EXPLORATORY RESEARCH

sesar JOINT UNDERTAKING

D2.1 –

Multi-Agent System for Routing

Deliverable ID:	**D2.1**
Dissemination Level:	**PU**
Project Acronym:	**AEON**
Grant:	**892869**
Call:	**H2020-SESAR-2019-2**
Topic:	**Innovation in Airport Operation**
Consortium Coordinator:	**ENAC**
Edition date:	**16 September 2022**
Edition:	**00.01.00**
Template Edition:	**02.00.05**

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Malte von der Burg / TUD	Task leader	05-09-2022
Alexei Sharpanskykh / TUD	Task contributor	05-09-2022
Jorick Kamphof / TUD	Task contributor	05-09-2022

Reviewers internal to the project

Beneficiary	Names	Date
ENAC	Mathieu Cousy, Cyrille Priou, Christophe Pierre	09-09-2022
DBL	Samuele Gottofredi	09-09-2022
Schiphol	Mick van Hattem	09-09-2022

Reviewers external to the project

Name / Beneficiary	Position / Title	Date

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Alexei Sharpanskykh / TUD	Task contributor	15-09-2022
Mathieu Cousy / ENAC	Project coordinator	16-09-2022
Paola Lanzi / DBL	Project member	16-09-2022
Mick van Hattem / Schiphol	Project member	16-09-2022

Rejected By - Representatives of beneficiaries involved in the project

Name and/or Beneficiary	Position / Title	Date

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.02	05/09/2022	draft	M. von der Burg / TUD	Internal review
00.00.03	12/09/2022	consolidated draft	M. von der Burg / TUD	Internal Review
00.01.00	16/09/2022	Release	M. von der Burg / TUD	Release

EUROPEAN PARTNERSHIP

Co-funded by the European Union

AEON

ADVANCED ENGINE-OFF NAVIGATION

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892869 under European Union's Horizon 2020 research and innovation programme.

Abstract

This document outlines the multi-agent system and its path planning algorithm that were developed as exploratory solutions to support the AEON concept of operations. We therefore give a description of the conceptual model underlying the multi-agent system based on the targets for, inputs to, and assumptions of the model as well as the core ideas behind it. Based on that, the multi-agent system with its environmental and agent specifications is delineated. Since the path planning algorithm is at the core of the multi-agent system, we provide a detailed explanation of it, including a summary of the undertaken verification and validation steps.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table of Contents

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Tables

List of Figures

EUROPEAN PARTNERSHIP

Co-funded by the European Union

1 Introduction

1.1 Purpose of the document

This document outlines the main ideas and core principles of the multi-agent system and the path planning algorithms utilized in AEON. Note that the ideas described in this document are possible solutions in line with the exploratory nature of AEON project and its low TRL level.

1.2 Intended readership

The intended audience of this report are mainly the AEON Consortium to be used as reference. However, the intended readership also includes:

- the key stakeholders targeted by the solution, in particular ground handlers, airport management, airlines, ATC operators and the industry providing green taxiing solutions, most of which are also represented in the AEON Advisory Board;
- the overall aviation community interested in the document, as it will be publicly available.

1.3 Related documents

This deliverable builds upon or relates to the following documents:

- D1.1 Initial Concept of Operations, providing the concept that has been assessed in the validation activities.
- D1.3 State-of-the-Art, as basis for the path planning algorithms that are outlined in this document.
- D2.2 Model for Optimal Allocation of Towing Vehicles, outlining the algorithm used for the tug allocation that obtains time and distance information from the path planning algorithm described in this document.
- D3.1 Representative Use Cases, detailing the use cases defined to design the AEON concept and system.
- D3.2 Supervision HMI, outlining the AEON solutions for the human-machine interface for ground control.
- D4.1 and D4.2, presenting the description of the platform used for the real-time simulation arranged as the final validation session.

1.4 Structure of the document

The next sections are structured as follows:

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- Section 2 outlines the conceptual model including assumptions and core ideas
- Section 3 describes the multi-agent system and brief link to the other AEON solutions
- Section 4 elaborates on the path planning algorithm
- Section 5 describes the verification and validation steps
- Section 6 contains references.

1.5 Acronyms and terminology

Term	Definition
AAS	Amsterdam Airport Schiphol
A-CDM	Airport Collaborative Decision Making
AEON	Advanced Engine Off Navigation
ATC	Air Traffic Control
ATCO	Air Traffic Controller
CONOPS	Concept of Operations
CTOT	Calculated Take-Off Time
DNE	Do-Not-Enter restriction
DNP	Do-Not-Persist restriction
EFB	Electronic Flight Bag
ELG	Electric Landing Gear, either installed in nose or main landing gear
HMI	Human-Machine Interface
HP	Holding Point
ICAO	International Civil Aviation Organisation
MAS	Multi-Agent System
PBS	Priority-based Search
RET	Rapid Exit Taxiway
RMO	Runway Mode of Operation
SET	Single-Engine Taxiing
SI	Safe Interval
SIPP	Safe Interval Path Planning
TB	Tug-Enabled Taxiing
TLDT	Target Landing Time
TOBT	Target Off-Block Time
USI	Unsafe Interval

EUROPEAN PARTNERSHIP
Co-funded by the European Union

2 Conceptual Model

In this section, the requirements from AEON's CONOPS (D1.1) are translated into targets for (section 2.1) and inputs to (section 2.2) the path planning algorithm, and necessary assumptions (section 2.3) are formulated. Furthermore, the core ideas behind the routing system are described in section 2.4.

