexity of motion planning, sampling-based motion planners were developed as an efficient means of discovering valid paths in the configuration space. These methods, such as the Probabilistic Roadmap Method (PRM) attempt to create a roadmap, or graph, approximating the configuration space. Paths are found by querying this roadmap. RRT is another sampling-based motion planning algorithm and MRdRRT is an RRT-based technique. ECBS-CT aims to solve the MAMP problem in the state lattice world representation, where the workspace is discretized into a grid, and then grid cells are connected using a predefined set of single agent motion primitives. It leverages using a state-lattice representation to map the agents' motions to a common workspace discretization. Thus, all the agents' motions can be incorporated into the same state-space representation.

Table 1 : An overview of the state-of-the-art MAPF and MAMP solvers ([126])

Algorithm	MAPF/MAMP	Coordination	Optimal	State representation
Composite-A* [125]	MAPF	Coupled	Yes	Grid
Decoupled-A* [171]	MAPF	Decoupled		Roadmap
CBS [112]	MAPF	Hybrid	Yes	Grid
MA – CBS [81]	MAPF	Hybrid	Yes	Grid
ECBS [116]	MAPF	Hybrid	Yes	Grid
M* [132]	MAPF	Hybrid	Yes	Grid
MRdRRT [172]	MAMP	Coupled	Yes	composite roadmap
Composite-PRM [173]	MAMP	Coupled	Yes	composite roadmap
Decoupled-PRM [173]	MAMP	Decoupled		roadmap
MRP-IC [174]	MAMP	Decoupled		composite roadmap
ECBS-CT [167]	MAMP	Hybrid	Yes	state-lattice
CBS-MP [126]	MAMP	Hybrid	Yes	roadmap

4.2.3 Recent advances in path and motion planning

Recent applications of A* include minimum dose path planning based on navigation mash ([175]), to avoid radiation in large and complex radiation environments. [176] solve the path planning problem of the automatic guided vehicle (AGV) sorting system on a mash topology map, using a two-stage algorithm. [177] apply several recently developed MAPF solution approach to the 3D Pipe Routing (PR) problem, which aims at placing collision free pipes from given start locations to given goal locations in a known 3D environment. Accordingly, a solution to a MAPF instance is a set of blocked cells in x-y-t space, while a solution to the corresponding PR instance is a set of blocked cells in x-y-z space.

A considerable amount of the recent literature includes the solvers dedicated to handling the combined problem of path finding and target or task assignment for the agents, or the problems where single agents have multiple goal locations and a sequencing of these is also needed. These are summarized in the following paragraphs.

[178] solve the combined Target-Assignment and Path-Finding problem (TAPF) which requires simultaneously assigning targets to agents and planning collision-free paths for agents from their start locations to their assigned targets. Instead of the Conflict-Based Search with Target Assignment (CBS-TA) which uses K-best target assignments to create multiple search trees and Conflict-Based Search (CBS) to resolve collisions in each search tree, [178] propose Incremental Target Assignment CBS (ITA-CBS) to avoid duplicated collision resolution in multiple trees and the expensive computation of K-best assignments. ITA-CBS generates only a single search tree and incrementally computes best assignments during search. Other variants of TAPF are presented by [179], [180], [181], [182], [183], [184], [185], and [186].

[179] adapt the Hungarian algorithm for solving the assignment problem with changing costs. [180] propose a novel approach called conflict-based Steiner search (CBSS) for solving MAPF in combination with Target-Sequencing which requires not only assigning targets to agents but also specifying the visiting order of targets. [181] deal with the problem of optimal target assignment and path finding for teams of agents by presenting the CBM (Conflict-Based Min-Cost-Flow) algorithm. On the low level, CBM uses a min-cost max-flow algorithm on a time-expanded network to assign all agents in a single team to targets and plan their paths. On the high level, CBM uses conflict-based search to resolve collisions among agents in different teams. [182] propose Task Conflict-Based Search (TCBS) algorithm to solve the combined delivery task allocation and path planning problem to optimality, which is to be used as a baseline for sub-optimal solvers. [185] introduces multi-goal multi agent path finding (MG-MAPF) problem. While the task in MAPF is to navigate agents in an undirected graph from their starting vertices to one individual goal vertex per agent, MG-MAPF assigns each agent multiple goal vertices and the task is to visit each of them at least once. To solve MG-MAPF, [185] suggests two novel algorithms: a heuristic search-based algorithm called Hamiltonian-CBS (HCBS) and a compilation-based algorithm built using the satisfiability modulo theories (SMT), called SMT-Hamiltonian-CBS (SMT-HCBS). [183] study the multi-goal task assignment and path finding (MG-TAPF) problem whereas many tasks as agents are given, and each task consists of a sequence of goal locations. Tasks have to be assigned to agents and each agent must follow the sequence of goal locations of the assigned task. The aim is to find collision-free paths to minimize flow time. Authors prove that the problem is NP-hard using a reduction from a specialized version of the Boolean satisfiability problem to the MG-TAPF problem and propose the Conflict-Based Search with Task Assignment with Multi-Label A* algorithm (CBS-TA-MLA) that solves the problem to optimality. The algorithm uses the best first search CBS-TA on the high level to assign tasks and resolve conflicts, and multi-label A*, MLA ([187]), on the low level to find the time-optimal path of each agent that visits a sequence of goal locations of its assigned task. They also extend CBS-TA-MLA to a bounded-suboptimal version, called ECBS-TA-MLA, using ideas from the bounded suboptimal version of CBS. [184] use Answer Set Programming for generalized target assignment and path planning problem. [186] analyze the problem of allocating and sequencing goals for each agent while simultaneously producing conflict-free paths for the agents. They introduce an

exact algorithm called MS* which computes an optimal solution by fusing and advancing state of the art solvers for multi-agent path finding (MAPF) and multiple travelling salesman problem (mTSP).

4.2.4 Comparison of solvers

In this section, the state-of-the-art solvers are classified as (i) optimal and complete, (ii) bounded suboptimal and complete, (iii) unbounded suboptimal with no completeness guarantee, (iv) complete and non-optimal, and (v) SIPP variants which might contain the characteristics of any of the former groups, and evaluated in terms of performance, solution quality and completeness.

Optimal MAPF solvers can be divided into four categories: A*-based, increasing cost tree search (ICTS) based, conflict-based search (CBS)-based, and reduction-based. The optimal solver also satisfies completeness.

The reduction-based solvers are optimal and complete. For small-scale MAPF problems with dense obstacles and agents, the reduction-based solver can solve the problem quickly. The difficulty for a reduction-based solver is proof of the correctness of the reduction process, which usually requires complex mathematical reasoning.

Optimal A* based solvers perform search in the k-agent state space. The drawback is that as the map size and the number of agents increases, the state space grows. All successor nodes are added to the OPEN list, regardless of whether they will be expanded. Both the joint state space and the joint branching factor grow exponentially as the number of agents increases. Thus, computational cost is high, and scalability is limited. M* is an improved version of optimal A* based solver which scales better than A* since joint branching factors are established only between conflicting agents. Unlike A*, M* does not need to add every neighbour to the OPEN list. M* initially uses decoupled planning to generate a low-dimensional search space. As robot-robot collision are found in the search space, the local dimensionality of the space is locally increased. When there is no conflict between agents, the state space is expanded to only one node every timestep, which contains the optimal actions of all single agents. For agents in conflict, the state space will generate all action combinations for them and combine them with the optimal actions of other agents. M* is proven to be complete and optimal. The worst-case computational cost of M* grows exponentially with the number of robots, however M* requires less time than A* to find paths for multirobot systems.

In the two-level search framework of ICTS, the high-level searches a tree with the exact path cost for each agent, while the low-level verifies to see whether there is a solution on each ICT node. If there exists a subset of m agents for which no valid solution exists, the low-level can immediately terminate. Although ICTS is faster than A*-based approach, it still works on the k-agent state space, which grows exponentially with the number of ICT levels. It is not efficient if the instance contains dense obstacles or agents.

When the agent density is relatively sparse, CBS can solve large-scale problems to optimality. In most of the instances, CBS performs better than ICTS and A*. However, in some instances with many path conflicts, it is worse than A∗-based solvers. As the number of agents increases, path conflicts increase rapidly, and solution efficiency decreases.

MA-CBS reduces the number of nodes in the constraint tree, by merging the agents into meta-agents when number of conflicts between them exceeds a given value and uses the A*-based MAPF solver to plan the path for the meta-agent at the low level, to speed up the search. ICBS applies Merge & Restart (MR) strategy that suggests re-establishing a root node to start searching after merging agents unlike the MA-CBS, which keeps expanding the constraint tree nodes. MR can save a significant amount of computing cost.

Optimal and complete MAPF solvers are given in Table 2 :

Table 2 : Optimal and complete MAPF solvers

Solver	Description	Search strategy	Computational time	Scalability
A* (coupled)	A* based	Search in joint state space of all agents	Exponential	Limited
M*	A* based	State space is reduced compared to A*	Exponential	Better than A*
ICTS	Two-level	Search in joint state space of all agents	Exponential	Significantly better than A*
CBS	Two-level	Binary tree search based on conflicts on the high level, fast single agent planner on the low level	Exponential	Significantly better than A* and ICTS for most of the instances.
MA-CBS	CBS based	Agents are merged into meta-agents, thus, the number of nodes in the search tree is reduced compared to CBS.	Exponential	Better than CBS
ICBS	CBS based	Prioritization of conflicts and Merge & Restart strategy speed up search	Exponential	Better than CBS and MA-CBS
MILP, SAT, ASP	Reduction based	Reduction to MILP, SAT, ASP	Exponential	Small scale problems with dense obstacles and agents

Bounded sub-optimal solvers can give some guarantee of the quality of the solution. Bounded sub-optimal solvers are generally derived from optimal MAPF solvers. Bounded sub-optimal A* based solvers trade-off between optimality and search efficiency using inflated heuristics. Optimal A* based solvers can all be transformed into bounded sub-optimal solvers by introducing a sub-optimality factor (inflation rate). The dynamic potential search (DPS), which is a special case of focal search, is also an A*based bounded sub-optimal solver. All Agent Costs (AAC), which is a bounded sub-optimal variant of ICTS, increases the cost of all agents by one at each subsequent node in the search tree, while ICTS only increments the cost of a single agent from a parent to a child in the increasing cost tree. CBS-based bounded sub-optimal solvers include BCBS and ECBS, which use focal search in both levels of CBS, reducing the number of collisions to be solved. ECBS guarantees bounded sub-optimal costs for each path in each node of the constraint tree. MA-ECBS reduces the search space of ECBS, using meta-agents. However, a joint-state-space MAPF solver makes resolving collisions within meta-agents inefficient. NECBS overcomes this resolving the collisions within meta-agents with ECBS. NECBS is also a complete and bounded suboptimal solver.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

Table 3 presents a brief comparison of bounded sub-optimal and complete solvers:

Table 3 : Bounded sub-optimal and complete MAPF of solvers

Solver	Description	Search strategy	Computational time	Scalability
Inflated M*	A* based.	Heuristic search where deviation from optimality is bounded	Lower than M*.	Higher when sub-optimality bound is larger, in which case solution quality is lower
DPS	A* based	A special case of focal search where the nodes in the OPEN list are expanded based on dynamic potential value function that includes a sub-optimality factor.	Lower than A*	Higher than A*
AAC	ICTS based	Increases the cost of all agents by one at each subsequent node in the search tree.	Lower than ICTS	Higher than ICTS
BCBS	CBS based	Focal search is applied in both levels of CBS. Suboptimlity bound is the product of the bounds of the two levels.	Lower than CBS	Higher than CBS
ECBS	CBS based	Focal search is applied in both levels of CBS. High and low levels share a joint suboptimality bound.	Lower than CBS	Higher than CBS. More than 50% of the instances on DAO maps with 250 agents were solved with 1% optimality gap, while CBS can perform well up to 50 agents.
NECBS	CBS based	Meta-agents applied to ECBS (MA-ECBS). ECBS is also used to resolve collisions in the joint state space within meta-agents to speed up MA-ECBS	Lower than ECBS	Higher success rate than ECBS and its variants

Unbounded sub-optimal MAPF solvers can get solutions faster and generally have a higher success rate, which include search-based, sampling-based, rule-based solvers. Search based solvers include priority-based decoupled search solvers, where the priorities are used to resolve conflicts between the agents' independent paths. Sampling-based algorithms randomly sample a fixed workspace to generate sub-optimal paths. Rule-based algorithms have agent-specific rules in place for different scenarios. They usually do not include a massive search like search-based algorithms. Rule-based solvers usually guarantee to find a solution very fast, but those solutions are in most cases far from optimal.

