Delft University of Technology

To70

A Delay Analysis in Airport Engine-off Towing Operations Using Hierarchical Multi-Agent Path Planning

Master of Science Thesis

Joost Soomers

A Delay Analysis in Airport Engine-off Towing Operations Using Hierarchical Multi-Agent Path Planning

Master of Science Thesis

by

Joost Soomers

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 02-06-2022.

Student number:	4430875	
Project duration:	01-01-2021 - 02-06-2022	
Thesis committee:	Dr.ir. B.F. Bruno Lopes Dos Santos	(TU Delft, Chairman)
	Prof. dr. Daan Pool	(TU Delft, Examiner)
	Prof. dr. O.A. Sharpanskykh	(TU Delft, Supervisor)
	Ir. P.M. Fernandez	(To70, Company Supervisor)

Preface

The document in front of you concludes my life's chapter of seven years at Delft University of Technology. It reports on a final research carried out to become where those seven years were all about: a Master of Science in Aerospace Engineering. The work focuses on understanding the implications of sustainable taxiing to aircraft ground delays. Sustainable taxiing has many potentials, but often require some complex operational changes. Let's hope this research will enlighten the aviation community with new information and possibilities to enable sustainable taxiing in the near future.

I can start with no one else but Alexei Sharpanskykh. My supervisor guided me through the research and never failed to stimulate my enthusiasm. He provide me with fundamental insights and was always there when I needed him - even in the most horrifying times. I sincerely respect and admire his passion for students and research in general. I am grateful for what I have achieved, but I am more grateful for Alexei who helped getting me here.

I thereafter would like to thank two fellow students: Malte von den Burg and Jorick Kamphof. Malte was a very pleasant and informative partner to work with, both for his knowledge in the technical details of coding and in conceptualising new and difficult matter. Jorick's critical thinking, inspiring discussions, and ability to detect and address critical team issues brought our work to the next level and kept our group together. Jorick and Malte, we have been a great team. I developed myself more than you think; all thanks to you. Finally, I would like to thank the aviation consultant agency To70 including Xander Mobertz and Patricia Martin Fernandez. Your expertise and advice in general aviation and airport operations has taken this research to an exceptionally high level. I hope to meet and possible collaborate with both of you again in future.

This report would never have existed if no educational journey preceded it. Every part of the journey could count on endless encouragement and support from my parents. I - quite literally - owe much debt to them for getting where I am right now. When I was a kid playing in the garden, I remember my father saying from behind his desk: "just wait little boy, one day these roles will be reversed". Well dad, today is that day. My career is about to take-off, and the flight is made possible by you - thank you.

Joost Soomers
Delft, May 2022

Contents

List of Figures

List of Tables

List of Abbreviations

AAS	Amsterdam Airport Schiphol
A-CDM	Airport Collaborative Decision Making
ALDT	Actual Landing Time
APU	Auxiliary Power Unit
ATC	Air Traffic Control
CBS	Conflict-Based Search
CPDSP	Collaborative Pre-Departure Sequence Planning
CTOT	Calculated Take-Off Time
DNE	Do Not Enter
DNP	Do Not Persist
DNW	Do Not Wait
IAI	Israel Aerospace Industries
MAPF	multi-agent path finding
PBS	Priority-Based Search
RMO	Runway Mode of Operation
SAS	Smart Airport Systems
SI	Safe Intervals
SID	Standard Instrument Departure
SIPP	Safe Interval Path Planning
TSAT	Target Startup Approval Time
USI	Unsafe Intervals
WTC	Wake Turbulence Category

List of Symbols

Symbol	Description	Units
a	Acceleration	m/s^2
α	P-value cutoff for statistically significant difference	-
b	Vehicle width	m
d	Distance	m
c_a	Activity weight coefficient	-
c_d	Distance weight coefficient	-
c_t	Time weight coefficient	-
d_o	Vehicle shape offset distance	m
d_t	Minimum take-off distance	m
d_s	Vehicle safety distance	m
E	Absolute simulation time of entering an edge	s
$\eta_{3,4}$	Absolute edge traversal simulation timepoints, calculated with type 4 edge length	s
GD_x	Average ground delays in aircraft set x	min
i, j, k	Subscriptions used to indicate aircraft, scenario, and activity number	-
L	Absolute simulation time of leaving an edge	s
l	Vehicle length	m
μ	Sample mean	-
n	Number of statistical tests performed in a scenario	-
p	Statistical value used to validate null hypothesis	-
ρ	Sample standard deviation	-
r_p	Planning radius	m
r_a	Agent shape radius	m
$r_{a_{p,d}}$	Radius of prioritised (p) and deprioritised (d) agent	m
$r_{e_{p,n}}$	Radius of previous (p) and next (n) traversed edge	m
r_{turn}	maximum turn radius at v_{max}	m
t	Time duration	s
t_t	Duration of Taxibotting activity	s
t_u	Duration of uncoupling activity	s
t_m	Duration of multi-engine taxiing activity	s
t_r	Duration of Taxibot return movement activity	s
t_c	Duration of multi-engine taxiing activity in the conventional scenario	s
$\tau_{1,2,3,4}$	Absolute edge traversal simulation timepoints, calculated on edge type 1	s
v	Velocity	m/s
v_f	Final motion velocity	m/s
$v_{f,lim}$	maximum final motion velocity	m/s
v_i	Initial motion velocity	m/s
v_{turn}	maximum turn velocity	m/s
v_{max}	maximum taxi velocity	m/s
w	Path planning search window	s
$wtc_{l,f}$	Separation minima with leading (l) and following (f) aircraft	s

I

Scientific Paper

A Delay Analysis in Airport Engine-off Towing Operations Using Hierarchical Multi-Agent Path Planning

J.A.L. Soomers, *MSc Student, Delft University of Technology*
Ir. X.R.I. Mobertz, *Aviation Consultant, To70 Aviation The Hague*
Ir. P.A. Martin Fernandez, *Aviation Consultant, To70 Aviation The Hague*
Dr. O.A. Sharpanskykh, *Assistant Professor, Delft University of Technology*

Abstract

The increased pressure to reduce aircraft emissions and the growing demand for air travel causes airports to consider alternatives for fuel-inefficient aircraft taxiing. External towing tugs can be deployed to effectively reduce on-ground fuel burn of outbound aircraft. These new operations are likely to aggravate ground delays as a result of flow-disturbing uncoupling processes on taxiways or holding platforms. The size of these delays has not been previously investigated. The goal of this research is to analyse the impact of various uncoupling scenarios on ground delays. Scenarios are created by varying the uncoupling duration and the available uncoupling points. A computational model is created with Amsterdam Airport Schiphol (AAS) and Taxibot as reference airport and towing system. Vehicle-specific operations are optimised by a hierarchical multi-agent path planning algorithm. The algorithm comprises augmented versions of a Priority-Based Search (PBS) algorithm and a Safe Interval Path Planning (SIPP) technique. Simulated conventional operations are compared with Taxibot-enabled operations for traffic samples corresponding to high traffic demands. Minor delays are found for inbound traffic. For outbound traffic, most delays occur during towing and are mainly determined by the arrangement of uncoupling points, the uncoupling duration, and the traffic demand. Delays are furthermore, but to a lesser extent, determined by the changed kinematics of aircraft towing and the runway's position. Delays resulting from uncoupling only on taxiways or holding platforms vary per runway as a result of different arrangements in uncoupling points. For every arrangement of uncoupling points, the number of points and the presence of alternative paths around the points are key in mitigating delays resulting from long uncoupling processes. The intensity of delays can be reduced by tuning search objective settings in the algorithm, in particular for uncoupling points arranged in series.

1 Introduction

The historic annual 5% growth of air transportation has led airports to approach boundaries in terms of capacity and emissions [1]. Aircraft ground movements are one of the most dominant sources of airport emissions [2]. It is estimated that short and medium range aircraft burn around 5-10% of their fuel on the ground because of a fuel-inefficient combustion in the aircraft main engines and long taxi times [3]. This share has risen over the past years as airports are expanding in size to increase infrastructural capacity [4]. Increasing airport capacity while reducing aircraft ground fuel burn therefore remains a complex issue in conventional taxi operations. The urgency of solving this issue has been overshadowed by the COVID-19 pandemic in recent years. Despite the evidence that the pandemic had severe consequences for air transportation, the industry has shown to be resilient all the time. By 2027, the industry is expected to be fully recovered to 2019 levels after which the historic growth is likely to continue [5]. It therefore remains highly relevant to anticipate on this foreseen issue.

Past research considered alternative systems to replace conventional fuel-inefficient ground movements. A general distinction is made between systems installed onboard of aircraft, requiring the Auxiliary Power Unit (APU) as power source, and systems located outside of aircraft [6]. Electric onboard taxi systems generate thrust by powering the aircraft nose or main landing gear. After installation, low operational costs and smooth ground movements are expected as no additional staff is needed and no uncoupling of external systems is required [7, 8]. A major drawback is the additional aircraft weight resulting from structural modifications needed to the aircraft landing gear. This would increase airborne fuel burn making such systems counterproductive for long-haul fights. Additional drawbacks including lower taxi speeds and more extensive APU requirements make onboard systems a questionable solution.

External systems could be an alternative to limit aircraft modifications. Because these systems replace ground propulsion by aircraft engines and do not increase airborne fuel burn, they show better performance in terms of fuel and emission reduction [9]. Malicki et al. [10] proposed an airport rail system in which aircraft are towed via small carts moving through channels nested in taxiways. Major airport modifications and substantial financial investments are, however, required for such system. The use of regular pushback trucks for the entire taxi phase seems an obvious option to prevent high investment costs. Towing full aircraft over long distances, however, is generally not permitted as the forces exerted on the nose wheel would damage the aircraft's structure [8]. A financially viable solution are tugs specifically designed for towing full aircraft over long distances. Israel Aerospace Industries (IAI) own the only certified tugs, called "Taxibot" [11]. Feasibility studies in India and the Netherlands show a potential reduction in fuel usage between 50-85% in comparison with regular operations [12, 13]. Besides environmental benefits and the unneeded aircraft modifications, Taxibots are likely to increase operational efficiency at aprons by removing the need for regular pushback truck uncoupling and engine warm-up. However, airport wide Taxibot operations have not yet been deployed. It is expected that scaling this concept would disturb aircraft flows because of the need to uncouple Taxibots near runways [13]. The size of these disturbances have not been previously investigated, but are expected to primarily depend on the uncoupling location and duration. The disturbances would increase Air Traffic Control (ATC) workload and put pressure on the already limited infrastructural airport capacity, increasing delays [14].

Autonomously coordinated vehicles may offer a solution. The approach replaces conventional centralised air traffic coordination by shifting the role of ATC officers from guiding and coordinating to supervising and intervening. Morris et al. [14] studied existing self-driving technology to enable automated taxi systems at busy airports. Morris concluded that downsides of increased operational complexity of towing tugs can be partially or completely mitigated through automated movements and decision making. Similarly, Chua et al. [15] considered the impact of path suggestions and autonomous taxiing tugs on airport operations and ATC workload. Practical results indicate that the inclusion of such automated technology assistance can improve ground movement efficiency and reduce ATC workload. Existing research focused on these automated ground movements by using agent-based computational models. Such models are suitable to specify complex intelligent and dynamic systems with many interacting entities [16]. Therefore, it is used, for example, in road traffic behaviour [17] and control systems of large energy grids [18]. It was found that distributed, decentralized control systems were effective in achieving safe, resilient and efficient operations in conventional airport ground movements [19, 20]. Subsequently, it was shown that this type of control is similarly able to safely and efficiently guide aircraft and Taxibots over airport ground surfaces with increased traffic complexity [21]. Relatively recent research in the field of multi-agent path finding (MAPF) demonstrates, in contrast to the research above, the potential of using centralized offline search algorithms to optimise conflict-free trajectories [22, 23]. Including a hierarchy in multi-agent systems allows for the implementation of these centralized algorithms. Because such structures plan paths from a centralized perspective, solutions tend to be more optimal in complex and disturbed environments. To the best of our knowledge, hierarchical multi-agent structures have not been used to investigate aircraft flow disturbances in tug-enabled airport ground movements before. Therefore, the objective of this research is:

> *To investigate how different uncoupling locations and durations influence ground delays when fully outbound towing is deployed by using hierarchical multi-agent path planning.*

Taxibot and Amsterdam Airport Schiphol (AAS) were chosen for the case study considered in this research. The study was done in collaboration with To70 Aviation, AAS and Smart Airport System (SAS) experts.

This research first establishes a set of generic modelling components for Taxibot-enabled ground movements. Then, AAS is analysed to develop a conceptual framework suitable for the airport. The MAPF model is subsequently established based on this framework by elaborating on four components. First, the environment, representing the layout of AAS, is created by using actual taxiway centrelines, runways, service roads, holding platforms, and gates. Second, the Routing Agent and the Moving Agents are defined. Third, Priority-Based Search (PBS) and its associated conflict avoidance mechanisms are established. Fourth, the Safe Interval Path Planning (SIPP) algorithm, used to assign uncoupling point and to find routes, is specified. Delays are finally analysed by comparing simulated conventional operations with simulated Taxibot operations using real-world data corresponding to high traffic demands.

This paper is structured as follows. A conceptual model of Taxibot-enabled operations is presented in Section 2. The hierarchical multi-agent system with a particular focus of those aspects needed for the augmented path planning algorithm is established in Section 3. The augmented path planning algorithm is subsequently discussed in Section 4. Section 5 elaborates on the approach for verification and validation. Section 6 introduces the experimental setup, and Section 7 considers the results of the experiments. A method to tune the path planning algorithm for minimum ground delay is given in Section 8. A discussion of the methods and results are provided in Section 9. The conclusion of this study is presented in Section 10.

2 Conceptual Model of Taxibot-enabled Ground Movement Operations

This section proposes a conceptual framework used for modelling Taxibot-enabled aircraft movements at AAS with a particular focus on uncoupling operations. The framework is presented in two stages. First, existing procedures for outbound Taxibot operations are translated into a set of modelling components. The translation and all associated assumptions are elaborated in Section 2.1. Second, the established components are aligned with the situation of AAS allowing for a complete concept of operation for outbound towing in Section 2.2. The operational concept serves as the foundation of the computational multi-agent system used for simulation and analysis, covered from Section 3 onward.

2.1 Modelling of Outbound Taxibot Operations

A chronological visualisation of Taxibot-enabled outbound aircraft ground movements is given in Figure 1. The left and right column contain the existing and conceptual (simulated) outbound Taxibot operations. A timeline with expected durations is given in the middle of the two columns. In the actual operations, Taxibotting happens from gates to designated uncoupling points. The movements are entirely controlled and executed by pilots. Upon arrival at one of the uncoupling points, a sequence of four fixed events occur: control handover to Taxibot operator, automatic unloading, intercom cable disconnection, and Taxibot drive-away. The expected procedural uncoupling durations according to Taxibot product owner Smart Airport Systems (SAS) experts is given left of the timeline [24]. When all events have been completed successfully, a clearance signal is given by Taxibots indicating that aircraft are allowed to continue their movement towards the runway. The designated location from which the signal is sent is called the all-clear point. From this location, Taxibots proceed to a new mission or return to a parking facility.

The existing procedures have been translated into a set of four simulated operations, indicated by yellow in the right column. The operations include: a) Taxibotting, b) uncoupling, c) multi-engine taxiing, and d) Taxibot return movements. All four operations have an associated duration, given in the timeline at the left of the column. Operations are performed according to a set of associated heterogeneous kinematic values, including finite acceleration and deceleration rates and maximum velocities for both straight and curved segments.

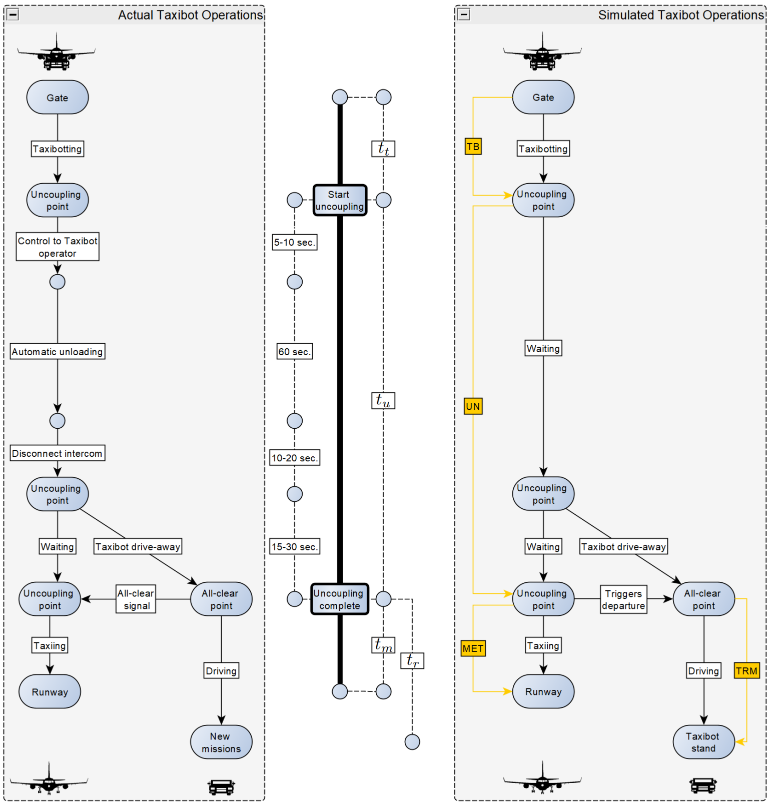


Figure 1: Chronological top-down diagram of existing (left) and simulated (right) outbound Taxibot operations. The four simulated operations (referred to as activities in the agent-based model) are indicated in yellow. Abbreviations: MET, multi-engine taxiing; TB, Taxibotting; TRM, Taxibot return movements; UN, uncoupling; sec, seconds.

Furthermore, all operations are modelled as a single two-dimensional shape. The shape and kinematic properties are dependent on the vehicle(s) executing the operation. Finally, all operations are planned by the path planning algorithm according to three operational-specific search objective coefficients. The coefficients express the relative importance of operations and the cost of traversing a distance and a time.

Taxibotting: during Taxibotting, aircraft and a Taxibot are jointly routed from the gate to an uncoupling point. All outbound traffic makes use of Taxibots. Taxibot coupling is not simulated and the vehicles are assumed to be attached to aircraft at departure. Sufficient Taxibots are available to meet all demand, meaning that departures are not delayed by the unavailability of Taxibots. Taxibot movements from parking facility to gate are excluded from the model. Specific kinematic Taxibot-towing properties apply. The shape of the two connected vehicles is assumed as similar to the single aircraft during the entire operation.

Uncoupling: during uncoupling, aircraft and Taxibot jointly wait at the uncoupling point. The end of uncoupling triggers the start of the multi-engine taxi and Taxibot return movement operation. The Taxibot movement from the uncoupling point to the all-clear point is included in the uncoupling time. It is assumed that all engines are entirely warmed-up at the end of uncoupling. It is furthermore assumed that no unsafe situations arise because of the necessity for Taxibot operators to disembark the Taxibot at all-clear points. The shape of the two uncoupling vehicles is assumed as similar to the single aircraft during the entire operation.

Multi-engine taxiing: during multi-engine taxiing, aircraft taxi on their own power. The operation is performed by inbound aircraft and outbound aircraft after uncoupling. For outbound aircraft, the movement is performed from the uncoupling point to a set of designated runway entrances. Aircraft are only directed to runway entrances that allow for sufficient take-off distance. They are subsequently only allowed to enter if separation with preceding aircraft adheres to wake turbulence restrictions. For inbound aircraft, gate occupancy is not considered, meaning that all aircraft can enter gates directly upon arrival. Aircraft-specific shapes and kinematic properties apply.

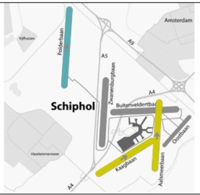
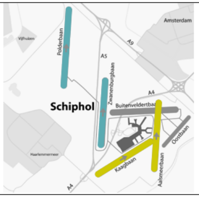
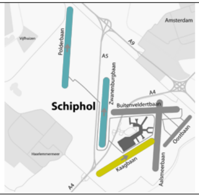
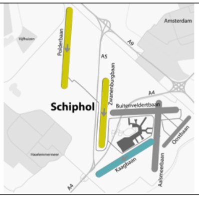
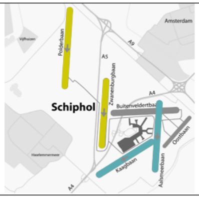
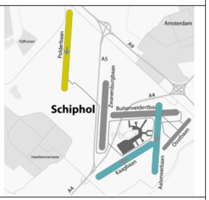
Taxibot return movements: during Taxibot return movements, Taxibots are routed from all-clear points to the parking facility. The parking facility is assumed to have enough capacity to facilitate all Taxibots. Taxibot-specific shapes and kinematic properties apply.

2.2 Outbound Taxibot Operations at AAS

The conceptual Taxibot model has several important differences compared to conventional operations. To get a clear understanding of delays that might result from the outbound Taxibot-enabled operations, the focus is set on the four most frequently used departure runways. These runways are part of Runway Mode of Operation (RMO) North and South, as visualised in Table 1. From the four simulated operations established in Section 2.1, Taxibotting and multi-engine taxiing are performed on regular aprons and taxiways readily suitable for these operations. For uncoupling, locations and associated all-clear points must be appointed for all departure runways. For Taxibot return movements, the movement area between all-clear points and the parking facility must be established.

Fifteen uncoupling and associated all-clear points are selected according to eight criteria. Their locations, criteria, and associated assumptions are visualised in Appendix A [26]. To investigate different uncoupling scenarios, three uncoupling strategies are established. The uncoupling locations, positioned either on taxiways

Table 1: Three phases of the two RMOs considered in this research. Active departure and arrival runways indicated in blue and green. Runway crossing to and from 36L is closed when 18C/36C is active. Map retrieved from Schiphol24 [25].

RMO North			RMO South		
Arrival peak	*Transition*	*Departure peak*	*Arrival peak*	*Transition*	*Departure peak*
					
36L / 06 + 36R	36L + 36C / 06 + 36R	36L + 36C / 06	24 / 18C + 18R	18L + 24 / 18C + 18R	18L + 24 / 18R

or holding platforms, determine the strategies; uncoupling points located on taxiways are part of the taxiway strategy, and vice versa, uncoupling points on holding platforms are part of the holding platform strategy. Finally, all uncoupling points are part of the mixed strategy. A set of runway-specific uncoupling points in the same strategy form an uncoupling configuration. As runway 24 does not have nearby holding platforms, 10 uncoupling configurations exist. A summary is provided in Table 2.

Taxibots must return to the parking facility after uncoupling. One parking facility, located near the H-platform, is selected for this research. AAS has service roads to facilitate movements of small vehicles. A circular service road network is located in the center of the airport and connected to the parking facility. The roads have enough capacity to facilitate all Taxibots for outbound flights. This network is accessible through entries located at taxiway Alpha. To reach these locations, Taxibots are directed over taxiway Alpha, Bravo and some service roads in the outer area of the airport. The location of all service roads and associated assumptions are visualised and explained in Appendix A [26]. Separation amongst Taxibots is not maintained on roads wide enough to allow for bidirectional traffic. This applies to all service roads and taxiways.

Table 2: Uncoupling configurations and strategies. Locations visualised on airport maps in Appendix A [26]. *Taxiway strategy used for runway 24 in simulation.

Uncoupling strategy	**Uncoupling configuration**	**Expected flow disturbance**	**Runway**	**RMO**
Holding platforms*	P1, P3	Limited	18L	South
	P4, P5	Full	36C	North
	P6, P7	Limited	36L	North
Taxiways	E6-I, E6-II	Full	18L	South
	B(A25-A26), Q	Partial	36C	North
	V4-I, V4-II	Full	36L	North
	B(A6-A7), B(A7,A8), B(A9-A8)	Partial	24	South
Mixed*	P1, P3, E6-I, E6-II	Partial	18L	South
	P4, P5, B(A25-A26), Q	Partial	36C	North
	P6, P7, V4-I, V4-II	Partial	36L	North

3 Multi-Agent Model[1]

The conceptual framework outlined above was translated into a hierarchical multi-agent model. This section first elaborates on the environment of the model in Section 3.1. The agents, their local properties, and their interactions are discussed in Section 3.2.

3.1 Environment Specification

The environment is fully deterministic and consists of three objects: a) a graph representing the airport surface of AAS, b) agent templates, and c) the flight schedule. All objects are described below.

Airport graph: the graph facilitates the common space in which the combined plans of aircraft and Taxibots are optimised. It was obtained by translating the ground movement area of AAS into a set of 1124 nodes and 3150 unidirectional edges, as depicted in Figure 2. Nodes are critical locations on the aerodrome that represent gates, holding platforms, parking facilities, stopbars, runway entries or exits, or locations from which different paths can be taken. Adjacent nodes are connected by two unidirectional edges - creating a bidirectional road - that represent taxiway centrelines, runways, or service roads. A zoomed-in visualisation of the uncoupling areas shaded in red is presented in Appendix A [26]. Various graph-related information is available as part of the environment. Each edge contains characteristic information such as the type, width, length, radius, and heading. Constraints can be included to restrict taxiway segments or waiting areas. Surrounding edges and nodes including their relative positions are similarly stored in each edge segment. All information needed for runway planning including entries, stopbars, uncoupling nodes and all-clear points is likewise present. The graph, accessible by the Routing Agent, is dynamic due to the opening and closing of runway entrances and crossings.

Agent templates: the agent templates define the general properties of vehicles in all activities established in Section 2.2. These include kinematics, dimensions, take-off distances, and separation safety factors on both taxiways and runways. Seven templates exist; six for aircraft and one for Taxibots. The static templates are accessible by the Routing Agent.

[1] This section builds upon the collaborative work with equal contribution by three students from Delft University of Technology. Team members: Malte von den Burg (PhD candidate), Jorick Kamphof (MSc) and Joost Soomers (MSc).

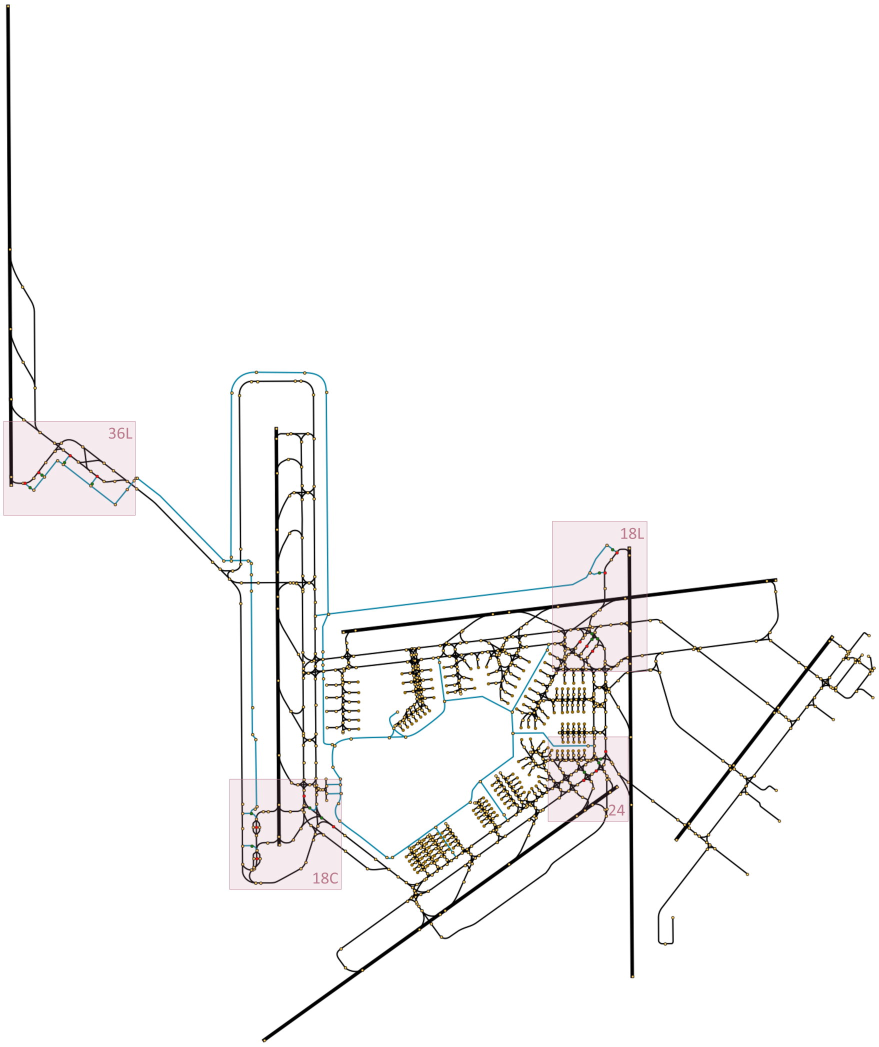


Figure 2: The graph representing the layout of AAS. The graph contains: taxiway centrelines (thin black line), runways (thick black line), and service roads (thin blue line). The Taxibot parking facility is located at the dead-end node in the north of the circular central service road network. Uncoupling sections are shaded in red. Zoomed figures of these sections, include all node and edge types, are provided in Appendix A [26].

Flight schedule: the flight schedule contains information of each flight. The following information is present for every flight: a) an identification number, b) an aircraft type, c) a flight direction, d) a runway, e) a gate, and f) a start time. The flight schedule is based on historical real-world data, more extensively discussed in Section 6. The static schedule is accessible by the Routing Agent.

3.2 Agent Specifications

The model has a hierarchical architecture with a Routing Agent at the top and Moving Agents at the bottom. This section provides a specification of both agents. Their characteristics are given first. Thereafter, their local properties together with their mutual and environmental interactions are discussed.

1) Routing Agent: the Routing Agent first creates all required Moving Agents. It does so using the flight schedule, the agent templates, and its uncoupling scenario characteristics. The Routing Agent subsequently has the task to optimise the required activities for all Moving Agents using the internal augmented path planning algorithm. The characteristic and the two properties are discussed below.

(Characteristic) Uncoupling Scenario: This characteristic includes the active uncoupling points and the uncoupling duration. The information is used by its *Create Moving Agent Property* to set the goal of the Taxibotting activity and the waiting time of the uncoupling activity. It will become apparent in Section 6 that this characteristic is used to created different uncoupling scenarios.

(Property) Create Moving Agents: this property involves interactions between: a) the Routing Agent and the environment, and b) the Routing Agent and the Moving Agents. The property is executed at a fixed search window w. At first, the flight schedule is observed from the environment for all flights in the search window. Based on the aircraft type, a matching aircraft template is obtained from the environment and a Moving Agent is created. For Moving Agents representing departing aircraft, a Taxibot template is obtained and an additional Moving Agent representing a Taxibot is created. All Moving Agents are assigned initial conditions and a set of activities. The initial conditions represent the start state of Moving Agents. The conditions are obtained from the flight schedule. Therefore, they vary amongst agents. Arriving aircraft agents are given: a) a start node based on a fixed exit of the arriving runway, b) the start time as in the schedule, and c) a start velocity equal to maximum taxi velocity. Departing aircraft agents are given: a) a start node based on the gate, b) the start time as in the schedule, and c) a start velocity equal to zero. The initial conditions of Taxibot agents are determined during path planning, discussed in Section 4.4. The activities represent the four operations established in Section 2.1. These similarly vary amongst the agents. Arriving aircraft agents are given one multi-engine taxiing activity. Departing aircraft agents are assigned three consecutive activities: Taxibotting, uncoupling and multi-engine taxiing. Taxibot agents are assigned one Taxibot return movement activity. If at least one Moving Agent is created, then the agent executes the *Augmented Path Planning Property.*

(Property) Augmented Path Planning: this property involves cooperative coordination between the Routing Agent and the Moving Agents. The property is executed after at least one Moving Agent is created in the *Create Moving Agents Property.* At the start, models of all Moving Agents are obtained. The models contain the initial conditions and the set of activities that must be planned for. The internal augmented path planning model is subsequently deployed to optimise all activities collectively. This algorithm determines the main behaviour of Moving Agents and is therefore discussed in detail in Section 4. As soon as the algorithm finds a valid solution, the *Obtain Routes Property* of the Moving Agents is triggered. If no valid solution is found, the agent-based model is unfeasible and terminates.

2) Moving Agents: The Moving Agents represent aircraft or Taxibots and are routed by the Routing Agent. The agents were given some initial conditions and a set of activities by the *Create Moving Agent Property* of the Routing Agent. Each activity is described by specific details summarised for the k^{th} activity below.

$$ACT_k = \begin{bmatrix} Goal_k & Kin_k & Dim_k & Obj_k \end{bmatrix}$$

The properties are more elaborately discussed as activity-specific characteristics of the Moving Agent below. The property to obtain the routes from the Routing Agent is discussed at the end.

(Characteristic) Goals: the goal of the Taxibotting activity is to reach one of the uncoupling points - determined by the uncoupling scenario - with a velocity of zero. The goal of the uncoupling activity is to wait for a time t_u - similarly determined by the uncoupling scenario - at one of the locations reached in the Taxibotting activity. The goal of the multi-engine taxi activity varies for arriving and departing aircraft agents. For departing aircraft agents, the goal of the multi-engine taxiing activity is to reach one of the runway entries that allows for sufficient take-off distance with any velocity. For arriving aircraft agents, the goals is to the gate node (obtained from the schedule) with a velocity of zero. Finally, the goal of the Taxibot return movement activity is to reach the parking facility with a velocity of zero.

(Characteristic) Kinematics: in each activity, a path is planned according to specific kinematic properties. The properties depend on the vehicle type(s) performing the operation. Paths are planned by motions performed between adjacent nodes connected by edges. Motions are subjected to finite constant acceleration and deceleration rates and a maximum velocity. For acceleration rate a, the equations of motion are given by eq. (1-3).

$$\text{distance} \quad d = v_i\, t + \frac{1}{2}\, a\, t^2 = \frac{v_i + v_f}{2}\, t \quad \text{in [m]} \tag{1}$$

$$\text{traversal time} \quad t = \frac{2d}{v_i + v_f} = \frac{v_f - v_i}{a} \quad \text{in [s]} \tag{2}$$

$$\text{final velocity} \quad v_f = v_i + a\, t \quad \text{in [m/s]} \tag{3}$$

In these equations, v_i and v_f indicate the initial and final motion velocities. The velocity profiles are thereby calculated monotonically for each edge on the path, meaning that a switch from acceleration to deceleration can only happen at nodes. The duration of motions are measured in continuous time, allowing for accurate planning and execution. In regular cases, the velocity profile is optimised for minimum traversal time. As a consequence, edge acceleration ($v_{f,lim} > v_i$) occurs at the beginning of the edge until a defined maximum velocity is achieved and then remains constant. Similarly for edge deceleration ($v_{f,lim} < v_i$), speed reduction is performed towards the end of the edge. If, however, additionally a given traversal time has to be exceeded, for instance to avoid conflicts with other agents (see Section 4), the velocity profile is optimised for maximum final velocity. As a result, routes are being traversed as fast as possible after resolved conflicts. For curved taxiway segments, additional speed reductions are imposed according to Equation (4).

$$v_{f,lim} = \begin{cases} v_{turn}, & \text{if } min(r_{e_p}, r_{e_n}) \leq r_{turn} \quad \text{in [m/s]} \\ v_{max}, & \text{if } min(r_{e_p}, r_{e_n}) > r_{turn} \quad \text{in [m/s]} \end{cases} \tag{4}$$

In this equation, r_{e_p} and r_{e_n} express the radius of curvature of the previous and next traversed edges attached to the node.

(Characteristic) Dimensions: all Moving Agents are modelled as two-dimensional shapes. Shapes are assumed circular with a radius determined by the length (l) or width (b) with an additional offset (d_o), as given by Equation (5). On top of the offset, additional separation requirements are imposed, as elaborated in Section 4.

$$r_a = max(l, b) + d_o \quad \text{in [m]} \tag{5}$$

(Characteristic) Objectives: as stated in Section 2.2, the four activities are planned according to three activity-specific search objective coefficients. The activity weight coefficient (c_a) expresses the relative importance of activities. The other two represent the weight of traversing a path in terms of time (c_t) and distance (c_d). The cost of traversing a distance d within a time t during a certain activity is thus given by Equation (6).

$$cost = c_a \cdot (c_t \cdot t + c_d \cdot d) \tag{6}$$

(Property) Obtain Routes: this property involves interactions between the Moving Agents and the Routing Agent. The property is executed once a valid solution is found by the *Augmented Path Planning Property* of the Routing Agent. The Routing Agent communicates the optimised operations as instructions to all Moving Agents. All instructions are subsequently assumed perfectly executed. Execution is therefore not further considered in this research. It must be noted that it is nevertheless implemented for verification purposes. This is more elaborately discussed in Section 5.

4 The Augmented Multi-Agent Path Planning Algorithm[2]

This section elaborates on the augmented path planning algorithm. The algorithm is used in the *Augmented Path Planning Property* of the Routing Agent. First, the main principles of the algorithm are given in Section 4.1. Second, a conflict detection and avoidance mechanism is provided in Section 4.2. Third, individual path planning and all its extensions are discussed in Section 4.3. Finally, the special case of handling Taxibots is discussed in Section 4.4.

4.1 Main Principles

The path planning algorithm is based on a cooperative setting in which models of Moving Agents combine their efforts to accomplish and optimise their private goals and objectives. A PBS, inspired by Hang Ma et al [23], is used to optimise a conflict-free solution. A solution contains a set of individual plans, one for each agent. Likewise, a conflict-free solution contains a set of conflict-free plans. These individual plans, correspondingly consisting of a set of activities, are determined by an extended version of the SIPP algorithm, proposed by M. Phillips and M. Likhachev [27].

The PBS algorithm searches for a conflict-free solution by constructing a priority order amongst all agents in conflict. Deprioritised agents must adapt their plans for prioritised agents. Each solution has a cost value attached, equal to the sum of the costs of all individual plans. When two agents are in conflict, two new solutions are found in which each of the agents is given a higher priority with respect to the other. The algorithm then chooses which agent should be given a higher priority based on the solution with the lowest cost. A depth-first

[2] The underlying concepts, tables and figures of this section are based on collaborative work with equal contribution by three students from Delft University of Technology. Team members: Malte von den Burg (PhD candidate), Jorick Kamphof (MSc) and Joost Soomers (MSc).

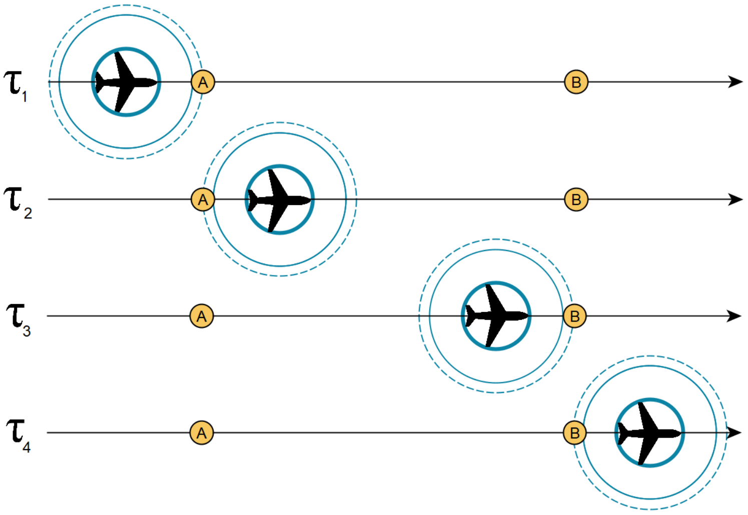


Figure 3: An illustration of the four relevant edge traversal timepoints τ_1, τ_2, τ_3 and τ_4 on edge AB. Yellow circles and black lines represent nodes and edges. The thick blue line indicates the shape of the prioritised agent, r_{a_p}. The thinner blue line indicates the combined shape of the prioritised and deprioritised agents, $r_{a_p} + r_{a_d}$. The dashed blue line indicates the planning radius.

search is adopted, meaning that one of the two previously generated solutions is further explored. This cycle repeats until all conflicts are resolved, or no valid conflict-free solution is found.

Individual path planning avoids prioritised agents by using constraints. Two types of constraints exist. The first type is Unsafe Intervals (USI). These intervals restrict the presence or entrance of deprioritised agents at nodes or edges. The second type is regular constraints. These constraints comprise minimum or maximum edge traversal times to prevent overtaking conflicts.

The SIPP algorithm searches for individual plans through space and time while respecting the imposed constraints. At start, USI are merged and inversed to a finite set of Safe Intervals (SI). These SI thus represent conflict-free time windows in between the presence of two agents at a given location. By using SI to define time in states, the time domain, and thereby the size of the entire search space, is reduced. Just like regular single agent path finding algorithms, SIPP searches through states by exploring motions to states on adjacent nodes at a given cost. The objective is to find a path with the lowest sum of cost while respecting the constraints set by PBS. States with the lowest cost plus an additional admissible heuristic are continuously explored first.

Constraints are created based on the shapes of agents. One of the main design choices was to include the shape of both the prioritised and deprioritised agent(s) in constraints. The advantage of this decision is that SIPP does not have to account for the shape of the agent for which it plans. SIPP is thus allowed to consider agents as points, rather than objects with physical bodies. If the point-based paths subsequently respects imposed constraints, both shapes are guaranteed not to overlap. This point-based SIPP technique drastically reduces search complexity for valid motions. A not outweighing computational drawback of this method is that the imposed constraints are only valid for a specific pair of agent-shapes. All constraints must therefore be recomputed for every agent-shape combination.

4.2 Conflict Avoidance

Conflicts are detected and resolved based on agent kinematics and shapes. A conflict on runways is defined as the violation of Wake Turbulence Category (WTC) constraints. A conflict on taxiways, aprons, and service roads is defined as the overlap of two shapes plus an additional safety margin. To express relative positions amongst agents, a planning radius is used, as given by Equation (7). In this function, r_{a_p} and r_{a_d} indicate the radius of the prioritised and deprioritised agent. d_s indicate the separation safety margin.

$$r_p = r_{a_p} + r_{a_d} + d_s \quad \text{in [m]} \tag{7}$$

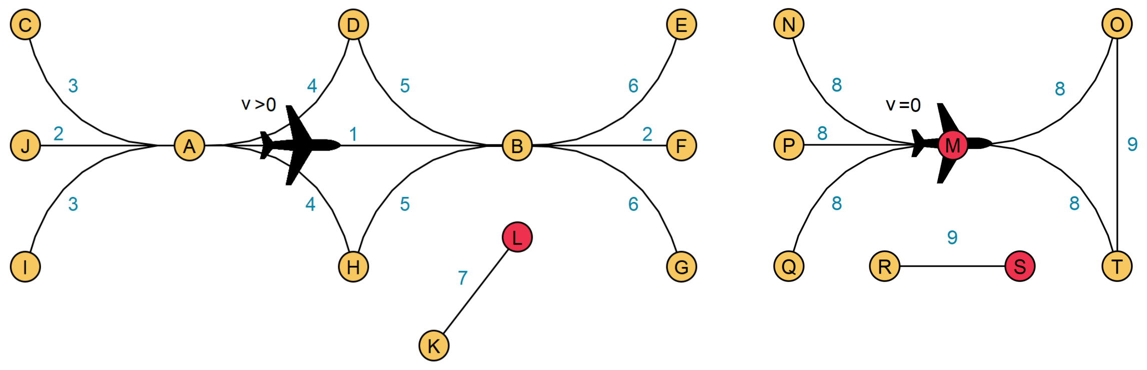
The planning radius is used to express relative positions in terms of four timepoints for each consecutive edge on the path, as visualised in Figure 3 for edge AB. The timepoints are calculated by looking at the exact velocity profiles over the edge and the two adjacent edges. At τ_1, the planning radius reaches the beginning of the edge at the front. At τ_2, the planning radius touches the beginning of the edge at the back. At τ_3, the planning radius hits the end of the edge with the front. Finally, at τ_4 the planning radius leaves the end of the edge. The edge-specific timepoints for the entire path are subsequently used to avoid conflicts with prioritised agents by placing constraints on surrounding edges and waiting nodes accordingly. This works differently on runways.

1) Non-runway Conflicts: Conflicts with a prioritised agent can easily be avoided by constraining all vicinity edges in both directions based on the edge traversal timepoints presented above. However, it is often not needed to constrain edges in both directions. Moreover, constraining a small segment of edges is generally sufficient. Constraining entire edges in both directions therefore usually leads to overconstrained situations and suboptimal results. To limit overconstrained segments, a conflict avoidance system is developed comprising two components. First, surrounding edges are grouped based on their arrangement. Second, all edges within the same group are assigned specific constraints. The system is active on the entire graph with the exception of runways. As waiting always happens at nodes and motions are performed over edges, constraints must be set from a node perspective during waiting and from an edge perspective during motion. Therefore, two classification systems are developed. Classification system A is active during motion ($v > 0$) and considers seven surrounding edge types, as visualised in Figure 4(a). Classification system B is active when waiting ($v = 0$) and considers two surrounding edge types, as visualised in Figure 4(b). The surrounding edges are grouped as such to allow for generic constraint setting while limiting the number of overconstrained areas. Traversal timepoints are always calculated for edge type 1. When waiting, τ_3 and τ_4 do not exist.

Classification system A assigns USI and regular constraints to edges according to Table 3. USI are set either in the same direction of the agent's movement (that is for all edges in Figure 4(a) from left to right) or in the opposite direction. When the edge is not connected to the traversed edge, the direction is irrelevant as the imposed USI are similar. Classification system B assigns USI at surrounding edges according to Table 4. These edges are defined either outwards or inwards with respect to the waiting node. When the edge is not connected to the waiting node, the directions are again irrelevant as the imposed USI are similar. Finally, both systems detect surrounding waiting nodes. These nodes are constrained by USI according to Table 5.

USI are expressed by three variables. The first two indicate a time window in which the constraint is active. The third variable indicates one of the three imposed limitations: Do Not Enter (DNE) intervals restrict entering the edge, Do Not Persist (DNP) intervals restrict presence at the edge, and Do Not Wait (DNW) intervals restrict waiting at the node. Even though these variants have different implications, they are collectively used to set SI on both edges and waiting nodes for the SIPP search, described in Section 4.3.

Regular constraints are imposed in classification system A to prevent overtaking from the back or at the front. These constraints express two conditions. If the first condition holds, the second must hold as well. E and L indicates the time of entering and leaving the edge with the center of the agent. Consider the first regular constraint for edge 1 as an example. A deprioritised agent that entered the edge after τ_2 must also leave after τ_4. As a result, the prioritised agent cannot be overtaken by deprioritised agent from the back. For type 4 edges, η_3 and η_4 indicates τ_3 and τ_4 but calculated with the length of the type 4 edge, rather than the length of the type 1 edge. η_3 and η_4 thus represent the time points at which the prioritised agent travelled the length



(a) Classification system A. Example for traversed edge AB on path JABF. 1) traversed edge. 2) other edges on path within the planning radius from any point on the traversed edge. 3) incoming edge at beginning of the traversed edge. 4) outgoing edge at beginning of the traversed edge. 5) incoming edge at end of the traversed edge. 6) outgoing edge at the end of the traversed edge. 7) all other edges of which any point is within the planning radius from any point on the traversed edge. Node L is a possible waiting node for other agents.

(b) Classification system B. Example for waiting at node M. 8) all edges connected to waiting node M. 9) all other edges of which any point is within the planning radius from waiting node M. Node M is the active waiting node, on which the agent waits. Node S is a possible waiting node for other agents.

Figure 4: Classification systems for the surrounding graph. Yellow circles and black lines represent nodes and edges. Classification system A, active when $v > 0$, is presented in Figure 4(a). Classification system B, active when $v = 0$, is presented in Figure 4(b).

Table 3: Constraint settings per edge type for classification system A. Edge traversal timepoints and edge types can be found in Figure 3 and Figure 4(a).

	Same direction		**Opposite direction**
Edge type	***USI***	***Regular constraints***	***USI***
1	$[\tau_1, \tau_2, DNE]$	$[E >= \tau_2 : L > \tau_4]$, $[E <= \tau_1 : L < \tau_3]$	$[\tau_1, \tau_4, DNP]$
2	-	-	-
3	-	-	-
4	$[\tau_1, \tau_2, DNE]$	$[E >= \tau_2 : L > \eta_4]$, $[E <= \tau_1 : L < \eta_3]$	$[\tau_1, \tau_4, DNP]$
5	$[\tau_1, \tau_4, DNP]$	-	$[\tau_1, \tau_4, DNP]$
6	-	-	-
7	$[\tau_1, \tau_4, DNP]$	-	$[\tau_1, \tau_4, DNP]$

Table 4: Constraint settings per edge type for classification system B. Edge traversal timepoints and edge types can be found in Figure 3 and Figure 4(b).