2.1 Path Planning Targets

Due to the exploratory nature of this project, we formulate the requirements outlined in D1.1 as targets for the path planning algorithm. Information obtained in the other related documents are added to the following list as well:

- path planning results in safe, i.e., conflict-free, routes
- path planning takes the various new taxiing techniques outlined in D1.1 into account
    - single-engine taxiing (SET)
    - tug-enabled taxiing (TB)
    - electric landing gear (ELG), either installed in nose or main landing gear
- path planning takes coupling/decoupling, engine-start, and pushback operations into account
- path planning can be done for both aircraft and tugs using fixed kinematic values per taxiing technique for maximum speed, maximal speed in turns, acceleration and deceleration rates
- path planning outputs the speed profile of each vehicle, and the time point at which the engines have to be started for aircraft using engine-off taxiing.
- path planning is automatically re-done in case the executed route deviates substantially from the planned one
- route suggestions are posed to ATCOs, who are able to influence and modify the path planning
    - prioritization levels of e.g. inbound vs. outbound flights or tugs vs. aircraft
    - standard taxiway directions can be set
    - waypoints can be selected
- the routing system interacts with other AEON solutions like the human-machine interface to pass information

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2.2 Inputs to Path Planning

In accordance with the requirements defined in D1.1, the path planning algorithm needs the following information as input:

- A-CDM data or flight schedule including TLDT or TOBT, CTOT, default gate and runway locations (can be based on a set of possible locations, e.g. viable runway entries), usable taxiing techniques (for consistency checks), assigned taxiing technique (to select corresponding kinematic values from internal database)
- runway assignment in the flight schedule has to match the runway mode of operation (RMO, defining active vs. inactive runways). Changes to the RMO shall be sent to the MAS as early as possible including the time that the new RMO becomes effective.
- task assignments of tugs from tug allocation module
- vehicle properties of all aircraft and tugs, e.g., length, width/wingspan, distance needed for landing / take-off, etc. as well as kinematic values, i.e., maximal velocity, reduced velocity for turns together with corresponding radius of curvature for which this is applicable, fixed acceleration / deceleration rates per vehicle type
- process times: e.g. coupling / decoupling time, engine-start time (including the warmup-time), time for switching direction (e.g. in case of pushback)
- once a route is (partially) cleared, its cleared path needs to be provided to the routing module
- current speed and position for taxiing vehicles
- any changes to the information above

2.3 Modelling Assumptions

Given the above defined targets for and inputs to the path planning algorithm, we formalize the following modelling assumptions:

- digital means of communication via a datalink such as AeroMACS are the norm throughout the operations to facilitate data sharing, e.g. providing speed profiles to pilots and tug drivers
- routing for aircraft takes place between a ramp and the stop-bar of a runway for outbound aircraft or vice versa for inbound aircraft. The start location has to be specified, while the goal locations can be expressed as a set of possible locations, such as multiple runway-entries or decoupling locations.
- in path planning, all aircraft are categorized as one of the 6 aircraft types from the ICAO aerodrome reference codes [1]. They are assumed to have a circular size with one of the following shape diameters:

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table 1: Categorization of aircraft sizes in path planning

ICAO-type	shape diameter [m]	exemplary aircraft type
ICAO-A	12	Cirrus Vision SF50
ICAO-B	25	Cessna, Learjet
ICAO-C	40	A320, B737, EMB 170/190
ICAO-D	54	A300F, B767
ICAO-E	72	A350, B747, B787
ICAO-F	80	A380, B747-800F

- path planning is performed with kinematic values for its maximum speed, maximal speed in turns, fixed acceleration rate, and fixed deceleration rate. This set of kinematic values is specific to the vehicle type and taxiing technique of the corresponding aircraft type.
- aircraft cannot pass each other on the same taxiway. In comparison to that, the service road infrastructure, if existent, does allow for tugs to do so.
- every departure aircraft with a push-back or push-pull manoeuvre has either a pushback-truck or a tug coupled to it at the respective TOBT used in path planning.
- all outbound aircraft use standard pushback paths if available (e.g. as for Amsterdam Airport Schiphol, see [2]). It is assumed that the pushback-truck or tug for the pushback-manoeuvre is already coupled to the aircraft when routing commences.
- when using tug-enabled taxiing, aircraft can start their engines during taxiing. The coupling or decoupling process is simplified to a pre-defined time duration in path planning.

2.4 Translation into Core Features of the Path Planning Algorithm

Multi-Agent Planning with Priorities

To deconflict the routes of all vehicles travelling concurrently on the airport surface, multi-agent planning is deployed. When a conflict in the initial paths of two vehicles is found, it is solved by assigning priority to one of the two conflicting vehicles, building a priority tree. Priority is assigned based on the smallest influence on the weighed sum of all objective functions, each yielding a vehicle-specific cost for its current route. The route of the deprioritized vehicle has to be adapted, either by changing its path or altering the speed profile along the path.

Online System

Since new aircraft continuously land and depart from the airport, we perform online path planning. For that, a rolling-horizon scheme with a planning window of w_{plng} and replanning period h_{plng} is used. Only those flights that are within w_{plng} from the current time point of planning are part of the corresponding planning round. Replanning can be triggered at any time but is done latest after h_{plng}.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Planning with Constraints and Preferences

During path planning, constraints and preferences for taxiway-usage are taken into account. Constraints limit the possible set of taxiway segments that can be traversed, either in both directions or one-way. They can be used to represent blocked taxiway-segments such as a crossing of an active runway, specified by the current runway mode of operation (RMO), or to assign fixed directionality of taxiways. In comparison to that, preferences influence the chosen path by assigning weights to the cost of traversing this taxiway segment. With that, the algorithm is able to suggest an alternative that might deviate from the standard traffic rules (cp. D4.1 for further details), but results in an improved route from the perspective of the overall routing.

The prioritization level of e.g., inbound vs. outbound flights or tugs vs. aircraft can be modified with the priority weight factors described above. This will influence the priority order between different vehicles that is created during path planning. Additionally, the vehicle-specific objective functions can be adjusted to influence the computed path. These are currently based on a linear combination of taxi time and distance but may later be extended with other cost-parts.

Planning with Kinematics

Since heavy vehicles such as aircraft entail finite acceleration and deceleration, we plan with kinematics. For that, a set of kinematic values is used: maximum speed v_{max}, maximal velocity in turns v_{turn}, velocity when exiting the runway via rapid-exit-taxiways (RETs) v_{RET}, constant acceleration rate acc, and constant deceleration rate dec. This has to be defined for each vehicle type and each taxiing technique. An exemplary speed profile for an inbound aircraft leaving via an RET is shown in Figure 1.