The search strategy, scalability and performance of unbounded sub-optimal MAPF solvers with no completeness guarantee are summarized and compared in Table 4:

Table 4 : Unbounded sub-optimal MAPF solvers without completeness guarantee

Solver	Description	Search strategy	Computational time	Scalabillity
LRA*	A* based local repair	Each agent searches for a route using the A* algorithm, ignoring all other agents except for its current neighbours. Agents follow their routes until a collision is imminent and replan the remainder of the route.	Faster than optimal and bounded suboptimal solvers.	Higher than optimal and bounded suboptimal solvers. Path lengths are more than twice the optimal lower bound for 100 agents.
CA*	A* based (decoupled)	For each agent the search is performed in 3D space that includes a wait move, while the planned routes of other agents that are stored in a reservation table are avoided.	Faster than optimal and bounded suboptimal solvers. Slower to initialize compared to LRA*.	Higher than optimal and bounded suboptimal solvers. Solution quality is better than LRA* with 20% deviation from optimal lower bound. The order of agents might affect the solution quality, which might be dealt using Prioritized Planning.
HCA*	A* based (decoupled)	Improves performance using a heuristic	Faster than optimal and bounded suboptimal solvers. Slower to initialize compared to LRA*. Slightly faster than CA*.	Higher than optimal and bounded suboptimal solvers. Solution quality is better than LRA* with 20% deviation from optimal lower bound.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

WHCA*	A* based MAPF solver. Priority based search.	Dynamic windows that limit the space-time search to a fixed depth.	Lower than 0.6ms per agent. Faster than CA* and HCA*. Performance depends on window size. With large window, initialization time increase and behaves like HCA*. With small window size, behaves like LRA* with lower solution quality.	Suitable for real-time use.
CBSw/P	An adaptation of CBS. Priority based search.	The whole prioritization space is explored using best-first search.	More efficient than CBS, better solution quality compared to CA*, HCA*, WHCA*. Obtains optimal and near optimal solutions.	Higher than the optimal solver CBS and usually finds optimal or near-optimal solutions.
PBS	Priority based search	The whole prioritization space is explored using depth-first search.	More efficient than CBSw/P. Solves well-formed Instances with six hundred agents in less than a minute. Finds solutions for many instances where standard prioritized algorithms cannot.	Remains near optimal and efficient for more than one hundred agents.
Sampling-based solvers	A* based search is replaced with sampling	The joint state space is searched using sampling methods	More efficient than A* based search. Exploring the joint state space is not efficient for large scale instances.	Small scale instances.
Rule based solvers	Agent-specific rules	Based on specific rules with no extensive search	Very fast	High scalability with low solution quality

BIBOX and PUSH AND ROTATE in Table 5 are reduction-based and rule-based solvers for which completeness is guaranteed for the class of instances with two unoccupied locations in biconnected graphs. Solutions are generated in polynomial time and non-optimal.

Table 5 : Complete non-optimal solvers

Solver	Description	Search strategy	Computational time	Scalability
PUSH AND ROTATE	Rule based solver.	Special movement rules are used in search	Fast and complete with large deviation from optimality.	Solves large-scale instances with low solution quality.
BIBOX	Reduction to pebble motion	The problem is reduced to pebble motion problem and solved using a polynomial algorithm.	Fast and complete for bi-connected graphs with two unoccupied vertices.	Scales well in highly connected 2D and 3D space.

We compare SIPP variants in Table 6, even though they belong to different classes in terms of optimality and completeness.

Table 6 : SIPP variants

Solver	Description	Optimality and completeness	Computational time and scalability
SIPP	Single agent path planner with dynamic obstacles.	Optimal complete.	Outperforms HCA* in terms of computational time and success rate. The search space is reduced compared to A* since continuous time intervals are used rather than discrete timesteps.
WSIPP	Weighted SIPP speeds up SIPP by sacrificing solution quality	Bounded suboptimal, complete	Outperforms SIPP in terms of computational time and scalability. Solution quality is lower than SIPP.
$WSIPP_d$	Improves the search strategy of Weighted SIPP	Bounded suboptimal, complete	Outperforms SIPP
$WSIPP_r$	Improves the search strategy of $WSIPP_d$	Bounded suboptimal, complete	Outperforms SIPP
Focal SIPP	Applies focal search	Bounded suboptimal, complete	Outperforms SIPP
ASIPP	Any time SIPP. Each obstacle is treated as a sphere with a radius and a trajectory.	Bounded suboptimal, complete	Large maps with 50 dynamic obstacles are solved in short time. Finds an initial solution quickly. Works well in dynamic environments.
AA-SIPP(m)	Decoupled prioritized planner that applies Any Angle SIPP to multiple agents.	Complete under well-defined conditions. No optimality guarantee	Significantly better than decoupled SIPP(m) (SIPP for multiple agents) and the coupled CBS based solvers ICBS and ECBS. Success rate is higher than 97%.
AAt-SIPP + Constraint path following control	Path and motion planning. Handles not only the translation moves but also the rotation (turn-in place) moves. The robot is modelled as an open disk of radius r =0.5l, where l is the size of the grid cell. Translation and rotation velocities are also considered.	Obtained trajectories are not always collision-free	Performed on 46 × 70 grid with 128 dynamic obstacles.

Motion planning is the extension of path planning. Path planning aims at finding the path between the origin and destination in workspace by strategies like shortest distance or shortest time, therefore path is planned from the global metric or topological level. Motion planning, however, aims at generating interactive trajectories in workspace when robots interact with dynamic environment, therefore motion planning needs to consider kinetics features, velocities and poses of robots and dynamic objects nearby when robots move towards the goal. On one hand, motion planning must consider short-term optimal or suboptimal reactive strategies to make instant or reactive response. This is achieved by rotary or linear control in hardware from the perspective of robotic and control engineering. On the other hand, motion planning should achieve long-term optimal planning goals as path planning when robots interact with the environment [115].

[115] classify the traditional motion planning algorithms as graph search algorithms, sampling-based algorithms, interpolating curve algorithms, and reaction-based algorithms. *Graph search algorithms* include the algorithms based on depth first search, best first search, breadth first search, such as Dijkstra, A*. *Sampling based algorithms*, the RRT and the probabilistic roadmap method (PRM), are two algorithms that are commonly utilized in motion planning. The RRT constructs a tree that attempts to explore the workspace rapidly and uniformly via a random search. The RRT algorithm can consider non-holonomic constraints, such as the maximum turning radius and momentum of the vehicle. The PRM algorithm is normally used in a static scenario. It is divided into two phases: learning phase and query phase. In the learning phase, a collision-free probabilistic roadmap is constructed and stored as a graph. In query phase, a path that connects original and targeted nodes is searched from the probabilistic roadmap. *Interpolating curve algorithms* use a set of mathematical rules to draw trajectories. Mathematical rules are used for path smoothing and curve generation. Typical path smoothing and curve generation rules include line and circle, clothoid curves, polynomial curves, Bezier curves and spline curves. *Reaction-based algorithms* are about making reactions or doing local path planning quickly and intuitively, rather than searching global solutions. Examples of reaction-based algorithms are potential field method (PFM) which uses vectors to represent behaviours and combine vectors to produce an emergent behaviour, velocity obstacle method (VOM) which relies on current positions and velocities of robots and obstacles to compute a reachable avoidance velocity space (RAV), and selecting a proper avoidance maneuverer (velocity) to avoid static and moving obstacles, and DWA which is about is about choosing a proper translational and rotational velocity (v, w) that will maximize an objective function that includes forward velocity of the robot, distance to the next obstacle on the trajectory and a measure of progress towards a goal location. Disadvantages of PFM include oscillation of motion when robots navigate among very close obstacles at high speed, impossibility to go through small openings. Collisions with obstacles still exist when using velocity obstacle method in complex scenarios like dense and dynamic cases. In addition to traditional motion planning algorithms*, classical machine learning algorithms* that are used for motion planning are listed by [115] as three supervised learning algorithms, SVM, LSTM, CNN, and a reinforcement learning algorithm, Monte-Carlo tree search (MCTS). [115] present the key characteristics of traditional and learning based motion planning algorithms in Table 7 and Table 8.

Table 7 : Traditional motion planning algorithms

Classification	Example	Input	Key features	Output
Graph search alg.	Dijkstra (1)	Graph or map.	Best-first search (1)	Trajectory
	A* (1), (2)		Heuristic function for cost estimation (2)	
	PRM (1)		Random search (1)	

EUROPEAN PARTNERSHIP
Co-funded by the European Union

<table>
<tr><td>Sampling based alg.</td><td>RRT (1), (2)</td><td rowspan="6"></td><td>Non-holonomic constraint (2)</td><td rowspan="6"></td></tr>
<tr><td rowspan="5">Interpolating curve alg.</td><td>Line and circle</td><td rowspan="5">Mathematical rules
Path smoothing</td></tr>
<tr><td>Clothoid curves</td></tr>
<tr><td>Polynomial curves</td></tr>
<tr><td>Bezier curves</td></tr>
<tr><td>Spline curves</td></tr>
<tr><td rowspan="3">Reaction based alg.</td><td>PFM</td><td>Robot configurations i.e. position</td><td>Different potential field functions U for different targets, i.e. goal, obstacle</td><td>Moving directions</td></tr>
<tr><td>VOM</td><td>Positions and velocities (robot and obstacles)</td><td>Exhaustive/global, heuristic search w.r.t. U</td><td rowspan="2">Selected velocity</td></tr>
<tr><td>DWA</td><td>Robot's position, distances to goals/obstacles and kinematics of robot</td><td>Velocity selection according to objective U</td></tr>
</table>

Table 8 : ML algorithms

Algorithm	Input	Key features	Output
MSVM	Vector	Maximum margin classifier	None-sequential actions
LSTM	Vector	Cell (stack structure)	Time-sequential actions
MCTS	Vector	Monte-Carlo method/Tree structure	Time-sequential actions
CNN	Image	Convolutional layers/ Weight matrix	None-sequential actions

Analytical comparison of traditional and learning based path and motion planning algorithms are presented by [115] in Table 9. Accordingly, graph search algorithms plan their path globally by search methods (e.g., depth-first search, best-first search) to obtain a collision-free trajectory on the graph or map. Sampling-based algorithms samples local or global workspace by sampling methods (e.g., random tree) to find collision-free trajectories. Interpolating curve algorithms draw fixed and short trajectories by mathematical rules to avoid local obstacles. Reaction based algorithms plan local paths or reactive actions according to their objective functions. MSVM and CNN make one-step prediction by trained classifiers to decide their local motions. LSTM and MCTS can make time-sequential motion planning from the start to destination by performing their trained models. Velocity criterion denotes the ability to tune the velocity when algorithms plan the paths, and safe distance criterion denotes the ability to keep a safe distance to obstacles.