Edge type	**USI node outwards**	**USI node inwards**
8	$[\tau_1, \tau_2, DNE]$	-
9	$[\tau_1, \tau_2, DNP]$	$[\tau_1, \tau_2, DNP]$

Table 5: Constraint settings per waiting node when agents are moving (top row) or waiting (bottom two rows). Edge traversal timepoints and node types can be found in Figure 3, Figure 4(a), and Figure 4(b).

Node type	**Active for classification system**	**Example node**	***USI***
N_{close}	1 $(v > 0)$	L	$[\tau_1, \tau_4, DNW]$
N_{close}	2 $(v = 0)$	S	$[\tau_1, \tau_2, DNW]$
N_{wait}	2 $(v = 0)$	M	$[\tau_1, \tau_2, DNW]$

of the type 4 edge on the type 1 edge. As a result, deprioritised agents cannot hit prioritised agents from the side when present on a type 4 edge. Elaborate explanations for all constraint are given in Appendix A [26].

1) Runway Conflicts: runway conflicts are defined as a violation of the WTC separation minima on edges behind the stopbars. To avoid runway conflicts, the constraint below is set on all runway and runway-entry edges for deprioritised agents.

$$[\tau_1 - wtc_l, \tau_1 + wtc_f, DNP]$$

In this constraint, τ_1 is defined as the moment the planning radius reached the stopbar node. wtc_l and wtc_f express the time-based separation between the leading and following agent.

4.3 Safe Interval Path Planning

The SIPP search is expanded by two new mechanisms and an enhanced heuristic. The anticipated motion search mechanism is able to prevent situations in which motions are kinematically unfeasible for the imposed constraints. The activity-based planning mechanism allows SIPP to search for a sequence of consecutive activities, rather than a single path. As such, the set of activities given to Moving Agents are optimised in one SIPP search. Finally, the admissible precomputed directional heuristics increases computational efficiency of the search. The new features are elaborated at the end of this section.

To facilitate these features, SIPP states (introduced in Section 4.1) are identified by five variables, four of which define the configuration: 1) a node, 2) a next node, 3) a velocity, 4) an activity, and the fifth a safe interval. The safe interval is defined on the edge between the node and the next node. In the event of waiting, meaning that the state's velocity equals zero, the safe interval is defined on the node. States furthermore have dependent information attached that describe but not identify the state. This information includes: a) a cost value, b) a heuristic value, c) an arrival time, and d) a list of skipped path segments. The cost and heuristic values are used to optimise and guide the search. The arrival time is used to construct the final plan. Finally, the list of skipped path segments contains a list of nodes with associated arrival times and velocities skipped by the anticipated motion search mechanism.

Anticipated motion search. Every motion is by default optimised for time. As a result, states - defined at nodes - are always generated at maximum operational velocity. States at adjacent nodes require, however, often a lower velocity because of imposed restrictions or to meet arrival times in later SI (to avoid conflicts, for example. See Section 4.1). It is then plausible that the edge distance, the initial velocity and the kinematic properties do not allow for the desired motion. This happens, for example, when the edge is too short to decelerate to the required velocity at maximum deceleration. This situation is prevented by an anticipation mechanism. The mechanism prevents unfeasible motions by anticipating on forthcoming states. As such, detected forthcoming unfeasible motions are generated by starting those motions from earlier states, increasing the distance over which the motion is performed. This means that motions can be performed over a set of edges, rather than a single edge. When this happens, the intermediate nodes, arrival times and velocities are stored in the list of skipped path segments appended to the state. A more elaborate example of anticipation is provided in Appendix A [26].

Activity-based planning. As stated before, Moving Agents representing outbound aircraft have three sequential activities that must be planned for: Taxibotting, uncoupling, and multi-engine taxiing. The SIPP search optimises all activities simultaneously. This means that the uncoupling point and runway entrance is selected based on an optimal trajectory from gate to runway, via an uncoupling point. The search is performed by merging the search of two paths: one to any active uncoupling point, and one from the previously selected uncoupling point to any allowable runway entrance. The searches are merged by creating activity-specific states; the activity number is added to the state identifier. States that can finish an activity are copied and recreated as a start state of the new activity without deleting the unexplored states of the previous activity. SIPP is therefore able to continuously explore states in different activities. Note that both the kinematic properties and search objective coefficients vary between activities. As a consequence, the allowable kinematics during motions are constantly changed and the cost function of the optimisation, given by Equation (6), is parameterized differently throughout the search.

Admissible precomputed directional heuristic. The search is guided by an admissible precomputed directional heuristic. For each node in the graph, the shortest distance to all other nodes was determined in the direction of all adjacent nodes. As states are characterised by the next node, the shortest distance to any other node in the corresponding direction can be retrieved from the precomputed data. To ensure an admissible heuristic, the maximum operational velocity is used with the shortest distance in Equation (6) to compute the heuristic to the end of the activity. When the plan contains a sequence of activities, the heuristic from the end of each activity to the end of the entire plan is precomputed at the start of the path planning algorithm for all goal nodes in the activity. This precomputed activity-specific heuristic is subsequently added to the state-specific heuristic to obtain the heuristical value to the end of the last activity.

4.4 Taxibot Planning

Taxibots are routed from all-clear points to the parking facility. The initial conditions of this operation are, however, dependent on the plan of the attached aircraft agent. Taxibot agents are therefore by default planned after the aircraft, and replanned automatically once initial conditions change. Similarly as all other Moving Agents, Taxibot agents have their private position in the priority order constructed by PBS. The conflict avoidance system does not detect conflicts between Taxibots if roads are wide enough to allow for bidirectional traffic. This applies to all service roads and taxiways.

5 Verification and Validation

The verification and validation process for agent-based simulations proposed by F. Klügl [28] was adopted as guiding framework. At first, a run-able preliminary simulation was developed based on the conceptual model validated by experts from AAS, SAS and To70. Subsequently, additional building blocks were implemented one-after-another, after which unit-testing was completed and compile errors were solved. Verification steps taken from the framework were applied at every upgrade.

The offline path planning algorithm presented in this research is part of a more elaborate online multi-agent system.[3] The system comprises additional agents involved in the execution of airport operations, and a graphical user interface to dynamically display individual agent behaviour and agent interaction. This allowed for the visual confirmation that aircraft and Taxibot operations optimised by the offline path planning algorithm were planned as expected. Three other methods were used in addition. First, static heatmaps visualising traffic densities were created to verify aircraft movements in a single map. Second, uncoupling- and take-off schedules were depicted as a block diagram to visualise the distribution of selected uncoupling points and verify conflict-free uncoupling and take-off (WTC constraints). Third, individual trajectories were displayed by both distance and velocity plots to verify that kinematic constraints are respected.

Operational settings and agent characteristics were validated by airport experts and vehicle manufacturers. Various analyses have been performed to test the sensitivity of the model to these input parameters. Results

[3] Multi-agent system developed based on collaborative work with equal contribution by three students from Delft University of Technology. Team members: Malte von den Burg (PhD candidate), Jorick Kamphof (MSc) and Joost Soomers (MSc).

from ranging input uncoupling operations are presented in Section 7. The performance and sensitivity of the augmented path planning algorithm was tested throughout development on three commonly used performance indicators for MAPF algorithms: a) success-rate, b) cost, and c) runtime. Finally, to verify that the behaviour of agents is optimised as desired, search trees were plotted and examined. All tests and associated tools mentioned in this section are elaborated in Appendix C [26].

6 Evaluating Uncoupling Operations

This section introduces the experimental setup to analyse ground delays in various uncoupling scenarios. The hypotheses are first presented in Section 6.1. Section 6.2 introduces the simulated scenarios. The key performance indicators of the scenarios are provided in Section 6.3. An approach to statistically evaluate hypotheses is given in Section 6.4. Finally, a sensitivity analysis to tune the path planning algorithm for minimum ground delay is proposed in Section 6.5. Results are presented from Section 7 onward.

6.1 Hypotheses

The experiments have the purpose to draw conclusions concerning the research hypotheses listed below. In the hypotheses, ground delays are defined as the differences in taxi times between the Taxibot scenarios and a conventional scenario. Hypotheses are accepted if a difference is found that is both statistically significant and relevant in size, as discussed in Section 6.4.

A1 No inbound ground delays are found.

A2 Outbound ground delays are found.

A3 Outbound ground delays are smallest in the mixed uncoupling strategy.

A4 Outbound ground delays are largest in the taxiway uncoupling strategy.

B1 Runway ground delays are found.

B2 Runway ground delays are increased for an increased uncoupling duration of 30 seconds.

The A. hypotheses measure and test ground delays for arriving and departing aircraft. These delays are called airport delays. The B. hypotheses measure and test ground delays for outbound aircraft departing from similar runways. These delays are referred to as runway delays.

6.2 Simulation Plan

This research defines ground delays as the increased taxi times in Taxibot scenarios compared to a simulated conventional scenario. No Taxibotting occurs in the conventional scenario, meaning that all outbound traffic performs the multi-engine taxiing operation from gate to runway. An additional 100 seconds is added to account for the not simulated apron pushbacks and uncoupling operations of regular trucks [29].

The simulation plan contains 10 scenarios, one of which models the convectional operations. The other nine scenarios evaluate the three uncoupling strategies (provided in Table 2) with three different uncoupling durations. The scenarios are created by changing the uncoupling scenario characteristics of the Routing Agent, as explained in Section 3.2. The uncoupling duration is based on SAS experts and AAS Taxibot trials [24, 13]. As visualised in Figure 1, 90 and 120 seconds are the best and worse case uncoupling duration according to SAS experts. Nonetheless, an average uncoupling duration of 150 seconds was found during Taxibot trials at AAS. This research therefore studies a short uncoupling duration of t_u = 90 seconds, a normal duration of t_u = 120 seconds, and a long duration of t_u = 150 seconds.

To evaluate worst-case delays, traffic samples corresponding to high traffic demand are used. Operational data of the 17^{th} and 18^{th} of July 2019, the two busiest days so far, was retrieved. A total of 2819 commercial passenger aircraft visited AAS during the two days. On 17^{th} of July, RMO North was constantly active, whereas on 18^{th} of July, RMO South was active. The data contains arrival and departure peaks handled by different phases in the RMO, as visualised in Table 1. To include pre-departure sequence planning and exclude as much influences of conventional ground operations, the Actual Landing Time (ALDT) and Target Startup Approval Time (TSAT) are extracted and used as start time for inbound and outbound flights. All kinematic values, dimensional properties, safety factors and separation requirements used in simulation are given in Appendix B [26].

The search objectives coefficients for the four activities are given in Table 6. To minimise aircraft delays, for the Taxibotting and multi-engine taxiing activity $c_t = 1$, and $c_d = 0$. $c_a = 1.1$ for the multi-engine taxi activity to stimulate outbound aircraft to uncouple closer to runways. To minimise Taxibot movement emissions, $c_t = 0$, and $c_d = 1$ for the Taxibot return movement activity. $c_a = 0$ to prevent aircraft waiting for Taxibots.

Table 6: Search objective coefficients for the four simulated operations (activities).

Activities	c_t	c_d	c_a
Taxibotting	1	0	1
Multi-engine taxiing	1	0	1.1
Taxibot return movement	0	1	0
Uncoupling	1	n/a	1

Results are obtained by deploying the augmented multi-agent path planning algorithm for every hour in the two days of data. This implies that all aircraft within the same hour are modelled as one group. The window of one hour is chosen to keep runtime within reasonable bounds and simultaneously reducing the consequences of the cut-off as much as possible.

6.3 Key Performance Indicators

The performance indicators of the 10 simulations are the durations of the operations performed by vehicles. As previously visualised by Figure 1, four operations are simulated: Taxibotting, uncoupling, multi-engine taxiing, and Taxibot return movements with duration t_t, t_u, t_m, and t_r, respectively. Because the focus of this research is set to aircraft delays, t_r is not evaluated. t_u is fixed during simulation but varied between scenarios (see Section 6.2). The duration of the Taxibotting and multi-engine taxiing operation remain:

- **Taxibotting duration** (t_t): the duration of the Taxibotting operation, as performed by outbound aircraft in the Taxibot scenarios;
- **Multi-engine taxiing duration** (t_m): the duration of the multi-engine taxi operation, as performed by inbound aircraft in the Taxibot scenarios, and outbound aircraft in the Taxibot scenarios after uncoupling;
- **Conventional multi-engine taxiing duration** (t_c): the duration of the multi-engine taxi operation, as performed by all aircraft in the conventional scenario during the entire taxi movement (similar to t_m but measured in the conventional scenario).

For every Taxibot scenario, individual aircraft ground delays are established by comparing $t_t + t_m$ with t_c. The average ground delay of n aircraft in set x and Taxibot scenario j is therefore calculated according to Equation (8). These aircraft sets can, for example, represent all arriving aircraft or aircraft departing from similar runways.

$$GD_{x,j} = \frac{\sum_{i\in x}\left(t_{t_{i,j}} + t_{m_{i,j}} - t_{c_i}\right)}{n} \tag{8}$$

In this equation, $t_{t_{i,j}} = 0$ if aircraft i is arriving. Note that $t_{u_{i,j}}$ is left out to exclude the direct consequence of longer uncoupling (varied by the scenarios) on delays. These average ground delays are used in this research to evaluate Taxibot scenarios but are not statistically sufficient to conclude upon the research hypotheses. Therefore, ground delays in aircraft set x and Taxibot scenario j are statistically evaluated by comparing the two sets below. Two test are used, discussed in Section 6.4.

$$\left\{t_{t_{i,j}} + t_{m_{i,j}}\right\}_{\forall i\in x} \leftrightarrow \left\{t_{c_i}\right\}_{\forall i\in x}$$

6.4 Statistical Evaluation of Delays

Delays extracted from simulation are measured as paired data (comparison of taxi times, see Section 6.3), in continuous time, and do not follow a normal distribution (Appendix E [26]). Hypotheses presented in Section 6.1 are accepted or rejected based on the outcome of two statistical tests. The non-parametric Wilcoxon signed-rank test is used to test for a statistically significant difference in taxi times. Statistical differences are accepted based on Bonferroni-adjusted alpha levels determined by a family-wise error rate of 5% [30]. This test is presented in Appendix E [26]. In the event that a statistical difference is found, the - similarly non-parametric - Vargha-Delaney A-test is adopted to test if the statistical difference is relevant in size. The A-test compares two samples for stochastic equality by returning a value between 0 and 1, indicating the probability that a randomly selected value from one sample is larger than a randomly selected value from the other [31]. A value of 0.5 thereby indicate equal medians. Values greater than 0.56, 0.64 and 0.71 indicate a small, medium and large difference, all of which are relevant in size. Similar intervals apply below 0.5 if the statistical relation is reversed.

6.5 Sensitivity Analysis Setup for Delay Minimisation

As described in Section 4, the activity-based SIPP mechanism selects one of the available uncoupling points for uncoupling. The selected location has many consequences for traffic flows and thereby delays. The search objective coefficients that guide this selection process can be tuned to minimise aircraft delays. A sensitivity analysis is proposed.

The selection of an uncoupling location is influenced by two factors. First, PBS ensures a cost-efficient solution and thereby distributes aircraft over the available uncoupling points such that delays are minimised. Second, SIPP finds an optimal path including an uncoupling point according to the objective coefficients. The latter, however, means that the selection is not based on the cooperative desire to minimise delays, but rather on the coefficients set in Table 6. More specifically, it is largely determined by the ratio of the Taxibotting and multi-engine taxiing weight coefficients (c_a): increasing the cost of the multi-engine taxi activity with respect to the Taxibotting activity increases the probability that an uncoupling point closer to the runway is selected (Appendix D [26]). Therefore, a sensitivity analysis is performed to determine optimal ratio of Taxibotting and multi-engine taxiing weight coefficients. The taxiway uncoupling strategy is analysed as the phenomenon is most dominant for serial uncoupling points. A peak outbound hour between 11:00 and 12:00 on July 17^{th} is studied at runway 36L. The analysis is presented in Section 8.

7 Results

The multi-agent system and associated augmented path planning algorithm were developed in Python 3.9. Simulations were done using a Windows 10 machine equipped with 16GB RAM and an Intel Core i7-10750H processor. Simulating the 10 scenarios took around 48 hours to complete.

Ground delays are evaluated for inbound and outbound traffic in Section 7.1. These general airport delays relate to the A. hypotheses. Ground delays are subsequently evaluated for outbound aircraft departing from similar runways in Section 7.2. These runway delays relate to the B. hypotheses. Several scenarios of the most remotely located runway 36L, at which the environmental benefits are expected largest, are thereafter analysed to clarify the findings concerning the hypotheses more thoroughly. Section 7.3 evaluates ground delays in the mixed uncoupling strategy throughout a day of operation. Section 7.4 analyses the durations of specific operations amongst various Taxibot scenarios.

7.1 Airport Delays

Table 7 presents the mean and standard deviation of taxi times in the conventional scenario and the Taxibot scenarios. A normal uncoupling duration is used in the Taxibot scenarios. Statistical tests are provided in Table 8 and Appendix E [26]. Results and statistical tests for short and long uncoupling durations are provided in Appendix D and E [26].

In the conventional scenario, the average outbound taxi time is two and a half minutes longer than the average inbound taxi time. The reason is threefold. First, outbound traffic in conventional operations require regular apron tug pushback and uncoupling, taking on average 100 seconds [29]. Second, outbound traffic must respect WTC constraints at take-off, possibly causing runway queues and thereby increased outbound taxi

Table 7: Taxi time simulation results of the conventional scenario and Taxibot scenarios with normal uncoupling duration (t_u = 120 seconds). Times given in minutes for arriving (Arr) and departing (Dep) aircraft.

			Taxibot scenarios		
Samples		**Conventional scenario**	Holding platforms	Taxiways	Mixed
Arr	μ	5.98	6.02	6.00	5.99
	σ	3.38	3.42	3.39	3.39
Dep	μ	8.47	12.21	12.77	10.65
	σ	3.71	6.43	7.06	4.51

Table 8: Vargha-Delaney A-test values for ground delays in the Taxibot scenarios with normal uncoupling duration (t_u = 120 seconds). Values obtained by comparing taxi times of Taxibot scenarios with the conventional scenario. Results given for arriving (Arr) and departing (Dep) aircraft. Differences indicating increased delays coloured by red (medium difference). Abbreviations: n/s, not statically significant. *Determined by Wilcoxon signed-rank test in Appendix E [26].

	Taxibot scenarios		
Samples	Holding platforms	Taxiways	Mixed
Arr	0.50	n/s*	0.50
Dep	0.69	0.70	0.65

Table 9: Vargha-Delaney A-test values obtained by comparing the outbound taxi time of three different uncoupling strategies for a short, normal and long uncoupling duration. Samples given as: treatment-control (values higher than 0.5 indicate increased delays in the first sample). Differences indicating increased delays coloured by light red (small). Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

	Taxibot scenarios		
Samples	$t_u = 90$ seconds (short)	$t_u = 120$ seconds (normal)	$t_u = 150$ seconds (long)
Hol-mix	0.55	0.56	0.56
Tax-mix	0.55	0.57	0.59
Tax-hol	0.51	0.52	0.54

times. Third, the initial velocity of arriving aircraft is v_{max}, whereas the initial velocity of departing aircraft is zero.

Minor delays are found for arriving aircraft in the Taxibot scenarios. The mean inbound taxi time is increased by not more than five seconds. The A-test confirms a negligible taxi time increase in all scenarios, including those with changed uncoupling duration. Hypothesis A1 is therefore accepted in all scenarios.

Larger delays are found for departing aircraft. The mean outbound taxi time in the Taxibot scenarios is increased by at least two minutes with respect to the conventional scenario. The A-test indicates medium taxi time differences in all three uncoupling strategies. When varying the uncoupling duration, several different results are found. For short uncoupling durations, a small taxi time difference is found in the mixed strategy. For long uncoupling durations, large taxi time differences are found in the taxiway and holding platform strategy. Hypothesis A2 is thus accepted in all nine Taxibot scenarios, as at least a small difference is detected in all samples. The cause of these delays are studied and discussed in the following Sections.

The largest delays are found in the holding platform and taxiway uncoupling strategies. The mean outbound taxi time in both scenarios is increased by at least one and a half minutes with respect to the mixed strategy. Table 9 shows the A-test values for outbound taxi time comparisons of the three uncoupling strategies with a short, normal, and long uncoupling duration. The test indicates a small difference between the mixed strategy and the taxiway and holding platform strategy for normal and long uncoupling, but not for short uncoupling. Hypothesis A3 is therefore accepted for normal and long uncoupling durations and rejected for short uncoupling durations. The increased number of uncoupling locations in the mixed strategy allows for a more efficient aircraft distribution during uncoupling. This efficient distribution is more useful when uncoupling takes longer - hence accepting hypothesis A3 only in the scenario with normal and long uncoupling durations.

Minor differences in taxi times are found between the taxiway and holding platform uncoupling scenarios. The A-test indicates a negligible taxi times increase in the taxiway strategy compared to the holding platform strategy for all uncoupling durations. Hypothesis A4 is therefore rejected for all scenarios. When considering the standard deviation of departing taxi times in Table 7, the reason for this rejection becomes more clear. A larger spread of data is found in the holding platform and taxiway uncoupling strategy than in the mixed uncoupling strategy and the conventional scenario. This indicates a performance variation between aircraft departing from different runways in the holding platform and taxiway scenarios. Runway-specific delays are therefore analysed in Section 7.2.

7.2 Runway Delays

Figure 5 presents the average runway delays in all uncoupling scenarios. This means that delays are evaluated per uncoupling configuration, as given in Table 2, for all uncoupling durations. Statistical test values are provided in Table 10 and Appendix E [26].

Table 10: Vargha-Delaney A-test values for runway delays in the Taxibot scenarios. Values obtained by comparing taxi times of aircraft departing from similar runways in Taxibot scenarios with the conventional scenario. Differences indicating increased delays coloured by light red (small), red (medium), and dark red (large). Differences indicating decreased delays coloured by light green (small). Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed; n/a, not applicable; n/s, not statically significant. *Determined by Wilcoxon signed-rank test in Appendix E [26].

	Taxibot scenarios								
	$t_u = 90$ seconds (short)			$t_u = 120$ seconds (normal)			$t_u = 150$ seconds (long)		
Samples	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
18L	0.44	0.67	0.40	n/s*	0.73	0.42	n/s*	0.78	0.43
36C	0.76	n/s*	0.43	0.77	n/s*	n/s*	0.81	0.54	n/s*
36L	0.64	0.70	0.59	0.67	0.74	0.61	0.70	0.78	0.61
24	n/a	0.43	n/a	n/a	0.44	n/a	n/a	0.45	n/a

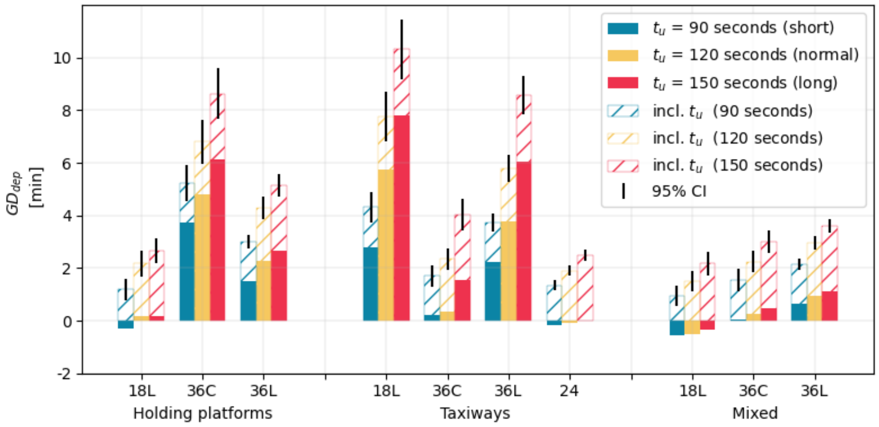


Figure 5: Average outbound ground delays per uncoupling configurations (runway- and scenario-specific) for all uncoupling durations. Ground delays defined positive for increased taxi times in the Taxibot scenarios compared to the conventional scenario. Uncoupling time included by striped bars. A 95% confidence interval is indicated by vertical black lines.

The results show negative ground delays in some scenarios caused by the unneeded uncoupling and pushback operations of regular pushback trucks. The effects are most dominant for runways close to the central gate area (18L and 24). For more remote runways (36C and 36L), the reduced maximum taxi velocity of Taxibotting causes larger delays. The severity of delays remain, however, mainly dependent on the associated uncoupling configuration. Hypothesis B1 is accepted for all runway- and scenario-specific samples with an A-value larger than 0.56 in Table 10. The hypothesis is rejected for all others. The uncoupling configurations with expected full flow disturbances given in Table 2 are indeed found to cause more delays. Similarly, the uncoupling configurations with expected partial and limited flow disturbances result in minor delays. This support the idea that delays are primarily caused by flow-disturbing uncoupling operations that entirely block paths for other traffic. The severity of delays are therefore mainly determined by the arrangement of uncoupling locations rather than the uncoupling strategy. It is for that reason that delays within similar uncoupling strategies cannot be generalised, as concluded upon by hypothesis A4. The arrangement of uncoupling points, including the presence of alternative paths around uncoupling points is therefore key in preventing delays.

To investigate the consequences of increased Taxibot uncoupling durations, the taxi times found in the scenarios with short and long uncoupling duration were compared to the taxi times in the scenarios with normal uncoupling duration. Results are given in Table 11 and Appendix E [26]. Hypothesis B2 is accepted for all runway- and scenario-specific samples indicated in red and green in Table 11. The hypothesis is rejected for samples indicated in white. It is found that longer uncoupling does increase the resulting delays, but depending on the configuration to a lesser extent. The reason is that delays resulting from longer uncoupling are not a direct consequence of the additional uncoupling time (the uncoupling time is not included in the results, see Section 6.3), but rather of congestion emerging as a result of the longer uncoupling process. The magnitude

Table 11: Vargha-Delaney A-test values obtained by comparing runway-specific outbound taxi times of the Taxibot scenarios with short (t_u = 90 seconds) and long (t_u = 150 seconds) uncoupling duration to the Taxibot scenario with normal uncoupling duration (t_u = 120 seconds). Differences indicating increased delays coloured by light red (small). Differences indicating decreased delays coloured by light green (small). Abbreviations: n/s, not statically significant. *Determined by Wilcoxon signed-rank test in Appendix E [26].

		Taxibot scenarios	
Samples		t_u = 90 seconds (short)	t_u = 150 seconds (long)
Holding platforms	*18L*	n/s*	n/s*
	36C	0.47	n/s*
	36L	0.47	0.54
Taxiways	*18L*	0.40	0.56
	36C	n/s*	0.55
	36L	0.44	0.57
	24	n/s*	n/s*
Mixed	*18L*	n/s*	n/s*
	36C	n/s*	n/s*
	36L	n/s*	n/s*

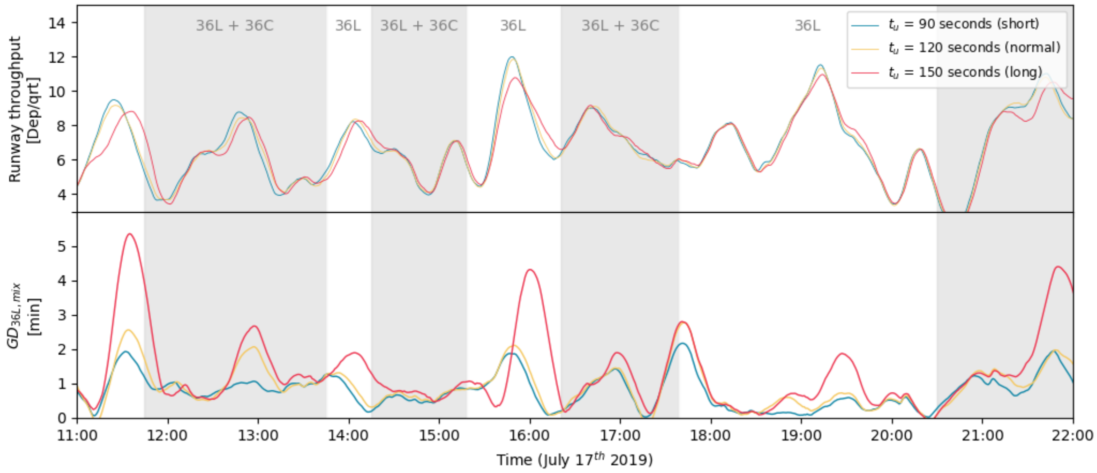


Figure 6: Ground delays and runway throughput for aircraft departing at 36L throughout the 17th of July for a short, normal and long uncoupling duration. Mixed uncoupling strategy is used. Top figure presents the runway throughput per fifteen minutes. The bottom plot presents the average ground delays in a time window of fifteen minutes. Grey areas indicate airport departure peaks, including the active RMO. A fourth degree polynomial Savitsky-Golay filter was used with a window length of one hour to smooth the data. [32].

of the additional congestion was found strongest for configurations in which uncoupling locations or runway entrances are arranged in series; the configurations with expected full flow disturbances in Table 2. Additional congestion for increased uncoupling time was found weaker than expected in the holding platform strategy at 36C. The reason is that - even though P5 is only accessible through P4 - both uncoupling locations have a direct path toward the runway entrance and therefore less mutual disturbance. Given the analysis presented in Table 11, it is concluded that the impact of longer uncoupling can be largely mitigated by a well established uncoupling arrangement.

When finally comparing Figure 5 and Table 11, it can be observed that A-test differences between normal and long uncoupling durations are small, even though average delays can be increased up to two minutes (taxiway uncoupling at 36L, for example). This suggests several large taxi time outliers. Section 7.3 therefore analyses the runway delays throughout a day of operation.

7.3 Runway Delays Through Time

To determine under what circumstances large delays occur, runway delays are analysed through time. Figure 6 presents runway throughput and delays of runway 36L for a short, normal and long uncoupling duration throughout the 17th of July. A mixed uncoupling strategy is active. A consistent positive delay can be observed due to the changed maximum velocity of Taxibotting and the remote location of runway 36L. The runway throughput and delays in the three scenarios behave in similar patterns. Peaks simultaneously occur at around 11:40, 13:00, 14:05, 16:00, 19:30, and 21:45 indicating a correlation between runway throughput and delays. The results of the three scenarios mainly deviate during peaks. This deviation is a result of delay formation. Most severe delays can be found at 11:40, 16:00, and 21:45 simultaneously with peaks in runway throughput. Delays are largest in the scenario with a long uncoupling duration. As delayed aircraft depart later, lags in runway throughput can be observed with respect to scenarios with lower uncoupling duration. Similar lags can be seen in delays as taxi times are recorded at take-off. Moreover, the late departures cause a drop in delays before the peaks, as observable at 11:15 and 15:45. This is because fewer aircraft take-off compared to the conventional scenario before delays start to emerge, causing a small temporary drop in delays. It can therefore be concluded that a steady increase in taxi time can be expected for runway 36L, with some extremes during outbound peak of which the magnitude depend on the uncoupling time. Similar patterns can be found for other runways, provided in Appendix D [26].

7.4 Runway Delays per Operation

Figure 7 depicts delay changes in Taxibotting and multi-engine taxiing when the normal uncoupling duration is decreased to a short duration or increased to a long duration. The same scenarios as Section 7.3 are considered. At first, one would expect aggravated delays in the scenario with increased uncoupling time, and vice versa

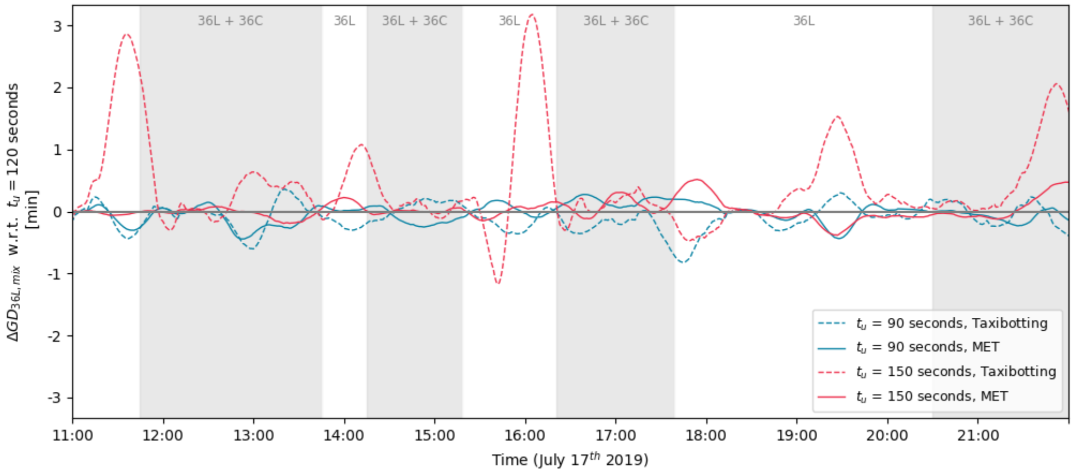


Figure 7: Effects on ground delays in the Taxibotting and multi-engine taxiing operation when the uncoupling time is changed from normal duration (t_u = 120 seconds) to short (t_u = 90 seconds)(blue) and long (t_u = 150 seconds)(red). Average taxi durations determined by a window of fifteen minutes. Results presented for aircraft departing at 36L throughout the midday of July 17^{th} (RMO North) in the mixed scenario (similar as Figure 6). Grey areas indicate airport departure peaks, including active RMO. A fourth degree polynomial Savitsky-Golay filter was used with a window length of one hour to smooth the data [32]. Abbreviations: MET, multi-engine taxiing.

weakened delays in the scenario with reduced uncoupling time. Increased delays can indeed be observed in the scenario with a long uncoupling duration at the three most dominant peaks around 11:40, 16:00, and 21:45 in Figure 6. However, no reduced delays are observed for the scenario with a short uncoupling time. This indicates that the delay consequences of increasing the uncoupling time from 120 to 150 seconds are more severe than from 90 to 120 seconds. It can therefore be concluded that, at a certain uncoupling time and aircraft throughput, queues start to form and large delays emerge.

Figure 7 furthermore reveals that the three most dominant peaks at 11:40, 16:00, and 21:45 in Figure 6 are mainly a result of increased Taxibotting time. Likewise during the smaller peaks present at 13:00, 14:05, and 19:30 the Taxibotting time is mainly increased, and no increase in multi-engine taxi time is observed. Similar

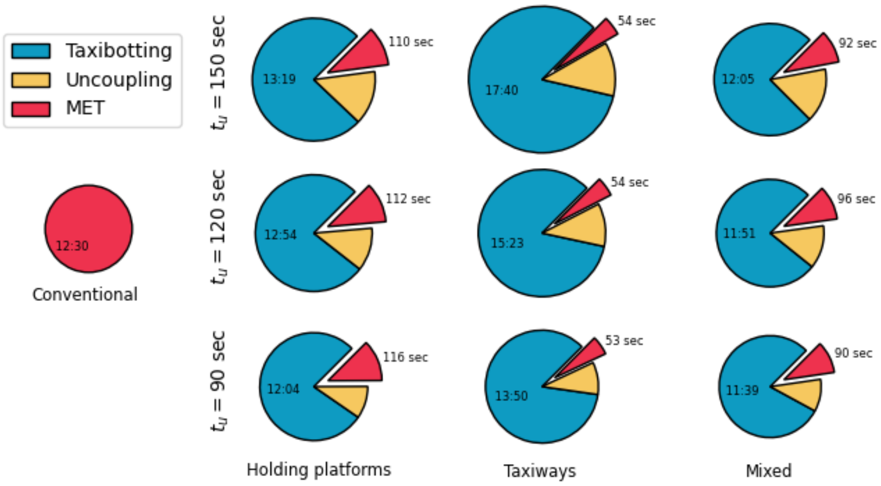


Figure 8: Mean duration of all operations for aircraft departing at 36L in all scenarios. Scenarios in the same row have identical uncoupling durations, as indicated at the left. Similarly, scenarios in the same column have identical uncoupling strategies, as indicated at the bottom. The radius of each chart is proportional to the taxi time increase with respect to the conventional scenario. Abbreviations: MET, multi-engine taxiing; sec, seconds.

observations can be made when looking at other uncoupling strategies, as presented in Figure 8. The figure depicts the mean duration of the operations for aircraft departing at runway 36L in all uncoupling scenarios. The general trend of increased taxi time visualised in Figure 6 can be observed by the size of the charts. For all uncoupling strategies in 36L, increasing the uncoupling duration does not lead to an increase in multi-engine taxi time. Similar findings are made when looking at other runways, provided in Appendix D [26].

When considering the durations of the operations presented in Figure 8 for 36L and in Appendix D [26] for the other runways, the individual advantages of uncoupling strategies can be observed. For 36L, the multi-engine taxi time can be reduced with on average 58 seconds when choosing taxiway uncoupling over holding platforms for normal uncoupling durations. This is, however, at the expense of an increased ground delay of on average one minute and 51 seconds. More extreme results can be found for runway 18L, at which the multi-engine taxi time can be reduced with 92 seconds at the cost of five minutes and 35 seconds delay. Contrary results were found for 36C, with holding platforms located adjacent to the runway. Though these platforms allow for uncoupling close to the entrance, the platforms are arranged such that one is only accessible via the other. The results of this strategy therefore behave similar as the flow-disturbing taxiway strategies at 36L and 18L, and vice versa; the multi-engine taxi time can be reduced with 28 seconds at the cost of on average four minutes and 28 seconds delay, when shifting from taxiways to holding platforms. The cost of reducing the multi-engine taxi time in terms of delay is thereby largest for 36C as a result of the holding platforms arranged in series. Section 8 presents a method to reduce delays resulting from these uncoupling arrangements.

8 Sensitivity Analysis for Delay Minimisation

The consequence of varying the multi-engine taxiing weight coefficient with respect to the Taxibotting weight coefficient for aircraft uncoupling in the Taxiway strategy at 36L is given in Figure 9. One hour of operations was repeatedly simulated in which the coefficient was varied by steps of 0.1. A total of 25 aircraft took off from runway 36L. The upper plot indicates how many aircraft are allocated to which uncoupling location; the sum of both lines always equals 25. The lower plot indicates the average taxi time of the 25 aircraft.

When planning for individual aircraft, increasing the cost of the multi-engine taxiing activity stimulates uncoupling close to the runway to minimise the consequences of the increased activity cost to the total cost of SIPP. Uncoupling location V4-I (closer to the runway) is therefore selected for high values of c_a, and V4-II (further from the runway) for low values of c_a. For $c_a = 1$, V4-II is selected because the higher allowable maximum velocity in the multi-engine taxi activity enforce early uncoupling to reduce total taxi time.

When planning for multiple aircraft, one would expect the path planning algorithm to distribute aircraft over the two available uncoupling locations regardless of the search objective coefficients. This was found not always true and the cause of increased delays for high and low values of c_a (Appendix D [26]). The reason is related to the behaviour of PBS when queues emerge. Due to the relatively close positions of queuing aircraft, a forthcoming conflict is detected and a priority order is created amongst all aircraft in the queue. To avoid

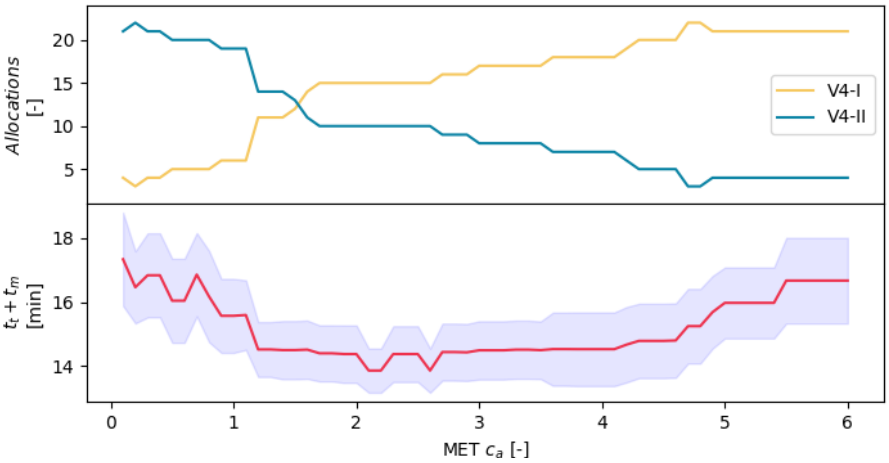


Figure 9: Consequences of varying the multi-engine taxi weight coefficient for aircraft departing at 36L between 11:00 and 12:00 at July 17^{th}. The Taxibotting weight coefficient remains one. Normal uncoupling time (t_u = 120 seconds) is used. Upper plot: number of aircraft allocations per uncoupling point. Lower plot: average aircraft taxi time including a 95% confidence interval in blue. Abbreviations: MET, multi-engine taxiing.

overtaking conflicts, rear aircraft are never prioritised over front aircraft. As a consequence, the priority order is synchronised with the order of the queue, regardless of the costs. Negotiation is thereby controlled by the position in the queue, meaning that aircraft in the front can fulfil their objectives without considering the cost of others. For low and high values of c_a, this results in V4-II (farthest from the runway) and V4-I (closest to the runway) being repeatedly selected. An optimum exists (in this example at around $c_a = 2.1$) in which uncoupling locations are selected such that total delays are minimal (Appendix D [26]).

The extent to which delays can be minimised by tuning c_a strongly depends on the uncoupling duration. It was found that delays are reducible with approximately three to four minutes for normal uncoupling durations and five to six minutes for long uncoupling durations. Negligible benefits can be derived in the scenario with a short uncoupling time because the uncoupling duration is short enough to prevent delays regardless of the uncoupling location. Statistical analyses of these delays are presented in Appendix E [26].

All simulations of which the results were presented in Section 7 used a multi-engine taxiing weight coefficient of 1.1 (see Table 6). It was found that delays between 11:00 and 12:00 can be reduced with approximately two and three minutes for the normal and long uncoupling scenario by tuning the coefficient from 1.1 to 2.1 (Appendix E [26]). It is therefore concluded that the delays during an entire day of operation are similarly reducible by tuning the weight coefficient accordingly. Similar behaviour is expected at other uncoupling configuration containing uncoupling points arranged in series.

The optimal ratio of the Taxibotting and multi-engine taxiing weight coefficients cannot be generalised for all uncoupling configurations. The reason is that the delay consequences of a changed ratio depend on the arrangement of the uncoupling points. In case of two serial uncoupling points located far from the runway, the weight of the multi-engine taxi objective coefficient contributes more to the final cost, meaning that a lower coefficient would be sufficient to obtain an uncoupling distribution that minimises delays. There is therefore a need to determine the optimal ratio between outbound activity weight coefficients for all uncoupling configurations, especially for those containing uncoupling pairs arranged in series. It is argued in Section 9 that such analysis must be performed in future studies using activity- and priority-based path planning algorithms to allocate waiting activities to points arranged in series.

9 Discussion

The obtained results are discussed in Section 9.1. A reflection on the proposed methodology is provided in Section 9.2. Recommendations for future work is given in Section 9.3.

9.1 Discussion on the Results

The obtained results indicate that the multi-agent path planning algorithm was able to successfully plan trajectories and uncoupling operations for all vehicles. It was found that the algorithm performs well in mitigating ground delays resulting from uncoupling operations in various scenarios. Additional remarks must, however, be made when interpreting the results to properly understand the implications.

This study compared simulated conventional operations with simulated Taxibots operations. To understand the practical use of the results, three key differences between actual operations and simulated conventional operations are important. First, it was assumed that all operations found by the path planning algorithm are executed perfectly. In practice, this is difficult to guarantee as complex systems influence real-life execution. Second, autonomous coordination and decision making was assumed. As a result, the role of conventional ATC is largely neglected and standard practices such as taxiway directions and preferred runway entrances are not included. Third, even though the model contains a set of realistic features, various assumptions, as reflected upon in Section 9.2, still had to be made to translate actual operations into a computational model. The comparison between the simulated scenarios thus demonstrates the expected delays in the event that outbound towing is deployed and airports shift towards autonomous coordination and execution.

This study has three general implications with respect to the main research question. The implications are usable for all airports considering towing systems. First, the foreseen flow disturbances and subsequent delays resulting from taxiway uncoupling in airport systems controlled by conventional ATC have the potential to be largely mitigated by autonomous execution and coordination controlled by the hierarchical multi-agent path planning algorithm. This is especially true for taxiway uncoupling points that can be avoided by alternative paths.

Second, a maximum aircraft throughput exists for every uncoupling configuration. The maximum throughput, or capacity, is determined by the uncoupling duration, the number of uncoupling points, and the arrangement of uncoupling points. When exceeding the capacity, queues emerge and delays stack up. Configurations with high capacity contain many uncoupling points that can be avoided by alternative paths. The capacity should be determined for every uncoupling configuration taken into consideration to avoid large delays.

Third, the hierarchical multi-agent path planning algorithm including the PBS + SIPP algorithm is found suitable for Taxibot-enabled airport movements concerning complex, dense, and detailed operations. For SIPP, activity-based planning has demonstrated to work well in assigning uncoupling locations during the search. No discrepancies were found in the developed anticipation mechanisms. Merging of the time domain in a set of SI was found effective for computational efficiency. PBS has proven to be fast and successful for the current problem.

9.2 Reflections on Proposed Methodology

The proposed augmented path planning model and its associated methodology has proved its capability to optimise the required operations for all vehicles. The methodology gains its potential from several relevant design choices. The accurate representation of AAS allowed for a detailed modelling of local uncoupling operations. The inclusion of various aircraft types increased the realism of the uncoupling operations further as it enabled the allocation of aircraft-specific runway entrances and accurate WTC separation minima. Operation-specific search objective coefficients subsequently allowed for various optimisation settings and behavioural control of the path planning mechanism. The modular activity-based setup finally allows for easy adaptation and expansion of the model for alternative research.

The methodology still has several limitations. First, the integration of the uncoupling operations in the broader picture of airport processes both before uncoupling (near gates and aprons), and after uncoupling (at runways and during take-off) is limited. Several important aspects like WTC separation minima and aircraft-specific gates are included, but more operations and standard practices exist and should be included to improve the model's realism. Aspects like aircraft holding, gate occupancy, gate pushback operations, Standard Instrument Departure (SID) routes, runway capacities, and Calculated Take-Off Time (CTOT) slots will all have individual implications on the uncoupling operations.

Second, the augmented path planning algorithm is not entirely conflict-free. The high complexity of the graph required a difficult conflict avoidance system that causes both over- and underconstrained situations for several locations on the graph. An underconstrained situations relevant for uncoupling was found in classification system B: the waiting node and adjacent outwards edges are constrained during uncoupling, but edges leading to the uncoupling node are not, causing queuing aircraft to closely approach and finally hit uncoupling aircraft. Adding a DNP USI to the approaching edges would solve this issue, but simultaneously restricts queuing on the edge - an undesired side-effect. These situations are, however, quite rare and do not include overtaking aircraft. Moreover, no runway or uncoupling conflicts have been found by the uncoupling and take-off schedules (Appendix C [26]). The consequences for delays are therefore expected to be limited.

Third, results were obtained by a sequence of offline simulations with a search window of one hour. As a consequence, the presence of aircraft from the previous hour is not considered at the start of a new hour. This means that delays were not able to stack up between hours, meaning that the actual delays might be higher than presented in the results. The consequences of the search window cutoff are likely noticeable for scenarios with large delays. For scenarios with medium or small delays, only minor differences are expected as no large queues were formed.

9.3 Recommendations for Future Work

This study provided a broad analysis on various uncoupling operations and resulting delays. For future research, it is firstly recommended to improve several features of the methodology and the developed simulation. It is secondly advised to improve the alignment of the uncoupling operations with other airport processes. Thirdly, it is recommended to conduct in-depth analyses for specific uncoupling operations. Other recommendations are made in Appendix F [26].

The results were obtained by a sequence of search windows of one hour. As a consequences, queues were not able to stack up between hours. It is therefore recommended to rerun the path planning algorithm as part of an online simulator to reduce the consequences of the search window cutoff. An example of such simulator was proposed in Section 5. The augmented path planning algorithm is, furthermore, not entirely conflict-free. The high complexity of the graph required a difficult conflict avoidance system that causes both over- and underconstrained situations for some specific locations on the graph. It is advised to solve these unique conflicts during execution. Self-separation, for example, can be considered in the multi-agent simulator as a last safety feature to prevent conflict.