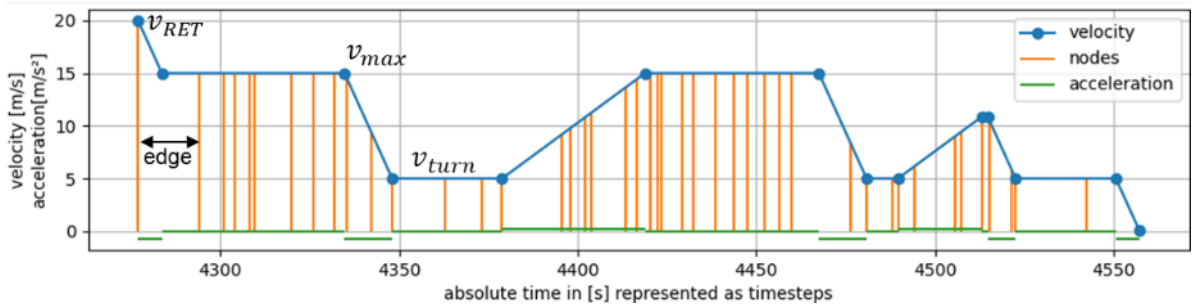


Figure 1: Exemplary velocity profile of the route of a landing aircraft. Orange lines: junction points in taxiway network (nodes). Vehicles travel along centrelines (edges).

Activity-Based Planning

To take the various operations such as pushback and coupling / decoupling of tugs into account in path planning, we express them as one of the following three activities:

- go-to activities comprise a start location and a set of goal locations, so that the planning algorithm gets two degrees of freedom: time and route. This activity is used as input to taxi from one point at the airport to another point at the airport, for example for regular taxiing.
- follow activities consist of a predefined list of path segments that must sequentially be part of the route. Therefore, time is the only remaining degree of freedom in the planning algorithm, and the path cannot be changed. This activity is used for instance for pushback and push-pull manoeuvres.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

- wait activities prescribe a waiting location and waiting duration that have to be accounted for in path planning. Operations such as coupling to or decoupling from a tug are examples of such.

Based on these activities, an activity sequence can be defined to express the necessary operations for both inbound and outbound flights for all specified taxiing techniques as well as the movements of tugs that are not coupled to an aircraft. As example, Figure 2 depicts this sequence for different taxiing techniques of an outbound flight.

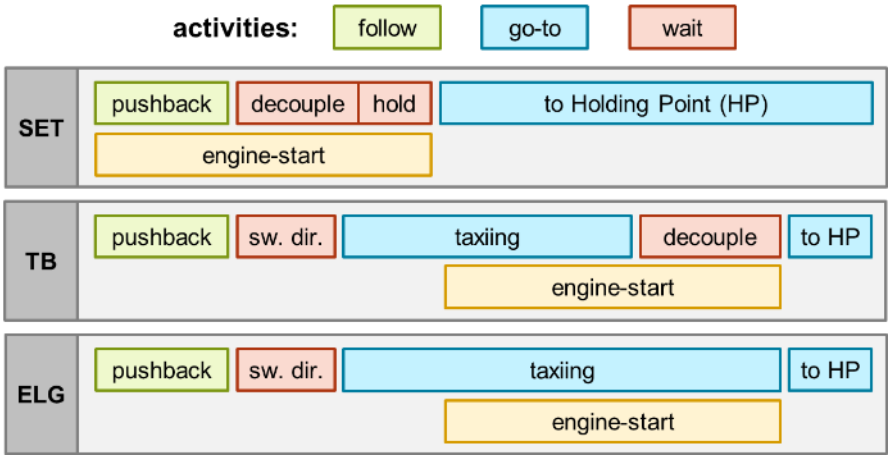


Figure 2: Exemplary activity sequence for different taxiing techniques of an outbound aircraft. "sw. dir.": switch direction

The warmup and cooldown of the engines represent a special case. The algorithm takes the warmup phase as part of the engine-start manoeuvre and on basis of the aircraft-specific engine-start duration as input value into account. Therefore, if this duration exceeds the time needed till decoupling from either tug or pushback-truck, additional waiting in form of holding is added to the route (cp. Figure 2, SET-case). We do not model engine cooldown, as it does not have an influence on the routing regarding the kinematics, since the engines are switched off after standstill at the gate or coupling point for tug-enabled taxiing.

If an ATCO has (partially) cleared a path of a vehicle, this segment is converted into a follow-activity. Thus, during replanning, the path planning will only allow the time spent on this segment to be altered, while the cleared route remains.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3 Multi-Agent System

In this section, the specification of the multi-agent system (MAS) is described and its interaction with other AEON solutions briefly outlined (cp. Figure 3). It comprises a distributed-hierarchical structure of both centralized and distributed agents. The centralized Routing Agent computes conflict-free routes for all vehicles that are scheduled to be taxiing within the predefined planning window w_{plng}. We use motion planning to account for vehicle kinematics in planning since heavy vehicles such as aircraft entail finite acceleration and deceleration. To ensure conflict-free paths, we deploy a two-level search based on Priority-Based Search (PBS) [1] with an augmented version of the Safe Interval Path Planning (SIPP) algorithm [2], which are outlined in section 4. Once a route is cleared by ATC, the Localized Agents, in their current form conceptualized to be positioned at every junction, monitor the execution of the cleared route and trigger central replanning if the deviation exceeds time thresholds.

The MAS can be utilized by the fleet management algorithm to obtain estimates for time and distance of paths, see D2.2 for details. Furthermore, the MAS interacts with the human-machine interfaces (HMIs, see D3.2) to facilitate co-design of the vehicle routes. For that, the Routing Agent takes constraints of the ATCOs into account during path planning and poses route suggestions to them. Additionally, the vehicle-specific objective functions can be adjusted to influence the path planning. These are currently based on a linear combination of taxi time and distance but may later be extended with other cost-parts. Besides that, the estimated arrival time and remaining distance per vehicle as well as the deviations during plan execution computed by the Localized Agents can be visualized in the HMI to keep stakeholders up to date.