Table 9 : Analytical comparison of traditional and ML algorithms for motion planning

Algorithm	Local/global planning	Path length	Velocity	Reaction speed	Safe distance	Time seq. path

Graph search	Global	Optimal	-	Slow	Fixed distance/high collision rate	No
Sampling based	Local/Global	Suboptimal	-	Slow	Fixed distance/high collision rate	No
Interpolating curve	Local	Fixed	-	Medium	Fixed distance	No
Reaction based	Local	Optimal	Optimal	Medium	Suboptimal distance	No
MSVM	Local	Suboptimal	Suboptimal	Fast	Suboptimal distance	No
LSTM	Local/Global	Suboptimal	Suboptimal	Fast	Suboptimal distance	Yes
MCTS	Local/Global	Optimal		Fast	Optimal distance	Yes
CNN	Local	Suboptimal	Suboptimal	Fast	Suboptimal distance	No

All things considered, for path planning, compared to optimal solvers, bounded sub-optimal solvers perform better in terms of computational time, while slightly decreasing the solution quality. On the other hand, unbounded suboptimal solvers generate solutions much faster than optimal and sub-optimal solvers, however the completeness of the obtained solutions are not always guaranteed even though a feasible solution exists. Thus, the trade-offs between solution quality, completeness and computational complexity should be considered while selecting the best solver. An interesting fact is that despite being optimal and complete, the single agent path planning solver, SIPP, is faster than some of the unbounded sub-optimal solvers with no completeness guarantees. For multi-agent path planning, when priority-based search is combined with the solvers such as CBS, larger scale instances can be solved with better solution quality compared to other unbounded sub-optimal solvers such as CA*, HCA*, WHCA*. One example is CBSw/P which usually finds optimal or near-optimal solutions, even though it is classified as an unbounded sub-optimal solver. A similar and improved solver is PBS, which solves well-formed Instances with six hundred agents in less than a minute, finds solutions for many instances where standard prioritized algorithms cannot, and remains near optimal and efficient for more than a hundred agents.

The state-of-the-art multi-agent motion planner, ECBS-CT, generates optimal or bounded suboptimal solutions. In the high-level search, it takes a problem instance and a suboptimality bound as input. The low-level search uses SCIPP, which is a generalization of SIPP that is suitable for focal search. A scalable and effective multi-agent safe motion planner is S2M that enables a group of agents to move to their desired locations while avoiding collisions with obstacles and other agents, with the presence of rich obstacles, high-dimensional, nonlinear, nonholonomic dynamics, actuation limits, and disturbances. A piecewise linear path is obtained for each agent such that the actual trajectories following these paths are guaranteed to satisfy the reach-and-avoid requirement. a collision-free path for each agent is found by solving Mixed Integer-Linear Programs and agents are coordinated using the priority-based search. S2M shows improvements over the solving time and the solution quality compared to two state-of-the-art multi-agent motion planners, ECBS-CT, in 2D and 3D scenarios with ground vehicles and quadrotors.

In ASTAIR, the goal is to generate realistic solutions in short computational time so that disruptions or changes in environmental conditions are addressed on time. Thus, rather than optimal and complex solvers, both high quality and efficient solvers are aimed to be integrated with human-machine interface to aid the planning and execution of instant changes. The problems are to be solved on complex airport surface layouts that are to be converted into graphs including nodes, edges, intersections. Thus, application of fast solution procedures becomes crucial for airport surface operations.

For path and motion planning, considering its efficiency and high-quality solutions in many of the instances, priority-based search, PBS, will be one of the tools that will be integrated with other approaches. One example of priority-based modelling for airport surface movements exists in the literature where access priorities of aircraft for a road section are adjusted by decreasing the priority of delayed aircraft. In combination with PBS, safe interval path planning solver, SIPP, is worth considering due to its computational efficiency, optimality and completeness. In addition to path finding, SIPP has also been combined with path tracking or following algorithms for motion planning or adapted to deal with agents with different shapes and sizes. Thus, combining SIPP with PBS and local level motion planning approaches to find trajectories by considering the speed profiles and other kino-dynamic constraints is a promising approach for dealing with path and motion planning problems in ASTAIR. The multi-agent safe motion planner, S2M, which combines mixed integer programming with priority-based search is also worth considering due to its scalability and some of its procedures could be integrated into the path and motion planning solutions of ASTAIR. Apart from these, the recent trend of combining path planning with task allocation or simultaneous target assignment and sequencing fits well to the scope of ASTAIR, where finding a tug allocation solution or dynamic assignment of tugs to aircraft while handling the path and motion planning at the same time is among the main interests.

4.2.5 Explainable AI for path and motion planning

In many of the safety critical applications (e.g., air traffic control, hazardous materials), planning is not fully automatic, and the plan is only suggested to a human supervisor, who may act upon it. In such settings, the plan has to be presented to the supervisor in a humanly understandable manner. In particular, the presentation should enable the supervisor to understand the paths taken by the agents, and to easily verify that the agents do not collide, as otherwise the supervisor would not necessarily trust the plan. Such a representation is called an explanation of the plan [188].

[188] propose an explanation scheme, *vertex-disjoint decompositions*, for MAPF, which bases explanations on simplicity of visual verification by human's cognitive process. The scheme decomposes a plan into segments such that within each segment, the paths of the agents are disjoint. The simplicity of a plan is measured by the number of segments required for the decomposition. Authors present a formal definition of the explanation scheme as follows: "An explanation scheme for a decision problem *P* is a mechanism that outputs, for a given input *I,* some information called an explanation, or outputs that no explanation is found". Accordingly, the following statements define the three properties of the explanation scheme: (i) If an explanation exists, then *I* is a *yes-instance*, i.e. I ∈ P (*Soundness*), (ii) If *I* is a *yes-instance*, then an explanation exists (*Completeness*), (iii) An explanation is easy to find and to verify if it exists (*Simplicity*). The *simplicity* requirement is context-dependent and not formal. The *soundness* and the *completeness* are the key requirements for the proof. The complexity of the problems that arise by the explanation scheme are studied and it is shown that finding optimal explanations for existing plans can be done efficiently, whereas planning for MAPF problems with simple explanations is NP-Complete. Additionally, the tradeoff between time-optimal plans and plans with simple explanations is analyzed. Experiments are performed in both continuous and discrete

settings. Furthermore, the practical difficulties that arise in implementing a search-based algorithm for planning with explanations are demonstrated.

Another study that addresses the Explainable Multi-Robot Motion Planning via disjoint decomposition is presented by [189]. They show that standard notions of optimality may create conflict with short explanations, and propose meta-algorithms, namely multi-agent plan segmenting-X (MAPS-X) and its lazy variant, that can be plugged on existing centralized sampling-based tree planners, represented by X, to produce plans with good explanations using a desirable number of images. We demonstrate the efficiency of the explanation scheme and evaluate the performance of MAPS-X and its lazy variant in various environments and agent dynamics. The study focuses on explanations for realistic robotic systems in the continuous space with kino-dynamical constraints. Explainability is treated as an additional concept on top of the multi-robot motion planning and incorporated into existing sampling-based algorithms. Due to the fact that there is often a trade-off between planning for short explanations and short paths, explainability might conflict with the state-of-the-art heuristics. To deal with this, generic meta-algorithms that search for optimally explainable plans using any centralized sampling-based algorithm are proposed. The performance of the proposed meta-algorithms is demonstrated by plugging them with classical motion planners such as rapidly exploring random trees (RRT).

[190] propose methods that generate explanations for the optimality of paths, focusing on the case of path planning on navigation meshes, which are heavily used in the computer game industry and robotics. The proposed methods are based on single inverse-shortest-paths optimization, and incrementally solving complex optimization problems. [190] show that scalability and performance of these methods are better than domain independent search-based methods. Although the domain-independent methods for Explainable AI such AS Model Reconciliation are also applicable to path planning, they lack the domain knowledge that would allow them to deal with large-scale path planning problems. Computation speed is a requirement for interactive interfaces such as human-in-the-loop designs, for safety-critical robots in dynamic environments, or when a speedy investigation of planner behaviour is desirable. [190] focus on explaining why a specific path is optimal rather than another path, unlike other studies for explainable multi-agent path finding which focus on explaining why the paths are not colliding or failures in motion planning. Inverse shortest path problem looks for a minimal change graph weights so that a desired path becomes optimal. Thus, it is a relevant problem to explanations of optimality of paths.

[191] investigate the explainability for multi-modal multi-agent path finding problem with resources (mMAPF), considering queries about the (in)feasibility and the optimality of solutions, as well as queries about the observations about these solutions. In real-world automated warehouses, the robots' battery levels change as they move around, and, in some parts of these warehouses, due to presence of humans or tight passages, the robots may need to move slowly to ensure safety. mMAPF is a general version of MAPF proposed by [191] to handle more realistic autonomous warehouse scenarios, considering multi-modal transportation, multiple objectives, resource constraints and waypoints. A flexible framework is proposed to solve the problem, using Answer Set Programming. Given a solution for mMAPF, the explainable framework is able to explain infeasibility or nonoptimality of the solution, confirm its feasibility and suggest alternatives, and provide explanations for queries. If a modified solution is found infeasible, then, an explanation regarding infeasibility of this modified solution can be "due to collisions with obstacles or other robots" or "due to low battery-level". An explanation regarding non-optimality can be "more time is needed to complete tasks" or "more

charging is required". If the modified solution is found feasible, alternative feasible solutions with better solution quality are obtained and returned to the engineer. Other queries may include why an agent is waiting too long at a location in which case the response can be "to avoid collision with another robot". The explainable framework is implemented using Python and the Answer Set Programming solver Clingo.

[192] provide a comprehensive outline of the different threads of work in Explainable AI Planning (XAIP). They present definitions and clarifications of the decision-making problem, explanation process, explanation artifacts, properties of explanations, algorithm-based explanations, model-based explanations such as Inference Reconciliation and Model Reconciliation, and plan-based explanations, as main concepts in XAI. They focus on automated planning as a subfield of decision-making problems. More specifically, [193] provide a taxonomy of concepts in the area of Interpretable Agent Behaviour. There has been significant interest in the robotics and planning community lately in developing algorithms that can generate behaviour of agents that is interpretable to the human (observer) in the loop. This notion of interpretability can be in terms of goals, plans or even rewards that the observer is able to ascribe to the agent based on observations of the latter. Interpretability remains a significant challenge in the design of human-aware AI agents. Authors introduce a general framework for describing problems in the space of "plan interpretability" and outline how existing works have addressed different aspects of this problems in cooperative settings. The planning problem, plan, computational model, completion function, observation model are formally defined and the concepts in cooperative settings which are relevant to motion planning are outlined and formulated based on the literature. These concepts are explicability, predictability, legibility and transparency. The concepts in adversarial setting which include privacy, plan-obfuscation, and security are also evaluated.

[194] introduce plan explicability and predictability for robot task planning so that intelligent robots can synthesize plans that are more comprehensible to humans. To achieve this, they must consider not only their own models but also the human's interpretation of their models. Humans understand agent plans by associating abstract tasks with agent actions (labelling). To compute the measures of explicability and predictability, [194] propose a model that learns the labelling scheme of humans for agent plans from training examples using conditional random fields (CRFs) and use the learned model to label a new plan. The measures of explicability and predictability are used by agents to proactively choose or directly synthesize plans that are more explicable and predictable to humans. The tests are performed on a synthetic domain with a physical robot.

[195] also focus on providing explanations for robot motion planning. Motion planners are traditionally not self-explanatory about their output. The result of running a motion planner is typically either a trajectory or a failure notice, so users may have problems understanding why a planner failed or why a trajectory is different from what was expected. However, notions of explanation in the existing motion planning literature are narrow. Thus, [195] introduce a new taxonomy of explanations in the context of motion planning and extend the concept to contrastive explanations and clarifications; propose methods for generating explanations and evaluate them on a user study; and elaborate on a comprehensive research agenda for explainable motion planning. Contrastive explanations explain why a trajectory A was returned by a planner, instead of a different trajectory B expected by the user. Optimization based and sampling based explainable motion planners which are capable of answering failure and contrastive questions are developed.

Explainable AI solutions for path and motion planning can be used in ASTAIR for explaining the motivations behind selecting certain paths over others to the users as part of human machine interactions and contributes to the concept of human-in-the-loop process in automation.

5 Conclusion and research directions

This deliverable presents the results of a comprehensive study on the state-of-the-art methodologies for Human-AI interaction, fleet management and path planning algorithms for operating a highly digitalised and automated airport that could be relevant to the ASTAIR project.

Based on this state-of-the-art, some research directions have been identified to be explored during the ASTAIR project that integrate the different aspects discussed in this document.

Regarding the Human-AI Interaction in ASTAIR, we will be targeting high levels of automation (Levels 2B and 3A according to the EASA's classification). Previous work on how to design efficient interactions for such high levels of automation is scarce and often studied within very narrow and controlled settings. According to previous literature, this specific context of AI-based system presents specific challenges related that we will have to consider during the project.