The alignment of uncoupling operations with conventional airport operations could be improved by including several important connect processes, as previously provided in Section 9.2. It is advised to include these operations in the online multi-agent simulator to model more realistic scenarios and obtain more accurate results.

The analyses for specific uncoupling operations could be twofold. Firstly, the path planning algorithm needs to be tuned for the arrangement of uncoupling points. This includes an analysis on the search objective coefficients used for outbound operations. The coefficients must be set in such a way that the allocation of uncoupling locations is not disturbed by priorities being synchronised with aircraft queues, as explained in Section 8. An alternative method is to replace PBS by a Conflict-Based Search (CBS) [22]. CBS is expected to mitigate delays more efficiently for uncoupling points arranged in series as it avoids priorities. However, combining CBS with the detailed graph and dense operations presented in this research is expected to heavily increase the number of computations. Secondly, the maximum aircraft throughput of the uncoupling configuration must be determined. One could subsequently consider adjusting runway schedules or temporarily opening additional uncoupling locations to prevent that capacity limits are exceeded during outbound peaks.

10 Conclusions

This research investigated how different Taxibot uncoupling scenarios for outbound aircraft influenced ground delays. An augmented path planning algorithm was developed as part of a hierarchical multi-agent system of autonomous Taxibots and aircraft operating at AAS. Actual taxiway centrelines were extracted to construct a graph in which four types of operations were planned. Operation were performed by vehicles equipped with heterogeneous kinematics, dimensions and objectives. The path planning algorithm utilises augmented versions of a PBS and SIPP technique to find a set of conflict-free operations while respecting the general properties of vehicles.

A short (90 seconds), normal (120 seconds), and long (150 seconds) uncoupling duration was used to evaluate ground delays resulting from uncoupling at holding platforms, taxiways, or both. Delays were analysed by comparing the duration of operations in Taxibot scenarios with a simulated conventional scenario for two days of real-world operational data. Minor delays were found for inbound traffic. For outbound traffic, delays were found mainly in the Taxibotting phase. Delays are smallest when uncoupling is allowed on both holding platforms and taxiways. Delays resulting from uncoupling only on taxiways or holding platforms differ per runway as a results of varying arrangements in uncoupling points, and can therefore not be generalised. When measuring ground delays for aircraft departing at similar runways, delays strongly depend on the runway traffic demand, the uncoupling duration and the uncoupling configuration. Every uncoupling configuration has a maximum aircraft throughput, or capacity, determined by the uncoupling duration and the arrangement of uncoupling points. When exceeding the capacity, queues emerge and delays stack up. Configurations with low capacity contain uncoupling points arranged in series that fully or partially block the flow of other traffic. Configurations with high capacity contain many uncoupling points that can be avoided by alternative paths. The search objective coefficients used by the path planning algorithm can be tuned to reduce delays, especially when uncoupling locations are arranged in series.

A broad picture of various uncoupling operations and resulting delays is provided in this research. Future research should further align the uncoupling operations with connecting airport processes. Research could subsequently analyse specific arrangements of uncoupling points. Such analysis should first be aimed at tuning the search objective coefficients of the path planning algorithm to minimise delays. Then, the capacity of the uncoupling points should be established to determine if the configuration allows for the desired runway throughput. If not, one could consider temporarily adjusting runway schedules or opening additional uncoupling locations during outbound peak to prevent delays to emerge.

References

[1] Airport Council International (ACI) Europe, "Sustainability strategy for airports (second edition)." [Online]. Available: https://www.aci-europe.org/industry-topics/industry-topics/28-airport-sustainability.html, November 2020. Accessed: 27-07-2021.

[2] Airport Operators Association (AOA), "Aircraft on the ground co2 reduction programme." [Online]. Available: https://www.sustainableaviation.co.uk/wp-content/uploads/2018/06/Aircraft-On-the-Ground-CO2-Reduction-Programme-Best-Practice-Guidance.pdf, June 2018. Accessed: 01-08-2021.

[3] I. Edema, O. Ikechukwua, I. Aniekan, E. Ikpeb, "The future of conventional aircraft ground propulsion systems in relation to fuel consumption and co2 emission," *International Journal of Thermal & Environmental Engineering*, vol. 13, pp. 91–100, Dec 2016.

[4] H. Khadilkar and H. Balakrishnan, "Estimation of aircraft taxi fuel burn using flight data recorder archives," *Massachusetts Institute of Technology Cambridge*, vol. 17, pp. 532–537, Oct 2012.

[5] Eurocontrol, "Seven-year forecast 2021- 2027 - main report." [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-forecast-update-2021-2027, June 2021. Accessed: 30-07-2021.

[6] J. Hospodka, “Cost-benefit analysis of electric taxi systems for aircraft,” *Journal of Air Transport Management*, vol. 39, pp. 81–88, July 2014.

[7] M. Lukic, P. Giangrande, A. Hebala, S. Nuzzo, M. Galea, “Review, challenges and future developments of electric taxiing systems,” *IEEE Transactions on Transportation Electrification*, vol. 5, pp. 1441 – 1457, Dec 2019.

[8] D. Fordham, M. Stephens, A. Chymiy, H Peace, G. Horton, C. Walker, M. Kenney, C. Morrow, M. Chester, Y. Zhang, and P. Sichko, “Deriving benefits from alternative aircraft-taxi systems,” Tech. Rep. Project 02-50, National Academy of Sciences, Transportation Research Board, Business Office, 500 Fifth Street, NW, Washington, Dec 2016.

[9] R. Guo, Y. Zhang, Q. Wang, “Comparison of emerging ground propulsion systems for electrified aircraft taxi operations,” *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 98–109, July 2014.

[10] S. Malicki, V. Howie, V. Bird, “Aircraft towing systems.” [Online]. Available: https://at-system.eu/, Dec 2017. Accessed: 23-07-2021.

[11] Smart Airport Systems (SAS), “The only certified & operational alternative taxiing solution.” [Online]. Available: https://www.taxibot-international.com/untitled, Nov 2020. Accessed: 12-07-2021.

[12] D. Bresser and S. Prent, “Sustainable taxiing and the taxibot.” [Online]. Available: https://www.schiphol.nl/en/innovation/blog/sustainable-taxiing-taxibot-trial/, May 2021. Accessed: 07-05-2021.

[13] D. Bresser and S. Prent, “Outcome Taxibot Feasibility Study Schiphol.” [Online]. Personal conversation, Jan 2021. Interview data: 20-01-2021.

[14] R. Morris, M. Chang, E. Cross II, J. Franke, R. Garrett, W. Malik, K. McGuire, and G. Hemann, “Self-driving aircraft towing vehicles: A preliminary report,” pp. 40–66, Jan 2015.

[15] Z. Chua, M. Cousy, M. Causse, and F. Lancelot, “Initial assessment of the impact of modern taxiing techniques on airport ground control,” pp. 1–8, Sept 2016.

[16] F. Klügl and A. Bazzan, “Agent-based modeling and simulation,” *AI Magazine*, vol. 33, pp. 29–40, Sept 2012.

[17] K. Hagera, J. Rauhb, W. Rid, “Agent-based modeling of traffic behavior in growing metropolitan areas,” *Transportation Research Procedia*, vol. 10, pp. 306–315, July 2015.

[18] C. Dou, D. Hao, B. Jin, W. Wang, N. An, “Multi-agent-system-based decentralized coordinated control for large power systems,” *International Journal of Electrical Power Energy Systems*, vol. 58, pp. 130–139, June 2014.

[19] H. Udluft, “Decentralization in air transportation.” [Online]. Available: https://repository.tudelft.nl/, Dec 2017. Accessed: 03-03-2021.

[20] K. Fines, “Agent-based distributed planning and coordination for resilient airport surface movement operations.” [Online]. Available: https://repository.tudelft.nl/, Nov 2019. Accessed: 03-03-2021.

[21] B. Benda, “Agent-based modelling and analysis of non-autonomous airport ground surface operations.” [Online]. Available: https://repository.tudelft.nl/, Nov 2020. Accessed: 03-03-2021.

[22] G. Sharon, R. Stern, A. Felner, N. R. Sturtevant, “Conflict-based search for optimal multi-agent pathfinding,” *Artificial Intelligence*, vol. 219, pp. 40–66, Nov 2015.

[23] H. Ma, D. Harabor, P. Stuckey, J. Li, and S. Koenig, “Searching with consistent prioritization for multi-agent path finding,” *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 219, pp. 7643–7650, Feb 2019.

[24] V. Metz and N. Girard, “Interview on taxibot uncoupling operations.” [Online]. Personal conversation, Nov 2020. Interview data: 13-11-2020.

[25] Schiphol24, “6 runways on amsterdam airport schiphol: de polderbaan, de kaagbaan, de buitenveldertbaan, de aalsmeerbaand, de zwanenburgbaan, en de oostbaan.” [Online]. Available: https://www.schiphol24.nl/en/runways/, March 2022. Accessed: 03-02-2022.

[26] J. Soomers, “A delay analysis in airport engine-off towing operations using hierarchical multi-agent path planning.” [Online]. Available: https://repository.tudelft.nl/, June 2022. Accessible from: 03-06-2022.

[27] M. Phillips and M. Likhachev, “Sipp: Safe interval path planning for dynamic environments,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5628–5635, June 2011.

[28] F. Klügl, “A validation methodology for agent-based simulations,” pp. 39–43, Jan 2008.

[29] CASPAR, Schiphol, “Data analysis for schiphol.” [Online]. Personal conversation, Feb 2019. Interview data: 20-01-2021.

[30] P. Sedgwick, “Multiple significance tests: the bonferroni correction,” *BMJ (online)*, vol. 344, pp. 509–509, Jan 2012.

[31] A. Vargha and H. Delaney, “A critique and improvement of the cl common language effect size statistics of mcgraw and wong,” *Journal of Educational and Behavioral Statistics*, vol. 25, pp. 101–132, July 2000.

[32] A. Savitzky, J. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, pp. 1627–1639, July 1964.

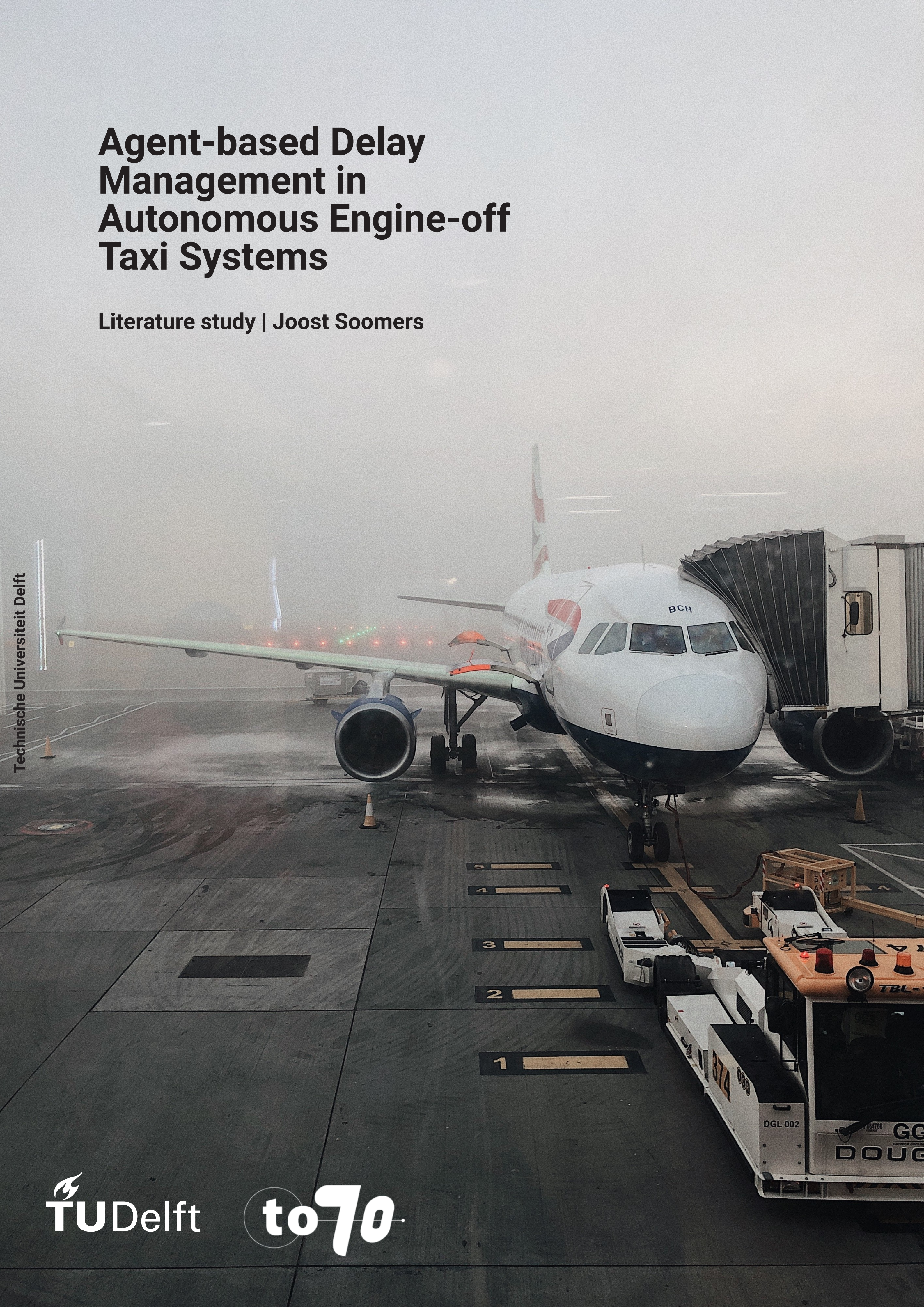
II

Literature Study

Previously graded under AE4020

Agent-based Delay Management in Autonomous Engine-off Taxi Systems

Literature study | Joost Soomers



Technische Universiteit Delft

[This page is intentionally left blank]

Agent-based Delay Management in Autonomous Engine-off Taxi Systems

Literature Study

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science
AEROSPACE ENGINEERING

AT THE DELFT UNIVERSITY OF TECHNOLOGY

by:
Joost Soomers

Supervisors:
Alexei Sharpanskykh *(TU Delft)*
Xander Mobertz *(To70)*

DRAFT VERSION

COVER IMAGE RETRIEVED FROM PEXELS [1]

Contents

List of Figures

List of Tables

Acronyms

(TAXI-CPDLC) Controller-Pilot Data Link Communication during taxiing.

A-CDM Airport Collaborative Decision Making.

A-SMGCS Advanced Surface Movement Guidance Control System.

AAS Amsterdam Airport Schiphol.

ABM Agent-Based Model.

ACERT Airport Carbon and Emission Reporting Tool.

ACI Airport Council International.

AEDT Aviation Environmental Design Tool.

AIBT Actual In-Block Time.

AIP Aeronautical Information Publications.

ALDT Actual Landing Time.

ANN Artificial Neural Network.

AO Aircraft Operator.

AOBT Actual Off-Block Time.

APSP All-Pairs Shortest Path.

APU Auxiliary Power Unit.

ARDT Actual Ready for Departure Time.

ASAT Actual Start-up Approval Time.

ASRT Actual Start-up Request Time.

ATC Air Traffic Control.

ATCo Air Traffic Control operators.

ATOT Actual Take-Off Time.

BCBS Boundec Conflict Based Search.

BF Bellman-Ford.

c.g. Center of Gravity.

CA* Cooperative A*.

CBM Conflict-Based Min-cost-flow.

CBS Conflict Based Search.

CBS + HWY Conflict Based Search with Highways.

CBS-TA Conflict Based Search with Task Allocation.

CBSw/P Conflict Based Search with Priorities.

CDQM Collaborative Departure Queue Management.

CDS Collaborative Departure Scheduling.

CO-WHCA* Conflict Oriented - Window Hierarchical Cooperative A*.

CO-WHCA*P Conflict Oriented - Window Hierarchical Cooperative A* prioritization.

CP Constraint Programming.

CPDSP collaborative Pre-Departure Sequence Planning.

CT Constraint Tree.

CTOT Calculated Take-Off Time.

ECBS Enhanced Conflict Based Search.

ECBS-TA Enhanced Conflict Based Search with Task Allocation.

EOBT Estimated Off-Block Time.

EXIT Estimated Taxi-In Time.

EXOT Estimated Taxi-Out Time.

FAA Federal Aviation Administration.

FAR Flow Annotation Replanning.

FBI Fuel Burn Index.

FOD Foreign Object Damage.

FtG Follow the Greens.

GCBS Greedy Conflict Based Search.

GDSP Goal-Directed Shortest Path.

GH Ground Handler.

HCA* Hierarchical Cooperative A*.

IAI Israel Aerospace Industries.

IATA International Air Transport Association.

ICAO international Civil Aviation Organisation.

LP Linear Programming.

LRA* Local Repair A*.

LVNL Luchtverkeersleiding Nederland.

MAPD Multi-Agent Pickup and Delivery.

MAPF Multi-Agent Path Finding.

MAPF-DP Multi-Agent Path Finding with Delay Probability.

MAPF-POST Multi-Agent Path Finding with POST processing.

MAPP Multi-Agent Path Planning.

MAS Multi-Agent Systems.

MILP Mixed Integer Linear Program.

MIT Massachusetts Institute of Technology.

MLG Main Landing Gear.

MTTT Minimum Turnaround Times.

NASA National Aeronautics and Space Administration.

NLG Nose Landing Gear.

NM Nautical Miles.

NMOC Network Manager Operations Centre.

PBS Priority Based Search.

PRC Pushback Rate Control.

PT Priority Tree.

RMO Runway Mode of Operation.

RRA* Reverse Resumable A*.

SARDA Spot and Runway Departure Advisor.

SAS Smart Airport Systems.

SE Single Engine-on.

SESAR Single European Sky ATM Research.

SIC Sum of Individual Costs.

SID Standard Instrument Departure.

SSSP Singe-Source Shortest Path.

TA Task Allocation.

TIBT Target In-Block Time.

TLD Teleflex Lionel Dupont.

TLDT Target Landing Time.

TOBT Target Off-Block Time.

TP Token Passing.

TPTS Token Passing with Task Swaps.

TSAT Target Start-up Approval Time.

TTOT Target Take-Off Time.

WHCA* Window Hierarchical Cooperative A*.

WTC Wake vortex separation.

1 Introduction

Ground surface movements of civil aircraft at airports contributes significantly to the fuel burn and environmental impact of the aviation industry. It is estimated that aircraft spend on average 10-30 minutes of their flight time taxiing and that a short/medium range aircraft uses between 5-10% of its fuel on the ground [27]. These numbers will most certainly increase. Global air traffic passenger demand has increased yearly with around 4.2% over the past few decades [28]. Even though the current decline in the total number of air passengers as a result of COVID-19 is enormous, International Air Transport Association (IATA) predicted in March 2021 that the number of short haul and long haul flights will be recovered to pre-COVID levels in 2024 and 2025 respectively [29]. The international Civil Aviation Organisation (ICAO) states that in the most likely scenario, the annual aviation growth will approach the historic rate of 4.2% again after the industry is fully recovered [30].

The rise in taxi time leads to higher environmental impact and capacity problems at airports. Usually, capacity problems are solved by additional infrastructure on and around airports. This is currently no option anymore as many airports, including Amsterdam Airport Schiphol (AAS), are approaching limits in terms of size and expansion leads to resistance by the surrounding population [31]. Moreover, increasing airport capacity does not solve the environmental problem of ground movements. In fact, studies have shown that larger airports generally have longer taxi times, leading to more environmental impact [32]. The call for environmental impact reduction and increased airport capacity might be answered by alternative taxi systems. For this reason, methods to replace thrust of aircraft main engine by alternative power sources in combination with different (and faster) operations is broadly studies over the last years.

A promising method identified in this study is the use of towing vehicles. Such vehicles tow aircraft to and from runways so that the need for ground propulsion by the main aircraft engines is eliminated. The only certified external towing vehicle on the current market is Taxibot. The concept owners of Taxibot, Israel Aerospace Industries (IAI), claims to reduce taxi emissions by 50-85% [33]. It is however evident that the need for uncoupling near runways and the additional vehicles on the taxiways increases the operational complexity of ground movements. This might lead to additional ground delays and reduced airport capacity. An autonomous system of aircraft and Taxibot might offer the needed relief. Autonomy has proven to increase efficiency, safety and security in many complex operational systems [34]. Research at the Delft University of Technology has previously demonstrated the power of autonomous control in conventional airport ground movements [10]. Such control systems allow for autonomous operations and thus replace the need for human interaction. The role of Air Traffic Control (ATC) shifts from controlling to supervising. This literature study is the start of a research that is focused on the development of an autonomous airport ground surface towing system in which cooperative coordination mechanisms are introduced to mitigate expected ground delays. A long-term perspective plan for AAS is developed that can be applied to other airports at later stages. The outcome of this research is relevant in the sense that the effectiveness of towing systems like Taxibot at big hub airports such as AAS depends on whether the expected congestion and ground delays can be mitigated sufficiently.

The main driver of this research is to make airport ground movements greener and more sustainable. This document contains the literature study performed prior to the research. The document is structured as follows. The first two chapter are established to gather background information related to the objective of more sustainable ground movements. Therefore, conventional airport ground movement operations are studied in chapter 2. In particular, airports using A-CDM and AAS are addressed. The environmental impact of these ground movements are subsequently quantified in chapter 3. Given the information of conventional taxi operations and its environmental impact, mitigation options can be considered. Therefore, chapter 4 and chapter 5 are devoted to study methods able to reduce environmental impact of ground movements by operational improvements and alternative systems. Based on these findings, a trade-off is preformed and the most promising alternative taxi system is selected. This method is then studied in more detail after which a research gap is presented. The set-up of the research is the development of a model that simulates ground surface movements. Therefore, Chapter 6 studies the given agent-based modelling method and reviews previous studies in the field of agent-based airport ground systems. At the end of the chapter, a model is selected to use as baseline. After this, chapter 7 studies cooperative coordination and planning techniques potentially applicable in this research. A trade-off between different mechanism is performed, and the most suitable one is selected. Based on all the literature, a detail research proposal is given in chapter 8. The report is concluded with chapter 9, related to research methodologies. In this chapter, the research strategy is discussed first. Then, all assumptions, simulation constants and independent variables are given in the research scope. finally, the outcome and relevance is highlighted and the research planning is presented.

2 Conventional Airport Surface Movement Operations

Before ground movement emissions can be analysed and alternative taxi systems can be found, the conventional ground movement operations are studied. These operations include those at the gates, aprons (or ramps), taxiways and runways. The different components of this system are visualised in the general schematic in Figure 2.1.

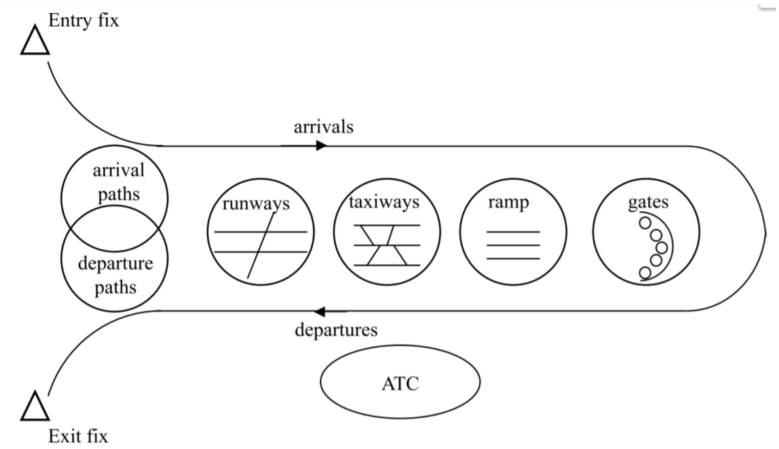


Figure 2.1: General schematic of an airport ground system [2].

All different components have associated operational protocols and are jointly executed by the Aircraft Operator (AO), Ground Handler (GH), ATC and the airport. To organise these joint operational procedures, most European hub airports, including AAS make use of the A-CDM concept. This concept is developed by EUROCONTROL to facilitate sharing of operational information and data to achieve better informed decisions to be made [3]. One of the key aspects of A-CDM is the milestone approach, an operational concept that marks significant events during the planning and operation of the arrival and departure of aircraft. This concept is discussed in section 2.1. The approach contains many elements, but only those relevant for this research are discussed. The collaborative Pre-Departure Sequence Planning (CPDSP), an important part in the milestone approach concept, is one of the key concepts of this research study. Therefore, it is discussed more elaborately in section 2.2. This section also contains another key concept of this research: the determination of the estimated taxi times. The most important stakeholders in the A-CDM process, are discussed in section 2.3. Four different types of Air Traffic Control operators (ATCo) are discussed that are together responsible for all the ground surface movements. AAS is used for the case study of this research. Therefore, its current ground operations and layout is analysed in section 2.4.

2.1. The Milestone Approach in A-CDM

The milestone approach marks significant events that occur during the planning and operation of the inbound, turn around and outbound process of a particular flight. The aim is to achieve a common situational awareness of all involved parties and to predict the timing of forthcoming events. At the completion of a milestone, the progress of other downstream events can be more accurately predicted and planned [3]. The approach, visualised in Figure 2.2, consists of three main components: inbound, turnaround and outbound. An estimation is made of the Estimated Taxi-In Time (EXIT), Minimum Turnaround Times (MTTT) and Estimated Taxi-Out Time (EXOT) in the respective components. The EXIT and EXOT are calculated by ATC and whereas the MTTT is agreed upon by GH and AO.

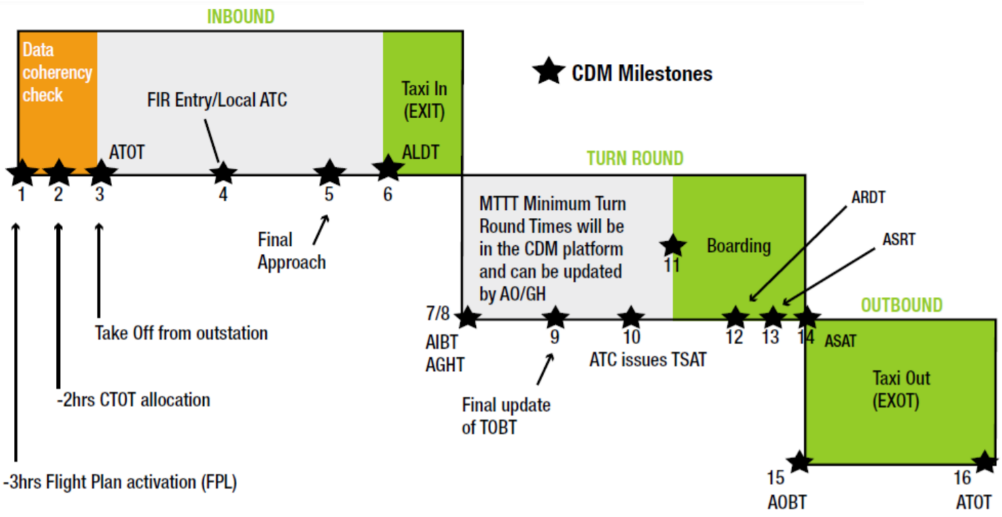


Figure 2.2: The milestone approach of A-CDM [3].

The most critical events and their location in the milestone approach are summarised in Table 2.1. Each event occurs at a certain time, denoted with A (Actual time). The timing of all events are first estimated before the turnaround process and logged in the flight plan, denoted with E (Estimated time). During the decision making process, all events get a more specific time stamp, denoted with T (Target time). This for example included arrival and departure sequencing (in contrast to estimated times).

Table 2.1: Critical events in the milestone approach (*terms that are not used).

Time (T) → ↓ **Type**	6 ★ Landing (LDT)	7 ★ in-block (IBT)	14 ★ Start-up approval (SAT)	15 ★ off-block (OBT)	16 ★ Take-off (TOT)
Estimated (E)	ELDT (or ETA)	EIBT	ESAT*	EOBT	ETOT (or ETD)
Target (T)	TLDT	TIBT*	TSAT	TOBT	TTOT
Actual (A)	ALDT	AIBT	ASAT	AOBT	ATOT

2.1.1. Inbound

The first four milestones indicate the timing of pre-arrival information sharing and are of less interest in this study. The flight enter the final approach usually between 2 and 5 minutes before landing. This is indicated with milestone 5. This milestone triggers GH to move equipment and vehicles needed during the turnaround process of the aircraft, such that this can immediately start upon arrival at the gate. ATC assigns a Target Landing Time (TLDT) and calculates the EXIT. With this new information, the Target In-Block Time (TIBT) can be updated and target times of other downstream events are adjusted. Milestone 6 indicates the landing of the aircraft. ATC stores the Actual Landing Time (ALDT) which might again lead to updated times of other downstream events. Milestone 7 indicate the arrival at the gate, and the Actual In-Block Time (AIBT) is stored in the system.

2.1.2. Turnaround

Milestone 8 represent the start of the turnaround process by the GH. If enough GHs are available and no other unusual situation occur, this process start directly upon arrival, meaning that milestone 7 and 8 occur at the same time. Throughout the turnaround process, the TOBT is updated by the AO and GH to the most up-to-date prediction. t_1 minutes before Estimated Off-Block Time (EOBT), at milestone 9, the final Target Off-Block Time (TOBT) is provided to ATC. The aim of this final TOBT is to provide ATC an accurate estimation at a certain fixed point in time, EOBT becomes fixed after arrival. This TOBT is needed to assign a pushback vehicle and to establish the pre-departure sequence. The pre-departure sequence is determined by ATC ATC using a CPDSP. This planning system is very relevant for this research and is therefore discussed in more depth in section 2.2. The outcome of this CPDSP is the Target Take-Off Time (TTOT) and Target Start-up Approval Time (TSAT). The TSAT is given in a 10 minutes window (TSAT +/- 5 min) and is given at milestone 10. Boarding starts at milestone 11. At milestone 12, the aircraft is completely ready to taxi upon ATC instruction, meaning that all doors are closed, boarding bridge is removed and the pushback vehicle is connected. This time is denoted as the Actual Ready for Departure Time (ARDT). Closely after ARDT, the aircraft request the start-up, at milestone 13. This time is denoted with the Actual Start-up Request Time (ASRT). If the ARDT is within the TSAT window, the aircraft maintains its position in the departure sequence and waits for the start up approval. Should the ARDT not be within the TSAT window, the CPDSP determines a new TTSAT window resulting in a slightly delayed departure time. At milestone 14, the start-up is approved and the Actual Start-up Approval Time (ASAT) is stored in the A-CDM system.

2.1.3. Outbound

During normal procedures, a couple of seconds after milestone 14, the aircraft departs from its gate, the Actual Off-Block Time (AOBT) is stored into the A-CDM system and (as usual) other downstream events like the TTOT is updated if necessary. This event marks milestone 15. As soon as the aircraft arrives at the runway and departs, the Actual Take-Off Time (ATOT) is recorded, marking the end of the milestone approach.

2.2. Collaborative Pre-Departure Sequence Planning (CPDSP)

An important element of the A-CDM system is the CPDSP, visualised in Figure 2.3. The planning is performed at fixed time intervals and determines the TTOT (and thus also the takeoff sequence) of all aircraft that do not have a TTOT assigned or are not able to meet its previous assigned TTOT. Because the CPDSP schedules departure times for multiple aircraft, it is used at fixed time intervals and thus milestone 10 does not follow directly after milestone 9.

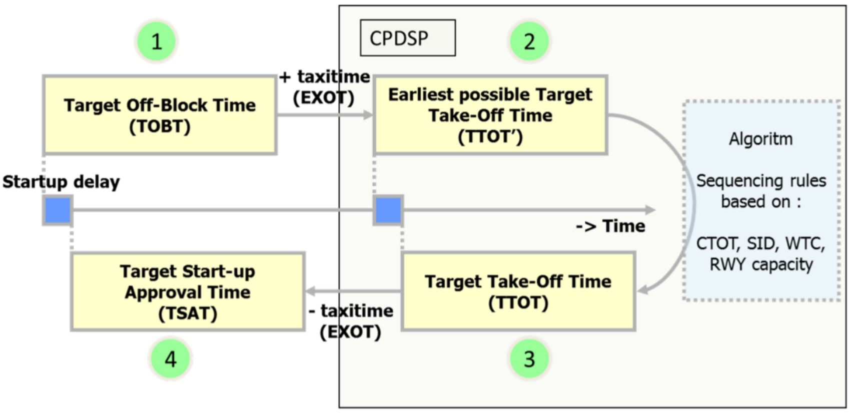

1
2
3
4
CPDSP
Target Off-Block Time (TOBT)
+ taxitime (EXOT)
Earliest possible Target Take-Off Time (TTOT')
Algoritm
Sequencing rules based on :
CTOT, SID, WTC, RWY capacity
Startup delay
-> Time
Target Take-Off Time (TTOT)
- taxitime (EXOT)
Target Start-up Approval Time (TSAT)

Figure 2.3: Collaborative pre-departure sequence planning (between milestone 9 and 10) [3].

Before the CPDSP runs, the EXOT is calculated for each flight by ATC. This estimated time is added to the received TOBT at milestone 9 to determine the earliest possible TTOT (denoted with TTOT'). This TTOT', together with the TTOT' of other aircraft, is the input to the algorithm that optimises the departure sequence. The objective of this optimisation algorithm is to minimise the startup delay (TSAT - TOBT) in combination with the optimisation of certain bottlenecks in the airport system. These bottlenecks might include: apron, gates, taxiways, de-icing areas or runways. The algorithm takes the following constraints into account:

- **TSAT - t$_{now}$ ≥ 0**: TSAT cannot be in the past [35].
- **TTOT = TSAT + EXOT**: The TTOT (determined by the optimisation) must be equal to the TSAT (also determined by the optimisation) plus the EXOT [35].
- **TSAT ≥ TOBT**: The aircraft can not be pushed back before it is ready. Note that, in combination with constraint 2, this also implies that TTOT ≥ TTOT' [35].
- **CTOT − 5 ≤ TTOT ≤ CTOT + 10**: ATC has to comply with Eurocontrol instruction. To manage bottlenecks in airspace, Eurocontrol may issue a Calculated Take-Off Time (CTOT) for flights. This CTOT is a time window of 15 minutes (CTOT -5/+10 min). This windows is a hard constraint; if the flight is not able to depart within this window, a new CTOT has to be requested, resulting in additional delays and associated costs [3].
- **WTC**: Wake vortex separation (WTC) rules must be respected. The time based separation minima for departing aircraft depends on the weight category of the preceding aircraft and following aircraft. Minima set by Eurocontrol are given in Table 2.2 [26].
- **SID**: Depending on the Standard Instrument Departure (SID) of a flight, two adjacent flights are not allowed [3].
- **Radar separation**: The radar separation must be respected at all times. This separation minima is applied for the radar to be able to distinguish the radar pulses reflected by two aircraft. Therefore, ICAO standards state that a separation between two aircraft in terminal control areas should be at least 5 Nautical Miles (NM). Depending on the surveillance systems' capabilities at the airport, this minimum may be reduced by ATC to 3 NM and 2.5 NM on final approach and departure respectively (within 10 NM of the runway threshold) [36].
- **Runway Capacity**: Extra attention needs to be given to the runway capacity in case the runway is used for both arrivals and departures [3].

The output of this optimisation is the departure sequence, in which each aircraft has its own assigned TTOT and TSAT, given to them in milestone 10. Note again that this is a continuous process; TOBTs from aircraft arrive continuous and therefore, this optimisation process is done repeatedly. The window of the TSAT is 10 minutes wide (TSAT +/- 5 min). This means that every time the AO or GH updates the TOBT to a new time later than 5 minutes, the CPDSP needs to recalculate a new TSAT and TTOT. Other reasons for recalculations are runway reconfigurations or a change in meteorological conditions.

Finally, note that the CPDSP is extremely dependant on a correct estimation of EXOT. This is also a relevant feature in this research. Therefore, extra attention is given to the establishment of EXOT (and therefore also EXIT) in subsection 2.2.1.

Table 2.2: RECAT-EU departure separation minima [seconds]. RECAT-EU is developed by Eurocontrol and is based on the ICAO wake turbulence longitudinal separation. It is a re-categorisation of the ICAO standards to optimise departure throughput [26].

Follower → **↓ Leader**	Super Heavy (A)	Upper Heavy (B)	Lower Heavy (C)	Upper Medium (D)	Lower Medium (E)	Light (F)
Super Heavy (A)		100	120	140	160	180
Uper Heavy (B)				100	120	140
Lower Heavy (C)				80	100	120
Upper Medium (D)						120
Lower Medium (E)						100
Light (F)						80

2.2.1. Calculating EXIT and EXOT

Accurate taxi time predictions are essential for the determination of downstream events in the turnaround process. Especially for the functioning of CPDSP, a correct calculation of EXOT is very important. At large airports like AAS the configuration of the runways and layout of terminals can result in significant different taxi times. The following aspects are used to estimate these times:

1. **Default taxi times.** Most small airports today attribute a single taxi time to each runway configuration. These values apply to all types of aircraft, all weather conditions and all parking stands.

2. **Operational expertise.** As partners like aircraft operators, ground handlers and ATC regularly deal with taxi times, their expertise is often used in the determination of the taxi time.
3. **Aircraft type.** The aircraft type should be taken into account as a variation in taxi speed profiles exists between different types of aircraft. Possible alternative taxi systems (e.g. towing systems) should be taken into account.
4. **Historical data.** Estimated taxi times could be based on historical data, if other factors such as taxi route, meteorological conditions and aircraft type are nearly equal.
5. **Operational conditions.** Deteriorated weather conditions normally increase taxi time.
6. **Taxi route.** Taxi times can be estimated by dividing the length over the route by the average speed.

2.3. Aerodome Control Parties

The broader outlines of the A-CDM approach and in particular the elements in the key aspects of the milestone approach and the CPDSP are clear. However, the main aim of the A-CDM concept is the facilitate sharing of operational data to achieve better situational awareness of all parties involved. This section briefly analyses these involved parties, and studies their respective roles in the process. The analysis is done by the socio-technical representation given in Figure 2.4. The representation is created based on data given in previous MSc studies [10–12]. For reasons that become more clear throughout in the report, this MSc study is mainly focused on departing aircraft. Therefore, the analysis in this section is performed for a departing aircraft at an airport that uses the A-CDM concept. The analysis is done in chronological order of a departing aircraft.

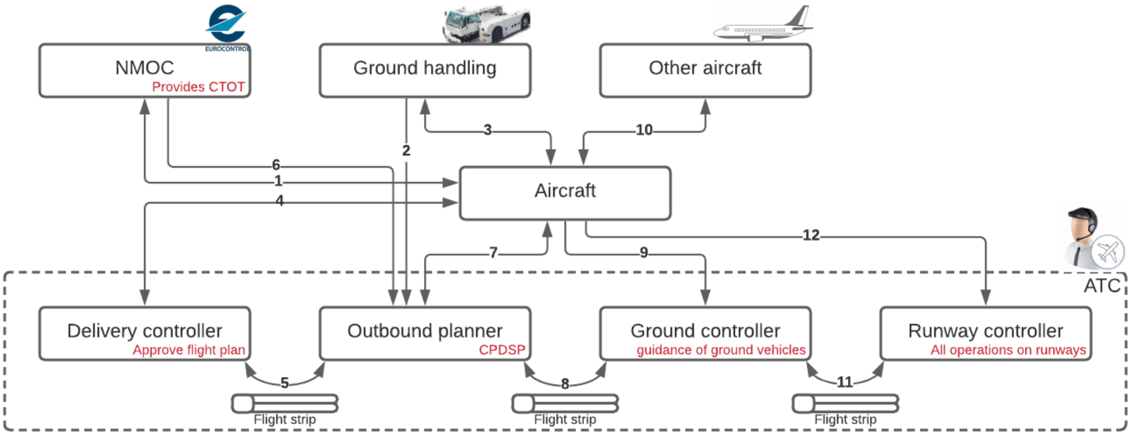


Figure 2.4: Socio-technical system of airport ground movements.

Aircraft and Ground handler

Two hours before EOBT, at milestone 2, the aircraft operators request and subsequently receive a CTOT from Network Manager Operations Centre (NMOC) at EUROCONTROL *(link 1)*. The AOs and the GHs take care of the turn around process. This process starts at milestone 7 and ends at milestone 14. Throughout this process, the GH and AO constantly communicate the most up-to-date estimation of the TOBT to the Outbound planner *(link 2,7)*, who is responsible for the CPDSP. The GH and AO are constantly in touch about the progress of the turn around process. The AO interact with the GH to communicate the onboard status of flight preparation, such as boarding and cleaning. Visa-versa, the ground handler communicates information related to off board activities such as refuelling and catering *(link 3)*.

Delivery controller

The delivery controller checks the flight plan and provide clearance to execute the plan *(link 4)*. This clearance is usually given 30 minutes in advance. If the flight plan is approved, the delivery controller forwards the flight strip (containing all data of the flight) to the outbound planner. In case changes occur such that the clearance must be revised, the flight strip is returned to the delivery controller *(link 5)*.

Outbound planner
The main responsibility from the outbound planner is to grant the TSAT. This is determined by the CPDSP. The inputs for this planning process is the TOBT *(link 2,7)* and the CTOT *(link 6)*. The outbound planner is also responsible for a accurate determination of EXOT. Ones the TSAT is determined, the outbound planner communicates this to the AO *(link 7)* and forwards the flight strip to the ground controller *(link 8)*.

Ground controller
The ground controller is responsible for the guidance of all ground vehicles at the airport. This includes movements on the taxiways, ramps and gates. During pushback and taxiing, the ground controller is constantly in contact with the aircraft to provide instruction related to the aircraft movement *(link 9)*. Depending on the situation, conflicts between ground vehicles are avoided either by ATC instruction or by visual separation *(link 10)*. Ones the aircraft has arrived at the runway, the flight strip is forwarded to the runway controller *(link 11)*.

Runway controller
The runway controller is responsible for all the movements on and around the runway. The controller gives clearance for departing and arriving aircraft to take off and land. It does so by respecting the separation requirements stated earlier. The runway controller is also responsible for guidance and instructions if runways needs to be crossed *(link 12)*.

2.4. Ground System of AAS

In this MSc research, AAS (IATA: AMS, ICAO: EHAM) is used as a case study. With more than 71 million passengers and around 500.000 aircraft movements, AAS was the third-busiest airport in terms of passenger volume and the busiest in terms of aircraft movements in Europe in 2019 [37]. Due to the historic background and the fast growth of the airport, AAS currently features a very complex and interesting ground system. This section studies all relevant areas and details of the airport system for this research.

As presented in the schematic of the airport ground system in Figure 2.1, the components of the turnaround process are the runways, taxiways, ramps and gates. The layout of all these segments on AAS are important throughout this study. Therefore, three maps have been added to Appendix D. These maps have been obtained from the Aeronautical Information Publications (AIP) of Luchtverkeersleiding Nederland (LVNL) [38]. The first map highlights the runway system at AAS, and is discussed in more detail in subsection 2.4.1. The second map is focused on the taxiway system of the airport. This is studied in more detail in subsection 2.4.2. Finally, the ramps, gates and parking areas are given in the third map of the appendix and is discussed in subsection 2.4.3.

2.4.1. Runway Usage

AAS features six runways. Five runways for international aviation, and a short runway for private jets and general aviation. The runway configuration (usually referred to as Runway Mode of Operation (RMO)) is the combination of runways open for arrivals or departures. It is determined by ATC and depends on factors such as the weather conditions, traffic density and other agreements. The runway configuration can change up to 14 times a day [11]. The configuration exists of multiple runways that are open in one of the two directions for arrival or landing. The two most common RMO at AAS are RMO North (departure: 36L + 36C, arrival: 06 + 36R) and RMO South (departure: 18L + 24, arrival: 18C + 18R) [4]. Figure 2.5 shows the forecast of runway usage for the year 2021, expressed in the number of runway movements. The spread around the expected runway use represent the degree of uncertainty due to uncertain traffic scenarios and changing weather conditions. As these number represent a forecast for next year, the effect of COVID-19 is taken into account. This research is more focused on the long term perspective possibilities at AAS. Therefore, pre-COVID-19 number are used throughout this study. The ratio in runway utilisation is not affected by COVID-19.

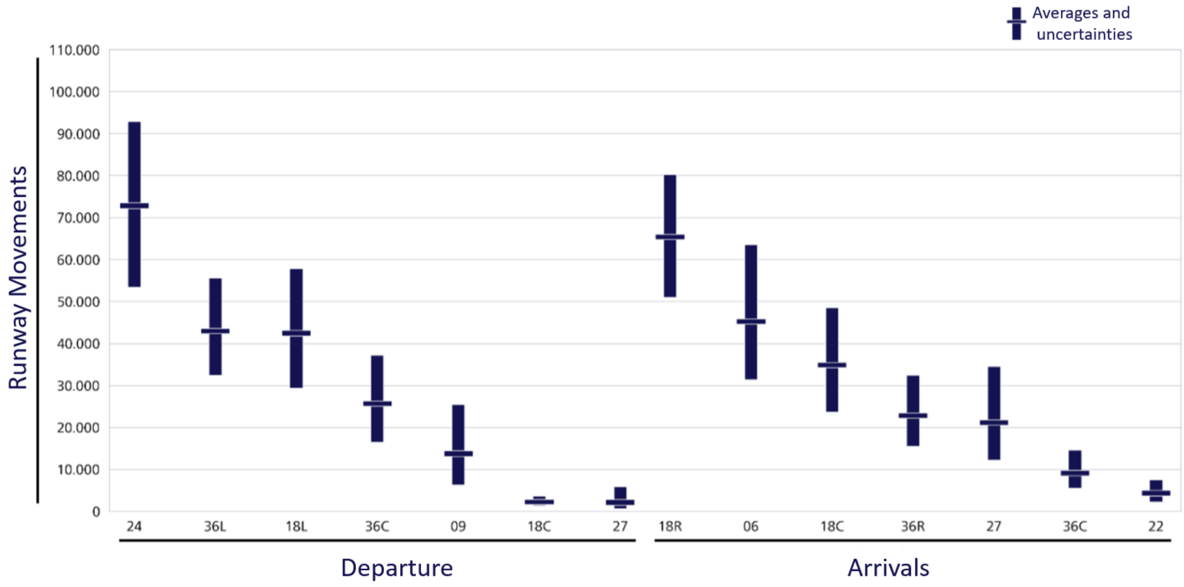


Figure 2.5: Forecast runway usage of AAS, 2021 [4].

For reasons that will become more clear throughout the report, the departure runways are most relevant in this study. These runways are 24, 36L, 18L and 36C.

2.4.2. Taxiway System

The taxiway system is presented in the second map of Appendix D. Aircraft are guided by ATC during taxiing. ATC has many coordination techniques to prevent conflicts from occurring. One of such techniques is to use pre-defined rules. An example is the pre-defined regulations on taxiway Alpha, Bravo and Quebec. Generally, aircraft are supposed to taxi in clockwise direction on Alpha and in counter-clockwise direction on Bravo. The direction of Quebec depends on the active runway configuration [39].

2.4.3. Ramp, Gates and Parking Areas

The ramp, gates and parking areas are shown in the third map of Appendix D. Later in this report, it shall become more clear that the gates and the parking areas near runways are highly relevant for this study.

The next chapter elaborates upon the environmental impact of the ground movements studied in this chapter, and specifically those of AAS. The chapters thereafter discusses operational methods and technological systems able to mitigate the impact.

3 Environmental Impact Assessment

In chapter 2, conventional airport surface movements operations at airports using A-CDM (like AAS) were studied. This chapter assesses the environmental effects of these operations such that a proper comparison between alternative taxi systems can be made.

Aircraft ground movement operations throughout the world are generally not so diverse. The main different that causes varying emissions and fuel consumption is the duration of aircraft taxiing. At AAS, the average taxi times for outbound and inbound flights are around 14 and 9 minutes respectively [40]. Similar size airports in the USA generally have higher average taxi-out times from around 20 minutes, with JFK airport having the highest average taxi-out time of 37.1 minutes [41]. A schematic, with AAS used as a baseline scenario, of the conventional aircraft ground movements for outbound and inbound taxiing is given in Figure 3.1. Included in the taxi-out operations is the pushback performed by a pushback tug, a process that takes around two minutes. It must be noted that Single Engine-on (SE) taxiing is already quite commonly adopted to safe fuel on inbound taxiing. Experts from AAS estimate that this method is applied in around 40% of the inbound taxi movements [42]. In section 4.1, this technique is discussed more extensively.