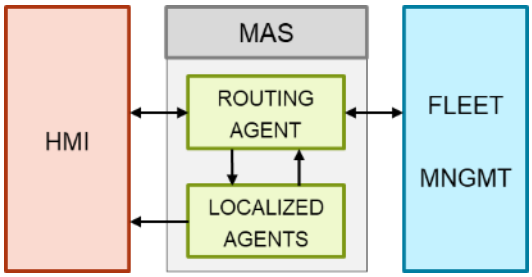


Figure 3: Multi-agent system and interaction with HMI and fleet management algorithm

3.1 Environment Specification

The airport taxiing infrastructure is represented by a graph G = (V, E) with nodes V and directional edges E, and is shown in Figure 4. Nodes represent taxiway intersections (yellow), aircraft ramps (red), or decoupling locations (blue). The latter are specific to each runway and are used for tugs to couple to or decouple from aircraft that taxi with Tug-Enabled Taxiing. Each bidirectional taxiway segment between two nodes is constructed from two unidirectional edges that connect the nodes. Taxiway edges (black) are obtained from the actual locations of these taxiways at Amsterdam Airport Schiphol. Ramp edges (red) represent the aircraft parking positions. The airport layout graph is fully accessible to the Routing Agent and remains static throughout the simulation. However, edges can be constrained to prohibit vehicles to be routed over these, e.g., in accordance with the current runway mode of operation (RMO) or due to temporally unavailable taxiway segments. Further details on the airport map and rules are provided in D4.1.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

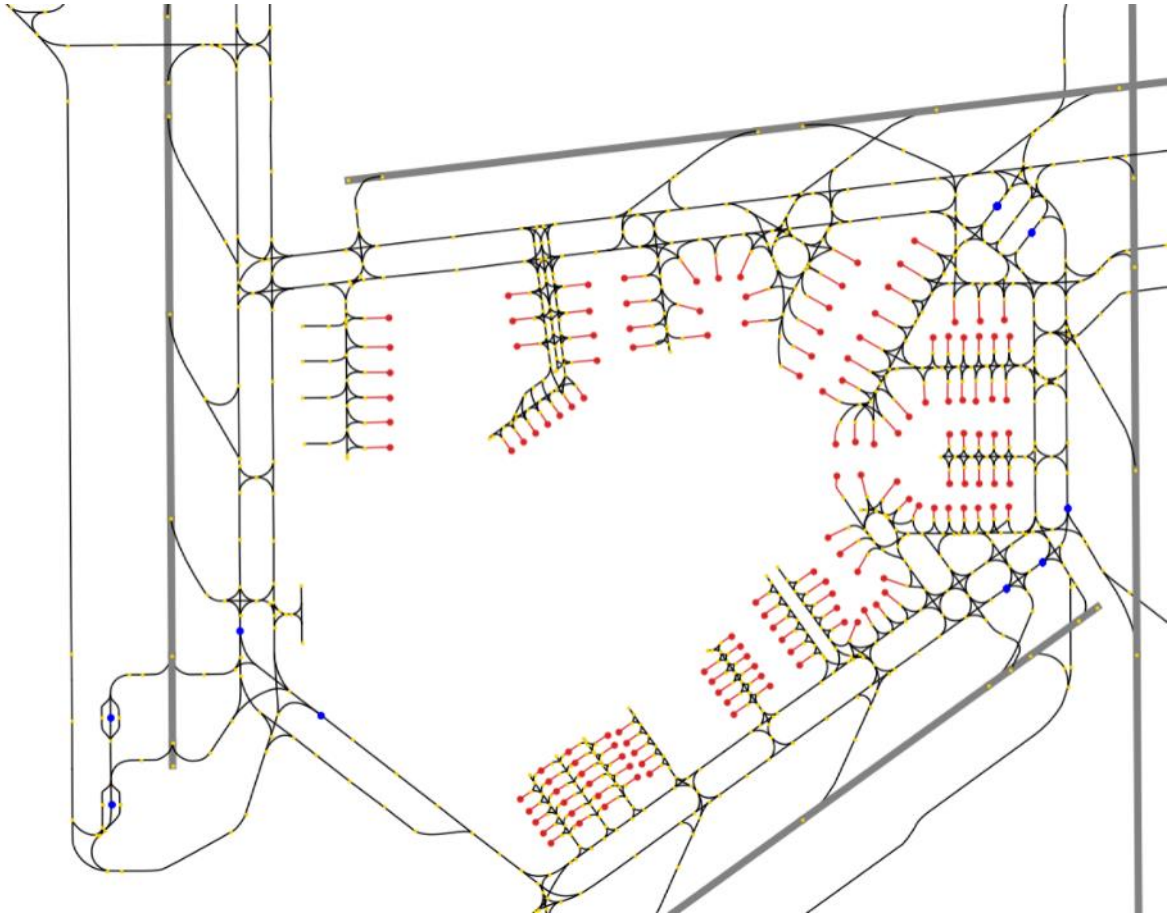


Figure 4: Part of the graph representation of Amsterdam Airport Schiphol. Taxiway edges (black), taxiway nodes (yellow), ramp nodes and edges (red), decoupling nodes (blue), and runways (grey)

3.2 Specification of Routing Agent

The role of the Routing Agent is to perform the central route planning. It generates a plan for all vehicles within the planning window w_{plng} at least every replanning period h_{plng}. For this, it uses the routing algorithm as specified in section 4. The Routing Agent obtains local information from the Localized Agents, performs central planning and returns the routes of all vehicles of that planning round. The properties of the Routing Agent are as follows:

P1. Check Replanning Property:
At each timepoint t, the Routing Agent checks on an internal counter if replanning is due, constraints such as the RMO changed, or if it received any replanning requests from one of the Localized Agents. If so, the internal counter is reset, and the path planning is commenced by executing P2.

P2. Get Flights Property:
If replanning is due, the Routing Agent adds the flights that are taxiing or that spawn within the planning window to the path planning. For each vehicle, the agent gets the ID, origin, initial velocity, destination, shape, and kinematic properties. Next, it executes P3 to create an activity sequence per vehicle and P4 to update its constraint-database. The actual path planning is then started with P5.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

P3. Create Activity Sequence Property:
The Routing Agent defines a sequence of activities for each vehicle to the planning. The path finding algorithm uses this sequence of activities as input. This is done so that activities can be forced to be on a vehicle's path in a predefined sequence, as outlined in section 2.4, "Activity-based Planning".

P4. Update constraints and ATCO preferences:
This property updates the Routing Agent's internal database for all constraints and ATCO preferences that may influence the planning round. Further details on the constraints and ATCO preferences are provided in section 2.4, "Planning with Constraints and Preferences".

P5. Generate Routes Property:
After executing the previously described properties, the Routing Agent uses the path planning algorithm outlined in section 4 to generate routes for all flights that are part of the current planning round.