First, we will need to investigate the roles and tasks allocation between AI and Humans as well as to identify relevant criterions to validate such allocation. We need to identify requirements for humans and AI so that they can share similar goals and constraints. As recommended by the literature review, we will use user centered design methods but also involve Ai researchers in the process to avoid over confidence in AI possibilities. This will enable us to invent new shared representations between humans and AI so that we can create successful conditions for Human-Automation Teaming.

Another important aspect that remains understudied in the identified related work concerns the transition between several levels of automation, either human initiated or system initiated. In ASTAIR, we want to explore how to transition between levels of automations according to user preferences or AI performances.

For path planning, compared to optimal solvers, bounded sub-optimal solvers perform better in terms of computational time, while slightly decreasing the solution quality. On the other hand, unbounded suboptimal solvers generate solutions much faster than optimal and sub-optimal solvers, however the completeness of the obtained solutions are not always guaranteed even though a feasible solution exists. Thus, the trade-offs between solution quality, completeness and computational complexity should be considered while selecting the best solver. An interesting fact is that despite being optimal and complete, the single agent path planning solver, SIPP, is faster than some of the unbounded sub-optimal solvers with no completeness guarantees. For multi-agent path planning, when priority-based search is combined with the solvers such as CBS, larger scale instances can be solved with better solution quality compared to other unbounded sub-optimal solvers such as CA*, HCA*, WHCA*. One example is CBSw/P which usually finds optimal or near-optimal solutions, even though it is classified as an unbounded sub-optimal solver. A similar and improved solver is PBS, which solves well-formed Instances with six hundred agents in less than a minute, finds solutions for many instances where standard prioritized algorithms cannot, and remains near optimal and efficient for more than a hundred agents. The state-of-the-art multi-agent motion planner, ECBS-CT, generates optimal or bounded suboptimal solutions. In the high-level search, it takes a problem instance and a suboptimality bound as input. The low-level search uses SCIPP, which is a generalization of SIPP that is suitable for focal search. A scalable and effective multi-agent safe motion planner is S2M that enables a group of agents to move to their desired locations while avoiding collisions with obstacles and other agents, with the presence of rich obstacles, high-dimensional, nonlinear, nonholonomic dynamics, actuation limits, and disturbances. A piecewise linear path is obtained for each agent such that the actual trajectories following these paths are guaranteed to satisfy the reach-and-avoid requirement. a collision-free path for each agent is found by solving Mixed Integer-Linear Programs and agents are coordinated using the priority-based search. S2M shows improvements over the solving time and the solution quality compared to two state-of-the-art multi-agent motion planners, ECBS-CT, in 2D and 3D scenarios with ground vehicles and quadrotors.

In ASTAIR, the goal is to generate realistic solutions in short computational time so that disruptions or changes in environmental conditions are addressed on time. Thus, rather than optimal and complex solvers, both high quality and efficient solvers are aimed to be integrated with human-machine interface to aid the planning and execution of instant changes. The problems are to be solved on complex airport surface layouts that are to be converted into graphs including nodes, edges, intersections. Thus, application of fast solution procedures becomes crucial for airport surface operations. For path and motion planning, considering its efficiency and high-quality solutions in many of the instances, priority-based search, PBS, will be one of the tools that will be integrated with other approaches. One example of priority-based modelling for airport surface movements exists in the literature where access priorities of aircraft for a road section are adjusted by decreasing the priority of delayed aircraft. In combination with PBS, safe interval path planning solver, SIPP, is worth considering due to its computational efficiency, optimality and completeness. In addition to path finding, SIPP has also been combined with path tracking or following algorithms for motion planning or adapted to deal with agents with different shapes and sizes. Thus, combining SIPP with PBS and local level motion planning approaches to find trajectories by considering the speed profiles and other kino-dynamic constraints is a promising approach for dealing with path and motion planning problems in ASTAIR. The multi-agent safe motion planner, S2M, which combines mixed integer programming with priority-based search is also worth considering due to its scalability and some of its procedures could be integrated into the path and motion planning solutions of ASTAIR. Apart from these, the recent trend of combining path planning with task allocation or simultaneous target assignment and sequencing fits well to the scope of ASTAIR, where finding a tug allocation solution or dynamic assignment of tugs to aircraft while handling the path and motion planning at the same time is among the main interests.

6 References

[1] A. I. R. 2. 0 EASA, “A Human-centric Approach to AI in Aviation 2.0,” Technical Report, EASA, 2023. [Online]. Available: https://www.easa.europa.eu/en/downloads/137919/en
[2] EASA, “Guidance for Level 1 & 2 machine learning applications,” Technical Report, EASA, 2023. [Online]. Available: https://www.easa.europa.eu/en/downloads/137631/en
[3] S. Berretta, A. Tausch, G. Ontrup, B. Gilles, C. Peifer, and A. Kluge, “Defining human-AI teaming the human-centered way: a scoping review and network analysis,” *Front. Artif. Intell.*, vol. 6, p. 1250725, Sep. 2023, doi: 10.3389/frai.2023.1250725.
[4] T. O’Neill, N. McNeese, A. Barron, and B. Schelble, “Human–Autonomy Teaming: A Review and Analysis of the Empirical Literature,” *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 64, no. 5, pp. 904–938, Aug. 2022, doi: 10.1177/0018720820960865.
[5] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Trans. Syst. Man Cybern. Part ASystems Hum.*, vol. 30, no. 3, pp. 286–297, 2000, doi: 10.1109/3468.844354.
[6] D. A. Grimm, M. Demir, J. C. Gorman, and N. J. Cooke, “Team Situation Awareness in Human-Autonomy Teaming: A Systems Level Approach,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 62, no. 1, pp. 149–149, Sep. 2018, doi: 10.1177/1541931218621034.
[7] S. W. A. Dekker and D. D. Woods, “MABA-MABA or Abracadabra? Progress on Human–Automation Co-ordination,” *Cogn. Technol. Work*, vol. 4, no. 4, pp. 240–244, Nov. 2002, doi: 10.1007/s101110200022.
[8] R. Parasuraman and C. D. Wickens, “Humans: Still vital after all these years of automation,” in *Decision Making in Aviation*, Routledge, 2017, pp. 251–260.
[9] R. Zhang, N. J. McNeese, G. Freeman, and G. Musick, “‘An Ideal Human’: Expectations of AI Teammates in Human-AI Teaming,” *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW3, pp. 1–25, Jan. 2021, doi: 10.1145/3432945.
[10] S. Conversy, J. Garcia, and D.-B. Vo, “State of the Art on Computer-Supported Collaborative Work (CSCW), Situation Awareness (SA) and Human-Automation Teaming (HAT) for the SESAR ER2 AEON (Advanced Engine-Off Navigation) Project,” PhD Thesis, Ecole Nationale de l’Aviation Civile, 2021. Accessed: Feb. 26, 2024. [Online]. Available: https://hal.science/hal-03930640/document
[11] A. E. Milewski and S. H. Lewis, “Delegating to software agents,” *Int. J. Hum.-Comput. Stud.*, vol. 46, no. 4, pp. 485–500, Apr. 1997, doi: 10.1006/ijhc.1996.0100.
[12] J. M. Jenks and J. M. Kelly, *Don’t do--delegate!* New York: F. Watts, 1985.
[13] M. Pinski, M. Adam, and A. Benlian, “AI Knowledge: Improving AI Delegation through Human Enablement,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, in CHI ’23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–17. doi: 10.1145/3544548.3580794.
[14] E. Park, Y. Jung, I. Kim, and U. Lee, “Charlie and the Semi-Automated Factory: Data-Driven Operator Behavior and Performance Modeling for Human-Machine Collaborative Systems,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, in CHI ’23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–16. doi: 10.1145/3544548.3581457.
[15] G. He, L. Kuiper, and U. Gadiraju, “Knowing About Knowing: An Illusion of Human Competence Can Hinder Appropriate Reliance on AI Systems,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, in CHI ’23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–18. doi: 10.1145/3544548.3581025.
[16] Y. T.-Y. Hou, W.-Y. Lee, and M. Jung, “‘Should I Follow the Human, or Follow the Robot?’ — Robots in Power Can Have More Influence Than Humans on Decision-Making,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, in CHI ’23. New York, NY, USA:

Association for Computing Machinery, Apr. 2023, pp. 1–13. doi: 10.1145/3544548.3581066.
[17] A. Fügener, J. Grahl, A. Gupta, and W. Ketter, “Cognitive Challenges in Human–Artificial Intelligence Collaboration: Investigating the Path Toward Productive Delegation,” *Inf. Syst. Res.*, vol. 33, no. 2, pp. 678–696, Jun. 2022, doi: 10.1287/isre.2021.1079.
[18] D. H. Pink, *Drive: the surprising truth about what motivates us*, First Riverhead trade paperback edition: April 2011. New York: Riverhead Books, 2011.
[19] N. J. McNeese, B. G. Schelble, L. B. Canonico, and M. Demir, “Who/What Is My Teammate? Team Composition Considerations in Human–AI Teaming,” *IEEE Trans. Hum.-Mach. Syst.*, vol. 51, no. 4, pp. 288–299, Aug. 2021, doi: 10.1109/THMS.2021.3086018.
[20] R. Kocielnik, S. Amershi, and P. N. Bennett, “Will You Accept an Imperfect AI? Exploring Designs for Adjusting End-user Expectations of AI Systems,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, in CHI ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–14. doi: 10.1145/3290605.3300641.
[21] T. Liao, R. Taori, I. D. Raji, and L. Schmidt, “Are We Learning Yet? A Meta Review of Evaluation Failures Across Machine Learning,” presented at the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Aug. 2021. Accessed: Jan. 15, 2024. [Online]. Available: https://openreview.net/forum?id=mPducS1MsEK
[22] S. Rismani *et al.*, “From Plane Crashes to Algorithmic Harm: Applicability of Safety Engineering Frameworks for Responsible ML,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg Germany: ACM, Apr. 2023, pp. 1–18. doi: 10.1145/3544548.3581407.
[23] Q. Roy, F. Zhang, and D. Vogel, “Automation Accuracy Is Good, but High Controllability May Be Better,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow Scotland Uk: ACM, May 2019, pp. 1–8. doi: 10.1145/3290605.3300750.
[24] T. Igarashi and J. F. Hughes, “A suggestive interface for 3D drawing,” in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, Orlando Florida: ACM, Nov. 2001, pp. 173–181. doi: 10.1145/502348.502379.
[25] A. Schmidt and T. Herrmann, “Intervention user interfaces: a new interaction paradigm for automated systems,” *Interactions*, vol. 24, no. 5, pp. 40–45, 2017.
[26] G. Calhoun, “Adaptable (Not Adaptive) Automation: Forefront of Human–Automation Teaming,” *Hum. Factors*, vol. 64, no. 2, pp. 269–277, Mar. 2022, doi: 10.1177/00187208211037457.
[27] S. Conversy *et al.*, “Vizir: A Domain-Specific Graphical Language for Authoring and Operating Airport Automations,” in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 2018, pp. 261–273.
[28] Y. Sun, Q. Song, X. Gui, F. Ma, and T. Wang, “AutoML in The Wild: Obstacles, Workarounds, and Expectations,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg Germany: ACM, Apr. 2023, pp. 1–15. doi: 10.1145/3544548.3581082.
[29] J. T. Liang, M. Arab, M. Ko, A. J. Ko, and T. D. LaToza, “A Qualitative Study on the Implementation Design Decisions of Developers,” in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, Melbourne, Australia: IEEE, May 2023, pp. 435–447. doi: 10.1109/ICSE48619.2023.00047.
[30] J. E. Allen, C. I. Guinn, and E. Horvtz, “Mixed-initiative interaction,” *IEEE Intell. Syst. Their Appl.*, vol. 14, no. 5, pp. 14–23, Sep. 1999, doi: 10.1109/5254.796083.
[31] E. Horvitz, “Principles of mixed-initiative user interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI ’99*, Pittsburgh, Pennsylvania, United States: ACM Press, 1999, pp. 159–166. doi: 10.1145/302979.303030.
[32] M. Neef, “A Taxonomy of Human – Agent Team Collaborations,” *Proc. 18th BeNeLux Conf. Artif. Intell.*, pp. 245–250.
[33] R. Iftikhar, Y.-T. Chiu, M. S. Khan, and C. Caudwell, “Human–Agent Team Dynamics: A Review and Future Research Opportunities,” *IEEE Trans. Eng. Manag.*, pp. 1–16, 2023, doi:

10.1109/TEM.2023.3331369.
[34] D. Tran Luciani, J. Löwgren, and J. Lundberg, “Designing fine-grained interactions for automation in air traffic control,” *Cogn. Technol. Work*, vol. 22, no. 4, pp. 685–701, Nov. 2020, doi: 10.1007/s10111-019-00598-9.
[35] J. M. Bradshaw, P. J. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok, “Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction,” in *Agents and Computational Autonomy*, vol. 2969, M. Nickles, M. Rovatsos, and G. Weiss, Eds., in Lecture Notes in Computer Science, vol. 2969. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 17–39. doi: 10.1007/978-3-540-25928-2_3.
[36] W. Xiong, H. Fan, L. Ma, and C. Wang, “Challenges of human—machine collaboration in risky decision-making,” *Front. Eng. Manag.*, vol. 9, no. 1, pp. 89–103, Mar. 2022, doi: 10.1007/s42524-021-0182-0.
[37] P. Ala-Pietilä *et al.*, *The assessment list for trustworthy artificial intelligence (ALTAI)*. European Commission, 2020. Accessed: Feb. 26, 2024. [Online]. Available: https://books.google.com/books?hl=fr&lr=&id=cu8dEAAAQBAJ&oi=fnd&pg=PA1&dq=THE+ASSESSMENT+LIST+FOR+TRUSTWORTHY+ARTIFICIAL+INTELLIGENCE+(ALTAI)+for+self+assessment&ots=CaVKymi4rY&sig=jGr0oYRkpLNxruu5D-7YRrwvL18
[38] B. Goodman and S. Flaxman, “European Union regulations on algorithmic decision-making and a ‘right to explanation,’” *AI Mag.*, vol. 38, no. 3, pp. 50–57, 2017.
[39] A. B. Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, pp. 82–115, 2020, doi: https://doi.org/10.1016/j.inffus.2019.12.012.
[40] R. Dwivedi *et al.*, “Explainable AI (XAI): Core Ideas, Techniques, and Solutions,” *ACM Comput Surv*, vol. 55, no. 9, Jan. 2023, doi: 10.1145/3561048.
[41] S. Lee, I. Hwang, and K. Leiden, “Flight deck human-automation issue detection via intent inference,” *Proceedings of the International Conference on Human-Computer Interaction in Aerospace*. Association for Computing Machinery, Santa Clara, California, p. Article 24, 2014. [Online]. Available: https://doi.org/10.1145/2669592.2669690
[42] R. J. Mumaw, N. Sarter, and C. D. Wickens, “Analysis of pilots’ monitoring and performance on an automated flight deck,” in *Proceedings of the 11th International Symposium on Aviation Psychology*, Columbus, OH, USA: Ohio State University, 2001.
[43] Y. D. Özkan, A. G. Mirnig, A. Meschtscherjakov, C. Demir, and M. Tscheligi, “Mode Awareness Interfaces in Automated Vehicles, Robotics, and Aviation: A Literature Review,” *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. Association for Computing Machinery, Leeds, United Kingdom, pp. 147–158, 2021. [Online]. Available: https://doi.org/10.1145/3409118.3475125
[44] S. S. Silva and R. J. Hansman, “Divergence Between Flight Crew Mental Model and Aircraft System State in Auto-Throttle Mode Confusion Accident and Incident Cases,” *J. Cogn. Eng. Decis. Mak.*, vol. 9, no. 4, pp. 312–328, Dec. 2015, doi: 10.1177/1555343415597344.
[45] B. d’enquêtes et d’Analyse, *Accident survenu le 20 janvier 1992 près du Mont Sainte-Odile (Bas Rhin) à l’Airbus A 320 immatriculé F-GGED exploité par la compagnie Air Inter*. FR: BEA, 1992.
[46] C. A. S. FAA, “Operational use of flight path management systems,” Federal Aviation Administration, 2013.
[47] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation,” in *2018 IEEE conference on computational intelligence and games (CIG)*, IEEE, 2018, pp. 1–8.
[48] I. C. Kaadoud, L. Fahed, and P. Lenca, “Explainable AI: a narrative review at the crossroad of Knowledge Discovery, Knowledge Representation and Representation Learning,” in *MRC 2021: Twelfth International Workshop Modelling and Reasoning in Context*, ceur-ws. org, 2021, pp. 28–40.
[49] A. Degas *et al.*, “A survey on artificial intelligence (ai) and explainable ai in air traffic

management: Current trends and development with future research trajectory," *Appl. Sci.*, vol. 12, no. 3, p. 1295, 2022.
[50] V. Battiste, J. Lachter, and S. Brandt, *Human-Automation Teaming : Lessons Learned and Future Directions*, vol. 1. Springer International Publishing, 2018. doi: 10.1007/978-3-319-92046-7.
[51] P. M. Salmon, N. A. Stanton, and D. P. Jenkins, "Distributed situation awareness: Theory, measurement and application to teamwork," 2017.
[52] L. E. Holmquist, "Intelligence on tap: artificial intelligence as a new design material," *Interactions*, vol. 24, no. 4, pp. 28–33, Jun. 2017, doi: 10.1145/3085571.
[53] S. Amershi *et al.*, "Guidelines for human-AI interaction," in *Proceedings of the 2019 chi conference on human factors in computing systems*, 2019, pp. 1–13.
[54] D. A. Norman, "How might people interact with agents," *Commun. ACM*, vol. 37, no. 7, pp. 68–71, Jul. 1994, doi: 10.1145/176789.176796.
[55] G. Cimolino and T. C. N. Graham, "Two Heads Are Better Than One: A Dimension Space for Unifying Human and Artificial Intelligence in Shared Control," in *CHI Conference on Human Factors in Computing Systems*, New Orleans LA USA: ACM, Apr. 2022, pp. 1–21. doi: 10.1145/3491102.3517610.
[56] D. Russell, Q. V. Liao, C. Kulkarni, E. L. Glassman, and N. Martelaro, "Human-Computer Interaction and AI: What practitioners need to know to design and build effective AI system from a human perspective," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, in CHI EA '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–3. doi: 10.1145/3544549.3574170.
[57] "User-Centered Design [Book]." Accessed: Jan. 31, 2024. [Online]. Available: https://www.oreilly.com/library/view/user-centered-design/9781449359812/
[58] W. Xu, M. J. Dainoff, L. Ge, and Z. Gao, "Transitioning to Human Interaction with AI Systems: New Challenges and Opportunities for HCI Professionals to Enable Human-Centered AI," *Int. J. Human–Computer Interact.*, vol. 39, no. 3, pp. 494–518, Feb. 2023, doi: 10.1080/10447318.2022.2041900.
[59] K. J. K. Feng, M. J. Coppock, and D. W. McDonald, "How Do UX Practitioners Communicate AI as a Design Material? Artifacts, Conceptions, and Propositions," in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, in DIS '23. New York, NY, USA: Association for Computing Machinery, Jul. 2023, pp. 2263–2280. doi: 10.1145/3563657.3596101.
[60] T. Bratteteig and G. Verne, "Does AI make PD obsolete? exploring challenges from artificial intelligence to participatory design," in *Proceedings of the 15th Participatory Design Conference: Short Papers, Situated Actions, Workshops and Tutorial - Volume 2*, in PDC '18. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–5. doi: 10.1145/3210604.3210646.
[61] Q. Yang, A. Steinfeld, C. Rosé, and J. Zimmerman, "Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, in CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. doi: 10.1145/3313831.3376301.
[62] T. Olsson and K. Väänänen, "How does AI challenge design practice?," *Interactions*, vol. 28, no. 4, pp. 62–64, Jun. 2021, doi: 10.1145/3467479.
[63] D. Council, "The 'double diamond'design process model," *Des. Counc.*, vol. 2, p. 1, 2005.
[64] M. Gillies *et al.*, "Human-Centred Machine Learning," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, in CHI EA '16. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 3558–3565. doi: 10.1145/2851581.2856492.
[65] J. P. Pita, C. Barnhart, and A. P. Antunes, "Integrated Flight Scheduling and Fleet Assignment Under Airport Congestion," *Transp. Sci.*, vol. 47, no. 4, pp. 477–492, Nov. 2013, doi: 10.1287/trsc.1120.0442.
[66] S. Ahmadi and A. Akgunduz, "Airport operations with electric-powered towing alternatives

under stochastic conditions," *J. Air Transp. Manag.*, vol. 109, p. 102392, Jun. 2023, doi: 10.1016/j.jairtraman.2023.102392.
[67] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser, "A Robust Fleet-Assignment Model with Hub Isolation and Short Cycles," *Transp. Sci.*, vol. 38, no. 3, pp. 357–368, Aug. 2004, doi: 10.1287/trsc.1030.0038.
[68] S. Venkatachalam, S. Acharya, K. Oba, and Y. Nakayama, "Prescriptive Analytics for Swapping Aircraft Assignments at All Nippon Airways," *Inf. J. Appl. Anal.*, vol. 50, no. 2, pp. 99–111, Mar. 2020, doi: 10.1287/inte.2019.1016.
[69] C. A. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi, "The fleet assignment problem: Solving a large-scale integer program," *Math. Program.*, vol. 70, no. 1–3, pp. 211–232, Oct. 1995, doi: 10.1007/BF01585938.
[70] J. Abara, "Applying Integer Linear Programming to the Fleet Assignment Problem," *Interfaces*, vol. 19, no. 4, pp. 20–28, Aug. 1989, doi: 10.1287/inte.19.4.20.
[71] C. Y. Justin, A. P. Payan, and D. N. Mavris, "Integrated fleet assignment and scheduling for environmentally friendly electrified regional air mobility," *Transp. Res. Part C Emerg. Technol.*, vol. 138, p. 103567, May 2022, doi: 10.1016/j.trc.2022.103567.
[72] Z. Dong, Y. Chuhang, and H. Y. K. H. Lau, "An integrated flight scheduling and fleet assignment method based on a discrete choice model," *Comput. Ind. Eng.*, vol. 98, pp. 195–210, Aug. 2016, doi: 10.1016/j.cie.2016.05.040.
[73] "AEON: Toward a concept of operation and tools for supporting engine-off navigation for ground operations - ENAC - École nationale de l'aviation civile." Accessed: Feb. 05, 2024. [Online]. Available: https://enac.hal.science/hal-03859908/
[74] S. Van Oosterom, M. Mitici, and J. Hoekstra, "Dispatching a fleet of electric towing vehicles for aircraft taxiing with conflict avoidance and efficient battery charging," *Transp. Res. Part C Emerg. Technol.*, vol. 147, p. 103995, Feb. 2023, doi: 10.1016/j.trc.2022.103995.
[75] M. Zoutendijk, S. J. Van Oosterom, and M. Mitici, "Electric Taxiing with Disruption Management: Assignment of Electric Towing Vehicles to Aircraft," in *AIAA AVIATION 2023 Forum*, San Diego, CA and Online: American Institute of Aeronautics and Astronautics, Jun. 2023. doi: 10.2514/6.2023-4219.
[76] E. V. Baaren and P. C. Roling, "Design of a zero emission aircraft towing system," in *AIAA Aviation 2019 Forum*, Dallas, Texas: American Institute of Aeronautics and Astronautics, Jun. 2019. doi: 10.2514/6.2019-2932.
[77] M. Soltani, S. Ahmadi, A. Akgunduz, and N. Bhuiyan, "An eco-friendly aircraft taxiing approach with collision and conflict avoidance," *Transp. Res. Part C Emerg. Technol.*, vol. 121, p. 102872, Dec. 2020, doi: 10.1016/j.trc.2020.102872.
[78] S. Zaninotto, J. Gauci, G. Farrugia, and J. Debattista, "Design of a Human-in-the-Loop Aircraft Taxi Optimisation System Using Autonomous Tow Trucks," in *AIAA Aviation 2019 Forum*, Dallas, Texas: American Institute of Aeronautics and Astronautics, Jun. 2019. doi: 10.2514/6.2019-2931.
[79] R. T. Bye, "A Receding Horizon Genetic Algorithm for Dynamic Resource Allocation: A Case Study on Optimal Positioning of Tugs," in *Computational Intelligence*, vol. 399, K. Madani, A. Dourado Correia, A. Rosa, and J. Filipe, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 131–147. doi: 10.1007/978-3-642-27534-0_9.
[80] R. T. Bye and H. G. Schaathun, "An Improved Receding Horizon Genetic Algorithm For The Tug Fleet Optimisation Problem," in *ECMS 2014 Proceedings edited by: Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani*, ECMS, May 2014, pp. 682–690. doi: 10.7148/2014-0682.
[81] B. Assimizele, J. Oppen, and R. T. Bye, "A Sustainable Model For Optimal Dynamic Allocation Of Patrol Tugs To Oil Tankers," in *ECMS 2013 Proceedings edited by: Webjorn Rekdalsbakken, Robin T. Bye, Houxiang Zhang*, ECMS, May 2013, pp. 801–807. doi: 10.7148/2013-0801.
[82] S.-T. Chen, G. Ermiş, and A. Sharpanskykh, "Multi-agent planning and coordination for automated aircraft ground handling," *Robot. Auton. Syst.*, vol. 167, p. 104480, Sep. 2023, doi:

10.1016/j.robot.2023.104480.
[83] S. Wang, Y. Che, H. Zhao, and A. Lim, “Accurate Tracking, Collision Detection, and Optimal Scheduling of Airport Ground Support Equipment,” *IEEE Internet Things J.*, vol. 8, no. 1, pp. 572–584, Jan. 2021, doi: 10.1109/JIOT.2020.3004874.
[84] J. Volt, S. Stojić, and P. Had, “Optimization of the Baggage Loading and Unloading Equipment,” *Transp. Res. Procedia*, vol. 65, pp. 246–255, 2022, doi: 10.1016/j.trpro.2022.11.029.
[85] K. Sheibani, “Scheduling Aircraft Ground Handling Operations Under Uncertainty Using Critical Path Analysis and Monte Carlo Simulation: Survey and Research Directions,” *Int. J. Bus. Strategy Autom.*, vol. 1, no. 1, pp. 37–45, Jan. 2020, doi: 10.4018/IJBSA.2020010103.
[86] D.-W. Bao, J.-Y. Zhou, Z.-Q. Zhang, Z. Chen, and D. Kang, “Mixed fleet scheduling method for airport ground service vehicles under the trend of electrification,” *J. Air Transp. Manag.*, vol. 108, p. 102379, May 2023, doi: 10.1016/j.jairtraman.2023.102379.
[87] A. Norin, D. Yuan, T. A. Granberg, and P. Värbrand, “Scheduling de-icing vehicles within airport logistics: a heuristic algorithm and performance evaluation,” *J. Oper. Res. Soc.*, vol. 63, no. 8, pp. 1116–1125, Aug. 2012, doi: 10.1057/jors.2011.100.
[88] D. Alonso Tabares, F. Mora-Camino, and A. Drouin, “A multi-time scale management structure for airport ground handling automation,” *J. Air Transp. Manag.*, vol. 90, p. 101959, Jan. 2021, doi: 10.1016/j.jairtraman.2020.101959.
[89] C. Evertse and H. G. Visser, “Real-time airport surface movement planning: Minimizing aircraft emissions,” *Transp. Res. Part C Emerg. Technol.*, vol. 79, pp. 224–241, Jun. 2017, doi: 10.1016/j.trc.2017.03.018.
[90] M. Weiszer, J. Chen, and P. Stewart, “A real-time Active Routing approach via a database for airport surface movement,” *Transp. Res. Part C Emerg. Technol.*, vol. 58, pp. 127–145, Sep. 2015, doi: 10.1016/j.trc.2015.07.011.
[91] S. Ravizza, J. Chen, J. A. D. Atkin, E. K. Burke, and P. Stewart, “The trade-off between taxi time and fuel consumption in airport ground movement,” *Public Transp.*, vol. 5, no. 1–2, pp. 25–40, Sep. 2013, doi: 10.1007/s12469-013-0060-1.
[92] J. Chen and P. Stewart, “Planning aircraft taxiing trajectories via a multi-objective immune optimisation,” presented at the 2011 Seventh International Conference on Natural Computation, IEEE, 2011, pp. 2235–2240. doi: 10.1109/ICNC.2011.6022587.
[93] X. Zou, P. Cheng, W. D. Liu, N. Cheng, and J. P. Zhang, “A two-stage taxi scheduling strategy at airports with multiple independent runways,” *Transp. Res. Part C Emerg. Technol.*, vol. 95, pp. 165–184, Oct. 2018, doi: 10.1016/j.trc.2018.07.005.
[94] G. Clare and A. G. Richards, “Optimization of Taxiway Routing and Runway Scheduling,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1000–1013, Dec. 2011, doi: 10.1109/TITS.2011.2131650.
[95] M. Samà, A. D’Ariano, P. D’Ariano, and D. Pacciarelli, “Optimal aircraft scheduling and routing at a terminal control area during disturbances,” *Transp. Res. Part C Emerg. Technol.*, vol. 47, pp. 61–85, Oct. 2014, doi: 10.1016/j.trc.2014.08.005.
[96] X. Wang and Q. Zuo, “Aircraft taxiing route planning based on airport hotspots,” presented at the MATERIALS SCIENCE, ENERGY TECHNOLOGY, AND POWER ENGINEERING I: 1st International Conference on Materials Science, Energy Technology, Power Engineering (MEP 2017), Hangzhou, China, 2017, p. 020088. doi: 10.1063/1.4982453.
[97] Y. Wang, S. Han, X. Tang, and X. Zhu, “Research on airport surface modeling and path planning algorithm based on CTPN,” in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Shanghai: IEEE, Nov. 2009, pp. 262–266. doi: 10.1109/ICICISYS.2009.5358176.
[98] Y.-L. Pan, J.-C. Jhang, and F.-R. Gu, “Using Advanced Surface Movement Guidance and Control System for Kaohsiung Xiaogang Airport Aircraft’s Route Assignment Planning Based on Petri nets,” in *2022 8th International Conference on Applied System Innovation (ICASI)*, Nantou, Taiwan:

IEEE, Apr. 2022, pp. 160–165. doi: 10.1109/ICASI55125.2022.9774454.
[99] C. Liu and Y. Qin, "Modeling the Aircraft Ground Operation on Airport Based on Timed Petri Net," in *2006 6th World Congress on Intelligent Control and Automation*, Jun. 2006, pp. 1304–1308. doi: 10.1109/WCICA.2006.1712558.
[100] X. Luo, Y. Tang, H. Wu, and D. He, "Real-time adjustment strategy for conflict-free taxiing route of A-SMGCS aircraft on airport surface," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2015, pp. 929–934. doi: 10.1109/ICMA.2015.7237610.
[101] S. Sun and K. Hua, "An Aircraft Sequencing Approach Based on Fuzzy Petri-net," in *2009 International Joint Conference on Computational Sciences and Optimization*, Apr. 2009, pp. 1008–1011. doi: 10.1109/CSO.2009.157.
[102] K. Uehara, K. Hiraishi, and K. Kobayashi, "Mesoscopic modeling of airport surface by object Petri nets," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2015, pp. 298–303. doi: 10.1109/CoASE.2015.7294095.
[103] X. Wang, H. Peng, J. Liu, X. Dong, X. Zhao, and C. Lu, "Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck," *Def. Technol.*, vol. 18, no. 2, pp. 238–248, Feb. 2022, doi: 10.1016/j.dt.2020.11.013.
[104] J. Liu, W. Han, C. Liu, and H. Peng, "A New Method for the Optimal Control Problem of Path Planning for Unmanned Ground Systems," *IEEE Access*, vol. 6, pp. 33251–33260, 2018, doi: 10.1109/ACCESS.2018.2846769.
[105] C. Liu and K. Guo, "Airport Taxi Scheduling Optimization Based on Genetic Algorithm," in *2010 International Conference on Computational Intelligence and Security*, Dec. 2010, pp. 205–208. doi: 10.1109/CIS.2010.50.
[106] Z. Shang, Z. Mao, H. Zhang, and M. Xu, "Collaborative Path Planning of Multiple Carrier-based Aircraft Based on Multi-agent Reinforcement Learning," in *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*, Jun. 2022, pp. 512–517. doi: 10.1109/MDM55031.2022.00108.
[107] L. Khammash, L. Mantecchini, and V. Reis, "Micro-simulation of airport taxiing procedures to improve operation sustainability: Application of semi-robotic towing tractor," in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Naples, Italy: IEEE, Jun. 2017, pp. 616–621. doi: 10.1109/MTITS.2017.8005587.
[108] D. Kornhauser, G. Miller, and P. Spirakis, "Coordinating Pebble Motion On Graphs, The Diameter Of Permutation Groups, And Applications," in *25th Annual Symposium onFoundations of Computer Science, 1984.*, Oct. 1984, pp. 241–250. doi: 10.1109/SFCS.1984.715921.
[109] M. Ryan, "Multi-robot path-planning with subgraphs," in *Proceedings of the 2006 Australasian Conference on Robotics and Automation, ACRA 2006*, 2006.
[110] R. M. Wilson, "Graph puzzles, homotopy, and the alternating group," *J. Comb. Theory Ser. B*, vol. 16, no. 1, pp. 86–96, Feb. 1974, doi: 10.1016/0095-8956(74)90098-7.
[111] P. Surynek, "A novel approach to path planning for multiple robots in bi-connected graphs," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3613–3619. doi: 10.1109/ROBOT.2009.5152326.
[112] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artif. Intell.*, vol. 219, pp. 40–66, Feb. 2015, doi: 10.1016/j.artint.2014.11.006.
[113] Q. Ma, X. Lei, and Q. Zhang, "Mobile Robot Path Planning with Complex Constraints Based on the Second-Order Oscillating Particle Swarm Optimization Algorithm," in *2009 WRI World Congress on Computer Science and Information Engineering*, vol. 5, Los Angeles, CA, USA, 2009, pp. 244–248. [Online]. Available: 10.1109/CSIE.2009.124
[114] R. Morris *et al.*, "Planning, Scheduling and Monitoring for Airport Surface Operations," presented at the AAAI-16 Workshop on Planning for Hybrid Systems, 2016. Accessed: Oct. 27, 2023. [Online]. Available: http://idm-lab.org/bib/abstracts/Koen16b.html
[115] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent

robots," *J. Intell. Manuf.*, vol. 33, no. 2, pp. 387–424, Feb. 2022, doi: 10.1007/s10845-021-01867-z.
[116] M. Barer, G. Sharon, R. Stern, and A. Felner, "Suboptimal Variants of the Conflict-Based Search Algorithm for the Multi-Agent Pathfinding Problem," *Proc. Int. Symp. Comb. Search*, vol. 5, no. 1, pp. 19–27, Sep. 2021, doi: 10.1609/socs.v5i1.18315.
[117] E. Boyarski *et al.*, "ICBS: Improved conflict-based search algorithm for multi-agent pathfinding," presented at the IJCAI International Joint Conference on Artificial Intelligence (IJCAI 2015), 2015.
[118] E. Boyarski *et al.*, "ICBS: The Improved Conflict-Based Search Algorithm for Multi-Agent Pathfinding: Extended Abstract," in *Proceedings of the 8th Annual Symposium on Combinatorial Search, SoCS 2015*, 2015.
[119] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, "The increasing cost tree search for optimal multi-agent pathfinding," *Artif. Intell.*, vol. 195, pp. 470–495, Feb. 2013, doi: 10.1016/j.artint.2012.11.006.
[120] G. Sharon, R. Stern, A. Fernel, and N. Sturtevant, "Meta-agent conflict-based search for optimal multi-agent path finding," in *Proceedings of the 5th Annual Symposium on Combinatorial Search, SoCS 2012*, 2012.
[121] M. Cap, P. Novak, A. Kleiner, and M. Selecky, "Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 835–849, Jul. 2015, doi: 10.1109/TASE.2015.2445780.
[122] K. Yakovlev, A. Andreychuk, and V. Vorobyev, "Prioritized Multi-agent Path Finding for Differential Drive Robots," in *2019 European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic: IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/ECMR.2019.8870957.
[123] B. D. Wilde, A. W. Ter Mors, and C. Witteveen, "Push and Rotate: a Complete Multi-agent Pathfinding Algorithm," *J. Artif. Intell. Res.*, vol. 51, pp. 443–492, Oct. 2014, doi: 10.1613/jair.4447.
[124] B. Wilde, A. Mors, and C. Witteveen, "Wilde, Boris & Mors, Adriaan & Witteveen, Cees. Push and rotate: Cooperative multi-agent path planning.," presented at the 12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013, 2013, pp. 87–94.
[125] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.
[126] J. I. Solis Vidana, J. Motes, R. Sandstrom, and N. Amato, "Representation-Optimal Multi-Robot Motion Planning Using Conflict-Based Search," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4608–4615, Jul. 2021, doi: 10.1109/LRA.2021.3068910.
[127] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, "Pruning Techniques for the Increasing Cost Tree Search for Optimal Multi-agent Pathfinding," *Proc. Int. Symp. Comb. Search*, vol. 2, no. 1, pp. 150–157, Aug. 2021, doi: 10.1609/socs.v2i1.18201.
[128] K. Yakovlev and A. Andreychuk, "Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm," *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 27, pp. 586–594, Jun. 2017, doi: 10.1609/icaps.v27i1.13856.
[129] S.-H. Chan *et al.*, "Nested ECBS for Bounded-Suboptimal Multi-Agent Path Finding," in *Proceedings of the IJCAI Workshop on Multi-Agent Path Finding*, 2020.
[130] D. Silver, "Cooperative Pathfinding," *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, vol. 1, no. 1, pp. 117–122, Sep. 2021, doi: 10.1609/aiide.v1i1.18726.
[131] B. Stout, "Smart Moves: Intelligent Pathfinding," 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:53832179
[132] G. Wagner and H. Choset, "M*: A complete multirobot path planning algorithm with performance bounds," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 3260–3267. doi: 10.1109/IROS.2011.6095022.
[133] M. Phillips and M. Likhachev, "SIPP: Safe interval path planning for dynamic environments," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5628–5635. doi:

10.1109/ICRA.2011.5980306.
[134] J. P. Gonzalez, M. Phillips, B. Neuman, and M. Likhachev, "Robust mobility in human-populated environments," presented at the SPIE Defense, Security, and Sensing, Baltimore, Maryland, USA, May 2012, p. 83870Z. doi: 10.1117/12.918948.
[135] K. Yakovlev, A. Andreychuk, and R. Stern, "Revisiting Bounded-Suboptimal Safe Interval Path Planning," *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 30, pp. 300–304, Jun. 2020, doi: 10.1609/icaps.v30i1.6674.
[136] A. Nash, S. Koenig, and C. Tovey, "Lazy Theta*: Any-Angle Path Planning and Path Length Analysis in 3D," *Proc. Int. Symp. Comb. Search*, vol. 1, no. 1, pp. 153–154, Aug. 2010, doi: 10.1609/socs.v1i1.18152.
[137] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-Angle Path Planning on Grids," *J. Artif. Intell. Res.*, vol. 39, pp. 533–579, Oct. 2010, doi: 10.1613/jair.2994.
[138] D. Harabor and A. Grastien, "An Optimal Any-Angle Pathfinding Algorithm," *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 23, pp. 308–311, Jun. 2013, doi: 10.1609/icaps.v23i1.13609.
[139] N. Hormazábal, A. Díaz, C. Hernández, and J. Baier, "Fast and Almost Optimal Any-Angle Pathfinding Using the 2k Neighborhoods," *Proc. Int. Symp. Comb. Search*, vol. 8, no. 1, pp. 139–143, Sep. 2021, doi: 10.1609/socs.v8i1.18442.
[140] V. Narayanan, M. Phillips, and M. Likhachev, "Anytime Safe Interval Path Planning for dynamic environments," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4708–4715. doi: 10.1109/IROS.2012.6386191.
[141] M. Likhachev, G. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems*, 2004.
[142] K. Yakovlev, A. Andreychuk, J. Belinskaya, and D. Makarov, "Combining Safe Interval Path Planning and Constrained Path Following Control: Preliminary Results," in *Interactive Collaborative Robotics*, vol. 11659, A. Ronzhin, G. Rigoll, and R. Meshcheryakov, Eds., Cham: Springer International Publishing, 2019, pp. 310–319. doi: 10.1007/978-3-030-26118-4_30.
[143] J. Yu and S. M. LaValle, "Multi-agent Path Planning and Network Flow," in *Algorithmic Foundations of Robotics X*, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds., in Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer, 2013, pp. 157–173. doi: 10.1007/978-3-642-36279-8_10.
[144] J. Yu and S. M. LaValle, "Planning optimal paths for multiple robots on graphs," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3612–3617. doi: 10.1109/ICRA.2013.6631084.
[145] J. Yu and S. M. LaValle, "Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1163–1177, Oct. 2016, doi: 10.1109/TRO.2016.2593448.
[146] J. Yu and S. M. LaValle, "Fast, Near-Optimal Computation for Multi-Robot Path Planning on Graphs," in *Late-Breaking Developments in the Field of Artificial Intelligence, Bellevue, Washington, USA, July 14-18, 2013*, in AAAI Technical Report, vol. WS-13-17. AAAI, 2013. Accessed: Oct. 27, 2023. [Online]. Available: http://www.aaai.org/ocs/index.php/WS/AAAIW13/paper/view/7011
[147] K. Murakami, "Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system," *Comput. Ind. Eng.*, vol. 141, p. 106270, Mar. 2020, doi: 10.1016/j.cie.2020.106270.
[148] P. Surynek, "Towards Optimal Cooperative Path Planning in Hard Setups through Satisfiability Solving," in *PRICAI 2012: Trends in Artificial Intelligence*, P. Anthony, M. Ishizuka, and D. Lukose, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 564–576. doi: 10.1007/978-3-642-32695-0_50.
[149] E. Erdem, D. Kisa, U. Oztok, and P. Schüller, "A General Formal Framework for Pathfinding Problems with Multiple Agents," *Proc. AAAI Conf. Artif. Intell.*, vol. 27, no. 1, pp. 290–296, Jun. 2013, doi: 10.1609/aaai.v27i1.8592.

[150] J. Li, P. Surynek, A. Felner, H. Ma, T. K. Kumar, and S. Koenig, “Multi-Agent Path Finding for Large Agents,” *Proc. Int. Symp. Comb. Search*, vol. 10, no. 1, pp. 186–187, Sep. 2021, doi: 10.1609/socs.v10i1.18483.
[151] H. Zhang, Y. Li, J. Li, T. K. S. Kumar, and S. Koenig, “Mutex Propagation in Multi-Agent Path Finding for Large Agents,” *Proc. Int. Symp. Comb. Search*, vol. 15, no. 1, pp. 249–253, Jul. 2022, doi: 10.1609/socs.v15i1.21776.
[152] S. Dergachev and K. Yakovlev, “Towards a Complete Multi-agent Pathfinding Algorithm for Large Agents,” in *Advances in Computational Intelligence*, vol. 13612, O. Pichardo Lagunas, J. Martínez-Miranda, and B. Martínez Seis, Eds., Cham: Springer Nature Switzerland, 2022, pp. 355–367. doi: 10.1007/978-3-031-19493-1_28.
[153] A. Andreychuk, K. Yakovlev, P. Surynek, D. Atzmon, and R. Stern, “Multi-agent pathfinding with continuous time,” *Artif. Intell.*, vol. 305, p. 103662, Apr. 2022, doi: 10.1016/j.artint.2022.103662.
[154] P. Raja and S. Pugazhenthi, “Path Planning for Mobile Robots in Dynamic Environments Using Particle Swarm Optimization,” in *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, Oct. 2009, pp. 401–405. doi: 10.1109/ARTCom.2009.24.
[155] M. R. Panda, R. Priyadarshini, and S. K. Pradhan, “Autonomous mobile robot path planning using hybridization of particle swarm optimization and Tabu search,” in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Chennai: IEEE, Dec. 2016, pp. 1–7. doi: 10.1109/ICCIC.2016.7919636.
[156] L. Qing, Z. Chao, X. Yinmei, and Y. Yixin, “Path planning of mobile robots based on specialized genetic algorithm and improved particle swarm optimization,” in *Proceedings of the 31st Chinese Control Conference*, Jul. 2012, pp. 7204–7209. Accessed: Oct. 27, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6391213
[157] J.-H. Zhang, Y. Zhang, and Y. Zhou, “Path Planning of Mobile Robot Based on Hybrid Multi-Objective Bare Bones Particle Swarm Optimization With Differential Evolution,” *IEEE Access*, vol. 6, pp. 44542–44555, 2018, doi: 10.1109/ACCESS.2018.2864188.
[158] Q. Ma, X. Lei, and Q. Zhang, “Mobile Robot Path Planning with Complex Constraints Based on the Second-Order Oscillating Particle Swarm Optimization Algorithm,” in *2009 WRI World Congress on Computer Science and Information Engineering*, Mar. 2009, pp. 244–248. doi: 10.1109/CSIE.2009.124.
[159] B. B. V. L. Deepak and D. R. Parhi, “Target seeking behaviour of an intelligent mobile robot using advanced particle swarm optimization,” in *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, Dec. 2013, pp. 1–6. doi: 10.1109/CARE.2013.6733749.
[160] R. Solea and D. Cernega, “Online path planner for mobile robots using particle swarm optimization,” in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2016, pp. 222–227. doi: 10.1109/ICSTCC.2016.7790669.
[161] X. Luo, J. Wang, and X. Li, “Joint grid network and improved particle swarm optimization for path planning of mobile robot,” in *2017 36th Chinese Control Conference (CCC)*, Jul. 2017, pp. 8304–8309. doi: 10.23919/ChiCC.2017.8028672.
[162] L. Tan, H. Zhang, J. Shi, Y. Liu, and T. Yuan, “A robust multiple Unmanned Aerial Vehicles 3D path planning strategy via improved particle swarm optimization,” *Comput. Electr. Eng.*, vol. 111, p. 108947, Oct. 2023, doi: 10.1016/j.compeleceng.2023.108947.
[163] R. C. Arkin, E. Riseman, and A. Hanson, “Visual Strategies for Mobile Robot Navigation,” 1987.
[164] R. R. Murphy, “Introduction to AI Robotics,” *Ind. Robot*, vol. 28, no. 3, pp. 266–267, Jun. 2001, doi: 10.1108/ir.2001.28.3.266.1.
[165] D. Sun, J. Chen, S. Mitra, and C. Fan, “Multi-Agent Motion Planning From Signal Temporal Logic Specifications,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3451–3458, Apr. 2022, doi: 10.1109/LRA.2022.3146951.