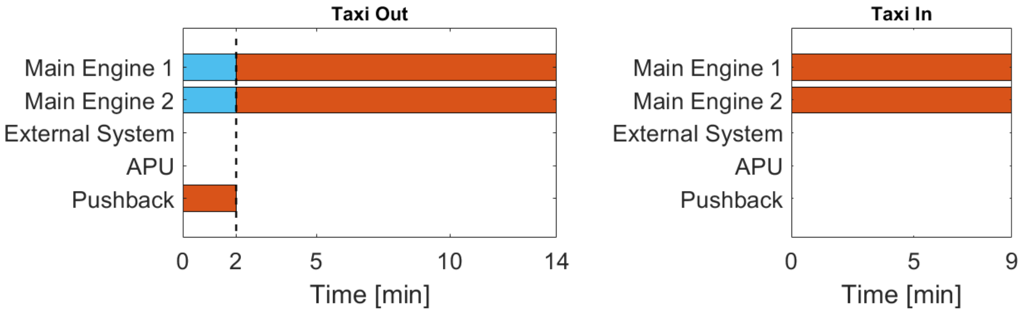


Figure 3.1: Schematic of conventional taxi operations. This scenario is used as a baseline throughout this report. Orange bar indicate that the main thrust for taxiing is provided by that system. A blue bar means that the system is active, but does not provide the thrust for taxiing.

The environmental impact results from the consumption of fuel, and the production of noise. The fuel consumption of conventional aircraft ground movement is shortly investigated and the two main mitigation methods are stated in section 3.1. Based on several tools and studies, section 3.2 elaborates on how the environmental impact of aircraft ground movements is defined in this report and why that is sufficient for this research.

3.1. Fuel Consumption

Aircraft engines used for ground propulsion is generally an extremely inefficient process [41]. The reason for this is that aircraft engines are optimised to power an aircraft in cruise conditions and not under taxi conditions. This inefficient process relates to a high Fuel Burn Index (FBI), relative to the obtained power. The relation of the FBI_i with the total fuel consumption FB_i of aircraft i is given in Equation 3.1 [32]. T_i and N_i are the runtime and the number of operating engines respectively.

$$FB_i = (T_i \cdot N_i) \cdot FBI_i \tag{3.1}$$

From this equation, one can easily observe that there are two main methods to limit the fuel consumption of conventional aircraft ground movements: reducing the FBI by making aircraft more efficient and reducing the engine runtime. Both methods are discussed below.

FBI: Fuel Burn Index

The fuel burned by aircraft engines during ground movements is, relative to the obtained power, extremely high in comparison with other ground propulsion systems. Generally, aircraft taxi with engine settings at 7% thrust, or idle [32]. According to 2010 ICAO engine database, this corresponds to a FBI ranging from 0.10 kg/s per engine for short/medium range aircraft to 0.38 kg/s for long range aircraft [43]. At AAS, the majority of aircraft (±80%) is short/medium range [44]. Therefore, a weighted average of the FBI for all aircraft at AAS is around 0.156 kg/s per engine. Many studies have been performed to increase the efficiency of aircraft engines, but that is beyond the scope of this study. For this research it is sufficient to know that the average fuel flow for taxiing aircraft at AAS is around 0.156 kg/s per engine and that this is extremely high relative to the obtained power in comparison with other ground propulsion systems [41].

Engine Runtime $(T_i \cdot N_i)$

The engine runtime is defined as the cumulative time one engine runs during aircraft ground movements. This research aims on limiting the fuel consumption of engines by reducing their runtime and replacing the propulsion of the ground movements by other, more efficient, methods. By observing the schematic in Figure 3.1, one can find that for conventional ground movements at AAS, the cumulative engine runtime $(T_i \cdot N_i)$ is equal to 40 minutes. chapter 4 and chapter 5 both studies new developments to reduce (or totally remove) this runtime by replacing the ground propulsion of aircraft by new, more efficient, methods.

3.2. Environmental Impact

The ability to quantify fuel consumption and emissions of aircraft during ground movements is of high importance for decision makers. Many studies have been done by aviation organisations, airlines and airports on emission quantification and reduction during aircraft taxiing. The ICAO developed an emission database of different type of engines to estimate fuel consumption and emissions under different operational conditions including taxiing [43]. Airport Council International (ACI) developed the Airport Carbon and Emission Reporting Tool (ACERT), a tool that can be used by airports to identify, quantify and manage their greenhouse gas emissions [45]. The Single European Sky ATM Research (SESAR) program came up with integrated surface management techniques to improve airport surface movement operations in terms of fuel use and emissions [46]. The Federal Aviation Administration (FAA) developed the Aviation Environmental Design Tool (AEDT), a software system that models aircraft performance to estimate fuel consumption, emissions, noise, and air quality consequences [47]. All these tools tend to qualify environmental impact from ground surface movement differently, but they all categorise environmental impact in three identical ways: noise, local air quality and emissions contributing to climate change.

In all the airport emission quantification methods stated above, it is generally recognised that ground noise is a significant lesser issue than airborne noise. However, it does affect some sensitive areas around the airport. Therefore, the reduction of noise is taken into account in the trade-off for alternative taxi systems, but shall be of less relevance than the reduction of emissions. This study thus primarily focuses on the emissions from jet engines during taxiing, leading to climate change and reduction in local air quality. Examples of these pollutant species are Carbon Dioxide (CO_2), Nitrogen Oxides (NO_X), Hydrocarbons (HC), Carbon Monoxide (CO), Sulfur Oxides (SO_2) and Particulate Matter (PM). The emitted water vapour (H_2O) is not considered as this only contributes to climate change if emitted at high altitudes. A first estimation of the quantity of an emitted species j for flight i can according to Equation 3.2[41]. In this equation, T_i presents the taxi time, N_i the number of engines running, FBI_i the fuel burn index and EI_{ij} the emission index.

$$E_{ij} = T_i \cdot N_i \cdot FBI_i \cdot EI_{ij} \tag{3.2}$$

In reality, the taxi time and number of operative engines are easily to determine, but the emission index depend heavily on factors such as throttle setting, engine type and meteorological conditions like ambient temperature and pressure [43]. With the ICAO emission database, an approximation of the quantity of the emitted species could be made and a CO_2 equivalent estimation for all emitted gasses can be calculated. This estimation could potentially be used in a trade-off between different alternative taxi systems. However, such estimation heavily depends on the location of the emitted species and do not include most feedback cycles present in nature. Furthermore, with this approach, only the contribution to climate change in terms of global warming is taken into account, not other forms of environmental impact. Assessing the environmental impact of airport taxiway systems thus require many sensitive variables.

To conclude, the impact of aircraft ground movements on the environment depends on the emitted quantity of different gasses, if measured in air quality and climate change contribution. The quantity of each emitted gas heavily depends on engine type, settings and meteorological conditions. Furthermore, the contribution of each gas to climate change and the decline in local air quality also varies with different meteorological and seasonal condition. The analysis of the emitted gasses and how they subsequently relate to the decline of the (local) environment is a study involving many aspects. Therefore, studies performed by Massachusetts Institute of Technology (MIT), aviation research programs or other aviation organisations or institutes all make use of databases from for example ICAO or ACI. Many other studies make the assumption that the total environmental impact is proportional to the fuel burned by the total system, as this is generally an adequate approximation for most research objectives. The latter will also be used in this study. The next chapters are therefore devoted to measures that contribute towards the reduction of fuel consumption by the total ground movement system. Chapter 4 introduces operational measures, and Chapter 5 alternative systems.

4

Alternative Taxi Operations

As stated in chapter 3, the two general ways of reducing environmental impact of ground movements is increasing engine efficiency or reducing engine runtime. This chapter focuses on the measures that have been taken so far to accomplish the latter. Many research to reduce this engine runtime is already done. This research led to the development of systems that increased efficiency in terms of communication and taxi procedures. Examples of developments in the field of communication efficiency are the use of Controller-Pilot Data Link Communication during taxiing ((TAXI-CPDLC)) and the Follow the Greens (FtG) concept, which is part of the Advanced Surface Movement Guidance Control System (A-SMGCS) [48, 49]. Though these initiatives all provide support for a more efficient and safe ground movement operation, they are not developments that contribute sufficiently to the environmental goals of the aviation industry. Therefore, these systems are not considered in more detail. This chapter considers only the developments that are relevant for this research and have the potential to contribute significantly to the reduction of environmental impact. Section 4.1 studies the concept of single engine-on taxiing and section 4.2 elaborates on current congestion management techniques.

4.1. SE: Single Engine-on Taxiing

SE taxiing means that only one engine is used for the propulsion of twin engine aircraft during ground movements [41]. Due to the lower number of operative engines, less fuel is consumed. When considering the average taxi-out and taxi-in time at AAS of 9 and 14 minutes, the engine runtime can be reduced with 6 minutes for inbound and 9 minutes for outbound taxiing. This scenario is visualised in Figure 4.1.

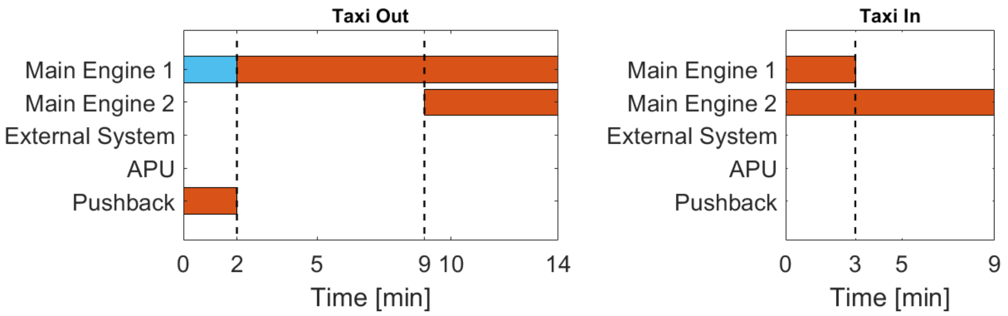


Figure 4.1: Schematic of singe engine-on taxi scenario. Orange bar indicate that the main thrust for taxiing is provided by that system. A blue bar means that the system is active, but does not provide the thrust for taxiing.

Note that the inoperative engine still requires a warm up time of five minutes before takeoff. Also, when an engine has been operated at high power levels for an extended period of time, some cool down time should be allowed before shut down. This cooling is done by operating the engine at low power setting for three minutes on average. During this cool down, the engine is still able to provide thrust for taxiing [32].

SE taxiing does have some important drawbacks mainly related to safety. Obviously, turning of one engine means that extra power is required by the other engine. This result in an extra strong jet blast by the operative engine. Such jet blasts in the ground surface area are dangerous for airport employees. Another drawback of the higher engine setting is the increased risk of Foreign Object Damage (FOD) to operative engine, leading to higher maintenance costs. Finally, the asymmetric trust leads to difficult or impossible turns. This is especially problematic in wet or slippery taxiway conditions [41].

These drawbacks all become more dominant for outbound taxiing because aircraft are heavier as it still contains the fuel needed for cruise. A heavier aircraft required more power for acceleration leading to stronger jet blasts, higher risks of FOD and more asymmetric thrust in turns. Moreover, when the aircraft moves away from a stationary position, the breakaway thrust would require jet blasts that exceed the safety limits. Especially when the aircraft is on a gradient, is heavy or has meteorological conditions against it (ice, wind etc.), this is a procedure that is way to dangerous.

All these drawbacks make SE taxiing for outbound taxiing in many situations nearly impossible and too dangerous. However, for inbound taxiing, it is already quite commonly adopted. A research survey under commercial airline pilots showed that more than 75% of the time pilots endeavour to taxi with only one engine on after landing [50]. Airlines such as Qatar and Iberia have said to fully adopt the SE taxi-in strategy and are aiming to use this procedure 95% of the time when the situation allow for it [51, 52].

It can thus be concluded that the SE taxi technique is not the complete solution for the environmental problem of ground movements. However, it is already often used for inbound taxiing, and it has the potential to be used even more, if the procedures are better established and adopted. This opens the possibility to combine SE taxi technique with other alternative system that performs taxiing for outbound movements. More on this in section 5.6.

4.2. Congestion Management

During periods of high departure demand at busy airports, queues of aircraft often form at the departure runways. This congestion results in inefficient operations which leads to increased fuel burn and emissions, higher operational costs and delays. The average fuel consumption of the taxi-out movements can be reduced by implementing surface congestion management techniques to limit the formation of these queues. Various techniques have been studied and developed to address this issue. All techniques are based on the principle visualised in Figure 4.2: As more aircraft taxi out (in the figure a 15 min time window is taken), the throughput of the departure runway increases because more aircraft are available in the departure queue. However, as soon as this number exceeds a certain threshold (at ±10 aircraft in this example), the departure runway capacity (normally dependant on WTC, SID, Radar separation and meteorological conditions) reaches its limits, and there is no additional increase in throughput, leading to growing runway queues.

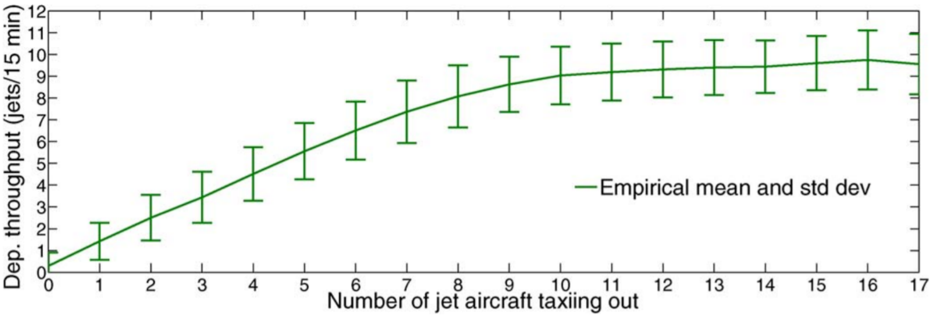


Figure 4.2: Variation of departure throughput with the number of aircraft taxiing out (Values taken from study at for the 22 L, 27 | 22 L, 22R configuration at BOS) [5].

The techniques manage the emergence of queues with a congestion control algorithm. This algorithm regulates the number of aircraft on the ground surface, so that the number of taxiing aircraft stays close to a specific value. To reduce the number of active aircraft on the taxiways, virtual queues are created by holding aircraft at gates or other designated areas during times of high congestion and expected runway queues. Another benefit of such congestion management techniques is that the predictability of the airport system is increased due to less active aircraft on the taxiways. This section outlines four of such congestion management techniques.

CDQM: Collaborative Departure Queue Management

The Collaborative Departure Queue Management (CDQM) concept manages the length of runway queues by regulating the number of active aircraft on the taxiways [53]. The CDQM algorithm predicts the departure throughput of the active runways and subsequently recommend a rate at which aircraft can be released from their spots. This is communicated by allocating time-window slots for aircraft to enter the taxiways through spots (the boundaries between the ramp area and the taxiway). The time window allocation is done by a "ration by schedule" approach, which effectively is a first-scheduled-first-serve approach [54]. The aircraft operator may use this given window to enter taxiways without coordination with other aircraft or ATC. Once the aircraft enters the taxiways, ATC coordinates the taxi route as in conventional procedures. CDQM was implemented and tested in the National Airspace System of the USA and at European airports that use A-CDM [53, 55]. It was found that CDQM resulted in reduced taxi times, fuel usage and emissions, while maintaining full use of departure capacity.

CDS: Collaborative Departure Scheduling

Collaborative Departure Scheduling (CDS), is a variant Congestion Management studied by the FAA in 2011 [53]. It is different from CDQM in the sense that it allocate time slots for takeoff instead of time slots to enter the taxiway system. This means that the formation of runway queue's into account in the pre-departure sequence planning and the rate at which aircraft are released from the gate or spot is not included.

PRC: Pushback Rate Control

The Pushback Rate Control (PRC) algorithm predicts the departure throughput a certain time in advance and provided ATC with an optimal pushback time for each aircraft, such that the queue at the runway is minimised. In contrast to the aforementioned techniques, PRC is a strategy that assigns a pushback window from the gates rather than from the spots [5, 56]. A 32-hour study performed at Boston Logan International Airport has shown that the average taxi-out time was reduced from 20.6 to 15.9 minutes [57].

SARDA: Spot and Runway Departure Advisor

The National Aeronautics and Space Administration (NASA) developed the Spot and Runway Departure Advisor (SARDA), a concept that control the release of aircraft from the gates and the spots by measuring the traffic at the gates, spots and runways [58]. The SARDA concept is therefore the most sophisticated technique of all. It provides an optimised departure sequence and controls the number of aircraft on the taxiways. SARDA consists of two stages. The first stage provides an optimal departure sequence. The second stage determines slots to release aircraft from gates or spots. According to NASA, SARDA is a very effective way to maximise runway throughput while reducing unnecessary fuel burn and emissions.

All techniques regulate the aircraft density on the taxiways system by allocating time slots either before taxiing, after taxiing or both. A overview is created and given in Table 4.1. Both tactics show benefits in terms of fuel reduction. However, they also both impose difficulties. At busy airports like AAS, gates are scarce and holding aircraft at gates might cause conflicts with arriving aircraft [42]. Implementing the formation of runway queue's in the departure planning like the CPDSP can create conflicts with the CTOT time window. Therefore, both tactics might be unsuitable, depending on the busyness of the airport and how much time can be shifted within the CTOT time slot.

Table 4.1: Summary of discussed congestion management techniques.

Strategy → **↓ Regulation**	CDQM	CDS	PRC	SARDA
Before Taxiing	Yes, from spot	No	Yes, from gate	Yes, from gate and spot
After Taxiing	No	Yes, departure time	No	Yes, departure time

Instead of different operations, alternative systems can also be used to reduce environmental impact of airport ground movements. This is discussed in the next section, after which a research gap is presented.

5 Alternative Taxi Systems

in section 3.1, it was concluded that aircraft engines are inefficient in the production of thrust for ground movement. By removing the need for aircraft engines during taxi operations, the environmental impact of airports can therefore be drastically reduced and operational expenses of airlines can be saved. This chapter studies ongoing developments in alternative aircraft taxi systems. Six alternative power source systems are reviewed and compared. The systems are indicated in blue in Figure 5.1 and are divided into two categories: onboard systems (1-3) and external systems (4-6). In case the alternative system is located on board, the aircraft is able to produce its own thrust and can therefore taxi autonomously. Visa-versa, external systems make the aircraft a non-autonomous vehicle. Both types of systems are discussed in section 5.2 and section 5.3 respectively.

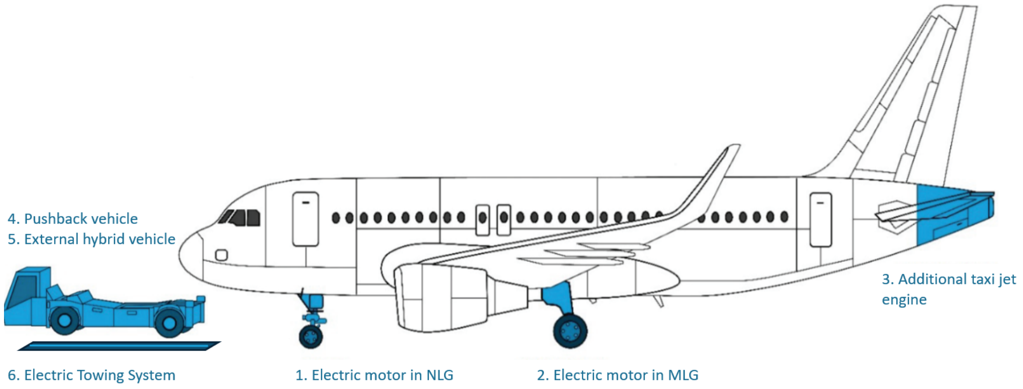

4. Pushback vehicle
5. External hybrid vehicle
3. Additional taxi jet engine
6. Electric Towing System
1. Electric motor in NLG
2. Electric motor in MLG

Figure 5.1: Six alternative aircraft taxi systems. Aircraft retrieved from [6].

Some key considerations for airports, airlines and ground service providers are established in section 5.1. These key considerations are used in section 5.4 to make a comparison between the alternative taxi systems. In section 5.5, the most promising method is discussed and general drawbacks are identified. Using this information, section 5.6 concludes with a general research gap for this MSc thesis.

5.1. Key Aspects

While selecting the most promising system, twelve key considerations in alternative taxi systems are identified as important. These consideration are based on reviewed literature from academics, airports and airlines [6, 59, 60]. The considerations shall be used to make an intercomparison between the systems. As this research is especially focused on a long term perspective plan for sustainable taxiing, economical and procedural aspects of the system (ownership, certification and communication) are of lower value in the trade-off.

Note that the implementation of alternative taxi systems lead to many common consequences, such as the need for engine warm up on taxiways and the reduced risk of FOD to aircraft engines. As these consequences are generally equal for all systems, they are not taken into account in the comparison of systems.

1. Climate change.
As discussed in chapter 3, one of the main reasons to implement alternative taxi systems is to reduce emissions that contribute to climate change. The reduction of these kind of emissions are hence a very important aspect.

2. Local air quality.
Another environmental issue related to aircraft ground movements, especially around big hub airports, are the emissions that lead to reduced air quality in the airport surroundings. Reducing this impact is therefore important in the choice of an alternative taxi system.

3. System noise.
As stated in chapter 3, noise of aircraft ground movements is generally a lesser issue then noise during takeoff and landing. However, it does affect some sensitive parts in and around the airport. It is therefore still taken into account in the trade-off between the systems.

4. Investment costs to purchase the system.
The investment costs of the system has an important role in the trade-off to determine whether a system is feasible. These costs includes the purchase of the system, modifications to aircraft and airport, and possible required trainings for employees.

5. Aircraft modifications.
Modifications necessary to the aircraft affect the aircraft weight and might cause a shift in the average location of the centre of gravity (Center of Gravity (c.g.)). It goes without saying that both affects are undesirable. Note that this criterion does not cover the associated costs of the modification.

6. required staff.
The additional staff needed to operate the system should be minimised in order keep the operational costs low.

7. Safety and visibility.
Safety is of major concern in aviation. The elimination of engine start-up near gates makes most alternative taxi systems safer then current operations. However, there are still safety concerns regarding systems that pilots do not directly control, or are not able too see. Also, braking failure of such systems can play an important role in the safety of a system.

8. Nose Landing Gear (NLG) landing-gear fatigue issue.
Virgin Atlantic and Boeing identified a major concern with regard to additional stress to the aircraft NLG caused by towing vehicles [6]. This additional stress reduces the wheel's fatigue life. This reduction in lifetime significantly increases the operational costs of the total system, as the wheel needs to be replaced more often.

9. Pilot workload.
As pilots need to fulfil a variety of tasks during taxiing, the workload during this flight phase should not be increased too much.

10. System applicability.
The alternative aircraft taxi system needs to be applicable for a wide range of aircraft. The more aircraft are able to use the system, the more effective the system is. The same principle hold for the possibilities to use the system for both taxi-in and taxi-out movements.

11. Acceleration and taxi speed.
In 2013, Airbus stressed that any alternative taxi system should be able to reach ground speeds of at least 20 knots [61]. The reason for this is that much lower taxi speed results in taxiway flow problems leading to increased operational costs and reduced airport capacity. Likewise, Boeing stated the need for aircraft to be able to reach reasonable taxiing speeds in a short time period [62]. These minimum acceleration and decel-

eration requirements are relevant for safety reasons and runway crossings. As a general guideline to prevent flow congestion, controllers state that aircraft should be able to cross runways and clear the safety area within 40 seconds [6].

12. Attachment and detachment time.
External taxi systems require attachment and detachment procedures. The location and the duration of this procedure can have operational consequences for other ground vehicles on the airport surface.

5.2. Autonomous Taxiing

Three types of autonomous taxi systems are considered. Two of them make use of an electric motor in either the NLG or Main Landing Gear (MLG) and are powered by the Auxiliary Power Unit (APU). A third system requires the APU to be replaced by an additional taxi jet engine.

Electric onboard taxi systems show the best potential of reducing local air quality in airport surroundings and show the highest reduction in noise levels [60]. Furthermore, as such electric onboard systems are able to provide forwards and backwards movement, the need for pushback trucks is eliminated. This results in less required staff, lower operational costs and faster turn around times as (de)coupling to pushback trucks is not needed anymore. A strong drawback of such systems is the additional weight added to the structure of the aircraft. This additional weight would lead to more fuel consumption during takeoff, climb, cruise and decent. This subsequently leads to increased operational costs and exhausted emissions. Moreover, depending on the location of the onboard system, the weight of the system can cause a shift in c.g. leading to an increase in drag and thereby increasing the amount of fuel burned. This balance between additional in-flight and reduced on-ground fuel burn is dependent on the range of the aircraft and the average taxi time. For low range flights with high taxi times (typically flights between two big cities in the same continent), such onboard systems are more effective then for long range flights.

The schematic of taxi operations using onboard electrical propulsion is given in Figure 5.2. The pushback truck is eliminated and the power is provided by the APU. As (de)coupling procedures are not required anymore, the taxi-out time is assumed to be reduced with one minute. The main engines still require a warm up and cool down time of five and three minutes respectively.

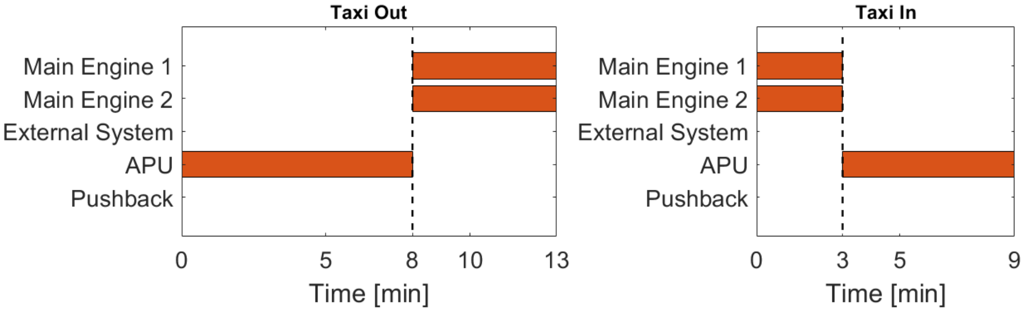


Figure 5.2: Schematic of taxi operations using onboard electric power systems (1 and 2).

5.2.1. Electric Motor in NLG

The most promising development in taxiing by means of an electric motor in the NLG is WheelTug, developed and owned by Borealis Exploration [63]. Its advantage is that is it a lightweight system compared to systems in the MLG. Moreover, as the system is located in the NLG, potential braking problems in the MLG are no issue [64]. Main disadvantage of the system is that acceleration is reduced under certain weather conditions. In the situation of slippery taxiways and a far afterwards c.g., acceleration is not possible at all, making the system dangerous and useless. The maximum speed of the system is also limited to around 10 kts [65]. As the system powers aircraft by the NLG, fatigue issues can result in a reduced lifetime of the wheel. Finally, because the weight of the system is located in front of the plane, the system creates a small continuous shift in the location of the c.g., resulting in a small increase of drag during flight.

5.2.2. Electric Motor in MLG

Two electric motor systems have been developed in the MLG: Electric Green Taxiing System (EGTS) by Honeywell and Safran and eTaxi by L3Technologies [66]. The main disadvantages introduced by the placement of the system in the NLG are eliminated by installing it in the MLG. However, these systems are almost two times as heavy the WheelTug, resulting in more in-flight fuel burn [6]. Finally, as this systems are located in the MLG, they may impose possible brake issues.

5.2.3. Additional Taxi Jet Engine

Some studies have been done to investigate the benefits of installing an additional jet engine in the rear of the aircraft just for taxi purposes [6]. In contrast to conventional jet engines, this taxi engine can be optimised to use for taxi purposes, reducing fuel consumption during this phase. However, compared to other alternative taxi systems, the engine still use conventional fuel and produces jet engine noise. Furthermore, pushback trucks are again required as the jet engine can only provide forwards movement, eliminating the benefits of not using these trucks. As thrust is provided to the rear of the aircraft, strengthening of this section is required leading to additional weight. This total additional weight of this system is estimated to be more or less equal to the weight of WheelTug, which is around half of the weight of EGTS.

5.3. Non-autonomous Taxiing

Three types of non-autonomous taxi systems are considered. Two systems make use of a towing vehicles (4-5), and one system uses a rail track to tow the aircraft (6).

The main advantage of external systems is that no weight is added to the aircraft, resulting in no extra fuel consumption during flight. Therefore, such external systems generally show the best performance in the reduction of emissions. Next to the environmental benefits, external systems will replace conventional pushback trucks, leading to benefits in terms of staff required, operational costs and turn around time. Drawbacks are major airport modifications and/or big changes in operational procedures. These airport modifications and procedural changes can require large investments cost and extra staff. Moreover, external systems can result in additional vehicles on the taxiway system which is not always operational feasible. Next to that, all external systems require some form of attachment and detachment, leading to longer taxi-times and new procedures. Furthermore, because all external systems power the aircraft by the NLG, fatigue issues may arise. Finally, as new situations arises, safety issues needs to be thoroughly considered.

A schematic of the taxi operations with external systems is given in Figure 5.3. The taxi-out and taxi-in duration is increased with one and two minutes respectively, as decoupling (for departures) and attachment (for arrivals) is expected to take longer compared to conventional operations. It should be noted that the additional (de-)coupling time depends heavily on the type of system, which shall be discussed in more detail below. As usual, an engine warm up and cool down time of five and three minutes is required. The warm up (and visa-versa the cool down) can also be performed simultaneously with towing operations.

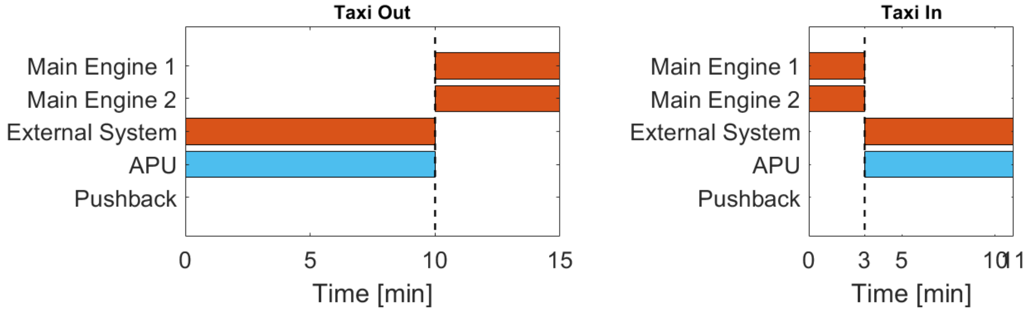


Figure 5.3: Schematic of taxi operations with external systems. Orange bar indicate that the main thrust for taxiing is provided by that system. A blue bar means that the system is active, but does not provide the thrust for taxiing. Note that the engine warm up and cool down can also be performed simultaneously with towing operations.

5.3.1. Pushback trucks

Intuitively, It makes sense to use conventional pushback trucks to perform the towing manoeuvre to the runway, instead of designing and manufacturing new vehicles. Indeed, it is estimated that the total investment costs to use conventional trucks as an alternative mean of taxiing is lower then any other external taxi system [6]. Moreover, as most of the equipment is already present at airports, it is a method that can be implemented quite quickly. A huge disadvantage is the increased fatigue load on the NLG. Pushback trucks are able to move a full aircraft over a short distance, or an empty aircraft over a long distance. A combination of the two (a full aircraft over a long distance) would damage the wheel too much. For AAS, this is enough reason to disregard this alternative taxi option [67]. Next to the decrease in NLG lifetime, this system might encounter problems with taxi- speed and acceleration requirements, depending on the pushback motor size and aircraft weight.

5.3.2. External Hybrid Vehicle

The two strong disadvantages in the usage of conventional pushback trucks in a complete towing system can be overcome by a new appropriate design, suitable for its new function. An example of such design is Taxibot, a concept owned by IAI [33]. IAI claims that Taxibot protects the NLG from exceeding maximum allowed fatigue load at all times. Furthermore, the company claims that Taxibot is able to tow fully loaded aircraft at current aircraft taxiing speed (23 knots).

5.3.3. Electric Aircraft Towing System

A very interesting and innovative concept is the aircraft towing system, where use is made of channels that are installed inside the taxiway. Aircraft are supposed to drive their NLG on a pull cart, that move through the channels, visualised in Figure 5.4[7]. As the system is does not require additional vehicles on the taxiway system, drawbacks in terms of additional staff, higher pilot workload and the need for attachment and detachment is eliminated. The concept owners furthermore claim safer operations, as less personnel and equipment in and around the ground operations is needed leading to fewer accidents and a reduction in human factor errors. Finally, as the system is total electric and no other vehicles are required, the system shows the best benefits in terms of noise and emissions. Major drawback of this system are huge airport modifications and high investment costs.

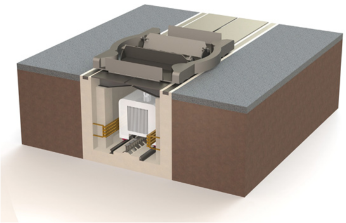


Figure 5.4: Cart in channel of aircraft towing system [7].

5.4. Intercomparison Alternative Taxi Methods

In section 5.1, 12 key considerations were established. Throughout this chapter, these considerations were discussed for the six alternative methods. These 12 aspects where ranked for each methods, and normalised compared to conventional taxi operations. The result is visualised in Figure 5.5. The aspects are located on the outer circle and the score for each method is indicated with a dot on the line towards the centre. The more outwards the dot is located, the better the score in that specific aspect (for example, a high score in the noise category leads to more noise reduction). Please refer to Appendix C for more information regarding this score system. The six methods are all given a colour and the conventional taxiing is given in black.

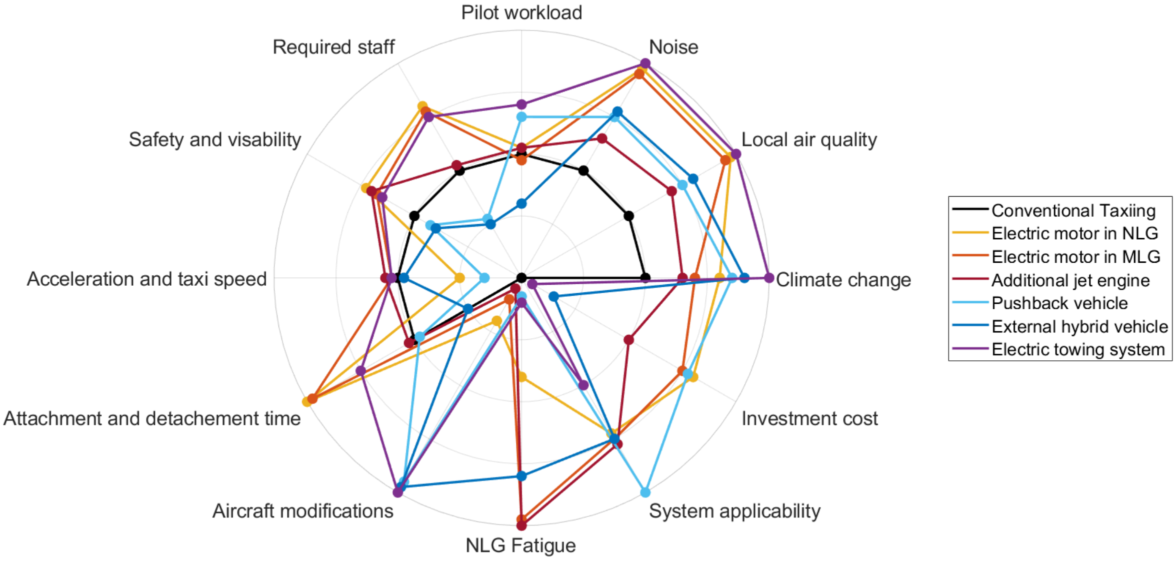


Figure 5.5: Intercomparison of six alternative taxi methods.

Given the intercomparison in Figure 5.5, a proper trade-off between all systems can be made, such that the most promising system can be selected for this research. In the trade-off, it is key to weight the advantages of a system against the disadvantages. However, even more relevant is to consider how this research can contribute towards the mitigation of disadvantages of promising systems.

The most best method in terms of environmental impact reduction is an electric towing system. It scores maximum on both noise, local air pollution and greenhouse gas reduction. Also, concept owners of such systems claim to need less staff, increase safety and decrease pilot workload while meeting speed and acceleration requirements. However, the investment costs and airport modifications are extremely huge. Implementing the system would require huge investments, risks and impact on ongoing airport traffic for years. Furthermore, the technological development stage of this system is far behind other systems. It is therefore concluded that this system is not considered further. The next available options are to use onboard systems for propulsion. The major drawback of these systems are the necessary aircraft modifications. Therefore, the estimation is made that such system will not be beneficial on many aircraft and do therefore not have the potential to achieve world-wide sustainable taxiing. The remaining methods are the use of external towing vehicles; either with existing pushback tugs or new developed vehicles. The new developed vehicles, like Taxibot, require some more investment, but solves issues related to acceleration, speed and fatigue. Is is assumed that these drawbacks are critical to solve. Therefore, the external hybrid vehicles as alternative taxi method is chosen for this research. However, this methods still has many challenges to overcome. An example is the increased operational complexity of ground movements due to the additional vehicles and changed uncoupling locations. These are challenges that, if not addressed properly, might prevent airports such as AAS from having these systems in future. This MSc thesis focuses on challenges related to the operations and coordination of ground movements at airports that use external towing vehicles. section 5.5 studies the only certified external towing vehicle on the current market, called Taxibot. A research gap related to the external towing vehicles is presented in section 5.6.

5.5. Taxibot

The only certified external hybrid towing vehicle on the current market is Taxibot [68]. Taxibot, shown in Figure 5.6 is a concept owned by IAI and is in operation since 2014 [33]. The concept is developed in collaboration with Teleflex Lionel Dupont (TLD), Smart Airport Systems (SAS), SIEMENS and Airbus, and is supported by Boeing. Taxibot claims 85% reduction in fuel and air pollution and many additional advantages in terms of reduced FOD, increased airport efficiency, high aircraft compatibility, NLG fatigue protection and no modifications to the aircraft.

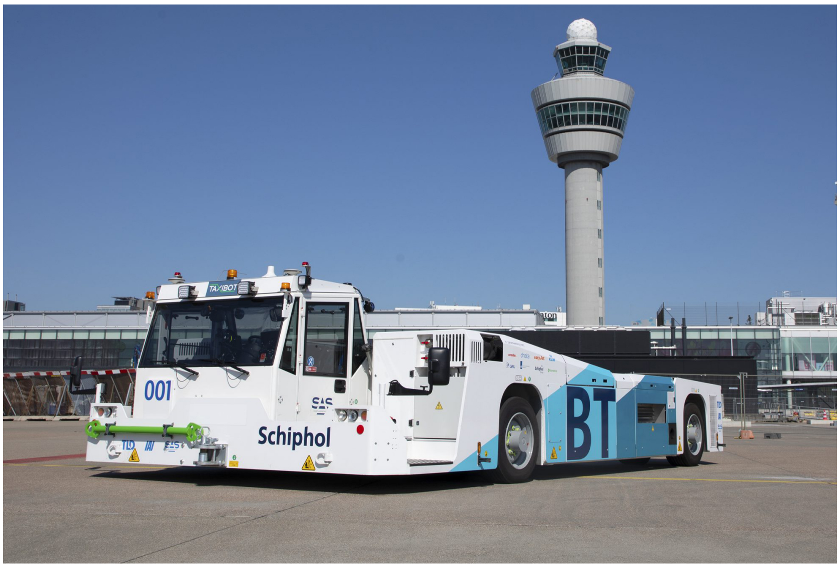


Figure 5.6: Taxibot at AAS during test in May 2020 [8].

Taxibot operations are claimed to be rather straightforward. For outbound movements, pushbacks are performed by Taxibots just like conventional pushback trucks. After pushback, Taxibots are switched into pilot mode to enable pilots to control the Taxibot movements while taxiing. At the last phase of the pull, and during uncoupling, engines are warmed up. Uncoupling is performed preferably as close as possible to the runway. After uncoupling, aircraft continue to taxi towards the runway using own power, while Taxibots return to the gate area. For inbound movements, Taxibots await aircraft near runway exits to attach and pull aircraft to the gate after landing. Aircraft engines are allowed to cool down while pulling and are turned off when finished. Just like outbound movements, pilots are in control during taxiing. At the gates, Taxibots uncouple and continue to another aircraft.

Taxibots have been tested at airports like Amsterdam (IATA: AMS, ICAO: EHAM), New Delhi (IATA: DEL, ICAO: VIDP) and Frankfurt (IATA: FRA, ICAO: EDDF) [69]. IAI claimed to have received good feedback from pilots, ATC and ground handlers. This claim is verified by experts from AAS, that conducted Taxibot test at the airport in May 2020 [42]. The observer environmental benefits based on a set of 7 departure trail is around 50% with respect to conventional taxiing. Other emissions reduced accordingly. The reasons for not being able to achieve the 85% fuel saving promised by the manufacturers is due to the imperfect execution of trail and small sample sizes. Simulation studies must be used to find the excepted fuel savings at AAS if the Taxibot system is implemented at full scale.

The Taxibot concept also comes with additional challenges. One of the major challenges can easily be identified by looking at the main operational difference: changed uncoupling points and moments. Where conventional pushback trucks where uncoupled after pushback, Taxibot is uncoupled near the runway entrance. The difference in uncoupling location creates a changed stopping point, affecting traffic flow on taxiways, especially during peak hours. On the other hand, uncoupling of conventional pushback trucks on aprons is prevented. The shift in uncoupling operations therefore makes the flow on aprons run smoother while increases complexity near runways. Next to the change in uncoupling location, the process is expected to take longer as well. Results from the Taxibot tests at AAS show an uncoupling duration of around 150 seconds, while uncoupling of conventional pushback trucks takes around 90 minutes, on average [42]. Two further interviews regarding this issue have been performed with SAS and TLD. TLD recognises the issue, but stresses that it is likely to only result in problems during peak departure periods when runway throughput is of main concern. SAS provided a break down analysis with the uncoupling procedures and their duration, given in Table 5.1. The aim is to have the uncoupling process done in around 90 seconds when operations run smoothly. However, SAS stress that especially at the beginning, uncoupling operations are more likely to be around 120 seconds, still 30 seconds shorter than test results from AAS. Whether the final uncoupling duration is 90, 120 or 150 seconds, a changed stopping point is a fact, and congestion problems near runway entrances are bound to occur during peak departures if tow operations are done on conventional infrastructure. These delays have huge affects on the operational efficiency at airport, and can result in increased cost, safety issues and reduced airport capacity. Furthermore, airlines are concerned that that ticket sales, network operations and aircraft & crew utilisation are negatively affected by an overall increase in taxi time. A classic approach to solve delay and capacity issues is to increase ground capacity by expanding the ground surface infrastructure used for the ground movements of aircraft and other vehicles. However, many airports are approaching limits in terms of airport size due to restrictions by the surrounding areas. Additionally, increasing capacity through expansion increases environmental impact, human workload and the overall complexity of operations. This is also the case for AAS [31].

Table 5.1: Uncoupling procedures and their respective duration. Based on operational tests in DEL.

Procedures	**Time (WB & NB) [sec]**
Switch pilot mode to driver mode	5-10
Actual unloading (automatic process)	60
Intercom cable disconnection	10-20
Drive away and thumb-up	15-30
Total	**90-120**

The changed uncoupling location requires towing vehicles to travel longer distances to the next mission. Instead of returning to gates from the bay area, Taxibots need to return from runway entrances. This results in more active vehicles on the ground movement area, leading to increased workload for ATC. Moreover, not all roads are suitable for Taxibots. At AAS, the Taxibots are physically too wide for the tight service roads. These roads are around five meters wide, whereas Taxibots are around four and a half meters wide [70]. This means that Taxibots cannot drive safely on these roads and no other traffic is able to pass the Taxibots. A second problem of these service roads are limited entrances to and from taxiways. Hence, Taxibots are forced to drive on certain taxiway pieces or even cross (possibly active) runways. Finally, note that conventional inbound movements do not require pushback vehicles at all, meaning that the complexity of operations is increased relatively more. Therefore, the effects of the above mentioned challenges are only worse for inbound movements.

Though a single Taxibot mission does not look very problematic, scaling up the concept to a complete airport imposes additional challenges. The underlying problem of these challenges is that airport infrastructure is not build and designed for large scale towing operations. These large scale operations would mean additional and different type of vehicles on taxiways and uncoupling near runway entrances instead of aprons, as explained previously. It is easy to imagine that these additional vehicles and alternative uncoupling locations can result in bottlenecks and problems on taxiways and service roads. As most airports are approaching limits in terms of size and capacity, the alternative taxi operations are hard or maybe impossible to manage by conventional ATC operators. A novel way of air traffic coordination and control might offer the necessary relief.

5.6. Conclusions & Research Gap

Over the past decades, autonomous vehicles have entered many industries. Amazon uses autonomous robots to pick-up and deliver packages in warehouses, the port of Rotterdam uses autonomous trucks to pick up and deliver container from cranes to stands and the first autonomous cars are expected to enter our streets soon [71][72]. Meanwhile, the rigid aviation industry is lagging behind. In the 'Autonomous Vehicles and Systems at Airports Report'[34], ACI state that automation in complex systems like airports provides key opportunities that have to be embraced. It could provide the needed relief to allow autonomous alternative taxi operations and maintain airport capacity. Automated air traffic ground control does not mean ATC operators are replaced. In fact, their role changes to a more proactive supervising task, that is more focused on solving problems rather than repetitive tasks.

Autonomous vehicles have proven to increase efficiency, safety and security in many complex operational systems. It can be expected that this is also true for airport towing systems like Taxibot, but notable evidence for this claim is lacking. This research develops a simulation to study the performance of autonomous airport ground movements, power by towing vehicles. The scalability of a towing system is analysed and tested for capacity, ground delays and bottlenecks. In this research, AAS is used as sample scenario. As for the towing system, note that Taxibot is currently the only certified towing vehicle. Therefore, this name is used in this research as a generic name, rather than a brand name. This is mainly done to avoid the longer description 'towing vehicle'. However, it is important to note that, when talking about 'Taxibot' the generic name is meant, and can thus also refer to other (still to be developed) systems. The parameters characterising Taxibot vehicles and the layout of AAS are developed to be easily adjustable such that the model is convertible and applicable to different airports or other towing system brands. The outcome of the simulation is sensitive to the assumptions made. The reason for the four most dominant assumptions are explained below. Other assumptions can be found in subsection 9.2.1.

Only outbound movements are towed
Three reasons underlie this decision. First of all, as stated in chapter 3, the environmental impact of inbound ground movements is considerably lower then outbound movements. Secondly, as stated in section 4.1, SE taxiing is already practised often for inbound taxiing, and has the potential to be applied more often in future. Finally, as explained in section 5.5, operational complexity is increased relatively more when towing inbound aircraft movements.

Uncoupling location and duration
One of the main operational differences identified in section 5.5 is the shift in uncoupling location and duration. Currently, aircraft are uncoupled on aprons after pushback, a process that takes on average around 60 seconds [42]. In the situation of towing vehicles, aircraft must uncouple as close as possible to the runway. The expected average uncoupling duration of Taxibot is still questionable, but all involved parties estimate this to be at least longer than 60 seconds. The uncoupling locations are discretized into three or four location per runway. The uncoupling duration is adjusted by means of a sensitivity study.

Suitability of service roads for Taxibots
As discussed in section 5.5, service roads are currently not suitable for a large scale use of Taxibots. At most locations, the roads are physically too narrow for the wide Taxibots. Furthermore, only limited entrances to the service roads from the taxiways exist. The roads around concourses (e.g. the roads between the terminal and aircraft) are tested to be suitable for Taxibot [42]. Therefore, the assumption is made that all Taxibot can only drive on taxiways, aprons, and the service roads around concourses. At the top of each concourse, an entrance to these service roads exist, such that Taxibots can exit the taxiways at multiple locations. A more detailed description of all locations is given in subsection 9.2.1.

Types of vehicles
To make the simulation as realistic as possible, different types of vehicles are taken into account. Taxibot have developed two types of Taxibot: Narrow Body (NB) and Wide Bode (WB) Taxibots. NB Taxibots are able to tow NB aircraft, and likewise, WB Taxibots tow WB aircraft. This result in six different types of vehicles in the system. Each type of vehicle contains its own kinematic properties in terms of taxi speed, turn speed, acceleration, deceleration and maximum deceleration. Furthermore, they have their own fixed separation requirements.