3.3 Specification of Localized Agents

While the Routing Agent plans the routes, the Localized Agents' role is to monitor the execution of the cleared routes. As optional feature, these agents are capable of providing text-based route instructions to pilots and tug drivers, when for instance the electronic flight bag (EFB, cp. D1.1) is not available. They have a fixed location on the map and are always placed at a node. The Localized Agents have the following properties:

P1. Get Positions and Status Property:
At each time point, the Localized Agents get the position, heading, and velocity of all vehicles under their control.

P2. Handover Control Property:
At each time point, a Localized Agent checks whether a vehicle passed itself during the previous time step. If so, it hands over the control responsibility to the next, closest, Localized Agent on the vehicle's route. If a vehicle reaches its goal node, the information connected to this vehicle is placed in storage.

P3. Automatic Instruction Property (optional):
In general, the Localized Agents can instruct the pilots or tug drivers automatically. In this case, all instructions for the next predefined time duration are communicated based on the distance from the current location where the instruction should be executed by the pilot / tug driver. Instructions can be of four types: velocity instructions, heading instructions, coupling/decoupling instructions, and engine-start instructions.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

4 Path Planning Algorithm

The path planning algorithm consists of a two-level search, the high-level and the low-level. In the high-level search, we adopt the search from the Priority Based Search (PBS) algorithm [3]. PBS maintains a priority tree and creates two new child nodes whenever it detects a conflict between two vehicles. In each child node, a new priority order is added, thereby constraining one of the two agents to avoid the other's path. Based on sum of cost of the paths of all agents in each child node, the algorithm then chooses which priority order results in the lowest overall cost and expands this node next. In our implementation, we define cost as the sum of taxi times. As in PBS [3], we use a depth-first search in which the algorithm only backtracks when it cannot find a solution in the explored branch. The high-level search continues until the algorithm expands a child node without any collisions.

To respect the priority order set in the high-level search, we translate all paths into a set of graph reservations. In short, all aircraft temporarily block a set of edges during each movement between one node and another. The blockage times and set of blocked edges are dependent on the agent's shape, velocity profile, and the shapes of other agents. The concept of graph reservations and conflict avoidance will be further discussed in section 4.1.

In the low-level search, we use an adapted version of the Safe Interval Path Planning (SIPP) algorithm [4]. SIPP represents moving obstacles as collision intervals and subsequently defines a set of Safe Intervals (SIs) per graph location, representing time intervals during which an agent can occupy that location. Our implementation of SIPP searches for a single-agent path, thereby respecting the constraints from the high-level PBS search and the activity sequence as defined by the Routing Agent. We set the graph reservations in such a way, that the shape of the agent in the low-level search can be disregarded. Before performing the SIPP search, we translate the graph reservations into a set of Safe Intervals (SIs) and restrictions relevant for that vehicle. This translation is described in more detail in section 4.2. With those SIs defined, the algorithm efficiently computes a route with the time dimension included. Note that this is different from the original PBS algorithm, which uses space-time A* to perform low-level search. The details of the low-level search are described in section 4.3.

The necessary replanning due to the rolling-horizon approach is outlined in section 4.4.

4.1 Graph Reservations and Conflict Avoidance

To efficiently account for the shapes of agents, we translate each vehicles' plan into a set of reserved edges. The concept of graph reservations is based on the idea of translating the movement of a dynamic obstacle to a set of Unsafe Intervals (USIs) for a lower prioritized vehicle. Philips and Likhachev [4] denote these as collision intervals. We use three main inputs for making graph reservations: 1) the combined shape of one agent a_i and another agent a_j and 2) the velocity profile of agent a_i and 3) the distances between edges in the layout. This works as follows:

The combined shape represents the shape r_{a_i} of agent a_i, the shape r_{a_j} of agent a_j, and a predefined safety distance D. We simplify the shape of each agent to be a circle with radius $r = max(W, L)$ where W is the vehicle's width and L the length. The combined shape r_s of two agents a_i and a_j is then calculated as follows:

$$r_s = r_{a_i} + r_{a_j} + D$$

EUROPEAN PARTNERSHIP

Co-funded by the European Union

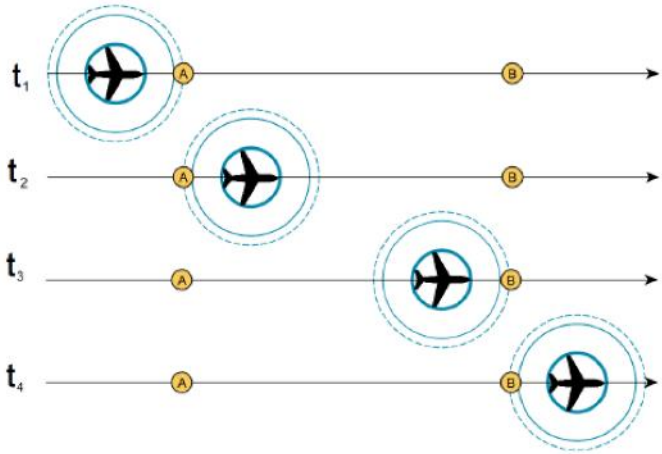
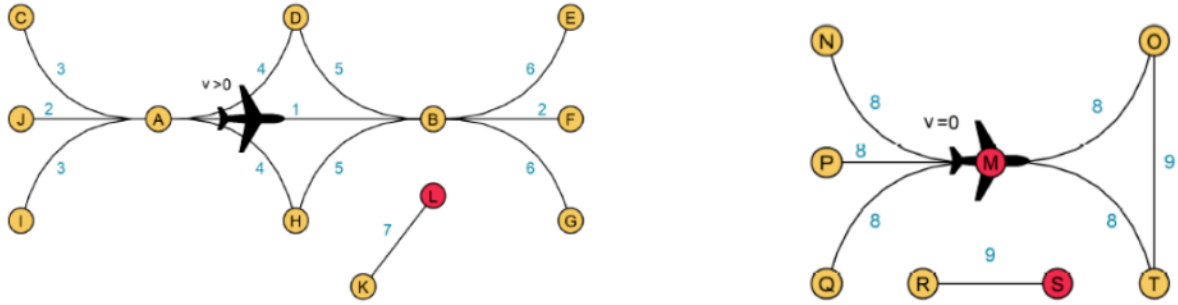


Figure 6: Calculated time points t_1, t_2, t_3, and t_4 relative to each edge on an agent's route. Inner circle represents area of agent with radius r_{a_i}, middle circle has radius $r_{a_i} + r_{a_j}$ representing the area of both agents, and outer dashed circle represents the combined shape with radius r_s.