[166] J. Chen, J. Li, C. Fan, and B. C. Williams, “Scalable and Safe Multi-Agent Motion Planning with Nonlinear Dynamics and Bounded Disturbances,” *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 13, pp. 11237–11245, May 2021, doi: 10.1609/aaai.v35i13.17340.
[167] L. Cohen, T. Uras, T. K. Kumar, and S. Koenig, “Optimal and Bounded-Suboptimal Multi-Agent Motion Planning,” *Proc. Int. Symp. Comb. Search*, vol. 10, no. 1, pp. 44–51, Sep. 2021, doi: 10.1609/socs.v10i1.18501.
[168] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, “Trajectory Planning for Quadrotor Swarms,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 856–869, Aug. 2018, doi: 10.1109/TRO.2018.2853613.
[169] M. Pivtoraiko, R. A. Knepper, and A. Kelly, “Differentially constrained mobile robot motion planning in state lattices,” *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, Mar. 2009, doi: 10.1002/rob.20285.
[170] A. Chari, R. Chen, and C. Liu, “Space-Time Conflict Spheres for Constrained Multi-Agent Motion Planning,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, Anchorage, AK, USA: IEEE, Jun. 2023, pp. 1–6. doi: 10.1109/IV55152.2023.10186586.
[171] J. P. Van Den Berg and M. H. Overmars, “Prioritized motion planning for multiple robots,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada: IEEE, 2005, pp. 430–435. doi: 10.1109/IROS.2005.1545306.
[172] K. Solovey, O. Salzman, and D. Halperin, “Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning,” *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 501–513, Apr. 2016, doi: 10.1177/0278364915615688.
[173] G. Sanchez and J.-C. Latombe, “Using a PRM planner to compare centralized and decoupled planning for multi-robot systems,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Washington, DC, USA: IEEE, 2002, pp. 2112–2119. doi: 10.1109/ROBOT.2002.1014852.
[174] M. Saha and P. Isto, “Multi-Robot Motion Planning by Incremental Coordination,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 5960–5963. doi: 10.1109/IROS.2006.282536.
[175] A. Zheng, T. Liu, Y. Song, X. Wang, and H. Yang, “Minimum dose path planning method for virtual nuclear facilities based on Multi-layer Adaptive Grid,” in *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, Apr. 2023, pp. 2153–2157. doi: 10.1109/ICSP58490.2023.10248516.
[176] K. Wang, W. Liang, H. Shi, J. Zhang, and Q. Wang, “Driving line-based two-stage path planning in the AGV sorting system,” *Robot. Auton. Syst.*, vol. 169, p. 104505, Nov. 2023, doi: 10.1016/j.robot.2023.104505.
[177] G. Belov, W. Du, M. Garcia De La Banda, D. Harabor, S. Koenig, and X. Wei, “From Multi-Agent Pathfinding to 3D Pipe Routing,” *Proc. Int. Symp. Comb. Search*, vol. 11, no. 1, pp. 11–19, Sep. 2021, doi: 10.1609/socs.v11i1.18530.
[178] Y. Tang, Z. Ren, J. Li, and K. Sycara, “Solving Multi-Agent Target Assignment and Path Finding with a Single Constraint Tree,” 2023, doi: 10.48550/ARXIV.2307.00663.
[179] G. A. Mills-tettey, A. Stentz, and M. B. Dias, “The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs,” Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, CMU-RI-TR-07-27, 2007. [Online]. Available: https://www.ri.cmu.edu/publications/the-dynamic-hungarian-algorithm-for-the-assignment-problem-with-changing-costs/
[180] Z. Ren, S. Rathinam, and H. Choset, “CBSS: A New Approach for Multiagent Combinatorial Path Finding,” *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2669–2683, Aug. 2023, doi: 10.1109/TRO.2023.3266993.
[181] H. Ma and S. Koenig, “Optimal target assignment and path finding for teams of agents,” in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems,*

AAMAS, 2016.
[182] C. Henkel, J. Abbenseth, and M. Toussaint, “An Optimal Algorithm to Solve the Combined Task Allocation and Path Finding Problem,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 4140–4146. doi: 10.1109/IROS40897.2019.8968096.
[183] X. Zhong, J. Li, S. Koenig, and H. Ma, “Optimal and Bounded-Suboptimal Multi-Goal Task Assignment and Path Finding,” in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 10731–10737. doi: 10.1109/ICRA46639.2022.9812020.
[184] V. Nguyen, P. Obermeier, T. C. Son, T. Schaub, and W. Yeoh, “Generalized Target Assignment and Path Finding Using Answer Set Programming,” pp. 1216–1223, 2017.
[185] P. Surynek, “Multi-Goal Multi-Agent Path Finding via Decoupled and Integrated Goal Vertex Ordering,” *Proc. Int. Symp. Comb. Search*, vol. 12, no. 1, pp. 197–199, Jul. 2021, doi: 10.1609/socs.v12i1.18582.
[186] Z. Ren, S. Rathinam, and H. Choset, “MS*: A New Exact Algorithm for Multi-agent Simultaneous Multi-goal Sequencing and Path Finding,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 11560–11565. doi: 10.1109/ICRA48506.2021.9561779.
[187] F. Grenouilleau, W.-J. van Hoeve, and J. N. Hooker, “A Multi-Label A* Algorithm for Multi-Agent Pathfinding,” *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 29, pp. 181–185, 2019, doi: 10.1609/icaps.v29i1.3474.
[188] S. Almagor and M. Lahijanian, “Explainable multi agent path finding,” in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, Virtual, Auckland: International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), May 2020, pp. 34-- 42.
[189] J. Kottinger, S. Almagor, and M. Lahijanian, “MAPS-X: Explainable Multi-Robot Motion Planning via Segmentation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China: IEEE, May 2021, pp. 7994–8000. doi: 10.1109/ICRA48506.2021.9561893.
[190] M. Brandão, A. Coles, and D. Magazzeni, “Explaining Path Plan Optimality: Fast Explanation Methods for Navigation Meshes Using Full and Incremental Inverse Optimization,” *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 31, pp. 56–64, May 2021, doi: 10.1609/icaps.v31i1.15947.
[191] A. Bogatarkan, “Flexible and Explainable Solutions for Multi-Agent Path Finding Problems,” *Electron. Proc. Theor. Comput. Sci.*, vol. 345, pp. 240–247, Sep. 2021, doi: 10.4204/EPTCS.345.40.
[192] 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*, Yokohama, Japan: International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, pp. 4803–4811. doi: 10.24963/ijcai.2020/669.
[193] T. Chakraborti, A. Kulkarni, S. Sreedharan, D. E. Smith, and S. Kambhampati, “Explicability? Legibility? Predictability? Transparency? Privacy? Security? The Emerging Landscape of Interpretable Agent Behavior,” *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 29, pp. 86–96, May 2021, doi: 10.1609/icaps.v29i1.3463.
[194] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. Kambhampati, “Plan explicability and predictability for robot task planning,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore: IEEE, May 2017, pp. 1313–1320. doi: 10.1109/ICRA.2017.7989155.
[195] M. Brandao, G. Canal, S. Krivic, and D. Magazzeni, “Towards providing explanations for robot motion planning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China: IEEE, May 2021, pp. 3927–3933. doi: 10.1109/ICRA48506.2021.9562003.

7 List of acronyms

The following table reports the acronyms used in this deliverable.

Term	Definition
A*	A - star
AAC	All Agent Costs
AA-SIPP	Any Angle Safe Interval Path Planning
AA-SIPP(m)	Any Angle Safe Interval Path Planning (Multi-agent)
AAt-SIPP	Any Angle Safe Interval Path Planning with Turn-in-place (rotation)
AEON	Advanced Engine Off Navigation
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
ARA*	Anytime Repairing A*
ASIPP	Any Time Safe Interval Path Planning
ASP	answer set programming
ASPN	Airport Surface Petri Nets
ASTAIR	Auto Steer Taxi at Airport
ATCO	Air traffic Controller
ATM	Air Traffic Management
AUCI	AI Usage Continuance Intention
BCBS	Bounded Conflict Based Search
BIBOX	Reduction Based Solver (Reduction to Pebble Motion)
CA*	Cooperative A*
CBM	Conflict Based Min Cost Flow
CBS	Conflict Based Search
CBS-MP	Conflict Based Search for Motion Planning with Continuous State Spaces
CBSS	Conflict Based Steiner Search
CBS-TA	Conflict Based Search with Target Assignment
CBS-TA-MLA	Conflict-Based Search with Task Assignment with Multi-Label A*
CBSw/P	Conflict-Based Search with Priority Based Search
CCBS	Continuous Conflict-Based Search
CNN	Convolutional Neural Network

CRF	Conditional Random Field
CSCW	Computer Supported Collaborative Work
CT	Conflict Tree
CTPN	Colored Timed Petri nets
CUG	Collision-free Unit-distance Graph
DKE	Dunning-Kruger Effect
DL	Deep Learning
DNN	Deep Neural Networks
DPS	Dynamic Potential Search
DWA	Translational and Rotational Velocity Selection Algorithm
EASA	European Aviation Safety Agency
ECBS	Enhanced Conflict Based Search
ECBS-CT	Enhanced Conflict Based Search for Motion Planning with State Lattice Representation
ECBS-MP	Enhanced Conflict Based Search for Motion Planning
ECBS-TA-MLA	Enhanced Conflict Based Search with Task Assignment with Multi-Label A*
ETV	Electric Towing Vehicle
FMEA	Failure Mode and Effects Analysis
FSIPP	Focal Safe Interval Path Planning
GCBS	Greedy Conflict Based Search
GH	Ground Handling
GSE	Ground Support Equipment
GSIPP	Generalized Safe Interval Path Planning
HAT	Human Automation Teaming
HCA*	Hierarchical Cooperative A*
HCBS	Hamiltonian Conflict Based Search
HCI	Human Computer Interaction
HMI	Human Machine Interface
ICBS	Improved Conflict Based Search
ICT	Increasing Cost Tree
ICTS	Increasing Cost Tree Search
IOT	Internet Of Things
IPA	Intelligent Personal Assistants
ITA-CBS	Incremental Target Assignment Conflict Based Search

k-QPPTW	k-Quickest Path Problem with Time Windows
LA-MAPF	Multi Agent Path Finding for Large Agents
LIDAR	Light Detection and Ranging
LLM	Large Language Model
LP	Linear Programming
LRA*	Local Repair A*
LSTM	Long Short Term Memory
MA-CBS	Meta Agent Conflict Based Search
MA-ECBS	Meta Agent Enhanced Conflict Based Search
MAMP	Multi Agent Motion Planning
MAPF	Multi Agent Path Finding
MAPF/C+POST	Multirobot Trajectory Planning with Continuous Refinement for Path Smoothing
MAPS-X	Multi Agent Plan Segmenting - X
MC-CBS	Multi Constraint Conflict Based Search
MC-CBS-MS	Multi Constraint Conflict Based Search with Mutex-based Symmetry-breaking
MCTS	Monte-Carlo tree search
MG-MAPF	Multi Goal Multi Agent Path Finding
MG-TAPF	Multi Goal Task Assignment and Path Finding
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MLA	Multi-Label A*
mMAPF	Multi Modal Multi Agent Path Finding
MPC	Model Predictive Control
MR	Merge & Restart
MRdRRT	RRT based Motion Planning Technique
MS*	Exact Algorithm based on MAPF and mTSP
MSVM	Multi-class Support Vector Machine
mTSP	Multiple Travelling Salesman Problem
MXP	Milan Malpensa Airport
NECBS	Nested Enhanced Conflict Based Search
NN	Neural Network
NP	Non-deterministic Polynomial
PAIA	Immune Inspired Multi Objective Optimization Algorithm

PBS	Priority based search
PFM	Potential Field Method
PPCP	Probabilistic Planning with Clear Preferences
PR	Pipe Routing
P&R-LA	Push and Rotate for Large Agents
PRM	Probabilistic Roadmap
PSO	Particle Swarm Optimization
PWL	Piece-wise Linear
RAV	Reachable Avoidance Velocity
RHGA	Receding Horizon Genetic Algorithm
RRT	Rapidly-exploring Random Trees
SAT	Boolean Satisfiability Problem
SCIPP	Safe Interval Path Planning with Focal Search
SIPP	Safe Interval Path Planning
SMT	Satisfiability Modulo Theory
SMT-HCBS	Satisfiability Modulo Theory – Hamiltonian Conflict Based Search
SOS	Swapper Optimization Suite
STL	Signal Temporal Logic
S2M2	Multi-agent Safe Motion Planner
STPA	System Theoretic Process Analysis
SVM	Support Vector Machine
TAP	Transportes Aéreos Portugueses
TAPF	Target Assignment and Path Finding
TCBS	Task Conflict Based Search
Theta*	Theta - star
TL	Temporal Logic
TSN	Time Space Network
U	Potential Field Function
UAV	Unmanned Aerial Vehicle
VOM	Velocity Obstacle Method
WHCA*	Windowed Hierarchical Cooperative A*
WSIPP	Weighted Safe Interval Path Planning
$WSIPP_d$	Weighted Safe Interval Path Planning with Duplicate States

$WSIPP_r$	Weighted Safe Interval Path Planning with Re-expansions
XAI	Explainable Artificial Intelligence
XAIP	Explainable Artificial Intelligence Planning
YUL	Pierre Elliott Trudeau International Airport
ZRH	Zurich Airport