Simulation schedule
To properly test the scalabilitiy of Taxibot, and the performance of the autonomous system in terms of capacity and delays, the most busiest days are simulated. Furthermore, the most common RMOs are considered, as discussed in section 2.4. More details related to the flight, gate and runway schedule can be found in subsection 9.2.1.

This MSc thesis is about delay management during peak departure hours in the novel concept of an autonomous engine-off outbound towing system. An agent-based model is developed to simulate the autonomous system of aircraft and Taxibot. In this model, cooperative coordination mechanisms are used to plan and coordinate uncoupling procedures and traffic routes in such a way that the resulting ground delays are minimised and airport capacity is preserved. A well developed model might even result in a positive effect on ground delays during certain periods of the day, as the advantage of not having conventional pushback uncoupling at aprons may outweigh the disadvantage of Taxibot uncoupling. The rest of this literature study is devoted to the methods used to study the objective of this research. In chapter 6 and chapter 7, the given agent-based method and cooperative coordination techniques are studied. Subsequently, a research proposal is specified in chapter 8 and the research methodologies, including scope, assumptions and expected results are given in chapter 9.

6
Agent-Based Airport Ground Surface Modelling

Previous chapters studied the conventional airport ground movements and identified operational methods and alternative system to make airport ground movements more sustainable. From now on, this literature study focuses on selecting and developing appropriate modelling methods to use in this research.

Chapter 5 concluded that engine-off taxiing by means of towing vehicles is a very promising technological development to reduce ground movement emissions. The introduction of this alternative taxi system changes operations on taxiways. The main operational differences are the different types and increased number of vehicles plus the need for uncoupling near runway entrances in stead of aprons. It is expected that these changed operations increases ground delays, and thereby reduces airport capacity. Replacing the conventional centralised ATC coordination by an autonomous system might be the solution to preserve airport capacity after a towing system is introduced. This research studies how such autonomous system can prevent capacity losses by the development of a model that simulates the alternative taxi operations. This chapter studies Agent-Based Modelling (ABM), the simulation method used in this research.

Many research in the modelling of airport ground networks and the feasibility of external towing systems in an airport ground network is already done. This research usually excluded an explicit study in the affects of towing systems on airport performance in terms of capacity and delay. The limited existing research in this area makes it a very challenging though relevant research. First, previous research in airport ground surface movement modelling is discussed in section 6.1. Subsequently, advantages and disadvantages of Agent-Based Model (ABM) in airport ground movement research is studied and conclusions are drawn regarding its suitability in section 6.2. Finally, a long-term research line at the Delft University of Technology is discussed and a baseline model is selected in section 6.3.

6.1. Research in Agent-Based Airport Surface Modelling

In "An Introduction to Multi-Agent Systems", Michael Wooldridge describes Multi-Agent Systems (MAS) as systems that consist of a number of agents which interact with one-another by cooperation, coordination and negotiation, based on different goals and motivations to satisfy a designed objective [73]. This approach allows to effectively specify a complex system with components at different levels and apply multiple scales of analysis to capture explicit interactions and unforeseen behaviour. The method is of often used in airport operation research as complex systems, uncertain environments and dynamic situations can be easily represented by agent-based models [74].

Agent-based modelling is thus a technique that makes it feasible to include details and constraints on a smaller level by pre-defining properties of the environment and the interacting agents. The interaction between agents can be done is such a way that the total system is coordinated according to some desired method. Typical methods are decentralized, centralized, (decentralized) distributed and hierarchical control [74]. This wide range of available coordination methods is what makes agent-based models interesting. Different coordination methods may lead to different kind of (unforseen) behaviour in the simulation. To find the best coordination methods, the behaviour of the simulation given agent characteristics and environmental properties under different coordination techniques is very interesting in airport taxi research. Visa-versa, to study the effect of small changes in the environmental properties of agent characteristics, the behaviour of the simulation given some kind of coordination method can be studied by changing some of the pre-defined properties. Agent-based modelling therefore opens a very wide range of possibilities in the research to more efficient airport ground operations.

Agent-based models are often used in airport operational research for different objectives. H. Udluft and T. Noortman [9, 10] used ABM to study the performance of decentralised control in airport ground movements. K. Fines [11] used different cooperative coordination mechanism to study the resilience of decentralised airport ground movements to runway reconfigurations. J. Siebers [75] developed an ABM to study the robustness to operational uncertainties in airport ground movements. S. Polydorou [76] implemented and studied the performance of a learning based cooperative multi-agent planning mechanism in airport ground networks. All of these studies are performed to increase efficiency of airport ground movement operations to obtain higher capacity, reduced ATC and pilot workload, less operational costs and lower environmental impact. There have also been some studies conducted that only considered capacity as a main performance indicator [77, 78]. Other previously studied objectives in airport operation research using ABM is aircraft deicing [79], increasing operations predictability [80] or a combination of objectives in safety, economy and sustainability [81]. Finally, B. Benda [12] used ABM to design and evaluate a novel taxi-concept for outbound traffic enabled by autonomous Taxibots.

ABM developed in previous research differ heavily in complexity and considered operations. Models range from simple hypothetical airport ground surfaces [9, 78] to actual existing airport surfaces [10–12, 75, 76] or even models that includes the Terminal Control Area (TMA) [81], the airspace surrounding major airports.

All the considered studies have modelled the behaviour and interaction of agents differently. Most studies simplify the airport ground system to a graph (or network) consisting of vertices (or nodes) and weighted edges (or links) [10–12, 76]. Such graphs can easily and accurately represent a complex airport system and are more often used to study simular objectives in network routing, route planning, traffic control, computer games and transportation systems [82]. In airport systems, the vertices usually represent intersections, gates and runway entries. Edges can represent taxiways, service roads, runway pieces, aprons or ramps. The representation of the airport ground system in such a network transforms the ground operations into a logistic system, in which coordination and planning can be done by many different techniques. This opens a wide range of possibilities to study different coordination techniques in airport operations like the use of search algorithms such as shortest path methods or multi-agent path finding techniques. This shall be more extensively discussed in section 7.1 and section 7.2. J. Siebers [75] uses Markov Decision Process to model agents behaviour in taxi systems. J. Borst implemented the use of linear programming for planning procedures [80]. Most studies use historic data as input or for assessment of the model performance.

6.2. Review of ABM for Airport Ground Surface Movements

A. Bazghandi [83] investigated all benefits and difficulties for ABM in traffic simulations. This section compares these results with the findings of previous research studied in section 6.1 and subsequently concludes upon the suitable of ABM in airport ground movement research.

The driving benefit of agent-based models is the ability to capture emergent behaviour. Emergent behaviour results from interaction of individual agents and describe, by definition, the behaviour of the whole system. A single ABM can result in many different emergent phenomena. The emergence of a specific phenomena is usually set as main objective of the study. All studies in airport ground movements therefore use this property to learn about the behaviour of the airport ground system in term of capacity, safety, resilience, operation costs, environmental impact, etc.. This behaviour can be studied and analysed by varying behavioural entities. ABM allows for a natural and effective description of these behavioural entities, which can be very complex in airport systems. Examples are different aircraft characteristics or ATC procedures. It is generally easy to add additional agents or tune their complexity in terms of rationality, abilities and interactions. This flexibility is a third major advantage of agent-based models. In previous airport ground movement research, this flexibility is usually used to study different kind of cooperative coordination techniques, interactions or characteristics. As Agent-based models represent a complex system by a combination of different entities, procedures, coordination methods or networks can be easily adjusted and reused by other researchers for different objectives. Finally, contrary to methods like Linear Programming (LP), the efficiency and computation speed of agent-based models is not significantly effected by scale or complexity.

The main challenge that needs to be addressed is the handling of complex factors and the level of detail required to fulfil the objective of the model. In airport ground movements, complex factors are unpredictable human behaviour in terms of driving and decision making and uncertain environments like changes in methodological conditions or accidents. The level of detail required for the model to serve its purpose must be throughout considered and well-founded. Depending on the methods and techniques used, the system does not always converge to optimal solutions. Finally, as agent-based models might consist of many different aspects, their development time might be very time-consuming.

The most important advantages and challenges for the use of ABM in airport taxiing operations are given in Table 6.1.

Table 6.1: Advantages and challenges of ABM in airport ground movement research.

Advantage	**Challenges**
1) Capture emergent phenomena	1) Many complex factors
2) Natural description of complex systems	2) Level of detail
3) High flexibility	3) Possible suboptimality
4) Model reusability and efficiency	4) time-consuming

As stated in section 5.6, the general objective of this study is to find methods to minimise aircraft ground delays in engine-off aircraft taxi systems. Cooperative coordination between ground vehicles, aircraft and ATC can offer the solution. To study these coordination methods, the development of a complex system with many parties and different interactions is required. The performance of the system must be analysed as a whole for different coordination methods. As concluded from the above review, agent-based models are an extremely efficient and powerful method to flexibly develop a complex system and analyse it for different emergent properties. The complexity of the airport system and the level of detail required must be well-founded and established in consultancy with experts closely involved with conventional ground movements of the considered airport. To preserve optimality as much as possible, different cooperative coordination techniques used in multi-agent systems are studied in chapter 7 and their applicability to airport networks is discussed. Finally, to mitigate the time-consuming development time, a baseline model is selected in section 6.3.

6.3. Baseline selection

The development of agent-based models can be very time-consuming. Especially large and complex models including with many elements require a long development time. According to section 6.2, this disadvantage can be addressed by using a previously developed model as baseline. Given the number of studies performed in the field of airport operational research and the fact that agent-based models are generally easy to reuse, it can be very time-saving to select a current existing model as a baseline instead of developing one from scratch. This section selects and studies the baseline model for this research. Before one is selected, some requirements are defined to make this research successful:

- The model must be accessible.
- The model must be clearly formulated and coded in understandable language.
- The model must be of an airport that is currently considering the use of external towing vehicles and foresees congestion problems due to decoupling procedures at taxiways.
- The model must be of an airport that provides easy accessible and clear data and information.

A long-term research line on agent-based modelling of taxiing operations at To70 and the faculty of Air Transport & Operation at the Delft University of Technology has result in a model that fulfils all the above mentioned requirements. The research line started in 2016 and have been continuously developed by MSc graduate student throughout the years. Udluft [9] completed the first model in 2017 of a fictional airport ground system, where departing aircraft agents are coordinated using decentralised control, shown in Figure 6.1. Noortman [10] extended the model in 2018 to a simplified version of the ground system of AAS and added arrival aircraft as well, presented in Figure 6.2. Fines [11] added a Conflict Based Search (CBS) algorithm to increase the resilience of the model in case of runway reconfigurations. He also tested algorithms that created highways in the taxiways to stimulate this desired resilience. His work is presented in Figure 6.3. Benda [12] extended the taxiway system of AAS with service roads and introduced a new kind of agents: Taxibot agents. An autonomous vehicle that tow departing aircraft agents from the gate to the runway. His model is given in Figure 6.4. simultaneously with Benda, Polydorou [76] analysed the performance of airport surface movement operations with a learning-based cooperative multi-agent planning mechanisms.

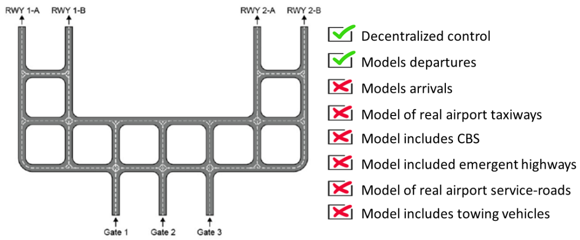

RWY 1-A
RWY 1-B
RWY 2-A
RWY 2-B
Gate 1
Gate 2
Gate 3
☑ Decentralized control
☑ Models departures
☒ Models arrivals
☒ Model of real airport taxiways
☒ Model includes CBS
☒ Model included emergent highways
☒ Model of real airport service-roads
☒ Model includes towing vehicles

Figure 6.1: H. Udluft (2016-2017)[9].

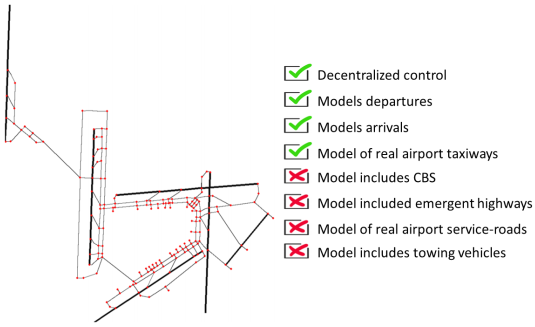

☑ Decentralized control
☑ Models departures
☑ Models arrivals
☑ Model of real airport taxiways
☒ Model includes CBS
☒ Model included emergent highways
☒ Model of real airport service-roads
☒ Model includes towing vehicles

Figure 6.2: T. Noortman (2017-2018) [10].

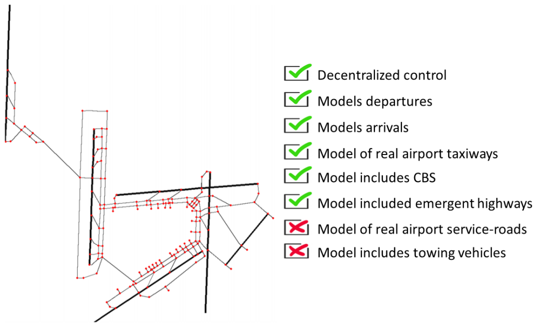

☑ Decentralized control
☑ Models departures
☑ Models arrivals
☑ Model of real airport taxiways
☑ Model includes CBS
☑ Model included emergent highways
☒ Model of real airport service-roads
☒ Model includes towing vehicles

Figure 6.3: K. Fines (2018-2019) [11].

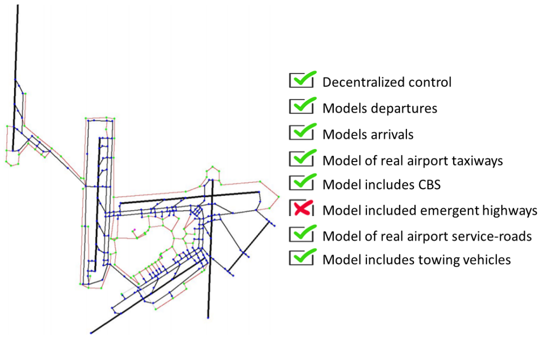

☑ Decentralized control
☑ Models departures
☑ Models arrivals
☑ Model of real airport taxiways
☑ Model includes CBS
☒ Model included emergent highways
☑ Model of real airport service-roads
☑ Model includes towing vehicles

Figure 6.4: B. Benda (2019-2020) [12].

Figure 6.5: Comparison of research projects at Delft University of Technology.

The development of the model throughout the years have let to the most recent model containing many elements created by different developers. Though some elements have been removed by other developers, the most recent model of Benda still contains many important and relevant elements for this research. This model is developed in Python, a high-level object oriented general-purpose programming language [84]. Given Python's strong capability, supportive tools, and the available background knowledge, this language is chosen to be acceptable for this research. As such, the model developed by Benda is used as a baseline in this study. The next chapter studies AI-planning techniques to use and apply in this baseline model to fulfil the objective of this research.

7
Coordination in Multi-Agent Systems

In the research gap presented in the final section of Chapter 5, it was concluded that autonomous ground movements are key in mitigating ground delays. The previous chapter subsequently reviewed the use of agent-based models to simulate these autonomous movements. It was concluded that agent-based modelling is a suitable technique for simulating the movements, and a baseline model was selected. This chapter elaborates on all mechanism usable to coordinate the autonomous movements, such that ground delays are mitigated as much as possible.

It was previously concluded that airport movement areas are generally simplified into a graph of vertices and edges. To make meaningful comparisons between different research, the most classical approach (also used in the baseline scenario) is considered. This approach simplifies the airport system as an undirected graph $G = (V, E)$ in which V represent a set of n vertices and E a set of m edges. Single vertices are indicated with $s \in V$ and edges $(s, s') \in E$. Each edge $(s, s') \in E$ has an associated weight w (or cost) which denotes travel time. A set of K agents are considered. Each agent $a_i \in K$ must travel via edges connected by vertices from a source vertex $s_0 \in V$ to a goal vertex $g \in V$. A valid single agent execution plan consists of a set of action a (wait or move) and is denoted π. If multiple agents are considered, the execution plan of all agents in K is denoted with Π.

The autonomous system must coordinate and plan uncoupling points and moments together with optimal paths for Taxibots and aircraft such that airport capacity is preserved and ground delays are minimised. In agent-based research, the objective of finding aircraft and Taxibot paths combined with the uncoupling task can be seen as a Multi-Agent Path Finding (MAPF) and Task Allocation (TA) problem, in which the goal is to find an optimum between paths and allocated tasks in the graph described above. These tasks represent the uncoupling procedure (in terms of location and time) whereas the paths are the routes from gate to runway and visa-versa. To study these problems in multi-agent systems, the basis of single agent path finding algorithms (or shortest path methods) needs to be understood first. Therefore, section 7.1 studies the most common shortest path algorithms, and elaborates upon the A* variant. As multiple agents are considered at the same time, conflicts needs to be avoided. The act of finding paths for multiple aircraft simultaneously is referred to as MAPF. This is discussed in section 7.2. The search for optimal uncoupling points and moments is generally known as a TA problem. Previous research in agent-based task allocation problems are studied in section 7.3. Naturally, all vehicles in the airport system have kinematic properties. Taking these constraints into account is important for the performance of the simulation, especially in a towing system where multiple types of vehicles are present on the same roads. This is discussed in section 7.4. In section 7.5, all different mechanisms are compared based on seven distinct criteria and the most applicable and promising MAPF and TA method is selected.

7.1. Shortest Path Problems

Naturally, in order to save time and costs, the shortest path must be found for all agents in the airport network. The weights w assigned to edges allow for the application of algorithms that are able to find a path between source s_0 ang goal g with the lowest sum of weights. The minimum sum of weights is often denoted as Sum of Individual Costs (SIC) [85]. Shortest path algorithm is usually used as a generic name for all such algorithms as in most cases it is the distance that is minimised. However, it is important to note that, when talking about 'shortest paths' the generic name is meant, and thus refers to the lowest sum of weights, rather than solely distance. Given the large body of literature in shortest path problems, Madkour and Ur Rehman [13] developed a breakdown of all existing variants. This breakdown is discussed and studied in subsection 7.1.1. The most commonly applied method, A* is subsequently studied in greater depth in subsection 7.1.2.

7.1.1. Problem Variants

Various settings of shortest path problems exist. Graphs can either be static or dynamic, contain directed or undirected edges or contain negative or only non-negative edge weights. Algorithms can give exact or approximate solutions. A breakdown, developed by Madkour and Ur Rehman [13], of all algorithms is presented in the taxonomy in Figure 7.1.

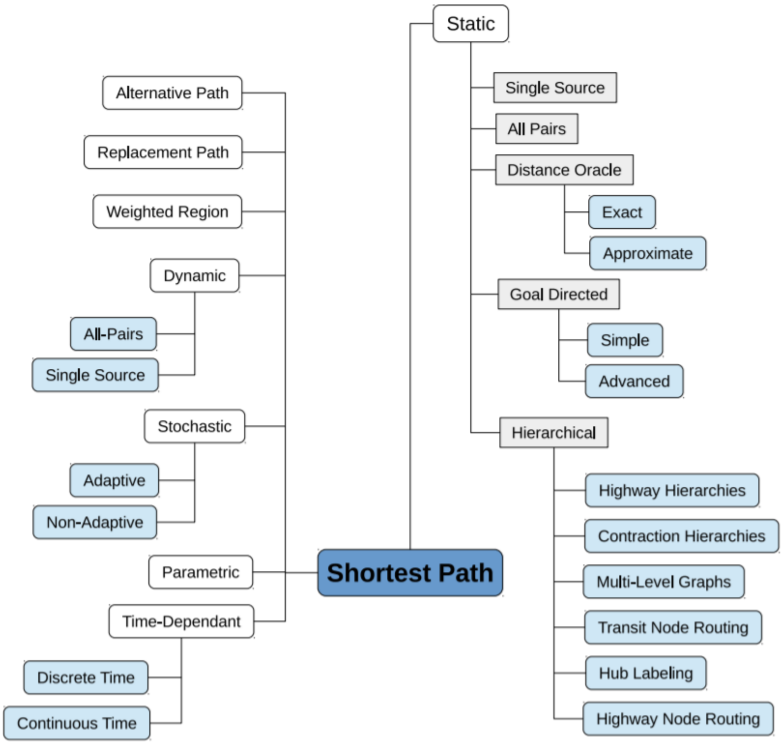


Figure 7.1: Taxonomy of shortest path algorithms [13].

Static Algorithms

In static graphs, vertices s, edges (s, s'), and the edges weight w do not change over time, i.e. they are fixed. In these graphs, a diversion can be made between Singe-Source Shortest Path (SSSP) and All-Pairs Shortest Path (APSP) algorithms. The SSSP algorithms computes the shortest path from a given vertex to all other vertices. Dijkstra's algorithm [86] is the most commonly used SSSP algorithm. The algorithm is used for directed graphs with non-negative weights. Dijkstra's algorithm picks unvisited vertex with the lowest distance to the start vertex, calculates the distance through it to each unvisited neighbor, and updates the neighbour's distance if smaller. Bellman [87] introduced the Bellman-Ford algorithm, a SSSP variant that is able to handle negative weights, unlike Dijkstra's algorithm. As this algorithm expends more nodes, it has a longer runtime then Dijkstra's algorithm. Furthermore, negative cycles should be addressed to avoid the solution going to minus infinity. Dijkstra's algorithm can also be used for APSP problems by computing the APSP for each vertex in the graph. Because the weights are fixed, some static algorithms perform precomputations over the graph to become a faster method then SSSP and APSP. Such algorithms make a trade-off between the query time compared to the precomputation and storage requirements. Thorup and Zwich [88] developed the distance oracle algorithm in which precomputed data structures are used to improve algorithm runtime. Goal-Directed Shortest Path (GDSP) algorithms are those that use additional information to allow the algorithm to determine which part of the search graph should be prioritised in the search. The most commonly known GDSP is A*, proposed by Hart et al. [89]. A* is essentially the same as Dijkstra's, except that there is an heuristic involved. The A* method is more elaborately discussed in subsection 7.1.2. Finally, hierarchical shortest path algorithms generate a hierarchical structure of vertices and edges in the preprocessing stage to reduce computation time in the shortest path search. An example is the generation of highways in dense areas. The idea is that important edges or vertices (those that overlap in various shortest paths) generate an overlay graph, resulting in faster algorithm runtime for future searches. Another interesting example is the use of hub labelling, introduced by Gavoille et al. [90]. In hub labelling, the algorithm labels vertices with certain distances to other vertices or goals vertices in the prepossessing stage. These labels are later used in the search stage to quickly determine the shortest path from a given source to a destination along one of the labelled vertices.

Alternative Path Algorithms
Alternative path algorithms compute the shortest path between vertices that avoids certain vertices or edges. The algorithms first determines the shortest path without considering the vertices or edges to avoid and subsequently finds alternative paths based on nodes along the path. In other words, the method computes the shortest path and reuses the result to find alternative paths for each 3 nodes along the path. This technique is introduced by Ko-Hsing and Wang [91] for multi-agent path planning, and is further discussed in subsection 7.2.8.

Replacement Path Algorithms
The replacement shortest path algorithms computes, just like the alternative shortest path algorithm, the shortest path between two vertices while avoiding certain vertices or edges. The difference is that for the replacement paths, the edge or vertices to avoid should be specified beforehand. Thus, where the alternative algorithm reuses the previously computed path with the unwanted edge or vertices, the replacement path instantaneously computes the shortest path for given edges to avoid. This technique is often used in conflict-based search algorithms. This will be more extensively discussed in subsection 7.2.13.

Weighted Region Algorithm
The weighted region shortest path algorithm constructs a sparse graph (named the pathnet) from the main graph. The pathnet links selected pairs of vertices together by locally optimal paths. Mata and Mitchell [92] proposed this technique to reduce query time at the cost of suboptimal solutions. A parameter controls the sparsity of the pathnet, and thereby the error with respect to the optimal solution and the reduction in query time.

Dynamics Graph Algorithms
In contrast to static graphs, dynamic graphs contain vertices and edges which are introduced, updated or deleted over time. Dynamic algorithms include both APSP and SSSP algorithms. Madkour and Ur Rehman [13] distinguish 3 types of dynamic algorithms. Fully dynamic algorithms are able to process both insertion and deletion. Incremental algorithms can only process insert operation, and no deletion. Decremental algorithms can, on the other hand, only process deletion, and no insertion. The latter two algorithms are therefore partially dynamic.

Stochastic Graph Algorithms
In a stochastic shortest path problem, the uncertainty associated with edges is modelled as a random variable. In other words, the weight of edges is dependent om some stochastic distribution: i.e. $w(v, v') = exp(\lambda)$. The objective is then to compute the shortest path based by minimising the expected costs. Usually, to model such decision making process, and choosing proper actions to meet pre-defined objectives, use can be made of Markov Decision Processes. In airport operation research, this method is usually used to correctly predict departure throughput and subsequently recommend a rate at which to release aircraft from their gates in order to control congestion [93–95]. A difference between adaptive and non adaptive algorithms is often made. The adaptive algorithms determines the best hop based on the graph at a certain time instance. Non-adaptive algorithms focuses on minimising the total expected cost of the graph.

Parametric Algorithms
Parametric shortest path algorithms compute the APSP based on a specific parameter. The weight of the edges depend linearly on this parameter: $w(v, v')(r)$. Mulmuley and Shah [96] used linear weight dependency to find parameter breakpoints: a combination of weights for which the shortest path changes.

Time-Dependent Graph Algorithms
Time dependent shortest shortest path algorithms processes graphs where the edge's weight is given as a function of time: $w(v, v')(t)$. Ding et al. [97] proposed an algorithm that determines the departure time at the source such that the total travel costs of a dynamic road network is minimised. Ding et al. makes a distinction between discrete and continuous dependency.

7.1.2. A*

A* is an optimal and complete goal directed shortest path algorithm, commonly applied to graph networks. It is more efficient than Dijkstra's algorithm as a heuristic is used to determine the next node to be expanded. For both methods, the next node to be expended is determined by the lowest F-value. Where Dijkstra's algorithm expends the node with the lowest costs to the initial location, often denoted as G, A* expends the node with the lowest cost to the initial location plus an estimation of the cost to the goal node, denoted with H. Thus, the F-value for Dijkstra's is determined by G, and for A* by $G + H$. As such, A* only scans the area in the direction of the destination, while Dijkstra's is expanding out equally in all directions. This heuristic search function usually makes A* algorithms more efficient then Dijkstra's [98].

The heuristic value can be determined by multiple approaches. Consider a graph in a Cartesian coordinate system, an initial node s_0, a node s at (x_s, y_s) and a goal node g at (x_g, y_g). The F-value of example vertex s, is determined by $F_s = G_s + H_s$, where G_s is the cost from s_0 to s and H_s the estimated costs from s to g, e.g. the heuristic. The following heuristics could be used:

Euclidean distance
The euclidean distance is the most commonly applied heuristic, as it gives an exact solution of the distance towards a goal node in Cartesian space, assuming that a straight road is available. The Euclidean distance can be determined by the Pythagorean theorem. The F-value for node s using a Euclidean heuristic is given in Equation 7.1.2.

$$F_s = G_s + H_s = G_s + \sqrt{(x_g - x_s)^2 + (y_g - y_s)^2} \tag{7.1}$$

Manhattan distance
The manhattan distance is commonly used in grid maps in which no diagonal movements are allowed. These grids could for example contain obstacles to represent, for example warehouses. The F-value for node s using a Manhattan heuristic is given in Equation 7.1.2.

$$F_s = G_s + H_s = G_s + \left|x_s - x_g\right| + \left|y_s - y_g\right| \tag{7.2}$$

Abstract distance
The abstract distance is the actual cost from s to g_s, found by previous searches. In single agent shortest path searches this heuristic does not make sense, as the actual cost to the goal is exactly the solution we are searching for. However, in multi-agent path finding, this heuristic can be used to efficiently find paths for single agents. Especially in small graphs with only a limited number of goals, the cost from each location on the graph to the goal can be computed in a preprocessing phase and stored in a lookup table.

Weighted heuristic
The weighted heuristic, proposed by Pohl [99] as weighted A*, multiplies the heuristic by a constant ε, as given in Equation 7.1.2. This method should thus be combined with another heuristic. The weighted heuristic is developed to significantly reduce computation time, at the cost of bounded suboptimal solutions [100]. A higher multiplier ε increases the weight of the heuristic and thereby reduces computation time, but also the solution quality.

$$F_s = G_s + \varepsilon \cdot H_s \tag{7.3}$$

7.2. Multi-Agent Path Finding

MAPF is an important type of cooperative coordination in Multi-Agent Systems (MAS). The main objective is to plan a path for all agents, with the key constraint that agents are able to follow these paths without colliding with each other. This objective requires algorithm to not only find paths for multiple agents, but also an execution plan. Therefore, the time dimension must be considered. At each timestep, an agent must perform an action by moving to another vertex via an edge or by waiting at a vertex to avoid collision. A valid execution plan is a set of action in which no conflicts occur.

This section is structured as follows. First, different MAPF problem characteristics are studied, and the variant needed for this study is discussed. Based on these findings, all commonly used MAPF approaches applicable in this research are studied. First, decentralised algorithm are discussed in section 7.2.2-7.2.10. These are algorithms in which paths are planned one after another rather than simultaneously. One of the main disadvantages of decentralised algorithms is that they cannot guarantee optimal solutions nor a solution quality with respect to a certain objective. Within these decentralised algorithms, a distinction can be made between complete and incomplete solvers. Incomplete solvers are first discussed in 7.2.2-7.2.7. These solvers cannot guarantee to converge to a solution, even if one exists. The advantage of these solvers is that they are generally very fast and efficient. These decentralised incomplete solvers can be further classified in online and offline solver. In 7.2.2, an online solver is discussed, in which agents do not solve conflicts in advance, but rather solve them when they are bound to occur in the next timestep. 7.2.3 and 7.2.4 are both offline solvers, meaning that they compute conflict free path of all involved agents in advance. 7.2.5-7.2.7 are solvers that are partially online and partially offline, depending on certain input parameters. Such parameters open the possibility to tune the model based on certain objectives. 7.2.8-7.2.10 discusses three complete decentralised online algorithms, in which 7.2.8 is only complete for problems that comply with specific requirements. 7.2.11-7.2.17 then consider centralised algorithms, in which coordination is performed in advance of the movement phases. This opens the possibility to optimally solve MAPF problems with respect to a predefined objective function. 7.2.11-7.2.13 make use of this ability and are all optimal and complete solvers. These solvers are often further developed into algorithms that give up optimally to improve success rates and algorithm runtime. four such methods are discussed 7.2.14 - 7.2.17 that all extend on the Conflict Based Search (CBS) method presented in 7.2.13.

7.2.1. Problem Variant

In airport ground network models, MAPF for vehicle routing is far beyond a classical MAPF problem. In classical problems, agent characteristics, interactions, objectives and the graph's layout can be much different than the situation of this research. These differences might require changes in the MAPF algorithm and should be taken into account while studying previous research. To make a good comparison with this research, five MAPF characteristics are discussed for the towing system of this research and compared with classical MAPF.

Objective Function

The objectives differ between the MAPF problems. There are three commonly used objectives [14]. One of them is to minimise the time at which the last agent arrives at its goal node, also known as the *makespan*. This methods, used by for example Hönig et al [24] is often used in applications where the time of the total system must be minimised, such as search and rescue, warehouse packing and assembly planning. Another objective is to minimise the total movements of agents. This objective is often referred to as the *sum-of-fuel*, and is used if fuel consumption or movement noise must be minimised. In this case, the weight of the link usually represent travel distance. The last objective is to minimise the sum of agent's action costs, known as the *sum-of-costs*, or *flowtime*. These action costs are often proportional to the total travel time. The objectives are mathematically formulated in Equation 7.4, Equation 7.5 and Equation 7.6 respectively. Refer back to the beginning of this chapter for the meaning of notations.

$$\max_{1 \leq i \leq K} |t(\pi^i)| \tag{7.4}$$

$$\sum_{i=1}^{K} |w(\pi^i)| \tag{7.5}$$

$$\sum_{i=1}^{K} |t(\pi^i)| \tag{7.6}$$

The objective of this research is to study and mitigate the resulting ground delays in an autonomous outbound towing system. To do so, a fast travel time is essential. Therefore, the MAPF objective in this study is to minimise the *flowtime*.

Conflicts

The definition of conflicts (or collisions) differs per study. Five commonly used type of conflicts in MAPF are given in Figure 7.2. Most research define a set of one of these conflict types as the occurrence of a conflict. This set usually always include conflict type B (vertex conflict, defined as $s_t^i = s_t^j$) and E (edge conflict, defined as $s_t^i = S_{t+1}^j$ and $s_{t+1}^i = S_t^j$).

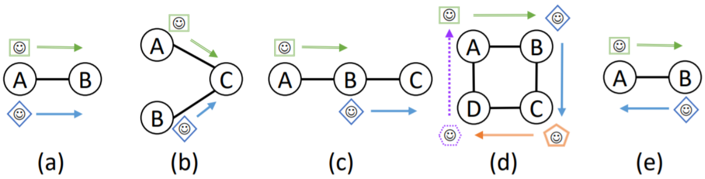


Figure 7.2: Illustration of common conflicts. From left to right: edge conflict, vertex conflict, following conflict, cycle conflict, and swapping conflict [14].

As for this research, the definition of conflicts depends on the involved agents and location. Conflict type b and e are always declared if a (towed) aircraft agent is involved. Two Taxibot agents are not restricted to conflict type b or e on taxiways and aprons, as they are able to overtake each other. Conflict and separation is further specified in the research scope of this literature study presented in section 9.2.

Agent Characteristics and Kinematic

In classical MAPF, agents are assumed to occupy exactly one vertex, they have no volume, no shape, and move at constant speed. In an airport towing system, the agents (aircraft, Taxibots or towed aircraft) move at different speed, have specific volumes, shapes, and can occupy multiple links at the same time. In other words, they are subjected to kinematic constraints. That is, the movement an agent can perform does not solely depend on its current location, but also on state parameters such as velocity and orientation. As the simulation needed for this study contains many different elements and kinematics, this issue is more extensively discussed in section 7.4.

Plan Execution

Most MAPF studies assume a perfect plan execution. That is, action a^i of agent i is either a perfectly executed move to another adjacent vertex, or no movement at all. As for this research, airport systems suffer from dynamic and uncertain situations due to (for example) new spawned aircraft, rerouted vehicles or RMO changes. Some MAPF algorithms are developed to absorb such imperfect plan execution to avoid (normally time-intensive) replanning. As this is an important issue in airport systems, it is more extensively discussed in section 7.4.

Graph Simplifications

Classical MAPF problems usually use a $\sqrt{n}X\sqrt{n}$ size raster grid [14]. These grids can get relatively large and might contain obstacles to represent, for example, warehouses. Moreover, weights are usually assumed to be equal to one: $w_{s,s'} = 1$ In airport ground networks, a graph is usually represented in Euclidean space. That is, every node represents a Euclidean point (x, y), and edges indicate allowed moves between the points. This is also true for this research, as the map visualised in Figure 6.3 is used as baseline. This map is comparable with maps containing many corridors, rather than open spaces.

One-shot or Lifelong

In classical MAPF, all agents are present simultaneously, have one specific goal, and stay at their goal node after reaching their destination [14]. Such problems are referred to as *one-shot* problems, and can easily be solved in one instance. However, there are many real life applications in which agents are assigned new goal nodes, disappear after reaching their goal, or spawn after a certain unpredictable time. These are called *lifelong* MAPF problems, and include airport ground systems [101]. There are three commonly used methods to solve these lifelong problems [101]. The first method is to use an offline solver, and solve the entire problem in advance as a whole. This method assumes perfect plan execution, and is not applicable on systems that contain uncertainties such as a uncertain arrival times of new agents. The second and third method is to use an algorithm that decompose the problem into a sequence of MAPF instances. These are online algorithms and make use of a plan-move cycle. This cycle can be used in two ways, either all paths of all agents are completely re-planned every cycle, or only paths are found for new agents. The latter method can lead to the most suboptimal solutions, as perfect plan execution is assumed, and other dynamic changes in the environment are neglected. In a dynamic and uncertain airport environment the most obvious method is therefore to use an online MAPF method that re-plans plans new paths for all agents. Offline algorithms can easily be used in an online fashion, by recursively solving the problem after a certain number of timesteps. This is usually referred to as the plan frequency, denoted with h. In other words, h is the number of timesteps in the move phase. At each plan phase, plans are generated for until a certain horizon w, called the search window. This means that conflicts are avoided for the next w timesteps. A straightforward setting is to use a plan frequency of 1 (meaning that paths are re-planned every timestep), and a search window that covers all timesteps until the last agent arrived at its node. Silver [17], was first to study the effects of different search windows and used the Hierarchical Cooperative A* (HCA*) algorithm to develop Window Hierarchical Cooperative A* (WHCA*). His work is elaborated in 7.2.5. Jiaoyang Li et al. [101] studied the use of windowed MAPF for other offline solvers, to make them applicable to lifelong systems. This is more extensively discussed in 3.2.13 - 3.2.17.

7.2.2. LRA*

The most obvious method to perform MAPF to simply push agents into the direction of their shortest path, and locally solve conflicts as they arrive. This method is called Local Repair A* (LRA*), introduced by Hart et al. [102]. It is a decentralised online solver used for its computational efficiency and its ability to adapt for uncertain and dynamic situations. In LRA*, each agents i searches for a shortest path with the A* algorithm from their source s_0^i to their goal g^i on the conventional 2-dimensional space graph, without considering any other agents but their neighbours. As long as no conflict with a direct neighbour is detected, the agents follow their route. As soon as a conflict is imminent, on of the agents recalculates their shortest path by avoiding the occupied node at the next time instance. The algorithm is finished when all agents have successfully arrived at their goal node.

The main advantage of this method is the reduced computational complexity compared to centralised offline solvers and the adaptive behaviour of agents, making the system resistant to dynamic and uncertain situations. A major drawback of LRA* is the inability to effectively handle difficult environments with obstacles and corridors. These locations are very sensitive to the emergence of crowded regions, or bottlenecks. These bottlenecks leads to many agents trying to escape the dense bottleneck by recomputing their path every turn. As each new path is computed independently of the previous, agents can get stuck in cycles in which the simular locations are visited repeatedly. This might result in extremely long paths, or agent not being able to reach their goal in the given timespan. Hence, the success rate and the solution quality of this algorithm is very low. Especially for airport ground systems (which generally contain many areas sensitive to the emergence of such bottlenecks), this algorithm performs insufficient. The next two methods are proposed to solve the emergence of bottlenecks and hence prevent the inefficient requirement to continuously reroute paths.

7.2.3. CA*

To prevent the emergence of bottlenecks (and hence reduced solution quality and success rate) in the LRA* mechanism, Latombe [103] proposed some kind of prioritised planning. An basic form of prioritised planning is often referred to as Cooperative A* (CA*). CA* is an offline decentralised algorithm than plans conflict free path consecutively for all agents. The shared 2-dimensional search space is turned into a 3-dimensional plan space, where time is the third dimension. A node in the plan space thus represents an agent being present at a certain vertex in the conventional 2-dimensional search space at a certain time (s, t). This is for two agents visualised in Figure 7.3, where each vertex (represented by blocks) is a function of coordinates: $s(x, y)$, and edges exists between (s, t) and (s', t') if and only if $t' = t+1$ and $v' = v$ or (v, v') exists. The 3-dimensional plan space can be seen as data structure in which agents are able to communicate their plans. This data structure is often referred to as a reservation table.

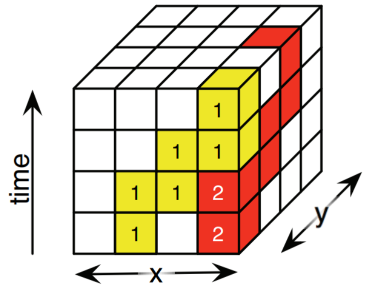


Figure 7.3: Example of a reservation table [15].

The first step in CA* is ordering all agents $i \in K$ based on a given priority: $1, ..., K$. The algorithm then finds a path for each agent based in the order of the priority in the 3-dimensional plan space. That is, a shortest path is found for each agents $i \in K$ with the constraint that the path does not collide with paths from agents ranked higher in priority. As soon as an optimal conflict free path is found, the path is stored in the 3-dimensional reservation table, to communicate the plan to other agents lower in priority. Lower prioritised agents must thus find conflict free paths based on the reservations made in the reservation table. In other words, agents with a higher priority are considered moving obstacles for agents lower in priority. To demonstrate the working principle of CA*, consider agent i and j with $s_0^i = A$, $s_0^j = B$ and $g^i = E$, $g^j = D$, visualised in Figure 7.4. In this problem, individual shortest paths would result in a conflict at node C at $t = 1$.

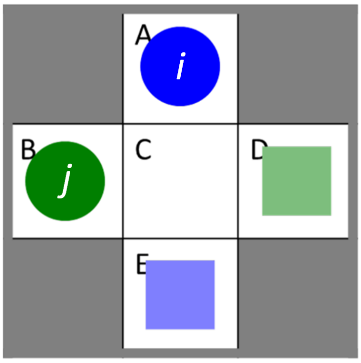


Figure 7.4: Elementary MAPF scenario [16].

As stated, the higher prioritised agents first determines the shortest path, and thus reserves vertex C in the reservation table at $t = 1$. The lower prioritised agents then calculates a path given the constraints made in the reservation by the previous agent, meaning that vertex C cannot be entered at $t = 1$, and the agent must wait at B. The search for the lower prioritised agent is given in Figure 7.5.

CA* provides a solid and fast solution for small networks. The plan space is linearly dependant on the number of timesteps and vertices. This means that the computational complexity compared to centralised algorithms is limited. A drawback of this method is that the solution quality is highly dependant on the priority order. Also, as the algorithm is offline it generally still requires some initialisation time, as it performs multiple full-depth search in the 3-dimensional plan space. The HCA* method, presented in the next section, is an extension to CA* developed to reduce the duration of this initialisation. The fact that the algorithm is offline means that it must be transformed to an online solver to make it applicable for this research. This is

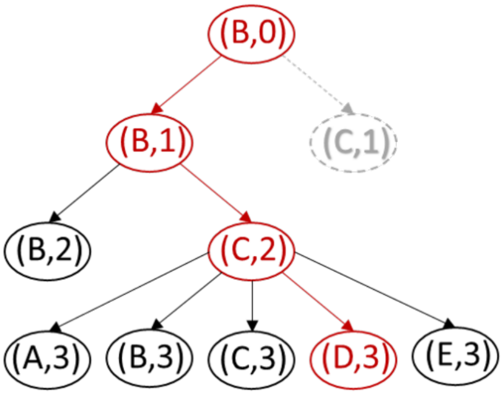


Figure 7.5: CA* path of lower prioritised agent for the elementary problem of Figure 7.4 [16].

discussed 7.2.5, in which WHCA* is introduced that uses search windows to make the algorithms usable for lifelong problems.

7.2.4. HCA*

The efficiency of CA* can be improved by a better heuristic. CA* uses the Manhattan distance (the sum of the absolute differences of the Cartesian coordinates), as heuristic. However, this can lead to low performance in complex environments. Silver [17] proposed HCA* as a more efficient method than his own CA* due to a more accurate heuristic. Instead of the Manhattan distance, he uses the abstract distance to the goal vertex as heuristic. The abstract distance is the shortest distance from any node to the goal node found in the 2-dimensional search space, rather than the 3-dimensional plan space. Thus, the abstract distance is shortest distance to the goal node while ignoring interactions with other agents. For smaller graphs, the abstract distance to goals can be precomputed and stored in a database. For larger graphs, Silver introduces Reverse Resumable A* (RRA*), an A* search in the 2-dimensional search space in the reversed direction.

Silver [17] concluded that a more sophisticated heuristic indeed reduces computation time of of CA*. However, both algorithms remain far from optimal, as the solution quality heavily depends on the priorities set to agents. The methods are neither complete, meaning that the algorithm does not always converge to a solution, even if one exists. An example of this issue is given in Figure 7.6. It can easily be visually determined that a solution exist. However, CA* and HCA* are unable to converge to a solution in this scenario as the planned path of the prioritised agent (doesn't matter whether this is i or j) blocks the other agent from finding a solution. Finally, as both algorithms are offline, they need to be transformed to an online algorithm to make it usable for this study. The next section covers WHCA*, an algorithm that makes HCA* online, more complete, and faster.

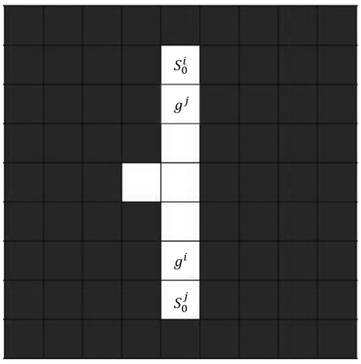


Figure 7.6: Scenarios that HCA* cannot solve [17].

7.2.5. WHCA*

Silver [17] developed the WHCA* to make the HCA* algorithm online, and solve several of its inefficiencies. The first is related to the incompleteness visualised in Figure 7.6. In this example, the higher prioritised agents blocks the path of the other agents after reaching its destination. To solve this issue, either the paths should be found simultaneously, or agents must continue cooperating after they reached their destination such that they can allow other agents to pass. The second inefficiency is the dependency of the solution quality on the priority ordering. A more ideal solution would be to randomly vary the order of priority throughout the search, such that paths can be found which are impossible to detect in a fixed priority order. Finally, HCA* determines a complete path in the 3-dimensional plan space. As such, plans are generated which are, due to dynamic and uncertain environment, unlikely to occur, meaning that the algorithm performs redundant computations.

Silver proposed a solution in which the 3-dimensional plan space (and thus the 3-dimensional reservation table) is limited to a fixed depth of timesteps, called the search window. The length of the search window, or the number of timesteps, is denoted with w. This means that, in contract to HCA*, agents plan their paths in the plan space up until w timepoints in the future, rather than the complete path. After w timepoints, agents compute the remainder of their path in the 2-dimensional search space to ensure their heading is in the right direction. This means that conflicts are only detected and resolved for w timesteps in the future, and ignored for the remainder of the search. After planning is done, the agents follow their reserved path up to timestep h. After h timesteps, the search window shifts forwards such that a new route is computed for the next w timesteps in the 3-dimensional plan space and in the 2-dimensional search space for the remainder of the route. In other words, the algorithm consists of a plan-move cycle, in which w timesteps are planned, and h timesteps are moved. Note that, since paths are only guaranteed to be conflict free up to w timesteps, the equation $h \leq w$ must hold. In fact, to avoid head-on conflicts, $h \leq \frac{w}{2}$ must hold.

The window size w is thus a parameter indicating the depth of which conflicts are avoided. If w is equal to the complete search time (that is, w is equal to the arrival time of the last agent at the goal), WHCA* behaves the same as HCA*, as all paths are completely planned in the 3 dimensional plan space. If w is equal to zero, no paths are planned in advance and only those conflicts occurring in the next timestep are avoided, meaning that WHCA* behaves as LRA*. The parameter w thus provides a spectrum between LRA* (an online algorithm) and HCA* (an offline algorithm), in which WHCA* behaves more like HCA* for larger windows and more like LRA* for smaller windows.

It is essential that w and h are chosen carefully. A smaller search window w means more short sighted decisions. This means that the computation time per turn is shorter, and that a smaller reservation table is required. Moreover, Less redundant conflicts are solved and the algorithm is more complete as it suffers less from unsolvable situations as visualised in Figure 7.6. The shorter search window goes at the cost of more turns that needs to be computed and a reduced solution quality resulting from the emergence of bottlenecks and frequent rerouting. If the window size become too small, the emergence of bottlenecks can drastically decrease solution quality and computation time and should therefore be avoided at all cost. An overview of the effects of a shorter search window is given in Table 7.1. Note that $h \leq w$ must hold to avoid conflicts. If $h \to 0$, the algorithm performs more turn computations, leading to better adaptive behaviour, increased success rates and a better solution quality at the cost of more turn computations. In terms of algorithm performance, the solution quality increases with search window size w. As for the success rate, there is an optimum search window size as a shorter window is more complete but leads to bottlenecks that result in agents not being able to arrive at their goal in the given timespan. The algorithm runtime is especially high at high w and low h.

Table 7.1: Effects of a short sighted decisions (shorter search window w)

Advantage	**Disadvantages**
1) shorter turn computation time.	1) More turn computations.
2) smaller reservation table.	2) Reduced solution quality.
3) solves less redundant conflicts.	3) Risk of bottlenecks.
4) More complete.	4) Frequent rerouting.