(a) Grouping of edges when $v \neq 0$

(b) Grouping of edges when $v = 0$

Figure 5: Edge reservations for zero and non-zero velocities.

Based on this r_s and the velocity profile of agent a_i, the start time and end time of the graph reservation are determined. We calculate four time points relative to each edge traversed by the centre of agent a_i: t_1, t_2, t_3, and t_4. Figure 6 shows these points for an example agent on edge AB. t_1 represents the time point that r_s touches the start node A of the edge A-B. t_2 is the moment r_s clears the start node of the edge. Finally, t_3 and t_4 represent the time points that the combined shape touches and clears the end point B of the edge A-B, respectively. If an aircraft stops at a node, we only define t_1 and t_2, i.e., the moment r_s touches the node and the moment r_s clears the node. These time points are computed for every path entry of agent a_i and for every combination of radii of flights in the flight schedule.

Once t_1, t_2, t_3, and t_4 are determined for each edge and waiting node in the plan, the surrounding edges and nodes are considered. For each edge in the route of agent a_i, the algorithm determines the neighbouring edges and the non-connected edges that are within a distance $r_{a_i} + r_{a_j}$ from an edge. Furthermore, any nodes in the vicinity on which a waiting activity can be performed are reserved. We developed two reservation mechanisms to constrain the neighbouring edges and nodes: 1) A reservation mechanism used when agent ai has a non-zero velocity $v \neq 0$ and 2) a reservation mechanism used when agent a_i has a zero-velocity due to an ongoing waiting activity. Furthermore, we distinguish between two types of graph reservations: Unsafe Intervals (USIs) and restrictions. USIs can prohibit other vehicles to enter the edge with a Do Not Enter (DNE) USI or to be present on the edge with a Do Not Persist (DNP) USI. Restrictions are additional constraints that must be respected

Co-funded by the European Union

Table 2: Edge types for reservation mechanism that is used when $v \neq 0$.
⇒: in direction of traversal, ⇐: in opposite direction, DNE: Do Not Enter, DNP: Do Not Persist.

Edge Type	Description	Unsafe Intervals (USIs)	Restrictions
1	Edge currently traversed by agent a_i.	⇒ $[t_1, t_2, DNE]$ ⇐ $[t_1, t_4.DNP]$	⇒ if entered before t_1, leave before t_3. ⇒ if entered after t_2, leave after t_4.
2	Any other edge on the route of a_i within 150 meters from the current edge.	-	-
3	Edges to the start node of the current edge that make an angle > 135° with current edge.	-	-
4	Edges from the start node of the current edge that make an angle < 135° with current edge.	⇒ $[t_1, t_2, DNE]$ ⇐ $[t_1, t_4.DNP]$	⇒ if entered before t_1, leave before t_3*. ⇒ if entered after t_2, leave after t_4*.
5	Edges to the end node of the current edge that make an angle < 135° with current edge.	⇒ $[t_1, t_4, DNP]$ ⇐ $[t_1, t_4.DNP]$	-
6	Type 4 edges of the next edge on agent a_i's route.	-	-
7	Any non-connected edge to the current edge within $r_{a_i} + r_{a_j}$ meters of any point on the current edge.	⇒ $[t_1, t_4, DNP]$ ⇐ $[t_1, t_4.DNP]$	-
Node L	Nodes on which a waiting activity can be performed within $r_{a_i} + r_{a_j}$ meters of any point on the current edge.	$[t_1, t_4, DNP]$	-

Table 3: Edge types for reservation mechanism that is used when $v = 0$.

Edge Type	Description	Unsafe Intervals (USIs)
8	Edges from the node where a_i is waiting.	⇒ $[t_1, t_2, DNE]$
9	Any non-connected edge to the current edge within $r_{a_i} + r_{a_j}$ meters of any point on the current edge.	⇒ $[t_1, t_2, DNP]$ ⇐ $[t_1, t_2.DNP]$
Node M	The node on which waiting is currently being performed by agent a_i.	$[t_1, t_2, DNP]$
Node S	Nodes on which a waiting activity can be performed within $r_{a_i} + r_{a_j}$ meters of any point on the current edge.	$[t_1, t_2, DNP]$

by lower prioritized vehicles to prevent overtaking. The two reservation mechanisms are further explained below.

1) Reservations with non-zero velocity $v \neq 0$

Figure 5 (a) shows the reservation mechanism used to set constraints when agents are moving. Reservations are always set from edges on the vehicle's route one-by-one. In this example, we consider the movement from node A to node B in Figure 5 as an example. We first group the edges around edge A-B based on their locations. This grouping is also shown in Figure 5, with the corresponding Unsafe Intervals (USIs) and restrictions per edge type defined in Table 2.

The edge labelled with a 1 always refers to the edge from which we are setting reservations. On this edge type 1, the constraining vehicle sets a DNE USI from $[t_1, t_2]$ in its own direction, prohibiting other vehicles to enter. In addition, it adds two restrictions for vehicles that use this edge to prevent overtaking: 1) if the other vehicle enters before t_1, that vehicle should leave the edge before t_3 and 2) if the other vehicle enters after t_2, that vehicle should not leave the edge before t_4. The first restriction prevents the other vehicle to be overtaken by the own vehicle, whereas the second restriction prohibits the other vehicle to overtake the own vehicle. To prevent head-on collisions, the USI on the opposite edge of type 1 is from $[t_1, t_4]$. This USI restricts other vehicles to be present on that edge between t_1 and t_4. The USIs on edge type 4 are the same: $[t_1, t_2]$ in the direction of traversal and $[t_1, t_4]$ in the opposite direction. Restrictions to prevent overtaking are also similar, but the t_3 and t_4 time points are calculated based on the distance on the type 4 edge. The USIs on edge types 5 and type 7 are from $[t_1, t_4]$ in both directions, restricting other vehicles to be present on those edges. Finally, all nodes on which a waiting activity can be performed and that are located within $r_{a_i} + r_{a_j}$ meters of any point on the current edge, are constrained from t_1 to t_4 with a DNP USI.