WHCA* has several important advantages for airport operations. The algorithm is efficient in dynamic and uncertain environments, as it recompute paths in the plan space regularly, and does not solve redundant conflicts. Furthermore, the algorithm is decentralised, and therefore generally faster than other centralised algorithms. The success rate, solution quality and computation time can be tuned by smartly adjusting w and h. Another advantage is that priorities can be set based on agent characteristics. A drawback of this method often mentioned in literature is that the algorithm is not complete, meaning that a solution is not guaranteed. In airport systems however, agents disappear after they reached their goal (as they take-off or park at a gate) and are always able to reduce speed to stay at their current node, meaning that the algorithm is complete as solutions can always be found. As the algorithm is decentralised and often runs based on random priority, the algorithm is not optimal. However, in a dynamic, uncertain and complex airport systems with many different types of agents, optimality is hard to define and guarantee. Moreover, optimality in airport ground movements is not an end in itself. It can be approximated very well by adjusting w, h and the priority of agents.

A number of variants of WHCA* are introduced to enhance the performance of WHCA*. Jansen and Sturtevant [104] proposed directional maps to drastically reduce the computation time. Wang and Botea proposed Flow Annotation Replanning (FAR), a method that uses directional columns and rows in grid maps to enhance the computational performance. Bnaya and Felner [18] developed Conflict Oriented - Window Hierarchical Cooperative A* (CO-WHCA*), an algorithm that is focused on agent coordination around conflicts. The former two methods are due to its directionality and graph assumptions not applicable for airport ground systems. The latter method is discussed in the next chapter.

7.2.6. CO-WHCA*

CO-WHCA*, proposed by Bnaya and Felner [18], extents on WHCA* by focusing on coordination around conflicts. In WHCA*, plan are generated for w timesteps and subsequently agents are moved for h timesteps into the planned direction. In CO-WHCA*, the search window w is flexible and equal to the time at which the first conflict occurs in the plan phase. Constraints in the reservation table are made by a single agents, only at those areas and timesteps where conflicts are detected. The advantage of this strategy is that the number of reservations is drastically reduced, avoiding many computational intergrations in the plan phase. A parameter W is used to determine the length of the reservation window (not to confuse with search window, indicated with a small w). This reservation window is set around conflicts after one is detected.

The plan-move cycle of CO-WHCA* is adjusted as follows. In the planning phase, individual paths are found for each agent $a_i \in K$ from its current location s_0^i to the goal g_i in the 3-dimensional plan space. During this planning, reservations made in previous cycles are taken into account. Clearly, in the first cycle, no previous reservations are prevent. The algorithm subsequently detects the first conflict. This conflict consists of a set of agents (usually two) $K_c \subset K$, a vertex s and a time t. A conflict owner $a_c \in K_c$ is randomly assigned. A reservation is subsequently made in the plan table for the conflict owner at vertex s for a timespan of W, the input parameter. This means that, vertex s is reserved for a_c from $t - \frac{w}{2}$ to $t + \frac{w}{2}$. As soon as this reservation is made, the plan phase is replaced by the moving phase. In the moving phase, all agents are moved along their planned path up to $t - \frac{w}{2}$. At that point, the planning phase starts and the modifications made in the reservation table in previous cycles are taken into account.

Bnaya and Felner compared CO-WHCA* with WHCA*, and demonstrated that this algorithm has improved computation time and success rates, without losing solution quality. To further increase solution quality and success rates, Bnaya and Felner replaced the random selection of conflict owner by a prioritisation algorithm, discussed in the next section.

7.2.7. CO-WHCA*P

To further increase solution quality and succes rates, Bnaya and Felner introduced Conflict Oriented - Window Hierarchical Cooperative A* prioritization (CO-WHCA*P). This algorithm extends CO-WHCA* by replacing the random selection of the conflict owner by a mechanism that detects the optimal priority. This prioritisation mechanism determines the optimal priority of agents involved in conflicts by simulating all scenarios in subsets of plan spaces and subsequently evaluating their outcomes based on a predefined objective. In most cases, two agents are involved and two subsets must be created. In the rare event that multiple agents are involved in a conflict, the number of subsets increases factorially with the number of involved agents.

Bnaya and Felner demonstrated that this prioritization mechanism further increases solution quality, but requires longer computation times for large number of agents. The resulting performance is very close to optimal, as individual conflicts are solved optimally, rather than all paths simultaneously. The performance of WHCA*, CO-WHCA* and CO-WHCA*P is compared in Figure 7.7. In this figure, the top row indicates success-rate, the middle row solution quality and the last row algorithm runtime.

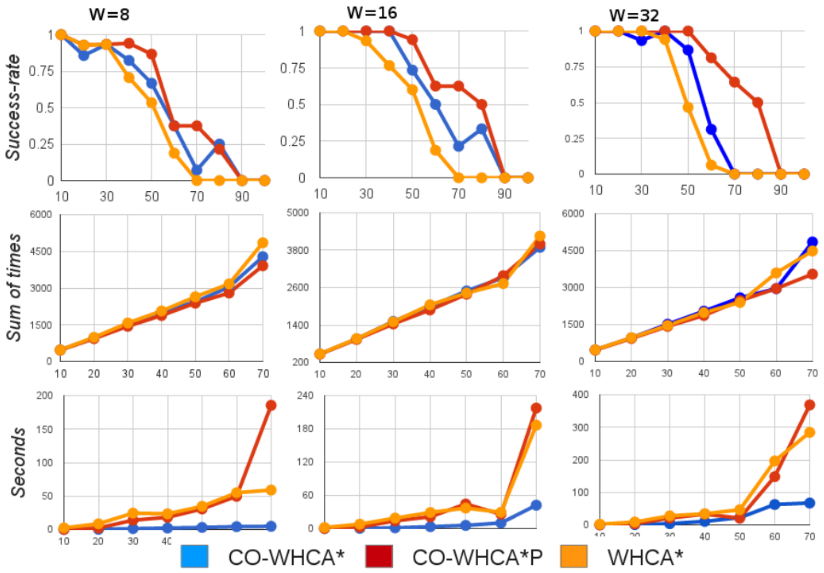


Figure 7.7: Comparison of WHCA*, CO-WHCA* and CO-WHCA*P [18].

7.2.8. MAPP

The algorithms mentioned so far all lacked the property of a guaranteed solution. Ko-Hsin and Wang [91] introduced Multi-Agent Path Planning (MAPP), a decentralised algorithm with guaranteed solutions for slidable path problems. In MAPP, shortest paths are computed for all agents $a_i \in K$ from a start node s_0^i to a goal node g^i using the alternative path algorithm, after which the agents are marked to have a slidable or unslidable shortest path problem. A shortest path problem is said to be slidable if the following three conditions are met:

1. For each three consecutive vertices along the path, an alternative path exists. This is visualised in Figure 7.8, in which an alternative path is shown for the vertex occupied at t. The alternative path for that vertex is indicated with Ω_t.
2. The start node s_0^i is unoccupied by other agents.
3. The goal node g^i does not interfere with shortest or alternative paths of other agents.

After the shortest and alternative paths are generated, the agents are pushed over their shortest paths. In case conflicts are detected at the next timestep, agents are diverted to the alternative paths for the node on which the conflict is expected to occur, based on a hierarchical order. This is repeated until all agents are at their designated goal nodes. The algorithm thus consist of one large plan phase, followed by a move phase. This is sufficient for an airport systems, in which the number of initial and goal nodes is finite. This means that (alternative) paths can be generated in advance from all possible initial nodes to all possible goal nodes, such that a complete day of simulation solely exists of a huge move phase.

Experiments have shown that MAPP determines conflicts free paths for 99.86 % of all agents, while reducing algorithm runtime compared to centralised approaches for large number of agents. K. Fines [11] discussed the practical implementation of alternative paths in a ground network of AAS. He stated that, due to the complex layout of AAS, the first requirement cannot be satisfied in many cases. Furthermore, Fines stressed that airport systems are uncertain, as new aircraft might appear. These aircraft could than suddenly block

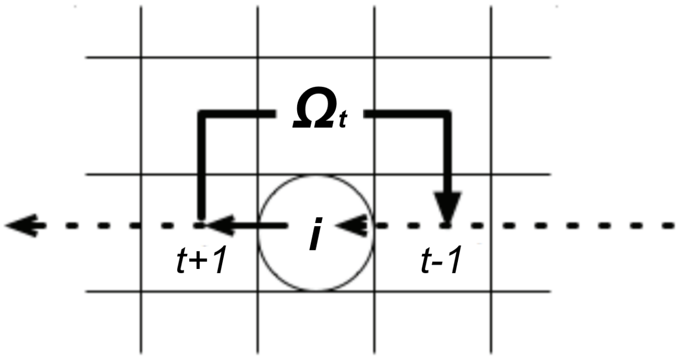


Figure 7.8: Alternative path example [19].

alternative paths, especially near runway entrances. Finally, Fines concluded that the algorithms capability to reduced runtime for large number of agents is of lesser value, as the maximum number of agents in the airport network does not increase beyond extreme values. Next to the observations made by Fines for his research, the objectives of this study must be considered. In this research, departing aircraft must uncouple at taxiways. This requires goal nodes to be located on taxiways, which might lead to violation of the third requirement. Given these observations, it is questionable whether MAPP is applicable in airport networks.

7.2.9. Push-and-Swap

The algorithms studies so far are decentralised and thereby usually efficient and fast. However, they lacked the property of being complete. MAPP was then discussed as a complete method method for slidable problem. This section discusses Push-and-Swap, a decentralised and complete method for all problems.

Push-and-Swap, introduced by Luna and Bekris [20], is a decentralised offline solver developed for its efficiency and guaranteed solution. First, all agents are order based on some priority ranking. Then, the algorithm finds shortest paths for each agent from a given source to a goal vertex in order of their priority. After the shortest path of an agent is found, the agent attempts to move forwards along its path. The resolution method of any conflict along the path depends on the priority of the other agent. If the agent is lower in priority (meaning that a path for the agent has not been found, yet), the agent is pushed away from the vertex along a shortest path to the nearest empty vertex, if possible. This operation is indicated in Figure 7.9, where agent 1 is highest in priority, and pushes agent 2 and 3 aside to reach its goal node g^1.

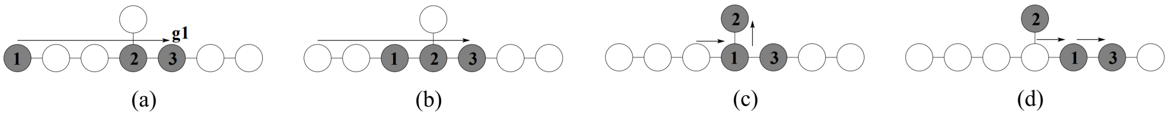


Figure 7.9: Illustration of a push operation [20].

If the push operation is not possible, or if the other agent is higher in priority (meaning that a path for the agent has already been found) a swap operation is performed. This operation involves four consecutive steps. The first step is to move both agents to the nearest vertex with a degree of three or higher. That is, a vertex with at least three adjacent vertices. The second step is to clear two adjacent vertices from other agents. Then, step three involves the actual exchange in position of both agents, by moving to the empty adjacent vertices. Finally, step four ensures that all agents are back on their initial location, expect the two swapping agents. The steps are visualised in Figure 7.10, where agent s and r are swapping locations.

Luna and Bekris [20] claims to have developed an efficient method, as conflicts are solved locally, rather than globally. The method is also claimed to be complete, as solutions are always found if possible. However,

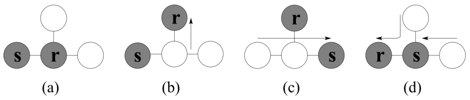


Figure 7.10: Illustration of a swap operation [20].

the methods has some serious drawbacks making its usability for this research questionable. First of all, the method is not applicable on polygon graphs (graphs with a single cycle). Also, as agents are "pushed" along their paths, deadlocks could arise. Furthermore, agents which already arrived in their destination might still be required to move as a result from swapping operations of two other agents. This results in many impractical movements, especially in airport ground operations. More impractical aircraft movements arise due to the fact that rotation at a vertex is not considered, which for aircraft operations is a big issue. Finally, when moving an agents back to its destination after a swap, new swap operation might be required. This result in recursive calls to swap. The former and the latter issue is solved by the Push-and-rotate method, discussed in the next section.

7.2.10. Push-and-Rotate

Push-and-rotate, developed by De Wilde et al. [105], is an extension on Push-and-swap that uses rotation of multiple agents to solve impractical or impossible scenarios. As such, the algorithm prevents recursive swapping operations and is applicable on polygon graphs. The extension remains efficient and complete. The main additions of this extension is the development of subgraphs in which rotation can occur. Subgraphs are developed in a preprocessing stage in which rotating is possible, meaning that each agent in the subgraph is able to reach any other position. Normal push and swap operation are performed, but redundant recursive swapping is eliminated and polygon graph deadlocks are solved by rotate operations. An example of a rotate operation is given in Figure 7.11, in which a cycle conflict (type D) is solved by swapping two other agents.

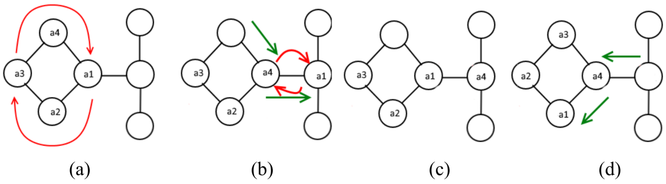


Figure 7.11: Illustration of a rotate operation [11].

De Wilde et al. [105] demonstrated that Push-and-rotate solves impractical issues of which Push-and-swap suffers from to improve solution quality, while maintaining high efficiency and completeness. Especially on larger graph with many agents, Push-and-rotate has a relatively higher success rate. The method however still suffers from many impractical problems in airport operations, such as assumed side and backward movements. Moreover, the method still requires agents to move that have already arrived at the goal node. This imposes many issues related to engine start up or the continuous required presence of towing vehicles. The methods discussed in the next sections will be offline and centralized, meaning that optimal and complete solutions can be found for offline problems.

7.2.11. A* in joint graphs

The A* algorithm, discussed in section 7.1, is often used in single agent path finding because of its completeness, optimality and efficiency. A very basic approach for MAPF is to use A* on an alternative graph, representing the multi-agent system. This is done by transforming a regular static 2-dimensional space graph, into a joint static 2-dimensional space graph, meaning that each node represent a combination of locations for all agents present, rather than just a single location. Consider again Figure 7.4. To solve for the conflict at $t = 1$, the graph (currently consisting of 5 nodes) can be extended to a graph in which nodes indicate the location of both agents and edges represent possible moves for both agents to perform simultaneously. Subsequently, A* can be used to find the shortest path from the start node (in which both agents are located in their initial location) to the goal node (in which both agents are located in their final location). This process is visualised in Figure 7.12, in which the first and the last letter in the vertices represent the location of agent i and j respectively. A shortest path (indicated in red) is found by A* in which agent i remains at node A at $t = 1$. Note that a second equally valid solution exists for $AB - CB - EC - ED$.

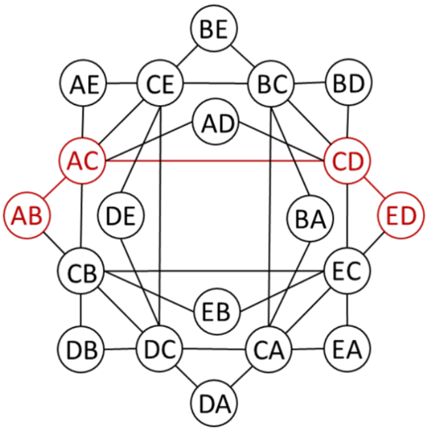


Figure 7.12: A* search space for elementary MAPF scenario [16].

As this algorithm computes all plans simultaneously in advance, the algorithm is centralised and offline. The latter means that recursive planning is required to make the method usable for this research. The advantage of this system is that the algorithm is optimal and complete. A serious drawback of this simultaneous planning strategy is the exponential growth of the search space with the number of agents. As one can easily observe from the given example, the search space of the alternative graph depends on the number of agents and nodes in the main graph. In fact, the number of nodes in the alternative graph is equal to n^K. The exponential growth of search space makes this method slow in practice, and inapplicable for this research as up to 20 agents can be present on the graph simultaneously. This is only reinforced by the fact that recursive planning is required to make this algorithm usable for airport systems. Two other methods that both solve MAPF problems optimally without exponential growth in algorithm runtime are discussed next.

7.2.12. CP

The A* method applied on the alternative joint location graph is centralised and offline, meaning that all plans are computed simultaneously in advance. This results in an algorithm that is optimal and complete, but suffers from exponential growth of computational complexity with the number of agents and the lack to account for uncertain and dynamic situations. Constraint Programming (CP) is a method in which a set of decision variables are bounded by constraints, and optimised for a predefined objective. An example of such method is LP, in which the goal is to achieve an optimal outcome in a mathematical model where the objective and constraints have linear relationships [106].

CP methods like LP are often used in airport operational research to find taxi routes of multiple aircraft simultaneously. All these studies use a combination of integers and binary variables, making it a Mixed Integer Linear Program (MILP). The basis of the variables in most studies are binary variables that are distinguished by a certain aircraft (or flight) i, being on location (normally node or a link) s at a time t (discretized, varying from 10-30 seconds) [107–110]. The variables are subjected to constraints. The most commonly used constraint is that two aircraft cannot be on the same node s at a given time t. This means that edge conflicts (type a) and vertex conflicts (type b) are avoided. Other constrains could be imposed to avoid other type of conflicts. Most studies develop this basic strategy further into different variant like including delay time [109], use time as a continuous variable [108] or make use of two nodes to account for travel direction [108]. Different objectives are used to compute the taxi route:

- Minimise: Total taxi time [107, 108].
- Minimise: Total taxi distance [111].

- Minimise: Departure delay [107–109].
- Minimise: Aircraft emissions [108].
- Minimise: Aircraft fuel consumption [110]

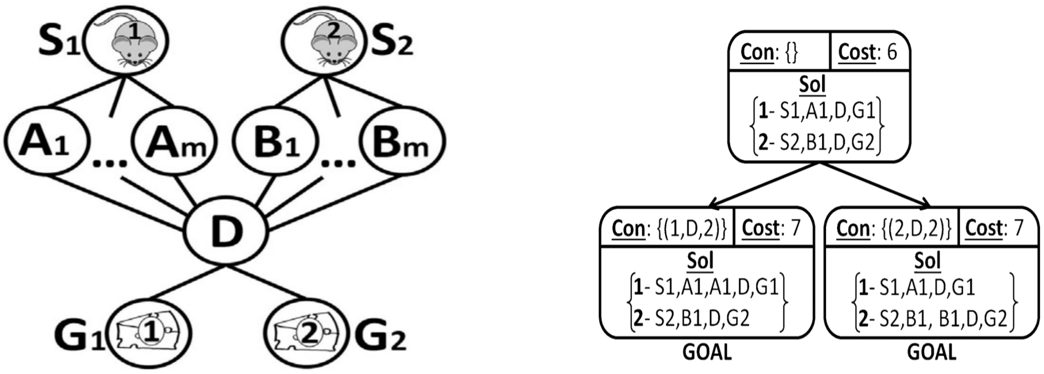
A disadvantage of LP is that the computation time is highly dependent on the problem size. As airports are rather complex systems, the number of variables and constraints (and thereby the problem size) is very large. This limitation is observed in all studies: many ground vehicles and discrete timesteps in combination with complex airport systems drastically increases problem size. This limitation imposes many restrictions in the model, and is the main reason that most studies exclude important operational details and are therefore forced to make assumptions that lack a proper representation of reality. Examples of such assumptions are the use of constant taxi speed (Meaning that no acceleration/deceleration is taken into account) and the exclusion of aircraft interaction or complex conflict types (like C en D) [109, 112]. As CP is a centralised offline solved, the optimisation is performed for a stationary equilibrium. Therefore, the model optimises paths simultaneously and thereby assumes that all operations are carried out perfectly. As a result, the entire model falls apart in case a vehicle fails to carry out their operation perfectly. This could be solved by decomposing the problem into a sequence of LP instances. However, this solution would be very impractical given the computationally costly optimisation. Another option would be to take delay probabilities into account. This is discussed in subsection 7.4.3.

7.2.13. CBS

CBS is a MAPF algorithm presented by Sharon and Stern [21] that minimises the flowtime. The basic principle of this approach is based upon constructing a structured search tree (named the Constraint Tree (CT)) to determine the optimal set of constraints to minimise flowtime. The algorithms consists of a high-level and low-level search. In the high-level search, a best-first search in the CT is performed to resolve collisions amongst agents while minimising flowtime. This means that, at each cycle, the CT node with the lowest sum of cost of all agents is explored. If conflicts are detected, the first conflict is selected and two child nodes are generated, both restricting one of the two agents from occupying the conflict node at the conflict time. Subsequently the low-level search finds individual optimal paths in each newly generated CT node by obeying the new constraint, and all constraints of the parent node. The total sum of cost of both child nodes is then calculated, such that the high-level cycles can repeatedly expand a new node with the lowest sum of cost. The principle is discussed using the example illustrated in Figure 7.13 developed by Sharon and Stern [21].

Two mouse agents $\{S_1, S_2\}$ must cooperatively find a path to their respective goals $\{G_1, G_2\}$, such that the total sum of cost is minimised and no collisions occur. A CT is explored to find the optimal set of constraints that minimises flowtime. The problem starts in the root node of the CT in which (per definition) no constraint is present. For both agents, the shortest path is found using one of the algorithms presented in section 7.1. It is then determined that both paths require three timesteps, making the total cost of this solution six. The high-level search algorithm now validates this solution and determines a conflict after two timesteps, in node D. Therefore, two child nodes are generated in which one node prevents agent S_1 to be at location D at timestep 2 (indicated with $\{1, D, 2\}$) and one that prevents agent S_2 to be at location D at timestep 2 (indicated with $\{2, D, 2\}$). The low-level search now determines for each child CT node a new shortest path for the agent that was imposed a new constraint. This results in agent s_1 and s_2 waiting at the second timestep for the left and right node respectively. The total sum of cost is now determined for both child nodes, such that the high-level cycles is able to select and explore the node with the lowest sum of cost. For this example, the high-level search finds that there are two nodes with the lowest sum of costs. The algorithm presented by Sharon and Stern [21] now prefers the node with the lowest number of collisions. As this is also equal, a random node is selected. It is then found that no collisions occur, meaning that the algorithm has found an optimal solution in the current node. Sharon and Stern have proved this claim to be true due to the fact that the CT node with the lowest sum of cost is always explored first.

Experimental results in [21] have demonstrated that CBS is always complete and cost-optimal. The algorithm is more effective when dealing with corridors and bottlenecks instead of open spaces, meaning that it is especially suitable for airport ground systems. Drawbacks of CBS are that the algorithm scales rather poorly, as the problem space is exponential to the number of conflicts. Also, as CBS is an offline solver, it must run



(a) Two mouse agents $\{S_1, S_2\}$ have to reach the cheese $\{G_1, G_2\}$.

(b) Constraint Tree (CT).

Figure 7.13: CBS example [21].

repeatedly at frequent time intervals, as airport ground systems have a dynamic and uncertain environment. Given the poor scalability, this issue might require many computations. Furthermore, as more extensively discussed in section 7.5, the advantage of fully optimal and complete solutions is of lower value in uncertain airport system. CBS also optimises for one global objective, which is to minimise the flowtime of all agents. However, in an airport towing system, many different types of agents are involved in which the timely arrival of some agents is more relevant than others. 7.2.14 and 7.2.15 introduce methods that can be implemented to reduce computation time by dropping the optimality requirement. Two other extension on CBS are given in 7.2.16 and 7.2.17 that both include prioritised planning in search trees to increase computational efficiency, success rates and allow for a pre-defined priorities, at the cost of only limited sub-optimal solutions.

Bounded-Horizon CBS
CBS is, just as the previous two methods, an offline solver. To make the method usable for lifelong problems like airport ground systems, a bounded-horizon (or search window) must be used. Jiaoyang Li et al. [101] studied the use of search windows for the CBS algorithm. He stated that the only modification needed in order to transform conventional CBS to a bounded-horizon variant is to adjust the conflict detection horizon by a limit w. Simulation results performed by Jiaoyang Li et al. indicate that small windows drastically reduce computation time as smaller CT size are needed. Solution quality is not substantially affected by the window size, until a certain threshold is reached and deadlocks may emerge. He recommended future research to study these threshold points such that more optimal search windows can be found based on expected congestion and planning time budget.

7.2.14. suboptimal CBS

In general, giving up optimality reduces computation significantly. Therefore, suboptimal variant of CBS are developed to reduce algorithm runtime at the cost of suboptimal outcomes. The working principle of these suboptimal variants is based on a relaxation of the high- and low-level searches. In classic CBS, Both search levels are performed optimal: the low-level search find the optimal shortest paths for single agents, and the high-level search finds the minimal cost of the CT. Just as with any other optimal search algorithm, ensuring optimality causes extra work due to the exploration of abandoning nodes. Suboptimal solvers allows for a more flexible high- and low-level search, such that only nodes are expanded that are more likely to produce a fast and valid (possible suboptimal) solution. Wagner and Choset [113] studied suboptimal CBS solvers and concluded that a runtime reduction of several orders of magnitude can be achieved while sacrificing only minor losses in solution quality. They distinguished two types: bounded and unbounded suboptimal algorithms. Bounded suboptimal algorithms guarantee a solution which is no larger than a given constant factor over the optimal solution cost. Unbounded variants do not guarantee such solution, meaning that it might be far from optimal. M. Barer et al. [100] continued this study and introduced three variants of the CBS algorithm: Greedy Conflict Based Search (GCBS), Boundec Conflict Based Search (BCBS) and Enhanced Conflict Based Search (ECBS).

GCBS is an unbounded suboptimal solver. The higher level search prioritise CT nodes that seems closer to a goal node by means of a heuristic function. The lower level search uses suboptimal shortest path methods, like the weighed A* method proposed by pohl [99]. The GCBS is completed if a first solution is found, without expending all possible nodes in the CT. GCBS can be extended to BCBS, where the lower- and higher-level search solution subjected to a given suboptimality bound. In other words, both solutions are guaranteed to be within an individual factor from the optimal solution. ECBS extends this version by subjecting the lower- and higher-level search to a joint suboptimality bound, meaning that a combination of the lower-and higher-level search must result in satisfactory solution. The advantage of ECBS over BCBS is that additional flexibility is provided to the high-level search when the low-level search find close to optimal solutions and visa-versa.

Research have demonstrated a substantial increase in runtime for all variants. M. Barer et al. [100] claims that GCBS is able to solve problems of 250 agents within the same timespan as CBS with only 50 agents. ECBS solves simular problems up to 200 agents where the solution (e.g. high- and low-level combined) is at most 1% worse than the optimal solution. Jiaoyang Li et al.[101] state that ECBS can be just as easily converted to bounded-horizon variants as CBS, and showed that the bounded-horizon ECBS variant (with a suboptimality bound of 1.1) is able to solve problem with 500 agents just as fast as the bounded-horizon CBS variants, for a window size and plan frequency of five timesteps.

7.2.15. CBS + HWY

Cohen et al.[114] concluded that the use of bounded-MAPF algorithms such as CBS often do not perform well in domains with many agents. Hence, they developed the sub-optimal algorithm named Conflict Based Search with Highways (CBS + HWY). This algorithm exploits the graph structure in a given MAPF problem to find paths for agents that include user-provided highways. These highways encourage or force a global behaviour of agents to avoid collision. Cohen et al. experimentally demonstrated that the CBS + HWY mechanism significantly decreases runtime at the cost of small sub-optimal solutions for high density agent areas. However, such highways might also lead to incomplete problems, as suitable paths became unavailable due to the additional highways.

K. Fines [11] studied the use of adaptive highways to improve the resilient behaviour of a MAS based on the layout of AAS. He concluded that the highway planning and coordination mechanism in high density traffic areas contributes toward the resilient behaviour of the system. From this result, it can be concluded that the use of CBS + HWY might be a promising method to effectively manage traffic flow at dense traffic areas such as the runway entrances where the uncoupling of Taxibots takes place. The implementation of highways still require CBS to be used in its bounded-horizon variant.

7.2.16. CBSw/P

Conflict Based Search with Priorities (CBSw/P), introduced by Hang Ma et al [22], is based on the same principles as conventional CBS: a high-level cycles explores a constraint search tree to find an optimal set of constraints that minimises the flowtime, and a low-level search solves individual shortest paths that obeys the constraints set in each CT node. The sum of cost of all individual paths can then be determined such that the high-level cycles can expend the new CT node with the smallest cost. This means that, each node in the CT contains a set of constraints, a set of paths that obey these constraints, and a cost equal to the sum of arrival times of all paths, which determines the next node to explore. Where CBS always generates two child CT nodes, CBSw/P only generate child nodes when the priority order of the parent node allow for it. Nodes in the CT of CBSw/P thus contain a priority order as a fourth element. The root node of the CT starts with an empty set of priorities.

This works as follows. Just like in conventional CBS, the CT node with the lowest sum of cost is explored, and the first conflict is detected (which can be either a vertex or an edge conflict). Conventional CBS subsequently generates child CT nodes for all agents involved in the conflict to prohibit the agent from occupying the conflict node at the conflict time. CBSw/P only generates a child CT node for the agents which is lowest in the priority order of the parent node, or not in there at all. If two child nodes are generated (meaning that one of the agents was not in the priority order of the parent node), then the corresponding priority pair of the child node is added to the priority order of the parent node. This means that child nodes inherit the updated plans, the sum of costs of all these plans, all constraints, and the priority order of the parent node, plus the additional priority pair (if the agents weren't in the priority order of the parent node already). No research is found on bounded-horizons CBSw/P. However, when looking at the main differences between CBSw/P and conventional CBS, it is expected that bounded-horizon CBSw/P improves the scalability in terms of success rates and runtimes compared to the bounded-horizon CBS introduced by Jiaoyang Li et al. [101].

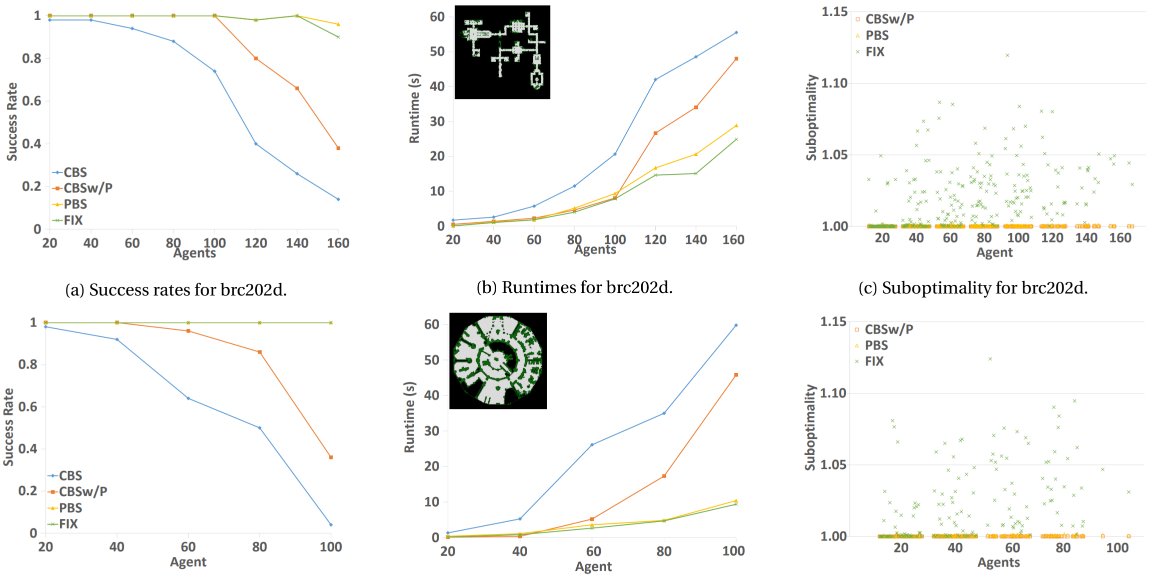
7.2.17. PBS

Hang Ma et al [22] proposed a second method to incorporate prioritization in CBS, named Priority Based Search (PBS). PBS performs a depth-first search in a Priority Tree (PT), rather than a best-first search in a CT. This means that the algorithm of PBS constructs a priority order, rather than a set of constraints to find a solution. The other principles of CBS remain largely unchanged, meaning that the nodes of the search tree still inherits a set of plans, and a cost value equal to the sum of cost of all paths in the plan. In the low-level of PBS, the priority order is, just like in the CA* algorithm discussed in 7.2.3, used to plan consecutively conflict free path in the 3-dimensional plan space. The main difference is that CA* uses a random priority that includes all agents, while the low-level of PBS uses a different priority for each node in the PT, that does not necessarily include all agents.

PBS uses depth-first search to find an efficient priority order. This works as follows. The lower-level cycle searches individual routes, based on a set of priorities, not on a set of constraint. Subsequently, the higher-level cycle validates this route for conflicts. If a conflict is found, it can be concluded that one of the two agents in not in the priority order of the current PT node, as otherwise the lower-level cycle would have solved for these conflicts. Therefore, two child nodes are generated, in which each agent is added to the priority order higher then the other. Individual shortest paths are then generated in both child PT nodes that obey the adjusted (slightly longer) priority order. The cycle then starts again, and the child node with the lowest sum of cost is explored. As child nodes are constantly explored, a depth-first search is used, rather than a best-first search. This means that a single priority order is constructed until no conflicts are found. If no solution is found, the algorithms backtracks to explore other branches.

In contrast to CBSw/P, the PBS algorithm allows for an initial priority ordering. However, as initial orders reduces search space, it is likely to result in more suboptimal solution, or no solution at all. The advantage of initial ordering is that it can be used to prioritise those agents which are more important too reach their goal first. Especially in system with different agent types and objectives, this possibility is a huge plus. In the extreme case that all agents are included in the initial priority order, the PT returns a solution in the root node, and PBS behaves exactly like CA*. Hang Ma et al. [22] named this PBS variant FIX.

Hang Ma et al. [22] compared the PBS algorithm with CBS (3.2.13), CBSw/P (3.2.16) and FIX (CA*, 3.2.3). The experiments where performed on game maps of Dragon Age: Origins (Sturtevant 2012). Two maps where taken as sample area, The brc202d map which contains many corridors, and the lak503d map mainly consisting of open spaces. Results are given in Figure 7.14. Figure 7.14a and Figure 7.14d show the success rate: the percentage of MAPF instances that where solved within the given time limit. It can be seen that both CBSw/P and PBS outperforms CBS in this area. The higher success rates can be explained by the reduced CT and PT size of CBSw/P and PBS, compared to CBS. As CBS explores most branches, it has a higher likelihood not to solve the problem in the given time limit. PBS explores even less branches than CBSw/P, as PBS adds priorities to agents whereas CBSw/P is focused on conflicts only, meaning that CBSw/P generally explores larger search trees. For problems with a high number of agents, FIX is in some instances unable to find a solution in the game map with narrow corridors. The success rate for PBS goes slightly down for very large number of agents, due to the time limits. Figure 7.14b and Figure 7.14e present the runtime of the algorithms. Algorithm runtime generally increases with the number of agents, bur remain low for small number of agents. As a result of the smaller CT and PT of CBSw/P and PBS, the runtimes are generally lower than that of CBS. Also, as PBS solves conflicts by prioritising agents rather than adding constraints around conflicts, the runtime does not increase so much for large number of agents as CBS and CBSw/P. FIX always shows the best performance in terms of runtime, as the algorithm returns a solution at the root node of the PT. Finally, Figure 7.14c and Figure 7.14f show the ratio of flowtime compared to optimal flowtime. As expected, FIX results in the worst performance, as the priority order is not optimised, but a fixed initial input. CBSw/P and PBS both converge to either optimal solutions, or very close to optimal solutions. CBS is not included in the graphs, as it always converge to an optimal solution.



(a) Success rates for brc202d.

(b) Runtimes for brc202d.

(c) Suboptimality for brc202d.

(d) Success rates for lak503d.

(e) Runtimes for lak503d.

(f) Suboptimality for lak503d.

Figure 7.14: Results of experiments with CBS, CBSw/P, PBS and FIX (CA*) on game maps. Subfigure a,b and c show results for the brc202d map, given in the top left of figure b. Subfigure d,e and f show results for the lak503d map, given in the top left of figure e. [22]

The partial inclusion of prioritised planning means giving up a guaranteed optimal and complete solutions. Interestingly enough, these results show that this strategy might lead to very high benefits in terms of success rates and computation times, at the cost of only very little (or no) suboptimal solutions. The value of guaranteed optimality and completeness must therefore be thoroughly considered in the search for suitable MAPF methods.

Bounded-Horizon PBS

As PBS is an offline solver, a bounded-horizon must be used to make it applicable for this research. Jiaoyang Li et al. [101] studied the use of search windows for the PBS algorithm. He concluded that the conflict detection horizon of the high-level search, and the conflict resolution by priority orders in the low-level search must be limited by w timesteps. Simulation results by Jiaoyang Li et al. [101] indicate that bounded-horizon PBS generate significant faster solutions than bounded-horizon CBS, ECBS and CA*, for the same number of agents and search window length. The method has improved scalability, and performs only little more worse in terms suboptimality compared to bounded-horizon CBS and ECBS.

7.3. Multi-Agent Task Allocation

In classical MAPF, each agent has one task, which is to get to its target. In this research, this is only true for arriving aircraft. Single Taxibots are supposed to exit the taxiways via the nearest gate, to enter the service roads around concourses. This means that Taxibots have multiple goals to chose from. Departing aircraft have the additional objective to uncouple at certain moments and places, meaning that these aircraft are assigned more then one task. In fact, departing aircraft are ought to find an optimal uncoupling location and time for a given runway entrance. This of often referred to as a multi-agent task allocation problem.

This section elaborates on seven possible multi-agent task allocation methods. 7.3.1-7.3.2 studies decoupled task allocation algorithms, in which tasks are assigned independently of path coordination. Ma et al. [115] presented two decoupled algorithms in which task are assigned based on a global shared block of data. Both variants are forms of multi-agent pickup and delivery algorithms and are discussed in 7.3.1. Then, the use of constraint programming is discussed in 7.3.2. Though these three methods provide solid solutions by themselves, they are decoupled from any path finding algorithm, leading to suboptimal global solutions. To schedule the uncoupling operations in such a way that ground delays are minimised as much as possible, a (close to) optimum between path finding and task allocation is desired. Methods in which MAPF and TA are jointly executed are generally able to provide more optimal solutions. Therefore, five coupled methods are discussed in 7.3.2-7.3.7. First, the conflict-based min-cost-flow algorithm, proposed by Hang Ma et al. [116] is discussed in 7.3.3. Then, 7.3.4 and 7.3.5 studies the use of task allocation method combined with (E)CBS for optimal and (bounded) suboptimal solutions. Finally, two methods are proposed that use shortest path variants for the allocation of tasks. As these variants are optimal, they can be used in the individual path algorithms of MAPF solvers such that the solution remains optimal. The two shortest path variants proposed are Bellman-ford and hub labelling, discussed in 7.3.6 and 7.3.7 respectively.

7.3.1. MAPD

Ma et al. [115] presented a decoupled Multi-Agent Pickup and Delivery (MAPD) algorithm, called Token Passing (TP) and Token Passing with Task Swaps (TPTS). In decoupled MAPD algorithms, each agent assigns itself to a task and computes its own collision-free path given global information of other agents. The two types are discussed below.

MAPD-TP: Token Passing

The TP algorithm is based on a shared block of memory in the global environment, called a token. This token contains the path of all agents, their respective current tasks, and a set of open tasks. For each timestep, the algorithm adds new tasks (if any) to the set of open tasks in the token. Subsequently, Any agent that does not have any assigned task (because they have not started, or just finished) requests the token. The algorithm thus sends the token to each agent that requests it, one after the other. To avoid deadlocks, the agent with the token chooses a task from the task set such that no paths of other agents ends in the pickup or delivery location of the task. If there are multiple tasks that fulfil this requirement, the agent assigns itself to the task that is closest to its current location. The task is then removed from the task set, and the agent updates its path in the token. If there is no such task available, then the agent does not assign itself to a task in the current timestep.

MAPD-TPTS: Token Passing with Task Swaps

TPTS is a method developed to make TP more effective. Both methods are simular except for the fact that the set of tasks in the token contains all tasks that have not been executed, instead of only the unassigned tasks. This means that an agent can assign itself to a task that is already assigned to another agents, but not executed by that agent (meaning another agent is still moving to the pickup location of the task). The agent that was previously assigned to the task is unassigned and no longer need to execute it. The agents receives the token from the agent that took over the task such that the agent is able to select another task. Ma et al. [115] experimentally demonstrated that this extension increases the flexibility of the allocation of tasks, leading to higher task throughput (e.g. a more optimal solution), but increased communication and agent movements.

The working principle of TP and TPTS is rather straightforwards, and is often used in autonomous warehouses, airports or harbours [115]. Benda [12] used the TPTS approach to assign Taxibots to aircraft ready for departure. The usability of this method to assign aircraft to uncoupling tasks is questionable, as the objective is not to fulfil all uncoupling task as efficient as possible, but instead to chose one (and only one) optimal task per departing aircraft. The method therefore requires some adaptions to make it usable for this research.

7.3.2. CP

CP like LP or MILP are often used in airport operational research to find optimal gate schedules, taxi routes or to allocate resources or staff to specific tasks [109]. It might therefore be an obvious option to use linear programming for optimal uncoupling allocation in the towing vehicle system. Several studies have been performed in which linear programming is used to compute centralised optimal schedules that are subsequently used as input to the agent-based models [117, 118].

Two main problems arise when using linear programming as input to agent-based models of airport ground movements. The first problems is that the environment of the ABM is dynamic and uncertain (conflicts arising, aircraft are spawning and runways changes). The optimal schedule can thus change over time. The linear program algorithm would need to run at frequent time intervals so that it adapts to the changing circumstances. This might again lead to unacceptable high computation times. The second problem is that optimality is lost when task allocation and path fining is performed independently. This problem is addressed in the next section, where joint MAPF and TA problems are discussed.

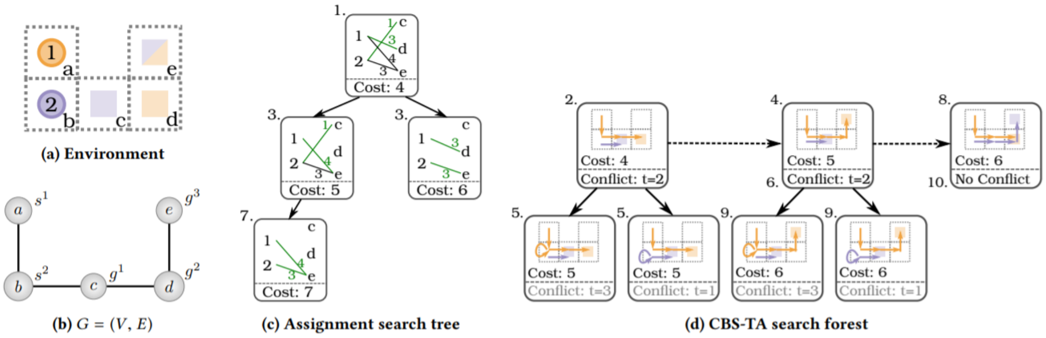
7.3.3. CBM

Hang Ma et al. [116] developed the Conflict-Based Min-cost-flow (CBM) algorithm, an algorithm that combined with conventional CBS, solves task allocation and path finding problems. The combined algorithm uses a hierarchical approach to to solve task assignment and path planning optimally. The algorithm is applicable on teams of agents which are assigned the same a number of target nodes as there are agents in the team. These target nodes ought to be divided amongst the agents and subsequently paths to these targets must be found. The objective of this hybrid solver is to minimise the makespan. The algorithm consists of two levels. The lower level cycle uses the min-cost, max-flow algorithm to assign agents within teams to the given targets and subsequently plans their paths such that there are no conflicts among the agents in the team. Then, the high-level cycle uses CBS to solve conflicts amongst agents in different teams.

The applicability of this algorithm for the objective of this research is questionable. Agents can be subdivided into teams that take off at the same runway. However, there is no single target node, as the aircraft agents are supposed to uncouple at a certain location and subsequently move to another location to take off. Furthermore, it is not true that all target nodes (the uncoupling locations) have to be assigned to an aircraft.

7.3.4. CBS-TA

Hönig et al. [23] introduced Conflict Based Search with Task Allocation (CBS-TA), an algorithm that comes closer to the objective of this research. CBS-TA is a variant of MAPF that seeks an optimal solution in TA and MAPF problems for agents on a general weighted graph while minimising the total sum of cost for all agents. The algorithm is based on the key idea of using a search forest created on demand, rather than a CT. The created forest avoids factorial expansion of possible task assignments. To understand the working principle of CBS-TA, consider Figure 7.15. The environment is shown in (a) and is represented by a network in (b). The search tree (c) visualises the search algorithm in terms of task allocation. Based on the goals assigned in the branches in this search tree, a search forest (d) is created to find optimal paths to these goals.



(a) Environment

(b) $G = (V, E)$

(c) Assignment search tree

(d) CBS-TA search forest

Figure 7.15: Working principle of CBS-TA. The environment (a) is represented as a graph in (b). The conventional CBS conflict tree (c) is replaced by a search forest (d) [23].

Consider agent 1, indicated with yellow, and agent 2, indicated with purple. The agents start at node *a* and *b* respectively. The yellow agent (1) is assigned goal node *d* or *e*, whereas the purple agent (2) is assigned goal node *c* or *e*. Agents are allowed to either move or stop at each discrete timestep, both at the cost of 1. The objective is to minimise the total sum of cost.

The algorithm start simular to CBS. First, the shortest path is found from the start nodes to the closest goal node, resulting in agent 1 going to node *d* and agent 2 to node *c*, with a SIC of 4. The root of the search tree and the root of the first tree in the search forest is created (step 1 & 2). The high-level validation algorithm subsequently finds a conflict between the agents at t = 2 at node *c*. In contrast to the CBS algorithm, the two new branches restrict the allocation of a specific task to an agent, rather than a node at a specific time (step 3). The lowest SIC of the two branches is subsequently selected, which in the example is the left branch. Based on these assigned task, the shortest path for both individual agents is found (step 4). As the cost of this path is higher than the root of the previous tree, the algorithm tries to solve the first conflict by delaying both agents at t = 2 (step 5). As this also results in conflicts at t = 3 and t = 1, the algorithms searches for conflicts in the second three (step 6). This cycles continuous until a solution is found (step 10). The SIC of this solution is automatically the optimal solution. In the example, the solution is found in the root of the third tree in the forest.

This methods comes closer to the objective of this study compared to the CBM algorithm, as individual agents are considered and the sum of individual costs is minimized rather than the total makespan. However, only one goal is considered, while in the situation of the towing system multiple goals should be taken into account. Therefore, it is questionable until what extend this method can be converted such that it is properly applicable in this research.

7.3.5. ECBS-TA

As explained in subsection 7.2.13, ECBS is a bounded suboptimal solver for MAPF problems. It is developed to drastically reduce runtime at the cost of only little suboptimal solutions. Honing et al. [23] implemented this strategy in the CBS-TA method to develop Enhanced Conflict Based Search with Task Allocation (ECBS-TA). ECBS-TA uses suboptimal shortest path methods, rather than A*. Furthermore, the algorithm uses an heuristics to determine the number of expected conflicts used for expansion in the search forest. By keeping track of the lower bound in both the high- and low-search cycle seperatly, ECBS-TA is able to guarantee an outcome within a single suboptimality factor.

7.3.6. BF

Another technique to combine TA with MAPF is based on the decrements algorithm introduced by Bellman [87] as the Bellman-Ford (BF) algorithm. As described in section 7.1, the BF algorithm (unlike Dijkstra's or A*), is able to process graphs with negative weights by exploring all vertices. This capability opens the possibility to force paths over certain locations. In other words, the task allocation is included in the shortest path search which, in combination with MAPF methods can lead to a global optimal solution. This is very useful for the objective of this research, as this method can be used to force aircraft over certain uncoupling locations.

Drawbacks of this mechanism is that the broad exploration of nodes drastically increased algorithm runtime, and is hence unsuitable for large graphs or a large number of agents. Furthermore, as multiple uncoupling locations are considered, and uncoupling needs to be done only once, the graph must be dynamic as the shortest path should only be forced to a single task. Therefore, if a task is selected, other tasks should disappear. Such dynamic graphs requires an additional iterations for each uncoupling point and hence increases runtime even more. It must finally be noted that this method can only be used in combination with MAPF method that use shortest path algorithms for individual path planning.

7.3.7. Hub Labelling

As stated in section 7.1, Gavoille et al. [90] introduced hub labelling to decrease computation time of the shortest path search algorithm by labelling vertices, named hubs, with distances to other vertices in a preprocessing stage. This method requires some preprocessing time and additional data storage to compute and save the labels for hub vertices, but ultimately takes advantage of it by reducing its overall computation time.

On a multi-agent level, this shortest path algorithm can be combined with MAPF solvers like CBS to optimally assign conflict free routes to multiple aircraft. The labels can be used in the lower level search of CBS to quickly generate routes via hubs based on given constraints. For the objective of this research, the hubs can represent the uncoupling locations for outbound movements, where aircraft are supposed to pass by for uncoupling. In the preprocessing stage of the graph, the shortest path of all hubs (e.g. uncoupling locations) to the entrance of the runway must be determined and labelled to each hub. Subsequently, during the search for a shortest path from the gate to the runway via an uncoupling location, the hub system is used to quickly search for shortest paths (using A*) via one of the uncoupling locations. In other words, the shortest path from the gate to the runway entrance is found using one of the pre-defined hubs, and pre-defined shortest paths from the hub to the runway entrance. The implementation of the hub system therefore replaces task allocation, and is, combined with for example CBS an optimal MAPF and TA algorithm.

The complexity of the method is expected to be rather low, as it smartly combines MAPF with a relative simplistic shortest path hub algorithm. Furthermore, the system has reduced overall computation time, which does not scale extensively with the number of agents, as reprocessed data is used, and the number of hubs are fixed. The mechanism is expected to be applicable for the objective of this research, as individual agents are considered, two goals are selected, and the sum of individual costs is minimised. Just like the BF algorithm, this method must be used in combination with a MAPF solver that uses shortest path algorithms for individual path planning.

7.4. Handling Kinematic Constraints and Uncertainties

Airport ground systems are far from typical. Agents have volumes, shapes, and changing kinematics. Multiple agents are allowed to be present on the same edge but must, depending on the type of agent, maintain separation. Edges represent taxiways between intersection, and are not uniform in length. The arrival time of aircraft is uncertain, until couple of minutes in advance. These are just a number of examples of how airport ground systems differ from typical MAPF problem. This section explains how agent-based models can find close to optimal solutions, given aircraft kinematics, uncertainties and unequal edge lengths. 7.4.1 presents the method used in the baseline model, described in section 6.3. 7.4.2 extend on this mechanism by using machine learning elements to improve its performance. In 7.4.3, an algorithm presented by Ma et al. that uses delay probabilities to account for uncertainties is presented. Finally, a post process algorithm is studied in 7.4.4. This algorithm uses any MAPF solver to subsequently take kinematics into account in a post processing state.

7.4.1. Forward Simulation

A general method to take vehicle kinematics into account is by estimating the travel time towards all locations along the shortest path and subsequently use this estimation in MAPF solvers. The CBS mechanism in the baseline model uses such forward estimation. The estimation is based on the taxi distance, taxi speed, and turn speeds for turn angles larger than a certain threshold. This means that interaction with other vehicles, or the emergence of new vehicles is not taken into account. This limitation might lead to situations in which either conflicts are not detected, or redundant conflicts are solved (meaning that conflicts are detected that will not occur). Both events leads to unnecessary rerouting and hence longer taxi times. A proper estimation of these estimated taxi times is therefore very important for the performance of the model.

Recent studied of Polydorou [76] concluded that the forward simulation in the baseline model performed much better than expected. However the inefficiency remains present, especially for conflicts expected to occur in the far future. Two methods can account for this inefficiency. The first method is to only use MAPF solvers for limited timesteps in the future, such as the search window used in 7.3.5-7.3.7. As these algorithms only searches for conflicts in the near future, less redundant conflicts are solved. A second method is to improve the prediction of taxi time by using machine learning. This problem is addressed by S. Polydorou [76], and discussed in the next subsection.

7.4.2. Machine Learning

As described in 7.4.1, to account for agent kinematics and edge lengths, a forward simulation can be used to predict taxi times, such that conflicts can be determined. This means that the detection of conflicts is limited by the performance of the forward simulation. As this forward simulation is suboptimal, S. Polydorou [76] introduced machine learning elements to improve its performance.

He based his strategy on research by Claes [119], who introduced learning methods to better estimate the travel time in multi-agent systems. His approach was to provide road users with traffic information by anticipating on the intentions of other road users. With this information, the road users can decide what routes are more optimal to take. This technique is named delegated multi-agent systems.

Claes defined two types of agents: vehicle and infrastructure agents. Infrastructural agents are placed on nodes between links. Vehicle agents are the road users and travel from node to node over the links. Claes provided the infrastructural agents with an Artificial Neural Network (ANN). This ANN predicts how much time it takes for road users to traverse a certain link adjacent to their node. This prediction is then communicated back to the vehicle agents such that they can anticipate on expected congestion and decide to take different routes. The working principle of ANN is visualised in Figure 7.16. ANN calculates the traversal time at time point t using the intention levels received from road users at and before time t, and the congestion information of downstream infrastructure agents.

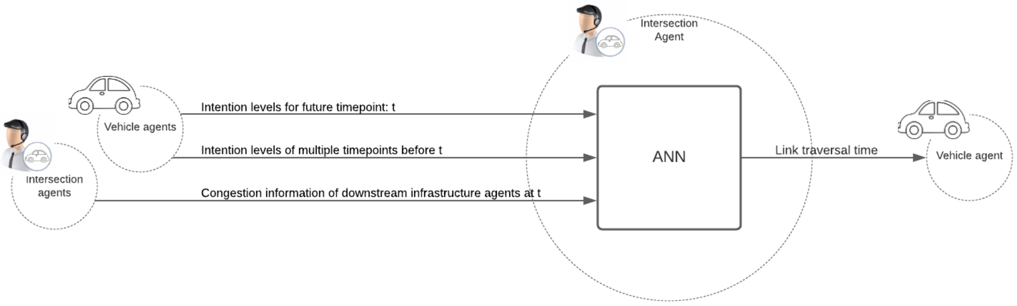

Intersection Agent
Vehicle agents
Intersection agents
Intention levels for future timepoint: t
Intention levels of multiple timepoints before t
Congestion information of downstream infrastructure agents at t
ANN
Link traversal time
Vehicle agent

Figure 7.16: visualisation of working principle artificial neural network.

The difference of this method compared to a more general forward simulation is that this method considers the dynamics between vehicles based on intention information. Using this intention information, the infrastructure agents is able to train the ANN. Furthermore, by monitoring the speed and vehicle inflow, the infrastructure agent can learn a threshold of intention levels after which their link becomes congested.

7.4.3. MAPF-DP

Taxi operations are often carried out imperfectly, resulting in optimal taxi schedules and routes being executed suboptimal. To take this inefficiency into account, Hang Ma et al. [120] introduced Multi-Agent Path Finding with Delay Probability (MAPF-DP). Delay probabilities are included to simulate uncertain behaviour in agent route execution. Just like previously studied MAPF methods, agents are supposed to travel over links from a start node to a goal node. In contrast to classical MAPF, MAPF-DP agents have a certain pre-defined constant probability not to move at each node. The agent can thus either perform a move or a wait action at each node. The MAPF-DP algorithm consist of two elements. The first element is to find the shortest path for each agent given a start and goal node and a sequence of wait and move actions. The second element controls the move and stop commands along the path of each agent such that no conflict occur during plan execution. The MAPF-DP algorithm proposed by Hang Ma et al. distinguishes three kinds of conflict, namely vertex collision (type a), edge collision (type b) and swapping conflicts (type e). Experimental results show that the model performs best in open environments. Increasing the number of objects in an open environment (where 0% is no objects and 100% is objects only) linearly increases the runtime and avoided conflicts for low object concentration. Increasing the number of agents almost exponentially increases runtime and conflicts.

7.4.4. MAPF-POST

Hönig et al [24] introduced Multi-Agent Path Finding with POST processing (MAPF-POST), an algorithm that uses a temporal graph to post process the outcome of MAPF algorithms, to develop an execution plan which takes uncertainties (e.g. imperfect plan executions), aircraft kinematic and variable edge lengths into account. The algorithm accounts for maximum velocity and turn speeds, and provides a safety distance between agents. Furthermore, the algorithm use slack variables to absorb imperfect execution of plans and in certain cases, prevent time-consuming replanning. The working principle can most easily be explained by the example of which the graph is given in Figure 7.17.

Table 7.2: Preliminary plan execution of the example in Figure 7.17.

Agent	t=1	t=2	t=3	t=4
1	A → B	B → C	C → D	D → E
2	B → C	C → F	F → C	C → D

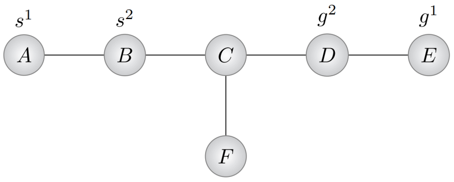


Figure 7.17: An example used in the Multi-Agent Path Finding with POST processing (MAPF-POST) algorithm of Hönig et al [24].

Consider as usual s^1, s^2 and g^1, g^2 as the source and goal vertex of agent 1 and 2. The agents are constraint with velocity limits of $1/4m/s$ and $1/16m/s$ respectively. First, the execution plan given in Table 7.2 is generated with any MAPF solver by assuming uniform edge lengths and constant, equal velocities. As this plan is nearly impossible to execute due to different kinematics, edge lengths and imperfect executions, the outcome is post-processed to develop a plan-execution schedule that accounts for these factors. The development of the plan-execution schedule is done by using a Simple Temporal Network (STN), given in Figure 7.18 for this example. In this example, X_s and X_f are the source and final state, respectively. The big vertices corresponds to an agent (top number) being at a certain node (the letter) at a certain time (lower number). These vertices are named events are are taken from the preliminary MAPF solution, given in Table 7.2. The job from this temporal network is to assign a more specific time to each event, such that factors as kinematics, edge length and imperfect executions are accounted for. To do so, safety markers (indicated by the smaller circles) are added before and after each main event (larger circles), to guarantee a safety distance between agents. The relative distance of these safety markers is set using an input parameter. Each edge in the network is allocated an STN bound, indicated with $[LB(e), UB(e)]$. The Lower Bound of edge e indicates that the next event must occur no earlier than $LB(s)$ timesteps after the previous event. Likewise, the upper bound of e indicates that the next even must occur no later than $UB(e)$ timesteps after the previous event. Note that in this example, A_0^1 and B_0^2 occur immediately after start. The values of the lower and upper bounds are based on the relative agent speeds. In the example given, the lengths between the edges is assumed to be 1, and the velocity limits are $1/4m/s$ and $1/16m/s$ for the first and second agent respectively. Hence, the cumulative travel time between two main events (bigger circles) are 4 and 16 for agent 1 and 2.

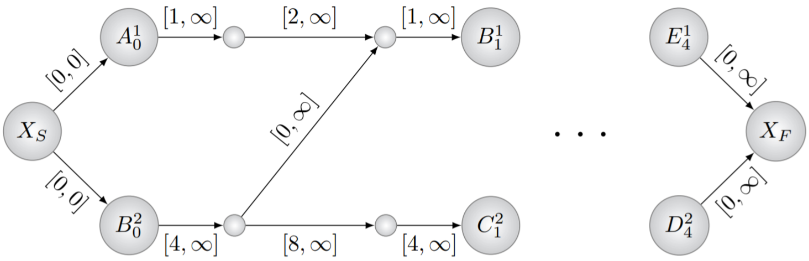


Figure 7.18: Simple Temporal Network (STN) for the plan execution in Table 7.2.

An execution schedule can now be determined by satisfying all edge conditions and $T(X_S) = 0$. Hönig et al. introduced several method for different optimization criteria. The first method is to convert the STN graph to a distance graph with opposite negative edge weights for the lower bound, and subsequently apply Bellman-Fort, discussed in subsection 7.1.1, to find the minimum cost-path. Another method is to use linear programming to minimise the cumulative sum of all arrival times by constraining all lower and upper bound edge conditions and $T(X_S) = 0$. Linear programming can also be used to minimise the total makespan, by slightly adjusting the objective function. Finally, Hönig et al. showed that linear programming can be used to maximise the global minimum velocity limit.

Hönig et al claims that MAPF-POST runs in polynomial time, and evaluated its performance on actual robots. One of the main drawbacks of this system is that the preliminary plan execution calculated by a MAPF solver remains fixed, and thus does not take kinematics or edge lengths into account. This especially leads to inefficiencies when there are big differences in agent velocities or edge lengths.

7.5. Comparison and Trade-off

This chapter started with a study in shortest path algorithms, then discussed MAPF and TA methods, and subsequently examined how kinematics and uncertainties can be taken into account in airport ground systems. This section presents a trade-off between all different coordination mechanisms based on the conclusion made throughout this chapter, and the objective set in section 5.6. Because the performances of these mechanisms are interdependent, decisions must be made by looking at all mechanisms simultaneously. Therefore, the outcome of the trade-offs between mechanisms are weighted concurrently, such that a combination of methods can ultimately be selected. Criteria are established in subsection 7.5.1 and scores are given in subsection 7.5.2. Conclusions are drawn for all mechanisms simultaneously in subsection 7.5.3.

7.5.1. Trade-off Criteria

Coordination mechanisms are evaluated based on the ease of implementation, performance and realism. These categories are subdivided into two or three lower level criteria. All criteria are ranked from 1 (low performance) to 5 (high performance).

Implementation

This category represents the ease of implementation of the mechanism. Two criteria are considered: complexity and availability & support. The complexity of the mechanism relates to the required amount of programming, steps, links and variables. Mechanisms that require a complex architecture or intense programming become less understandable and reproducible. Furthermore, The undesirable risk of not being able to finish this research within the given timespan of 9 months increases with higher complexity. Availability & support refers to the online and offline resources and tools available to support the implementation of the mechanism. For online support, this includes the available literature, sample codes and online assistance. The offline support refers to the knowledge and support available within the research group of this study.

Performance

This category evaluates the mechanism based on its performance. The method must be proven to work and to produce reliable and valid results. Three criteria are considered: scalability, optimality and success rate. Scalability refers to the relation between the number of agents and the computation time. A higher scalability means that an increased number of agents only leads to minor shifts in computation time. Generally, a linear relation between the number of agents and computation time is considered as good, whereas an exponential or logarithmic relation is more worse. Optimality refers to the ability of the mechanism to find optimal, or very close to optimal solutions. Finally, the success rate refers to the number of agents that are successfully coordinated without conflicts to their goal node in the given timespan. To make a proper comparison in terms of success rates between the mechanisms, a dense environment with many corridors is considered, if possible.

Realism

The final category assesses the mechanisms based on the realism in the final model. This is divided into two criteria: assumptions and feasibility. The model becomes less realistic if many operational assumptions are implemented. Such assumption can be related to vehicle kinematics like accelerations, turns and movements but also vehicle dimensions and plan execution. Feasibility refers to the applicability of the mechanism to the considered situation. In this research, the layout of AAS is considered. This layout is represented by a graph with nodes and links with only limited diversion possibilities. This means that algorithms efficient in open spaces or algorithms that require many space for diversion are less suitable for this research.

7.5.2. Trade-off Cooperative Coordination Mechanisms

This section provides the trade-off between all cooperative coordination mechanism suitable for the objective of this research, set in section 5.6. In essence, three decisions need to be made: a MAPF method, a TA method, and a way to account for kinematics and uncertainties must be chosen. It must be noted that combining optimal MAPF and TA algorithms does not necessary lead to a global optimal solution. Likewise, some strategies to account for kinematics and uncertainties are not applicable on some MAPF methods. Therefore, a trade-off is performed between all different mechanism, but conclusions are drawn simultaneously. The latter is done in subsection 7.5.3.

Many MAPF methods are discussed in this chapter. An overview of all methods applicable for this research is given in Table 7.3. In this table, the first and second column indicate the section and method respectively. The third column states whether the method is considered to be optimal. In theory, an optimal method is always favourable. However, optimal solvers generally have higher computation times as a larger search spaces are explored. Furthermore, the value of optimal solvers is lower in online systems that contain uncertain and dynamic situations as unaccounted behaviour makes the execution of plans different than expected. In fact, Jiri Svancara et al. [121] demonstrated that it is not possible to create an optimal solver for online MAPF problems. Therefore, solvers that do not guarantee an optimal solution but approaches it very good are just as valuable in uncertain systems as (offline) optimal solvers made online by recursive planning. The forth column indicate whether a certain solution quality can be guaranteed, meaning that the outcome is subjected to a given suboptimality bound. Clearly, this is always the case for optimal solver. Decentralized solvers can never guarantee this, as plans are generated in a sequence, rather than at once. The fifth column specifies if a mechanism is complete. Complete mechanism always converge to a solution, if one exists. It is questionable how valuable this property is in airport systems, as agents are always allowed to reduce speed and agents disappear from the network after arrival. Given that, all problems in an airport systems are complete, as long as deadlocks are avoided. Column six states whether mechanisms are online or offline. Online mechanism are those that contain a plan-move cycle, and in offline mechanism, all computations are done in advance after which the complete plan is executed. In uncertain airport systems (aircraft take-off and land at unpredictable times), a plan-move cycles must be used. Offline mechanism must therefore be used in an online fashion by recursive planning. However, this is generally computationally more expensive. The last column indicates whether a method is centralised or decentralized. Centralised method compute all paths simultaneously, decentralized methods don't. The advantage is that centralized methods open the possibility for optimal solutions, but are generally computationally more expensive. It must be noted that, comparing the MAPF solvers based on the black-and-white data in Table 7.3 can give a false picture of its suitability for this research, as in fact many aspects are involves. Therefore, a more in-depth trade-off is performed, not only based on the performance, but also on the ease of implementation and realism, in the next tables.

Table 7.3: Overview of all MAPF methods.

Section	Method	Optimal	Guaranteed quality	Complete	Offline/online	(de)Centralized
7.2.2	**LRA***	No	No	No	Online	Decentralized
7.2.5	**WHCA***	No	No	No	Online for w –>0. Offline for w–>t	Decentralized
7.2.6	**CO-WHCA***	No	No	No	Online for w –>0. Offline for w–>t	Decentralized
7.2.7	**CO-WHCA*P**	No	No	No	Online for w –>0. Offline for w–>t	Decentralized
7.2.8	**MAPP**	No	No	If sliceble	Online	Decentralized
7.2.9	**Push-and-swap**	No	No	Yes	Online	Decentralized
7.2.10	**Push-and-rotate**	No	No	Yes	Online	Decentralized
7.2.11	**A* in joint graphs**	Yes	Yes	Yes	Offline (search window required)	Centralized
7.2.12	**CP**	Yes	Yes	Yes	Offline (search window required)	Centralized
7.2.13	**CBS**	Yes	Yes	Yes	Offline (search window required)	Centralized
7.2.14	**GCBS**	No	No	Yes	Offline (search window required)	Centralized
7.2.14	**BCBS/ECBS**	No	Yes	Yes	Offline (search window required)	Centralized
7.2.15	**CBS+HWY**	No	No	No	Offline (search window required)	Centralized
7.2.16	**CBSw/P**	No	No	No	Offline (search window required)	Centralized
7.2.17	**PBS**	No	No	No	Offline (search window required)	Centralized

The scores of all MAPF methods for the established criteria are given in Table 7.4 and Table 7.5. ECBS and BCBS have been combined into one, as the mechanisms are very simular. LRA* and WHCA* are classified as the least complex, as the matter is quit straightforward, and limited search space is needed. As for Availability & support, CP and CBS score highest because of the assistance and knowledge available in the faculty and by other student. When looking at performance, the LRA*, GCBS and PBS performs best in scalability due to their limited (for LRA*) and efficient (for GCBS and PBS) planning and search. The highest score in the optimality criterion is only given to those mechanism that are validated as optimal and CBSw/P, as the latter is proven to approach optimality extremely close. The highest scores in success rates are given to CO-WHCA*P and PBS, as studies clearly demonstrated their improved performance over their variants. Finally, only PBS is ranked highest for feasibility because its ability to combine prioritise planning with efficient conflict resolution and its ability to allow initial priorities. When looking at the main categories, LRA*, MAPP, Push-and-swap and Push-and-rotate are all unfeasible because of its low realism, due to the complex required aircraft manoeuvres, unsuitable graph structures or intensive replanning. A* in joint graphs and CP are both concluded to be infeasible due to its exploding number of variables or graph structures for many agents, which resulted in extremely low scalability scores. This leaves two categories: A WHCA* or CBS, or one of their variants. By comparing their scores, one can observe that PBS has scored highest in both the performance and realism category. The scores on implementation is not outstanding high, but sufficiently high enough to still obtain the highest number of total points. Further conclusions regarding the most suitable MAPF method will be drawn simultaneously with other methods, discussed in subsection 7.5.3.

Table 7.4: Trade-off between multi-agent path finding algorithms, part 1.

	LRA*	WHCA*	CO-WHCA*	CO-WHCA*P	MAPP	Push-and-swap	Push-and-rotate
Implementation							
Complexity	5	5	4	3	3	3	3
Availability & support	3	4	3	3	2	2	2
Performance							
Scalability	5	3	4	3	2	4	4
Optimality	1	3	3	3	2	3	3
Success rate	1	3	4	5	3	4	4
Realism							
Assumptions	1	3	4	3	2	1	1
Feasibility	1	3	4	4	1	1	1
Total	17	24	26	24	15	18	18

Table 7.5: Trade-off between multi-agent path finding algorithms, part 2.

	A* in joint graphs	CP	CBS	GCBS	BCBS/ECBS	CBS+HWY	CBSw/p	PBS
Implementation								
Complexity	1	2	3	4	4	3	2	3
Availability & support	3	5	5	4	4	4	3	3
Performance								
Scalability	1	1	3	5	4	4	4	5
Optimality	5	5	5	3	4	4	5	4
Success rate	1	3	3	4	4	4	4	5
Realism								
Assumptions	2	1	3	3	3	3	4	4
Feasibility	2	2	3	3	3	2	4	5
Total	15	19	25	26	24	24	26	29

Scores are given to all studied TA mechanism in Table 7.6. Only those mechanism that result in optimal solutions when combination with another MAPF algorithm are scored maximum in optimality. The success rate category is not applicable on TA algorithms, as conflict resolution is not involved. TP, TPTS, CP and CBM all score relatively low on realism. This is mainly because of assumptions made in the mechanisms that are in conflict with the task allocation objective of this research, such as agent grouping, task fulfilment and target nodes. As for CP, it is expected that many assumptions are needed to limit the number of variables and constraints. BF and Hub Labelling score highest in realism as these methods do not require any additional assumptions, rather than small changes in the shortest path algorithm. The BF algorithm however has a low total score because of its costly implementation and weak scalability. This leaves (E)CBS-TA and Hub Labelling. Hub Labelling then scores slightly higher in all categories as a result of the ease of implementation, its applicability for this research, and its high performance.

Table 7.6: Trade-off between task allocation algorithms.

	MAPD-TP	MAPD-TPTS	CP	CBM	CBS-TA	ECBS-TA	BF	Hub Labelling
Implementation								
Complexity	5	4	1	2	3	4	1	4
Availability & support	3	4	5	2	2	2	2	3
Performance								
Scalability	4	3	1	5	3	4	1	5
Optimality	2	4	5	4	5	4	5	5
Success rate	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Realism								
Assumptions	3	2	1	1	4	4	4	4
Feasibility	2	2	2	1	3	3	4	4
Total	19	19	15	15	20	21	17	25

In section 7.4, four methods to account for kinematics, uncertainties and nonuniform edge lengths where discussed. Their scores are given in Table 7.7. This table does not present a trade-off, as the methods fulfil different function, or must be combined anyway. Rather, this table provides scores to indicate what mechanisms might be useful in this research.

Table 7.7: Scores to kinematic and uncertainty methods.

	Forward Simulation	Machine Learning	MAPF-DP	MAPF-POST
Implementation				
Complexity	5	1	3	2
Availability & support	5	3	3	4
Performance				
Scalability	4	2	3	1
Optimality	2	5	1	4
Success rate	n/a	n/a	n/a	n/a
Realism				
Assumptions	3	2	2	3
Feasibility	3	3	1	2
Total	22	16	13	16

Machine learning has a low score in complexity due to its required intensive programming. A forward simulation can be implemented using simple kinematic equations, and is therefore estimated to be the least complex. The forward simulation is also used in the baseline model, and scores therefore highest in availability & support. Also MAPF-POST scores high in availability & support, as CP is used as main solver in this method. Machine learning scores rather low in scalability, as the computation time of the method can get very long for large number of agents. MAPF-POST also scores low in scalability, as the search space increases faster with the number of agents compared to the other methods, and CP is (usually) used as solver. The forward simulation scores low in optimality, as it only takes nonuniform edge lengths and vehicle kinematics into account, and not uncertainties. MAPF-DP scores low because of its unnecessary added delays. MAPF-POST has a lower score due to its assumption to fix the preliminary plan execution meaning that only small changes in edge lengths or velocities are accounted for. Finally, MAPF-DP scores lowest in feasibility because the algorithm tends to perform better in open environments, rather than complex networks such as airport systems.

7.5.3. Conclusion in Cooperative Coordination Mechanisms

Considering the scores given to MAPF methods, it can be concluded that either a WHCA* or CBS variant must be used. The high scores given to WHCA* and its variants can explained by the efficient computations, effective search window tool, and the possibility to prioritise agents. Especially the latter is expected to be a big advantage for this research, as many types of agents with different interests are expected to be present at the taxiways simultaneously. The drawback of the WHCA* variants is that no active search is performed to determine the optimal agent priority order (hence the lower score in optimality). As the objective of this research is to mitigate ground delays as much as possible, this is something that must be included in the determination of paths. CO-WHCA*P (partially) did this by its prioritization mechanism. However, the method scores lower on realism due to its altered move actions. PBS also included this. In fact, from all the CBS variants, PBS is in many ways simular to the WHCA* variants. It is also very efficient, uses CA*/HCA* in the lower level cycle to find individual paths, and allows for initial priority order. The critical difference is that a depth-first search is used to explore the solution space to quickly find a (close to) optimal priority [22]. Given these characteristics, the similarities with variants, and the scores given to the established criteria, it is concluded that PBS is the most suitable MAPF method for this research.

As PBS is an offline solver, recursive planning is needed to make the algorithm usable for the online problem of this research. The search window of WHCA* (consisting of a window length w, and plan frequency h) is proven to be a very effective tool to prevent solving redundant conflicts and reduce computations [17]. This tool will be used in PBS to make the algorithm online, and applicable for this research. This combination is studied before by Jiaoyang Li et al. [101] and introduced as Bounded-Horizon PBS. The fact tat no optimal solution can be guaranteed by a Bounded-Horizon PBS is assumed to be a negligible issue, as the simulation considers an online problem, meaning that optimal solutions cannot be found anyway [121].

As for the task allocation mechanism, the hub labelling method wins due to its high scores in all categories. The main advantage of this method is its limited required algorithm adjustments, optimal outcomes, and small computations. The other task allocation options all score rather low on realism. This is because most algorithms make different assumptions in objectives, tasks and agent characteristics. Examples are grouped agents, multiple or compulsory groups of targets or different objectives. These differences require drastic changes in the algorithm to make it applicable for this research, hence increasing complexity and required assumptions. Only BF scores high on realism, but has a lower score for implementation and is computationally more complex. Therefore, hub labelling can be seen as a clear winner.

Finally, consider the scores given to method usable to account for uncertainties and kinematics. Forward simulation has by far the highest score. The only low score is for optimality, as the method does not account for agent interactions and hence does not take uncertainties into account. This can lead to inaccurate conflict times, especially for conflicts expected to occur in the far future. However, as a bounded horizon is used in PBS, the search window size w can be set such that this inefficiency is avoided as much as possible. The bounded horizon also solves the problem that uncertainties are not taken into account. By lowering the replan frequency h, the adaptive behaviour of agents is increased, meaning that uncertainties are dealt with better. The determination of the input parameters for the search window size and replan frequency (w and h) is therefore an important objective in this study. Given these conclusion, it is concluded that a forward simulation remains the main method to account for agent kinematics and edge lengths.

To conclude, a Bounded-Horizon PBS is used to plan conflicts free paths for all agents. HCA* is used in the lower level cycle of PBS to plan individual paths using an abstract heuristic. The Hub Labelling mechanism is used to assign outbound aircraft movements to uncoupling locations. Forward simulation is used to account for agent kinematics and changing taxiway lengths.

The literature review is now finished. The next chapter proposes the thesis research by means of an objective and several questions. In chapter 9, the research strategy, scope, planning and expected results are discussed.

8 Research Proposal

This MSc literature study started with a review in current airport taxi operations and its environmental impact. Subsequently, alternative taxi operations and systems have been studied. It was then concluded that the use of external towing vehicles is the most promising method to make airport ground movements greener and more sustainable. The introduction of this novel concept drastically changes operations mainly due to different uncoupling procedures and the presence of additional vehicles. It was found that these changed operations might lead to additional ground delays and reduced airport capacity. Autonomous coordination might be able to mitigate this problem. Hence, the final chapters of this report were devoted to study previous research in agent-based airport operations and to identify mechanism usable for autonomous coordination.

This section presents the research proposal for this MSc thesis. The proposal is specified through a research objective and several research questions in section 8.1 and section 8.2, respectively. The next chapter contains and elaborate discussion related to the research strategy, scope, outcome and planning.

8.1. Research Objective

It is expected that the alternative taxi operations of Taxibot results in changed traffic flows on taxiways and aprons. Especially near runway entrances, the traffic flow is expected to be denser and more complex compared to conventional operations. It is because of this reason that a simulation is developed to study these changed operations. This simulation is simultaneously a representation of an autonomous system, that can be used in future to manage the changed (more complex) alternative operations. To guarantee optimal airport capacity and efficient airport operations, cooperative coordination mechanisms are used to plan and coordinate traffic routes and uncoupling procedures. The aim of this research is thus to develop a simulation in which ground delays are mitigated as much as possible, such that airport capacity is persevered, and smooth operations guaranteed. The remaining ground delays are thereby the main performance indicators of the simulation. Hence, the objective of this research is formulated as follows:

> *"To analyze how well autonomous coordination mitigates ground delays in an airport outbound towing system by simulating the alternative taxi operations in conventional infrastructure."*

This research is focused on Taxibot movements in conventional infrastructure. As explained in section 5.5, some service roads are not suitable for Taxibot operations. The assumption that these roads are broadened and suitable for Taxibots is rejected on purpose. The objective of this research is to get a clear understanding of how autonomous coordination can contribute towards the implementation of towing systems given the current state of the airport. Making such huge infrastructural assumptions would thus give a distorted picture of the power of autonomy. A possible outcome can therefore be that autonomous coordination cannot mitigate ground delays sufficiently, and that infrastructural adjustments are needed. The additional result is than that the simulation can locate bottlenecks to conclude at what locations adjustments are needed.

The research assumes that only outbound movements are towed. As described in section 5.5, the use of SE taxiing, the lower environmental impact of inbound movements, and the operational complexity of inbound towing underlie this decision.

Several research projects in the feasibility of external towing systems at airports are previously done. This research is unique in the sense that it is completely focused on congestion and delay management resulting from alternative operations. Knowledge regarding these expected delays and the effects of different mitigation methods is generated and can be used in towing system feasibility studies by airports, ground handlers and airline. Therefore, the outcome of this research is relevant in the sense that the effectiveness of towing systems like Taxibot at big hub airports such as AAS depends on whether the expected delays can be adequately mitigated, to preserve airport capacity and ensure safe and fast operations.

Research question are established in section 8.2 to divided the work needed to fulfil the above stated objective. A complete strategy to fulfil the mentioned objective is given in section 9.1. Specific statements related to the scope of the research in terms of assumptions, simulation constants and independent variables can be found in section 9.2. The expected results in terms of ground delays and mitigation options are discussed in section 9.3. Also, the relevance of the outcome is addressed in this section. Finally, a research planning, contain the sub-goals and work packages necessary to fulfil the objective is given in section 9.4. Given the above objective, the required knowledge, programming skill, available hardware, and support within To70 and the Aerospace Engineering (AE)-faculty staff, it is estimated that it is possible to develop a sufficiently successful model to reach the objective and answer the research question in the given time span of 9 months.

8.2. Research Questions

As stated in section 8.1, the main objective of this research is to study how well autonomous coordination mitigates ground delays in an outbound Taxibot system. The main research question is therefore formulated as follows:

> *What additional ground delays can be expected in an autonomously coordinated distributed multi-agent engine-off outbound towing system at AAS?*

To answer the main research question, nine lower level key questions are established, each supported by several sub-questions. Together, these questions provide an answer to the main research question. The main research question, and several key questions are interdependent according to Figure 8.1. As one can observe from this figure, key question seven and eight combined provide an answer to the main research question, as they consider the conventional ground delays and the outcome of the simulation respectively. To establish representative sample scenarios and define ground delays, key question one and two are established. The first key question specifies what flight data input must be used, what day of operations is best to simulate, and what RMOs are considered. Key question two specifies the main performance indicator of the model, ground delay. Key question three and four are imposed to better identify the new operational condition in terms of additional vehicles and uncoupling on the taxiways. Key question five and six subsequently consider the mechanisms to mitigate ground delays in the autonomous system with regard to congestion management techniques and cooperative coordination mechanisms to plan and coordinate the uncoupling procedures and ground movements. noticeably, the latter four key questions are needed to properly establish the simulation. The last key question, number nine, is imposed to analyses the performance of the simulations in terms of safety and emergent behaviour.

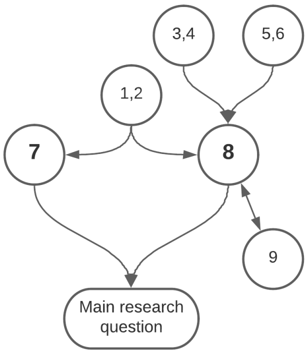


Figure 8.1: Relation research question and key questions

1. What sample scenario is used?
   (a) Which day of operation is modelled?
   (b) What type of flights are considered?
   (c) What flight input data can be obtained from To70, TU Delft or Schiphol?

   (d) What flight input data is most suitable and representative for this research and the developed model?
   (e) What RMOs are considered?

2. How can aircraft ground delays be specified?
   (a) How can ground delays be classified?
   (b) What types of ground delay are relevant for this research?
   (c) What types of ground delay are important in terms of airport efficiency and capacity?

3. How must tow vehicles behave in the airport ground system?
   (a) Do all uncoupling locations have possible routes for the towing vehicles after uncoupling?
   (b) What type of conflicts can occur if towing vehicles drive on taxiways?
   (c) What type of conflicts can occur if towing vehicles drive on service roads?
   (d) What service roads are suitable, or easily adjustable, for the use of towing vehicles?

4. What are the technological and operational specifications of the uncoupling procedure of Taxibot?
   (a) What is the minimum realistic uncoupling duration of Taxibot?
   (b) What are possible uncoupling locations?
      i. What is the minimum space necessary to complete the uncoupling of towing vehicles?
      ii. Are all taxiway roads suitable for uncoupling in a fully autonomous towing system if enough space is available?
      iii. What other limitations prevent the use of uncoupling locations?

5. How can congestion management methods be used to minimise aircraft taxi delays in an autonomous outbound towing system?
   (a) What existing congestion management technique is most applicable to an autonomous engine-off towing system?
   (b) What existing congestion management technique is most effective for an autonomous engine-off towing system?
   (c) How can existing congestion management techniques be adapted in such a way that they are applicable on the autonomous towing system?
   (d) What other effects do these congestion management techniques have on the operations of the airport?

6. What are the best cooperative coordination mechanisms to plan and schedule uncoupling procedures and ground movements such that ground delays are minimised?
   (a) What MAPF and task allocation methods are most suitable for to coordinate the uncoupling procedures of towing vehicles on taxiways?
   (b) How can task allocation and MAPF be combined without losing optimality?
   (c) How effective can conflict resolution methods be exploited for the goal of this research?
   (d) What strategies can be used to guide the flow of traffic around uncoupling events?

7. How much ground delays of what type occur in conventional ground movements?
   (a) What data can be obtained from AAS and To70?
   (b) How where ground delays previously tracked?
   (c) What ground delays form a strong foundation in the comparison with ground delays obtained from the simulation?

8. How much ground delays of what type occur in the outbound towing simulation?

   (a) What additional agents and interactions are needed?
   (b) How should the environment be altered with respect to the conventional model?
   (c) How can the congestion management and coordination techniques be implemented?

9. Is the performance of the simulation satisfactory?
   (a) Does the resulting simulation have an acceptable level of conflicts?
      i. What is a sufficient level of safety in an autonomous agent-based airport ground model?
      ii. How is a conflict between two vehicles defined?
   (b) Does the simulation produces results within acceptable time limits?
      i. What is an acceptable time limit?
      ii. How long does it take to simulate a full day of operation?
   (c) Are all coordination methods to mitigate ground delays implemented successfully?
      i. How do the coordination methods perform?
      ii. Are there any obvious improvements to reduce ground delays further?

9 Research Methodology

This chapter covers the research methodologies. In section 9.1, the strategy to achieve the objective and complete the research successfully is discussed. Then, section 9.2 states the scope of the research with regards to assumptions, simulation constant and independent variables. The expected results, and there relevance is outlined in section 9.3. The chapter concludes with a planning containing all work packages for this research.

9.1. Research Strategy

The objective of this research is to determine the effects on ground delays when conventional outbound taxi movements are powered by Taxibots, and all operations are coordinated autonomously. A distributed multi-agent simulation is developed to study this system, and to obtain ground delays for the comparison with historic data. Cooperative coordination techniques to plan paths and schedule uncoupling tasks form the core of the model.

To adhere to a more structured modelling approach, the mental simulation cycle for sociotechnical systems in agent-based modelling, developed by Stroeve & Everdij [25], is used. This method is originally developed to improve resilience in agent based systems and is therefore slightly adjusted to make it applicable for this research. This mental simulation consists of five main steps (the higher level cycle) and three sub-steps (the lower level cycle). The steps are visualised in Figure 9.1.

In the higher level cycle, a strong foundation for the objective and simulation is established. These steps are critical for the performance and analysis of the model. The higher level cycles begins with step 0, in which the scope and the objective of the study is formulated. This is previously done in the literature study and documented in this report. In step 1, the baseline operations are established. This means that conventional ground delays are obtained, studied and classified. Also, a sample scenario is established. This step is related to key question 1, 2. Step 2 addresses the main disturbances to the conventional taxi operations. These disturbances include the uncoupling operations and additional vehicles on taxiways. They are related to key question 3 & 4 respectively. In step 3, the capacity of the model to adapt for the new towing vehicle situation is analysed through the lower level cycle. This analysis is done by looking at the behaviour of the models in terms safety, delays and computation efficiency, related to key question 9. The lower level cycle keeps running as long as the answers given to the sub- key questions are not sufficient or a further reduction in ground delay is expected to be achievable. Finally, in step 4, the performance of the model is tested to conclude the research. If the ground delays are not mitigated sufficiently, the research can be extended by completing the cycle with additional measures. These measures could (for example) include assumptions related to infrastructure or the number of outbound tows.

The lower level cycle is done in the third step and includes everything related to the development and analysis of the actual agent-based model. In step 3a, Strategies that can be used to mitigate ground delays in the alternative situation are identified. This steps relates to key question 5 & 6. In some situations, this identification can be sufficient for the analysis in step 3. However, due to the complexity of the models, it is expected that this is not true for this research. Therefore, the next step is 3b. In 3b, the qualitative models are developed in

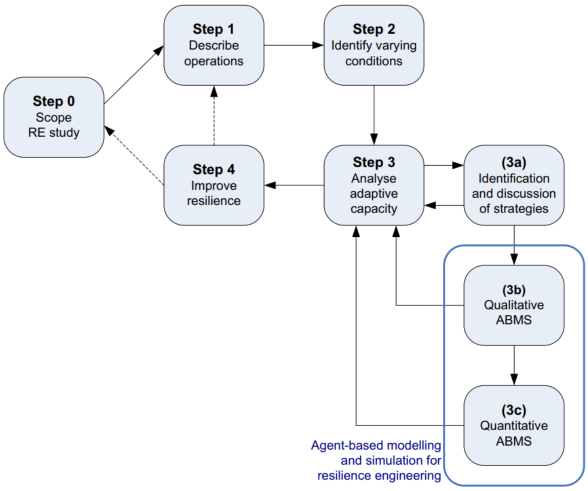


Figure 9.1: Steps in an agent-based modelling simulation [25].

which the agents, environments and interactions are described and formulated. This relates to key question 8. In some situations, these qualitative models can be sufficient for the analysis in step 3. However, with the same reasoning as before, it is expected that this is not true for this research. Therefore, the next step is 3c. In this final step, the qualitative agent-based models are further formalised into a complete mathematical model. This step includes all the programming and simulation work.

Given the above cycle and the linked research question, it is estimated that this study is doable in the given time span of 9 months. Furthermore, as quickly mentioned in section 8.1, The required programming skills, knowledge, hardware and support from To70 and the faculty are sufficient for the objective.

9.2. Research Scope

This main experimental set-up is the construction of an agent-based model. Chapter 6 studied these models and concluded that it is a very effective and efficient technique to model and study airport ground systems. A long-term research line in autonomous multi-agent ground movements at To70 and the Delft University of Technology has led to a model that is suitable as baseline. This model is developed in Python, a high-level object oriented general-purpose programming language [84]. Given Python's strong capability, the baseline model, supportive tools, and the available background knowledge, python is chosen as the main programming language of the model. This chapter summarises the set-up of the experiment in terms of assumptions, parameters and variables.

9.2.1. Assumptions

This section present all assumption made in this research.

1. General.

   (a) The model simulates an autonomous system of aircraft and towing vehicles. As a result, a driverless solution for towing vehicles is presented, meaning that no procedures are taken into account to protect truck drivers from hazardous situations and Taxibot plans are always executed perfectly.
   (b) The simulation assumes that all stakeholders are familiar with the system and execute their tasks perfectly. This also means that there are no deviations, mistakes or other errors in the execution of plans of aircraft or ATC.
   (c) Meteorological conditions are neglected.

2. Infrastructure.

   (a) The ground system of AAS is used. AAS has recently run multiple trials with Taxibots. The airport is very interested in the scalability of the Taxibot system, and the effect it has on ground delays [42].

Furthermore, as AAS is an international hub airport with a complex ground system, it is expected to be a suitable sample for other airports.

(b) The ground system is modelled as a graph with vertices and edges. The edges represent taxiways, runway pieces, service roads and aprons. The vertices represent intersections, gates and runways entrances. An example of such graphs is given in Figure 6.5.
(c) Gates are modelled as meta-gates. 22 meta-gates are present, representing combinations of gates.
(d) Service roads are inaccessible for Taxibots. The only service roads accessible are those around concourses, and those indicated in Figure 9.2a and Figure 9.2d.

3. Aircraft operations.

   (a) All departing aircraft are towed to the runway, and no arriving aircraft is towed to the gate. As described in section 5.5, the use of SE taxiing, the lower environmental impact of inbound movements, and the operational complexity of inbound towing underlie this decision.
   (b) Only Commercial air transport is considered. Two type of aircraft are included: NB aircraft (e.g. B737/A320) and WB aircraft (e.g. B747/B777).
   (c) Aircraft and towed aircraft must travel via taxiway or apron edges from a source to a goal node.
   (d) Departing aircraft must travel via one of the uncoupling location to the runway entrance.
   (e) The dimensions of aircraft are neglected, and aircraft have constant kinematic properties, given in subsection 9.2.2.
   (f) Engine warm up is performed during Taxibotting and is possible under all circumstances.

4. Taxibot operations.

   (a) Two type of towing vehicle is considered: a NB and WB Taxibot. The NB Taxibot is suitable for B737/A320 and WB Taxibot is suitable for B747/B777 type aircraft.
   (b) After uncoupling, towing vehicles must travel via taxiways, aprons and (limited) service roads from a source to a goal vertex.
   (c) Taxibots are not allowed to cross active runways.
   (d) The source vertex is always the uncoupling location, and the goal vertex is the closest meta-gate.
   (e) Taxibots leave the taxiway system via the closest meta-gate. There is always at least one gate available for the Taxibot to leave the apron, and enter the service roads around concourses.
   (f) Taxibots on service roads around concourses are not modelled.
   (g) Unlimited Taxibots are available.
   (h) The dimensions of towing vehicles are neglected, and towing vehicles have constant kinematic properties, given in subsection 9.2.2.

5. Taxibot coupling.

   (a) Coupling operations are not modelled, meaning that a Taxibot is coupled to aircraft when ready for departure.

6. Taxibot uncoupling.

   (a) The model uses discretized uncoupling locations. Uncoupling points are indicated by an aircraft in Figure 9.2.
   (b) All uncoupling points have a designated *all clear* point, indicated by a green dot in Figure 9.2.
   (c) After uncoupling, Taxibots must go to the designated *all clear* point.
   (d) After the all clear point is reached by Taxibot, aircraft can enter all runway entrances, and Taxibot must travel back to the nearest meta-gate.
   (e) Uncoupling duration is given with t_d, for both NB and WB aircraft.
   (f) Uncoupling tasks (consisting of an uncoupling location and time) are assigned to departing aircraft by the **Hub Labelling** mechanisms.

7. Flight, runway and gate schedule.

   (a) The traffic sample in the schedule corresponds to the busiest day of 2019. An example of a flight schedule is given below.
   (b) The schedule contains the aircraft ID (column 1).
   (c) The schedule contains the source node (meta-gate/runway-exit for departure and arriving aircraft) (column 2).
   (d) The schedule contains the goal node (runway-exit/meta-gate for departure and arriving aircraft) (column 3).

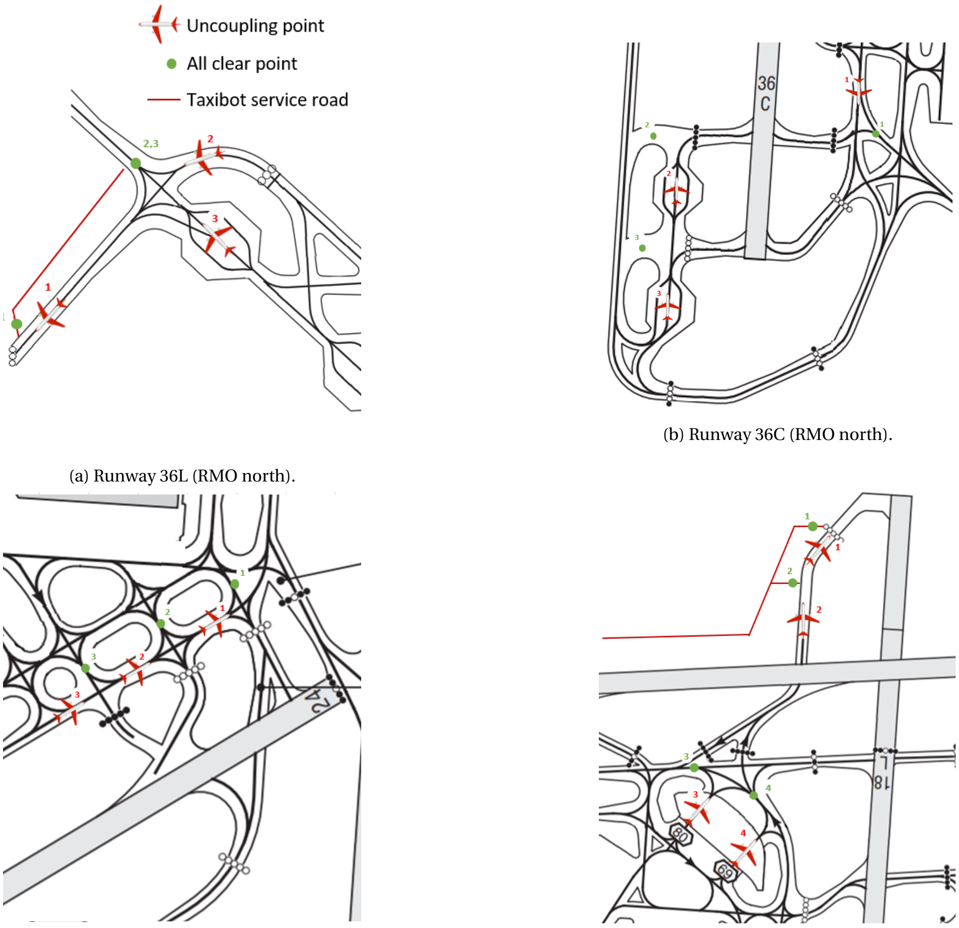
(e) For arriving aircraft, the schedule contains the arrival time. In the model, aircraft land at the scheduled arriving time and spawn directly afterwards in the system (column 4).