2) Reservations with zero velocity $v = 0$

Figure 5 (b) shows the reservation mechanism used when velocities are equal to zero, and the USIs for this case are listed in Table 3. We set USIs on all outgoing edges from the moment the vehicle arrives at the node (t_1) until the moment that it clears the node (t_2). This prevents other vehicles to enter

EUROPEAN PARTNERSHIP
Co-funded by the European Union

those edges during these intervals. In addition, edges within a distance $r_{a_i} + r_{a_j}$ from the node are constrained from t_1 until t_2 in both directions for other vehicles to be present.

4.2 Conversion of Graph Reservations to Safe Intervals and Restrictions

Before starting the low-level search, we collect all relevant graph reservations for the agent a_i. Given the graph reservations and the Routing Agent constraints, a set of Safe Intervals, restrictions, and no-go edges is computed for a_i. First, the Unsafe Intervals and restrictions set for the ICAO type of the a_i are collected from all agents higher in priority. Constraints set by the routing agent are also translated in USIs and added to this set. Overlapping intervals are joined together and merged. If the time interval between the end of one USI and the start of another USI is smaller than the edge length l divided by the agent's maximum velocity v_{max}, we also merge the USIs into one, as the agent will never be able to traverse that edge. Finally, we create all intervals between $[t, t + w_{plng}]$ that do not overlap with the USIs, where t is the time of planning and wplng is the planning window. The result is a set of Safe Intervals for this agent a_i and a set of restrictions that should be respected in the low-level search.

4.3 Low-Level Search

The low-level search is performed sequentially, based on the activity sequence generated by the Routing Agent. As described before, this sequence is created by the Routing Agent before planning and consists of go-to, follow, or wait activities. This section first presents the state definition and then elaborates on the activity-based search. In this search, we first generate a set of initial states for a single agent. From these initial states, we generate successors in two ways: regular successor generation and anticipatory successor generation. In regular successor generation we generate successors on the adjacent node of a vehicle's current node. In anticipatory successor generation, successors are generated on non-adjacent nodes that are within the agent's braking distance away from the current node.

State Definition:
In general, a state is defined by a configuration and a Safe Interval: (*cfg*, SI). We define the configuration *cfg* as a set of the current node, the next node, velocity, and the activity ID (N, N_n, v, act_{ID}). The safe interval SI corresponds to the edge (N, N_n) composed by the node and the next node. An exceptional case occurs for states with a velocity of 0 m/s. These are created with a safe interval on the node N instead of the edge. While not used for identification of the state, the cost, an arrival time, and a purpose (wait/move) are added to the state as additional information.

A. Initial State Generation:
Before starting the activity-based search, the algorithm determines the start state for each agent. If a vehicle is taxiing, the initial state is generated at the node that this vehicle approaches, with the corresponding configuration and SI at that point. If the vehicle has an initial velocity of 0 m/s, we check two additional conditions: 1) are engines needed and started and 2) can this vehicle still wait at the current node. If the engines are needed but not started yet, the algorithm checks whether the start time of the engines is within the current SI on the node. If so, it generates a state. Else, the algorithm returns without a solution, as engine start is not possible, and the vehicle is not able to perform any other motions without the engines started.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

B. Regular Successor generation:
Starting from the initial state, we generate successors based on the feasible motions that can be performed from this state. We assume that a vehicle always moves to a neighbouring node as quickly as possible unless there is an USI or restriction that prevents the vehicle from doing so. Furthermore, we assume that the velocity profile on an edge is monotonic. In case there are no USI or restrictions on the neighbouring edges of the next node, we optimize for minimum traversal time. Thereby we accelerate at the start of the edge and decelerate at the end of the edge. The result is a configuration and a SI. If the arrival time in that state is within a SI, we add the state as a successor state. In case there are USIs, we optimize for maximal final velocity on the edge and aim to arrive at exactly the start of the earliest Safe Interval. If there is no feasible motion, no successors are generated.

In the motion generation, we are bound to the vehicle's kinematic properties for the current activity and the velocity in the current state. A motion that is part of the follow-activity for push-back is for example constrained by a lower maximum velocity than regular taxiing in a go-to activity. In addition, vehicles that have maximum velocity in the current state, might not be able to decelerate enough to satisfy an USI on the next edge or node. In this case, it might be required to start decelerating on the edge before the current state. To efficiently account for this, we anticipate based on the vehicle's current velocity, braking distance, and USIs or restrictions within the braking distance.

C. Anticipatory Successor Generation:
Whenever the algorithm opens a state, it explores the direct neighbours first. To anticipate on future USIs, restrictions, or turns for which agents need to slow down, anticipated successors are also generated. To do this, the algorithm considers all edges within the braking distance away from the node in the explored state. All successors that can be generated from that state are then considered. If no feasible motion is found from the explored state to the potential successor state, the algorithm checks whether the potential successor state can be reached from the state that it opened. If so, it generates the potential successor state at a non-neighbour node from the opened state.

4.4 Replanning

As discussed before, replanning is performed every h_{plng} minutes for the next w_{plng} minutes. As we plan with continuous time, agents can be in between two nodes when replanning is performed. Localized Agents estimate the arrival time of the agent on the next edge and pass this as input to the Routing Agent. The USIs and constraints set for this agent that already started are copied and used as root constraints to the next planning. In this way, other agents avoid the edges that this agent is blocking already.

EUROPEAN PARTNERSHIP
Co-funded by the European Union

5 Verification & Validation

Verification and validation of the simulation model were performed in accordance with validation techniques and tests as described by Sargent [5]. To validate the conceptual model, operational experts from Amsterdam Airport Schiphol were consulted as well as manufacturers from the TaxiBots. During implementation, continuous verification was performed. The model was developed in different modules, allowing for the independent testing of the building blocks. In addition, assertion conditions were added to ensure correctness of the internal processes in the code and compiler errors were resolved. Visual animations were used to verify that the routes were executed as planned. Furthermore, small test scenarios were created to verify the model's behaviour in the bay areas for pushback, push-pull, and engine start manoeuvres, and at the decoupling locations to verify the decoupling processes. Furthermore, the activity-based path planning was verified with small test scenarios ensuring correct timings and kinematics. With these scenarios, face validation was performed to ensure that the model performance was as expected. Finally, individual agent behaviour was carefully followed throughout the system to ensure correctness.

The integration into the simulation platform helped to further verify the algorithm and validate the core ideas behind it. Not all of the features described above, such as anticipatory routing based on RMO changes, are implemented in the simulation platform so that they could not be tested. Additionally, the algorithm was mainly tested with the airport layout of Amsterdam Airport Schiphol. The one of Paris Charles de Gaulle Airport used automatically generated pushback-paths that posed difficulties for the path planning algorithm. Furthermore, the tests also showed current limitations of the path planning algorithm in coping with human input and quickly arising deviations to the planning. The latter had the consequence of increased replanning that overwhelmed the current runtime capabilities of the system. The rather low level of automation (e.g. datalink not compulsory) that was used during the final evaluation of the AEON solution (cp. D5.2) violated the input requirements for the routing, making it infeasible to use in the real-time simulation. Thus, more research is needed to address the multitude of special cases that arise when the MAS interacts with humans.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

6 Solution Maturity

The multi-agent system and the path planning algorithm were implemented as outlined above: all path planning targets defined in section 2.1 are met. However, as described in section 5, it was not possible to use this solution in the final evaluation study, since the required robustness for the real-time simulation with human interaction was not reached. Therefore, to provide additional insights into the maturity of the proposed solution, we draw upon past and ongoing research at TU Delft[1] in which an augmented version of the multi-agent system was developed to run stand-alone simulations without any human interaction.

While this simulator uses the same path planning algorithm as outlined in this document, assumptions deviating from those of the AEON solution (cp. section 2.3) were made:

- operations are fully automated, i.e. no human interaction with system
- static flight schedule
- no departure sequence at runways, no CTOT slots that have to be adhered to
- all outbound aircraft use tug-enable taxiing (TB), all inbound use single-engine taxiing (SET)
- unlimited amount of tugs available, i.e. no task assignment required; tugs return to base mainly via service roads after decoupling
- path planning is performed with exact knowledge of the vehicle kinematics
- standard taxiway directions do not exist, i.e. no traffic rules
- all vehicles are instructed by the Localized Agents on the spot (cp. its property P3), i.e. clearance for an entire route is not necessary
- all vehicles execute these instructions perfectly, i.e. no deviations to planned routes
- simulation is executed sequentially, i.e. paused when routes are planned

These assumptions give the path planning algorithm more freedom to optimize, limiting the risk of incomplete solution, i.e. situations in which the routing cannot be done. Furthermore, simulations were only done with the layout of Amsterdam Airport Schiphol. This layout is also more mature than the one of Paris Charles de Gaulle Airport (CDG), as for instance the standard pushback paths from airport manuals were integrated into the layout, in comparison to automatically generated ones for the CDG airport map.

Based on simulations using a flight schedule of the two busiest days at Schiphol in the year 2019, and enriched with findings during the AEON evaluation sessions, we made the following insights into algorithm maturity:

[1] references cannot provided at the time of writing, as the work is not published yet.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Scalability of Collision-free Routing

In general, the adopted routing algorithm was successful in collision-free routing of the scheduled flights in case studies with high traffic load. However, in rare cases with peak traffic load, the shapes of aircraft in queues overlapped for aircraft with a low velocity. Further research is needed to mitigate this.

Computational efficiency

The algorithm is implemented in Python, which leads in general to higher runtimes in comparison to compiling programming languages such as C or C++. In the simulation, two parameters influence the duration of path planning the most: the number of vehicles to be routed concurrently, and the time window in which conflicts have to be resolved. For the case of AEON, in which route suggestions were requested per vehicle, the path planning usually executed within a second. However, when routes of previously cleared vehicles had to be taken into account, the algorithm needed increasingly more time to execute. The stand-alone simulation was able to route approximately 100 vehicles for the next 30 minutes within 5 minutes of runtime. Since this varied greatly between replanning instances, more research is needed to optimize the computational efficiency further. However, as alternative already mentioned above, implementations in compiler-based programming languages have the potential for a manifold reduction of execution times.

Modularity

The modular design of the multi-agent system allows for later integration of additional aspects of airport surface movement operations as well as new features. The requirements for the MAS are listed in this document, and further documentation on e.g. the standardized airport maps and rules are supplied in D4.1. Modularity of the algorithm allows for enforcing these, for example respecting airport specific traffic rules. The activity-based path planning was found to be well suited for planning push-back, engine start, taxiing, and decoupling operations. We expect that further operations can also be represented due to its modular structure.

Limitations and Outlook

Due to the exploratory nature of the AEON project and its low TRL level, many aspects necessary to prove feasibility of this AEON solution could not be investigated so far. This is especially the case for disruptions in the operations, and safety-related topics except those of vehicle separation that is at the core of the path planning algorithm. Therefore, the suitability of the MAS also for operations in e.g. situations with low-visibility conditions, changing weather and weather extremes have to be studied with further research. Furthermore, as also seen in the real-time simulations of the final evaluation, more work is needed to guarantee the required robustness of the MAS itself and its interaction with other parts when it is meant to interact with humans. However, the modular structure of the MAS is designed to take occurring disruptions into account when assumptions are relaxed (e.g. triggering replanning when deviations to the planned route are too large).

Nonetheless, we hope to spark discussion around the ideas presented in this document and their potential towards sustainable airport surface movement operations.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

7 References

[1] International Civil Aviation Organization, *ICAO Annex 14 to the Convention on International Civil Aviation: Aerodromes. Volume I: Aerodrome Design and Operations*, 7th ed. 2016.

[2] 'Schiphol - Standaard pushback per positie', *Schiphol*. https://www.schiphol.nl/en/operations/page/sleep-en-pushbackbewegingen/ (accessed Aug. 30, 2022).

[3] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, 'Searching with Consistent Prioritization for Multi-Agent Path Finding', *arXiv:1812.06356 [cs]*, 2019, Accessed: Apr. 02, 2021. [Online]. Available: http://arxiv.org/abs/1812.06356

[4] 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.

[5] R. G. Sargent, 'Verification and validation of simulation models', in *Proceedings of the 2010 Winter Simulation Conference*, Baltimore, MD, USA, Jan. 2011, pp. 166–183. doi: 10.1109/WSC.2010.5679166.