(f) For departing aircraft, the schedule contains the departure time. Departing aircraft adhere to the departure sequence, and try to take-off as close as possible to their scheduled take-off time (column 4).

(g) The schedule indicates whether the aircraft is arriving or departing (column 5).

(h) The schedule contains the aircraft type (NB or WB).

(i) Two RMOs are considered. RMO north (36L + 36C / 06 + 36R) and RMO south (18L + 24 / 18C + 18R). They are evaluated separately.



(a) Runway 36L (RMO north).

(b) Runway 36C (RMO north).

(c) Runway 24 (RMO south).

(d) Runway 18L (RMO south).

Figure 9.2: Pre-defined uncouple spots at taxiways and P-platforms near runway entrances. At each uncoupling spot, an ATC uncoupling agent is located to guide the procedure and ensure a conflict free process.

$$F_i = \begin{bmatrix} ID_1 & Origin_1 & Dest_1 & Time_1 & AD_1 & Type_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_i & Origin_i & Dest_i & Time_i & AD_i & Type_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_n & Origin_n & Dest_n & Time_n & AD_n & Type_n \end{bmatrix}$$

8. Separation and conflicts.

(a) Conflicts between vehicles are avoided in advance, using the **Bounded-Horizon PBS** mechanism, as described in chapter 7.

(b) The initial priority order of Bounded-Horizon PBS is given in Table 9.1.

(c) Separation between vehicles is maintained by visual inspection.

Table 9.1: Initial prioritization settings of the Bounded-Horizon PBS mechanism.

Priority	**Vehicle**	**Reason**
1	Departing aircraft currently uncoupling at a point	Unable to move
2	Departing aircraft running late	Mitigate ground delays
3	Arriving aircraft	Reduce consumption of aviation fuel
4	Departing aircraft	Normal priority
5	Departing aircraft planning to uncouple at conflict location	To avoid runway queues
6	Taxibot	No passsengers

9.2.2. Simulation Constants

This section states all parameters used in this research.

1. Vehicle parameters.
   (a) Vehicles have a constant taxi speed, given in the first column of Table 9.2.
   (b) Vehicles do not have to decelerate for small turn angles, given in the second column of Table 9.2.
   (c) For large turn angles, vehicles must decelerate to a turn speed, given in the third column of Table 9.2.
   (d) Vehicles have a fixed deceleration, given in the fourth column of Table 9.2.
   (e) Vehicles have a fixed emergency deceleration, given in the fifth column of Table 9.2.
   (f) Vehicles have a fixed acceleration, given in the sixth column of Table 9.2.
   (g) Vehicles have a fixed visual range, given in the seventh column of Table 9.2.

2. Separation and conflicts.
   (a) Vehicles maintain visual separation. Table 9.3 provides the minimum separation requirement between vehicles at taxiways, aprons and the two service roads.
   (b) Five common types of conflicts are considered: edge conflict (a), vertex conflict (b), following conflict (c), cycle conflict (d), and swapping conflict (e). Table 9.4 provides what conflicts are allowed between vehicles at taxiways & Aprons and the two service roads.

Table 9.2: Conflicts between all three agents on two different road types.

	V $[m/s]$	$Turn_{max}$ $[\circ]$	V_{turn} $[m/s]$	d $[m/s^2]$	d_{max} $[m/s^2]$	a $[m/s^2]$	radar $[m]$
Taxiways & Aprons							
NB Aircraft	tbd_1	tbd_9	tbd_{17}	tbd_{25}	tbd_{33}	tbd_{41}	tbd_{49}
WB Aircraft	tbd_2	tbd_{10}	tbd_{18}	tbd_{26}	tbd_{34}	tbd_{42}	tbd_{50}
NB Taxibot	tbd_3	tbd_{11}	tbd_{19}	tbd_{27}	tbd_{35}	tbd_{43}	tbd_{51}
WB Taxibot	tbd_4	tbd_{12}	tbd_{20}	tbd_{28}	tbd_{36}	tbd_{44}	tbd_{52}
Towed NB aircraft	tbd_5	tbd_{13}	tbd_{21}	tbd_{29}	tbd_{37}	tbd_{45}	tbd_{53}
Towed WB aircraft	tbd_6	tbd_{14}	tbd_{22}	tbd_{30}	tbd_{38}	tbd_{46}	tbd_{54}
Service Roads							
NB Taxibot	tbd_7	tbd_{15}	tbd_{23}	tbd_{31}	tbd_{39}	tbd_{47}	tbd_{55}
WB Taxibot	tbd_8	tbd_{16}	tbd_{24}	tbd_{32}	tbd_{40}	tbd_{48}	tbd_{56}

Table 9.3: Separation requirements between all types of vehicles. *Separation on service roads.

	NB aircraft	WB aircraft	NB Taxibot	WB Taxibot	Towed NB aircraft	Towed WB aircraft
NB aircraft	tbd_{57}	tbd_{63}	tbd_{69}	tbd_{77}	tbd_{85}	tbd_{91}
WB aircraft	tbd_{58}	tbd_{64}	tbd_{70}	tbd_{78}	tbd_{86}	tbd_{92}
NB Taxibot	tbd_{59}	tbd_{65}	tbd_{71}/tbd_{72}*	tbd_{79}/tbd_{80}*	tbd_{87}	tbd_{93}
WB Taxibot	tbd_{60}	tbd_{66}	tbd_{73}/tbd_{74}*	tbd_{81}/tbd_{82}*	tbd_{88}	tbd_{94}
Towed NB aircraft	tbd_{61}	tbd_{67}	tbd_{75}	tbd_{83}	tbd_{89}	tbd_{95}
Towed WB aircraft	tbd_{62}	tbd_{68}	tbd_{76}	tbd_{84}	tbd_{90}	tbd_{96}

9.2.3. Independent variables

The simulation is performed under varying circumstances. There are two main varying indicators.

1. Uncoupling duration, t_d. As described in section 5.5, the uncoupling duration is a critical factor for the throughput of runways, and hence the capacity of airports.
2. Conflict search window parameters of the Bounded-Horizon PBS coordination mechanism. The search window w, and the plan frequency h.

Table 9.4: Conflicts between all three agents on two different road types. Note: there are no differences in conflict types for NB and WB Taxibots/aircraft.

	a	b	c	d	e
Taxiways & Aprons					
Aircraft - Aircraft	Yes	No	Yes	Yes	No
Taxibot - Taxibot	Yes	Yes	Yes	Yes	Yes
Aircraft - Taxibot	Yes	No	Yes	Yes	No
Aircraft - towed aircraft	Yes	No	Yes	Yes	No
Taxibot - Towed aircraft	Yes	No	Yes	Yes	No
Towed aircraft - towed aircraft	Yes	No	Yes	Yes	No
Service Roads					
Taxibot - Taxibot	Yes	No	Yes	Yes	No

9.3. Results, Outcome and Relevance

A supercomputer at To70 is available to run simulations and obtain results. This section start with an overview of what results are expected and what outcomes can be concluded from this. Finally, the relevance of the outcome is mentioned and the verification and validation method is discussed.

9.3.1. Results and Outcome

The objective of this research is to manage and study ground delays in an autonomous engine-off outbound taxi system implemented at AAS. Therefore, the main outcome of this study is the minimum ground delay reachable in the simulation. This outcome is studied under varying circumstances, given in subsection 9.2.3. The outcome is subsequently compared with conventional operations to conclude on the suitability of the autonomous system in terms of ground delays. Four different types of ground delay are considered:

1. **Taxi delay:** Delays as a result of congestion on the taxiways.
2. **Decoupling delay:** Delays caused by aircraft waiting for other aircraft to finish decoupling.
3. **Runway delay:** Delays caused by aircraft waiting in a runway queue as a result of an occupied runway.
4. **Take-off delay:** Difference between modelled take-off time and scheduled take-off time.

Another outcome are heatmaps of the AAS ground system. These maps shall be developed to provide insight in the emergence of bottlenecks. Based on these bottlenecks, decision makers can conclude on the construction of additional infrastructure to reduce ground delays by eliminating these bottlenecks, if needed.

9.3.2. Relevance

It is evident that major hub airports like AAS are not willing to reduce ground capacity or risk ground delays as a result of a towing system like Taxibot. At the same time, airports are under increased pressure to reduce their environmental impact. The outcome of this research is relevant because the decision to reduce airport ground emission by a towing system depends on whether ground delays can be adequately mitigated. If the resulting ground delays in the autonomously coordinated system are not within acceptable bounds, use can be made of heatmaps to conclude if additional infrastructure is needed. Likewise, the sensitivity analysis can be used to determine what uncoupling duration is required to mitigate ground delays sufficiently.

The research is assumed to be ethically justified. The main reason for this assumption is that the final outcome is intended to contribute towards efficiency, safety and security in a complex system. Furthermore, it is intended to significantly contribute towards sustainable innovation, which hopefully leads to increased happiness worldwide. One might argue that harm is done to ATC employees, as their work is replaced by an autonomous system. However, the autonomous system also leads to the emergence of new (e.g. supervising & maintenance) jobs. It is yet to be determined what the resulting net shift in workload is.

9.3.3. Result Verification & Validation

Verification and validation of the simulation is performed using an interface developed in Pygame. Pygame is a module designed for video games programmed in Python. This interface is used in several validation and verification methods.

Visual validation of the developed network is performed by the comparison of Pygame with the actual ground map of AAS. The network will be validated in terms of accuracy and dimensions in consultancy with AAS and To70 experts. Validation of the ground movements is performed by a slow simulation of the model, to observe whether all ground vehicles produce movement that represent actual operations.

Verification of the simulation is done at different steps, at different levels of the model. A distinction is made between low and high-level verification. Lower level verification is performed by testing and analysing individual blocks of code, by using simplified inputs, leading to predictable outputs. High-level verification is performed by decomposing and reconstructing the model. This works as follows. First, the model is broken down until only the environment is left. Then, modules and small number of agents are added to this simplified, predictable model. The behaviour of individual modules and agents can than be observed, to verify their correct functioning. As such, all modules are verified consecutively, until the model is complete constructed.

9.4. Research Planning

The planning of the project can be found in Table 9.5. Note that at the current time of writing, the literature study and the first work package is finished ([update: April 1^{st}]), meaning that step 0, 1 and 2 of the mental cycle presented in Figure 9.1 is completed. Most work performed in WP0 and WP1 is summarised in this document. In the second work package, the first cycle of step 3 is executed. Ground delay mitigation strategies developed in the literature study are improved and subsequently implemented in the model. The model capabilities are tested and findings shall be reported. In WP3, the model is trained such that ground delays are mitigated as much as possible. This is done by a continuous improvement of cooperative coordination methods and delay management techniques (lower cycles, step 3). As this work package contains continuous improvement and analysis, it is repeated until a sufficient level of delay is reached. WP4 analyses the performance of the model in terms of ground delays, bottlenecks and varying uncoupling time. In the last work package, the final thesis is written and the graduation process is finalised.

All project phases depends on the outcome of previous phases. This is visualised in the Gantt chart of Appendix B. In this chart, the sub-work packages and the main work packages are indicated in light and dark blue respectively. Milestones are indicated with red. At the end of work package 2, 3 and 4, some documentation time is taken into account. This is visualised with the dark slashes.

Table 9.5: Breakdown and duration of the work packages in this research, as of January 1, 2021 [update April 1st: WP1, finished].

Work Package	High-level task	Low-level task	Duration
WP 1	**Familiarize and Start up**		**~7 weeks**
WP 1.1	Start up To70	- Familiarize with company and employees - Recieve laptop, e-mail, etc..	~0.5 week
WP 1.2	Familiarize baseline model	- Experiment with and verify existing ABM tools of B. Benda and S. Polydorou - Clean code	~3 weeks
WP 1.3	Develop conventional model	- Collect variables for conventional operations - Adjust model to conventional operations	~1 week
WP 1.4	Specify towing model characteristics	- Decide upon exact towing vehicle operations - Collect data for towing operations - Specify ground delay requirements (KPI's)	~2.5 weeks
WP 2	**Model expension**		**~8 weeks**
WP 2.1	Improve and expend strategies to mitigate ground delays	- Develop the scheduling and routing module for uncoupling based on optimal MAPF and task assignment - Develop or improve MAPF strategies to manage additional vehicles on taxiways and flow around uncoupling points	~1 week
WP 2.2	Qualitative ABM development	- Specify environmental modifications - Specify agents, interaction and behaviour	~1 weeks
WP 2.3	Quantitative ABM development	- Extend conventional model with new environment and agents - Implement scheduling and routing module for uncoupling - Improve MAPF strategies	~3 weeks
WP 2.4	Analyse model capabilities	- Compare ground delays with conventional model (KPIs) - Test model in terms of safety and emergent behaviour - Verificate and validate model	~2 week
WP 2.5	Thesis report	- Report findings of WP 2	~1 week
WP 3	**Model training**		**~9 weeks**
WP 3.1	Review conventional model and varying conditions (if neccecery)	- Review conventional model - Review towing model characteristics	~1 week
WP 3.2	Review model and delay mitigation strategies	- Review developed model - Study alternative ground delay mitigation methods - Conclude upon model improvements (e.g. conflict resolution, CBS, ML methods)	~1 week
WP 3.3	Expand ABM	- Implement improvements and strategies	~5 weeks
WP 3.4	Analysis model capabilities	- Test model in terms of safety and emergent behaviour - Verificate and validate model - Compare ground delays with conventional airport data - Decide whether further reduction in ground delay is achievable	~1 week
WP 3.5	Thesis report	- Report findings of WP3	~1 week
WP4	**Final model analysis**		**~4.5 weeks**
WP 4.1	Familiarize PC	- Familiarize with supercomputer To70	~1 weeks
WP 4.2	Simulation	- Determine simulation scenarios - Carry out simulations	~2.5 week
WP 4.3	Analysis	- Sensitivity analysis - Statistical analysis	~1 week
WP 4.4	Conclusions	- Draw conclusions from simulations and analysis	~1 week
WP 5	**Complete graduation**		**~5 weeks**
WP 5.1	Thesis report	- Write, edit and complete MSc thesis	~4 weeks
WP 5.2	Graduation presentation	- Prepare graduation presentation	~1 week

Bibliography

[1] Lina Kivaka. [retrieved at 04/01/2021] from https://www.pexels.com/pl-pl/zdjecie/lot-pogoda-burza-mgla-2253921/.

[2] Husni Rifat Idris. Observation and analysis of departure operations at boston logan international airport. *Massachusetts Institute of Technology (MIT)*, 2001.

[3] EUROCONTROL Airport CDM Team. *Airport CDM Implementation.* EUROCONTROL, IATA, ACI, 5.0 edition, March 2017.

[4] W.H. Dalmeijer V.A. Bijsterbosch. Gebruiksprognose 2021. *SCHIPHOL*, September 2020.

[5] Melanie Sandberg Ioannis Simaiakis and Hamsa Balakrishnan. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 15(1):285–295, February 2014.

[6] Ashley Chymiy Hazel Peace Gareth Horton Charles Walker Mike Kenney Clint Morrow Mikhail Chester Yu Zhang Damon Fordham, Mia Stephens and Paul Sichko. Deriving benefits from alternative aircraft-taxi systems. *THE NATIONAL ACADEMIES PRESS*, 2016.

[7] Stan Malicki. The aircraft towing system. *http://www.at-system.eu/system.html*, December 2020.

[8] Metropolitan Airport News. Taxibot makes debut at schiphol airport. *Aviation News*, June 2020.

[9] H. Udluft. Agent-based simulation of decentralized control for taxiing aircraft. Tu delft repository, Delft University of Technology, October 2017.

[10] T. Noortman. Agent-based modelling of an airport's ground surface movement operation. Tu delft repository, Delft University of Technology, September 2018.

[11] K. Fines. Agent-based distributed planning and coordination for resilient airport surface movement operations. Tu delft repository, Delft University of Technology, September 2019.

[12] B. Benda. Agent-based modelling and analysis of non-autonomous airport ground surface operations. Tu delft repository, Delft University of Technology, November 2020.

[13] Amgad Madkour, Walid Aref, Faizan Rehman, Abdur Rahman, and Saleh Basalamah. A survey of shortest-path algorithms. 05 2017.

[14] A. Felner S. Koenig H. Ma T. Walker J. Li D. Atzmon L. Cohen S. Kumar E. Boyarski R. Stern, N. Sturtevant and R. Bartak. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Proceedings of the Symposium on Combinatorial Search (SoCS)*, (1):1–8, May 2019.

[15] M. Renee Jansen and Nathan R. Sturtevant. Direction maps for cooperative pathfinding. *IEEE Transactions on Systems Science and Cybernetics*, (2):185–190, 2008.

[16] Sven Koenig Wolfgang Hönig, Jiaoyang Li. Overview of multi-agent path finding (mapf). *Model AI 2020 Assignments: A Project on Multi-Agent Path Finding (MAPF)*, (1), 2020.

[17] David Silver. Cooperative pathfinding. *American Association for Artificial Intelligence*, 2005.

[18] Zahy Bnaya and Ariel Felner. Conflict-oriented windowed hierarchical cooperative a*. *Information Systems Engineering Department, Ben Gurion University, Israel*, 2007.

[19] Ko-Hsin Cindy Wang and Adi Botea. Tractable multi-agent path planning on grid maps. *NICTA and The Australian National University*, pages 1870–, 2011.

[20] Ryan Luna and Kostas E. Bekris. Efficient and complete centralized multi-robot path planning. *e Department of Computer Science and Engineering, University of Nevada*, 2010.

[21] Ariel Felner Nathan Sturtevant Guni Sharon, Roni Stern. Conflict-based search for optimal multi-agent path finding. *AAAI Conference on Artificial Intelligence*, (219):40–66, December 2014.

[22] Peter J. Stuckey Jiaoyang Li Sven Koenig Hang Ma, Daniel Harabor. Searching with consistent prioritization for multi-agent path finding. *AAAI, Association for the Advancement of Artificial Intelligence*, (3rd):7643–7650, May 2018.

[23] Andrew Tinka Joseph W. Durham Nora Ayanian Wolfgang Hönig, Scott Kiesel. Conflict-based search with optimal task assignment. *International Foundation for Autonomous Agents and Multiagent Systems*, 2018.

[24] Liron Cohen Hang Ma Hong Xu Nora Ayanian Wolfgang Honig, T. K. Satish Kumar and Sven Koenig. Multi-agent path finding with kinematic constraints. *International Conference on Automated Planning and Scheduling*, pages 477–485, 2016.

[25] Sybert H. Stroeve and Mariken H.C. Everdij. Agent-based modelling and mental simulation for resilience engineering in air transport. *Safety Science*, 93:29–49, 2017.

[26] RECAT-EU. *European Wake Turbulence Categorisation and Separation Minima on Approach and Departure.* EUROCONTROL, 1.1 edition, July 2015.

[27] Christophe Cros and Carsten Frings. Alternative taxiing means – engines stopped. *Airbus workshop on "Alternative taxiing means – Engines stopped"*, November 2008.

[28] E. Mazareanu. Global air traffic - annual growth of passenger demand 2006-2021. *STATISTA*, June 2020.

[29] IATA: International Air Transport Association Corporate Communications. Reconnecting the world. *PRESSROOM IATA*, March 2021.

[30] The International Civil Aviation Organisation (ICAO). Effects of novel coronavirus (covid-19) on civil aviation: Economic impact analysis. *Coronavirus economic impact TH*, October 2020.

[31] Job van der Plicht. Dit zijn de vijf knelpunten rond de uitbreiding van schiphol. *Nu.nl*, December 2018.

[32] Harshad Khadilkar and Hamsa Balakrishnan. Estimation of aircraft taxi fuel burn using flight data recorder archives. *Massachusetts Institute of Technology Cambridge*, September 2016.

[33] Smart Airport Systems (SAS). brochure: The only certified operational alternative taxiing solution. Israel Aerospace Industries report, 06 2020.

[34] Airport Council International (ACI). Autonomous vehicles and systems at airports report (2019). *ACI REPORT*, November 2019.

[35] Jason A.D. Atkin Juan Castro-Gutierrez Daniel Karapetyan, Andrew J. Parkes. Lessons from building an automated pre-departure sequencer for airports. *Annals of Operations Research*, May 2017.

[36] International Civil Aviation Organization (ICAO). Procedures for air navigation services. *DOC 444, Technical report*, ATM/501, November 2007.

[37] Airports Councils International. Aci reveals top 20 airports for passenger traffic, cargo, and aircraft movements. *aci.aero*, May 2020.

[38] Air Traffic Control the Netherlands. *EHAM — AMSTERDAM/Schiphol, AIP NETHERLANDS.* LVNL, February 2016.

[39] Schiphol group. integralsafetyschiphol: projects in progress. *SCHIPHOL*, November 2020.

[40] Schiphol. Schiphol and partners to begin sustainable aircraft taxiing trial. *SCHIPHOL NEWS*, April 2020.

[41] Indira Deonandan Hamsa Balakrishnan and Ioannis Simaiakis. Opportunities for reducing surface emissions through airport surface movement optimization. *AIAA: Massachusetts Institute of Technology, Cambridge*, pages 1–34, December 2008.

[42] Dirk Bresser and Simon Prent. Interview and/or classified documents. *SCHIPHOL GROUP*, Oktober 2020.

[43] International Civil Aviation Organization (ICAO). Icao aircraft engine emissions databank. *ICAO Annex 16*, August 2020.

[44] Schiphol News. Zo hoog en snel vliegen ze. *Schiphol.nl*, 2019.

[45] Airport Council International. *ACERT User Manual.* ACI, 5.1 edition, April 2018.

[46] ATM Operations. *SESAR 2020 Concept of Operations Edition 2017.* EUROCONTROL, 01.00.00 edition, November 2017.

[47] Federal Aviation Administration. *Aviation Environmental Design Tool (AEDT).* FAA, 2a edition, November 2012.

[48] EUROCONTROL. *Specification on Data Link Services*, 2.1 edition, January 2009.

[49] EUROCONTROL. *Advanced Surface Movement Guidance and Control System (A-SMGCS).* https://www.skybrary.aero/, September 2020.

[50] H. Balakrishnan Clewlow, R. and T. G. Reynolds. A survey of airline pilots regarding fuel conservation procedures for taxi operations. *International Airport Review*, 14(3), May 2010.

[51] Ben Schlappig. Qatar airways threatens pilots for taxiing with two engines. *onemileatatime*, October 2020.

[52] aviationpros. Iberia airlines taxiing program to reduce emissions at ord. December 2012.

[53] Steve Lent (Mosaic) Chris Brinton, Chris Provan and Susan Passmore (FAA) Tom Prevost. Collaborative departure queue managemen. *NinthUSA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, 2011.

[54] Gautam Gupta, Waqar Malik (University of California), and Yoon C Jung (NASA Ames Research Center). An integrated collaborative decision making and tactical advisory concept for airport surface operations management. *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, September 2012.

[55] STBO Project Team. Collaborative departure queue management (cdqm). *FAA AJP-67*, (2.0), September 2010.

[56] Yu Sun Hongfei Zhang Kai Wang Xinhua Zhu, Nan Li and Sang-Bing Tsai. A study on the strategy for departure aircraft pushback control from the perspective of reducing carbon emissions. *ENERGIES MDPI*, September 2018.

[57] H. Balakrishnan Simaiakis, M. Sandberg and R. J. Hansman. an massachusetts institute of technology, cambridge, ma, usa. *5th International Conference on Research in Air Transportation (ICRAT)*, 2012.

[58] National Aeronautics and Space Administration (NASA). Sarda fact-sheet: National aeronautics and space administration. *NASA: Information Technology and Software*.

[59] Dirk Bresser and Simon Prent. Sustainable taxiing and the taxibot factsheet. *Schiphol.nl*, 2020.

[60] Qing Wang Rui Guo, Yu Zhang. Comparison of emerging ground propulsion systems for electrified aircraft taxi operations. *ELSEVIER, Transportation Research Part C: Emerging Technologies*, pages 98–109, March 2014.

[61] Research Yann NICOLAS and Programme Leader Aircraft Control Airbus S.A.S. Technology. etaxi taxiing aircraft with engines stopped. *FAST | FLIGHT AIRWORTHINESS SUPPORT TECHNOLOGY*, January 2013.

[62] D. J. Paisley. Alternative taxi considerations. *International Air Transport Assosiation (IATA) Aircraft Taxiing Systems Conference*, February 4 2015.

[63] BRISTOL OFFICE. Wheeltug taxiing system. *Stirling Dynamics*, December 2020.

[64] Ahmed Hebala Stefano Nuzzo Michael Galea-Senior Member Milos Lukic, Paolo Giangrande. Review, challenges and future developments of electric taxiing systems. *Institute of Electrical and Electronics Engineers*, Oktober 2019.

[65] Thierry Dubois. Wheeltug, safran-honeywell and iai offer three rival solutions for airline engine-off taxiing. *ainonline*, February 2014.

[66] Thomas F. Johnson. Electric green taxiing system (egts) for aircraft. *IEEE TRANPORTATION ELECTRIFICATION COMMUNITY*, April 2014.

[67] Dirk Bresser and Simon Prent. What is sustainable taxiing? (part 1). *Schiphol.nl*, December 2020.

[68] Vincent Metz (SAS). Smart airport systems (sas) presentation: Taxibot. *Alvest*, 2020.

[69] Smart Airport Systems (SAS). The taxibot concept. *https://www.taxibot-international.com/*, November 2020.

[70] TLD-group. Taxibot to enter service in 2014. *TLD newsroom*, April 2014.

[71] SAM SHEAD. Amazon now has 45,000 robots in its warehouses. *Businessinsider INDIA*, January 2017.

[72] Loes Witschge. Rotterdam is building the most automated port in the world. *WIRED*, October 2019.

[73] MICHAEL WOOLDRIDGE. *An Introduction to Multiagent Systems*. Department of Computer Science, University of Liverpool, Uk', 2002.

[74] Alexei Sharpanskykh. *AE4422-19: Agent-based Modelling and Simulation in Air Transport*. Delft University of Technology, September/October 2019-2020.

[75] J.A. Siebers. Airport surface planning under uncertainty. Tu delft repository, Delft University of Technology, January 2017.

[76] Spyros Polydorou. A learning-based approach for distributed airport surface movement operations. 2020.

[77] Sun Junqing Sun Bin Yang Peng, Gao Wei. Evaluation of airport capacity through agent based simulation. *International Journal of Grid Distribution Computing*, 7(6):165–174, 2014.

[78] Memòria del Treball Fi de Grau en Gestió Aeronàutica. Taxiway management through multi agent system. Escola d'enginyeria sabadell, Universitat Autonoma de Barcelona, July 2013.

[79] Nico Roos Cees Witteveen Xiaoyu Mao, Adriaan ter Mors. Agent-based scheduling for aircraft deicing. *almede B.V.*, pages 1–8, 2006.

[80] J.A.P. Borst. A multi-agent operational planning model for airport stakeholders. *Royal Netherlands Aerospace Centre (NLR)*, (NLR-TR-2016-461), 2016.

[81] R. Curran S. Bouarfa, H.A.P. Blom. Airport performance modeling using an agentbased approach. *Air Transport and Operations Symposium 2012, TU Delft repository*, 2012.

[82] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *PNAS*, 99:7280–7287, May 2002.

[83] Ali Bazghandi. Techniques, advantages and problems of agent based modeling for traffic simulation. *IJCSI: International Journal of Computer Science*, 9(3):115–119, 2012.

[84] CHARLES M. MACAL MICHAEL J. NORTH. Agent based modeling and computer languages. *Argonne National Laboratory, Decision and Information Sciences Division, Center for Complex Adaptive Agent Systems Simulation (CAS), Argonne, USA*, 2:58–75, 2012.

[85] Eva Tardos Jon Kleinberg. *Algorithm Design*. PEARSON Addison Wesley, 2005.

[86] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[87] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics.*, 16:87–90, 1958.

[88] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM*, (52):1–24, 2005.

[89] N. Nilsson P. Hart and B. Raphael. Formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics*, pages 100–107, 1968.

[90] S. P ´erennes C. Gavoille, D. Peleg and R. Raz. Distance labeling in graphs. *J. Algorithms*, pages 85–112, 2004.

[91] Ko-Hsin Cindy Wang and Adi Botea. Mapp: a scalable multi-agent path planning algorithm with tractability and completeness guarantees. *Journal of Artificial Intelligence Researc*, (42):55–90, 2011.

[92] C. Mata and J. S. B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. *13th Annu. ACM Sympos. Comput. Geom.*, page 264–273, 1997.

[93] Ioannis Simaiakis and Hamsa Balakrishnan. A queuing model of the airport departure process (mit). *Transportation Science*, (94–109), December 2016.

[94] Hamsa Balakrishnan Ioannis Simaiakis, Melanie Sandberg. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE Transactions on Intelligent Transportation Systems*, February 2014.

[95] Lance Sherry Poornima Balakrishna, Rajesh Ganesan and Benjamin S. Levy. Estimating taxi-out times with a reinforcement learning algorithm. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2008.

[96] K. Mulmuley and P. Shah. A lower bound for the shortest path problem. *Journal of Computer and System Sciences*, 63(63):253–267, 2001.

[97] J. X. Yu B. Ding and L. Qin. Finding time-dependent shortest paths over large graphs. *roceedings of the 11th international conference on Extending database technology Advances in database technology EDBT 08*, page 205, 2008.

[98] Rishabh Luthra Ms. Neeti Sangwan Abhishek Goyal, Prateek Mogha. Path finding: A* or dijkstra's? *IJITE*, 2, January 2014.

[99] I Pohl. Heuristic search viewed as path finding in a graph. *Journal of Artificial Intelligence Research*, pages 193–204, 1970.

[100] Roni Stern Ariel Felner Max Barer, Guni Sharon. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. *, Association for the Advancement of Artificial Intelligence*, pages 19–27, 2014.

[101] Scott Kiesel Joseph W. Durham T. K. Satish Kumar Jiaoyang Li, Andrew Tinka and Sven Koenig. Lifelong multi-agent path finding in large-scale warehouses. *University of Southern California*, (arXiv:2005.07371v1), May 2020.

[102] N. J.; Hart, P.; Nilsson and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, (4):100–107, 1968.

[103] J. Latombe. Overview of multi-agent path finding (mapf). *Robot Motion Planning. Boston, MA: Kluwer Academic.*, 1991.

[104] M. Renee Jansen and Nathan R. Sturtevant. Direction maps for cooperative pathfinding. *Journal of Artificial Intelligence Research*, pages 185–190, 2008.

[105] Cees Witteveen Boris de Wilde, Adriaan W. ter Mors. Push and rotate: a complete multi-agent pathfinding algorithm. *Journal of Artificial Intelligence Research*, (51):443–492, 2014.

[106] Frederick S. Hiller (Standford University) and Gerald J. Lieberman (Late of Standford University). *Introduction to Operations Research*. McGRAW-HILL Inernational EDUCATION.

[107] N.J.F.P. Guillaume. Finding the viability of using an automated guided vehicle taxiing system for aircraft. Master's thesis, Delft University of Technology, April 2018.

[108] H.G. Visser C. Evertse. Real-time airport surface movement planning: Minimizingaircraft emissions. *Elsevier*, March 2017.

[109] P. C. Roling and H. G. Visser. Optimal airport surface traffic planning using milp. *International Journal of Aerospace Engineering*, page 11, April 2008.

[110] E.V.M. van Baaren. The feasibility of a fully electric aircraft towing system. Master's thesis, Delft University of Technology, May 2019.

[111] Gillian Clare and Arthur Richards. Optimization of taxiway routing and runway. *University of Bristol, Bristol, UK.*

[112] Angel Marin. Airport management: Taxi planning. *Annals OR*, 143(202):1–12, March 2006.

[113] G. Wagner and H. Choset. A complete multirobot path planning algorithm with performance bounds. *IROS*, pages 3260–3267, 2011.

[114] S. Koenig L. Cohen, T. Uras. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding. *Association for the Advancement of Artificial Intelligence*, 2015.

[115] J.; Kumar T. K. S.; Ma, H.; Li and S. Koenig. Lifelong multi-agent path finding for online pickup and delivery tasks. *International Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, page 837–845, 2017.

[116] Sven Koenig Hang Ma, T. K. Satish Kumar. Optimal target assignment and path finding for teams of agents. *International Foundation for Autonomous Agents and Multiagent Systems*, 2016.

[117] Bart Selman Michele Lombardi Willem-Jan van Hoeve, Carla P. Gomes. Optimal multi-agent scheduling with constraint programming. *Association for the Advancement of Artificial Intelligence*, pages 1813–1818, 2007.

[118] Mark F. Tompkins. Optimization techniques for task allocation and scheduling in distributed multi-agent operations. *Mark Freeman Tompkins, MMIII.*, pages 1–12, May 2003.

[119] Rutget Claes. Anticipatory vehicle routing | coordinating traffic using community generated traffic predictions. June 2015.

[120] Sven Koenig Hang Ma, T. K. Satish Kumar. Multi-agent path finding with delay probabilities. *Association for the Advancement of Artificial Intelligence*, 2017.

[121] Roni Stern Dor Atzmon Roman Bartak Jir Svancara, Marek Vlk. Online multi-agent pathfinding. *Association for the Advancement of Artificial Intelligence*, pages 7732–7739, 2019.

III

Supporting Work

A
Model Elaboration

This Appendix elaborates on the developed model. Some general comment related to the infrastructure are first given in Appendix A.1. The uncoupling locations, service roads and associated assumptions are subsequently provided in Appendix A.2 and Appendix A.3. The remaining operational assumptions are provided in Appendix A.4. The constraint setting for the conflict avoidance system is given in Appendix A.5. Finally, an example of SIPP anticipation is provided in Appendix A.6.

A.1. General Infrastructure

The ground movement area of AAS was translated into a graph comprising 1124 nodes and 3130 unidirectional edges. The graph was extracted from WorldEditor; an open source 2-d scenery and airport editor for the simulation game X-Plane [1]. In the graph, adjacent nodes are connected by two unidirectional edges, creating a bidirectional road. A section of the graph is visualised in Figure A.1.

Seven node types are important for this research: 1) regular nodes (yellow), 2) gates (dark yellow), 3) uncoupling points (purple), 4) node before uncoupling point (pink), 5) all-clear point (green), 6) stop bars (dark red), and 7) runways (red). The node before the uncoupling point is used by the algorithm to ensure uncoupling in the right direction.

Four type of edges are included in the model: 1) taxiways (thin black lines), 2) aprons (thin black lines), 3) runways (thick black lines), or 4) service roads (thin blue lines).

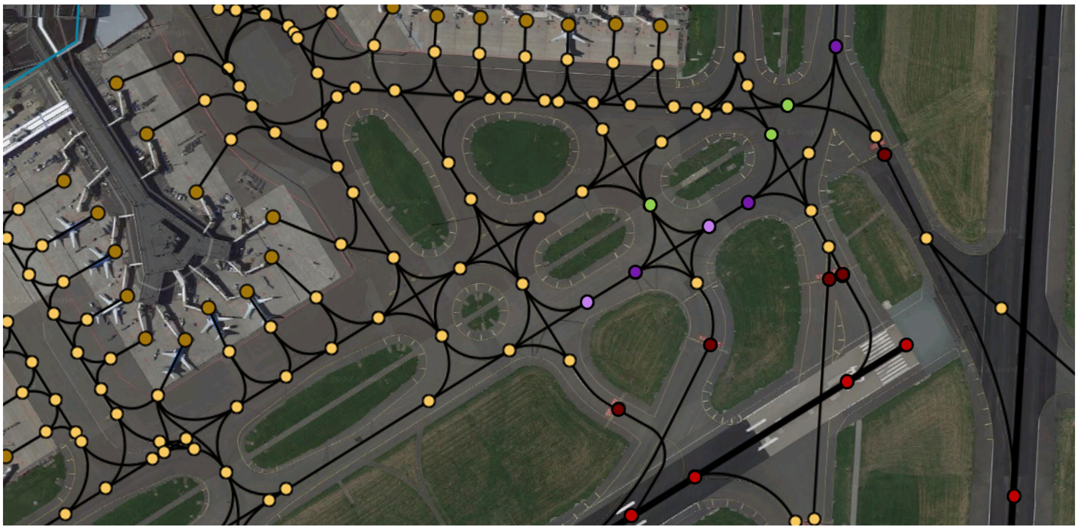


Figure A.1: Graph section at runway 24.

The graph - located in the environment - comprises many relevant information. The augmented path planning algorithm uses this information during search. Important information attached to nodes include:

node type, node coordinates, neighbor nodes, and vicinity nodes/edges. Important information attached to edges include: edge type, connecting nodes, length, heading start/end, width, max radius, vicinity nodes/edges, and heuristic to all other nodes. Depending on the edge or node type, more relevant information can be attached. For example uncoupling nodes have the associated all-clear point (to spawn the Taxibot at the right location and in the right heading), runway (to make sure the aircraft goes to the right runway after uncoupling), and node before uncoupling point (to ensure uncoupling in the right direction) attached.

A.2. Uncoupling Locations

Four departure runways are considered. At each runway, three uncoupling strategies are studied: a) uncoupling at holding platforms, b) uncoupling at taxiways, or c) uncoupling on both holding platforms and taxiways. Runway 24 does not contain any nearby holding platforms. We therefore only considered the Taxiway uncoupling strategy for this runway. For each runway, 2 uncoupling points were selected on taxiways, and 2 uncoupling points were selected on holding platforms (except for runway 24). A set of uncoupling points at the same runway and in the same strategy are called an uncoupling configuration. An overview of all locations is given in Table A.1. The uncoupling location selection process was based on 8 criteria (see below) and performed in collaboration with AAS.

The locations of all uncoupling locations are visualized for runway 36L, 36C, 18L and 24 in Figure A.2, Figure A.3, Figure A.4, Figure A.5 respectively. Edges in these figures represent: taxiway centrelines (thin black line), runways (thick black line), service roads (thin blue line), and stop bars (thin green line). Nodes in these figures represent: regular nodes (yellow), uncoupling nodes (red), all-clear nodes (green). Uncoupling points starting with a "P" are located at holding platforms. All other points are located at taxiways. The allowable heading during uncoupling is given in parentheses in the caption of each figure.

Table A.1: Uncoupling configurations and strategies. *Taxiway strategy used for runway 24.

Uncoupling strategy	Runway	Uncoupling configuration	Expected flow disturbance
Holding platforms*	18L	P1, P3	Limited
	36C	P4, P5	Full
	36L	P6, P7	Limited
Taxiways	18L	E6-I, E6-II	Full
	36C	B(A25-A26), Q	Partial
	36L	V4-I, V4-II	Full
	24	B(A6-A7), B(A7,A8),B(A9-A8)	Partial
Mixed*	18L	P1, P3, E6-I, E6-II	Partial
	36C	P4, P5, B(A25-A26), Q	Partial
	36L	P6, P7, V4-I, V4-II	Partial

AAS established eight criteria for uncoupling locations [4]. At this stage, no location is found that fully meets all criteria. Assumptions were therefore made when selection the uncoupling points. The criteria and corresponding research specific-assumptions are given below.

1 **The uncoupling point should have as little negative influence on the flow on taxiways as possible.** This criteria is studied in this research. It was therefore not considered when selecting uncoupling points. The expected flow disturbances are given in column four of Table A.1.

2 **The uncoupling point requires a straight nose wheel, for the Taxibot to uncouple.** This criteria is implies that at least 10 meters of straight driving is available before the uncoupling point. The criteria is met for all selected locations except for V4-I. It was considered to move V4-1 to the stopbar 15 meters ahead. This would however require Taxibots to pass aircraft on the taxiway to reach the all-clear point and moreover, violate criteria 7. Another option was to move V4-1 in front of the turn. However, this would require two additional service road segments, instead of one (see Appendix A.3). A proper trade-off must be made between the two option. The nose wheel problem is therefore neglected for V4-1 in this research.

3 **The pilot is able to identify the right location to stop, also in the dark or with reduced visibility conditions.** This criteria is assumed to be solved for all locations (either by proper lights and markings, or by autonomous operations).

4 **The truck driver is able to safely exit the Taxibot at all times.** In the current (conventional) uncoupling process, the Taxibot driver must disembark the vehicle to unplug the communication cable from the aircraft. This mainly cause safety issues for uncoupling points located on taxiways. It is assumed that this safety issue is resolved by one of the following solutions: 1) a system that automatically unplugs the communication cable, 2) a communication system that makes the cable redundant, or 3) a method developed to mitigate the risk of truck drivers related to other traffic and weather.

5 **The truck driver is able to safely drive towards its all-clear point.** The Taxibot movement from the uncoupling location to the all-clear point is assumed to be part of the uncoupling operation and not specifically modelled. It is for all uncoupling point assumed that safe procedures for driving to the all-clear points are developed.

6 **The truck driver is able to position its Taxibot at a location clear of all traffic, without interrupting other (car) traffic.** This criteria, similarly to criteria 1, addresses flow disturbance issues. This is the main subject studied in this research and therefore not considered when selecting the locations.

7 **The truck driver and pilot are able to have visual communication, also in the dark or with reduced visibility conditions, for the safe all-clear signal.** This is true for all all-clear points selected.

8 **The Taxibot is able to move away (safely) from the uncoupling area for its next mission.** In this research, Taxibots are allowed to move over active taxiways if no service road is present. The conflict detection and avoidance system ensures safe movements.

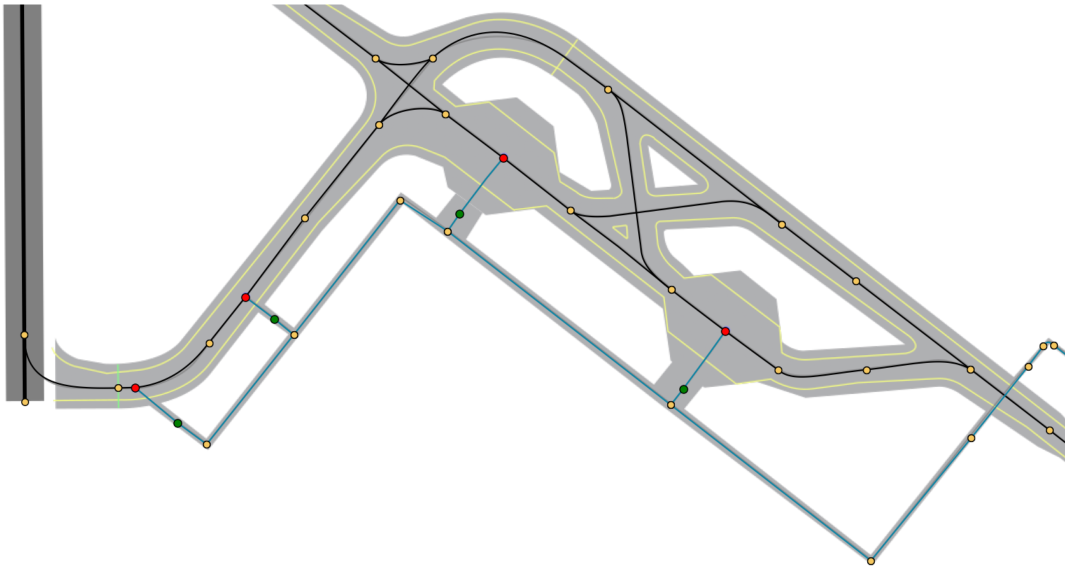


Figure A.2: The graph representing the layout of AAS at runway 36L. Uncoupling nodes from left to right: V4-I (west), V4-II (south-west), P7 (north-west), and P6 (north-west).

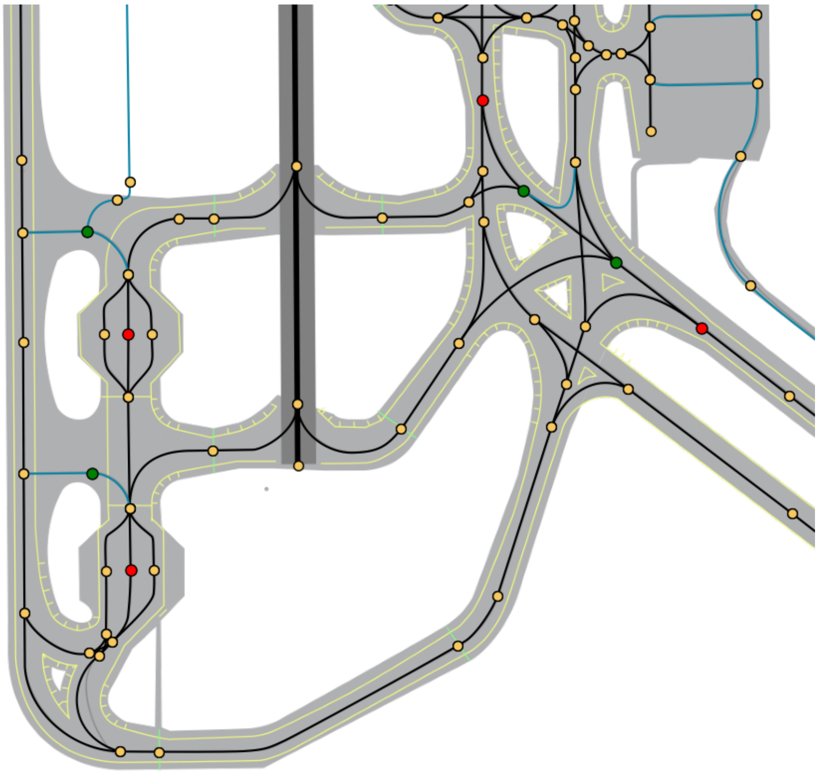


Figure A.3: The graph representing the layout of AAS at runway 36C. Uncoupling nodes from left to right: P4 (down)(north), P5 (top)(north), B(A25-A26)(south), Q(north-west).

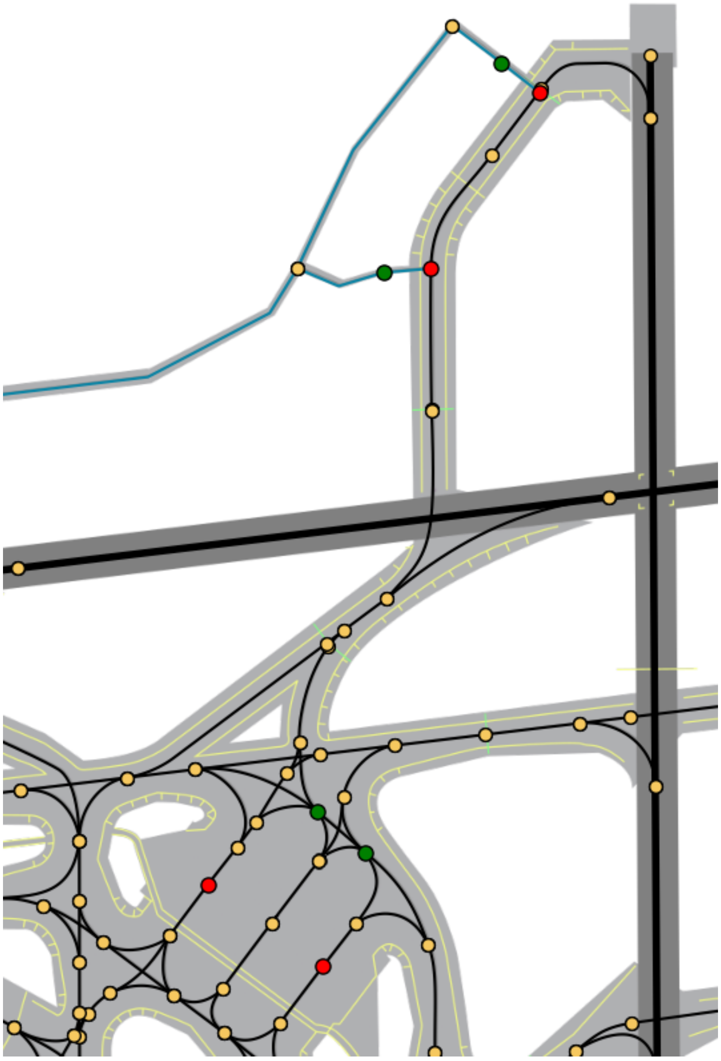


Figure A.4: The graph representing the layout of AAS at runway 18L. Uncoupling nodes from left to right: P1 (north-east), P2 (north-east), E6-II (north), E6-I (north-east).

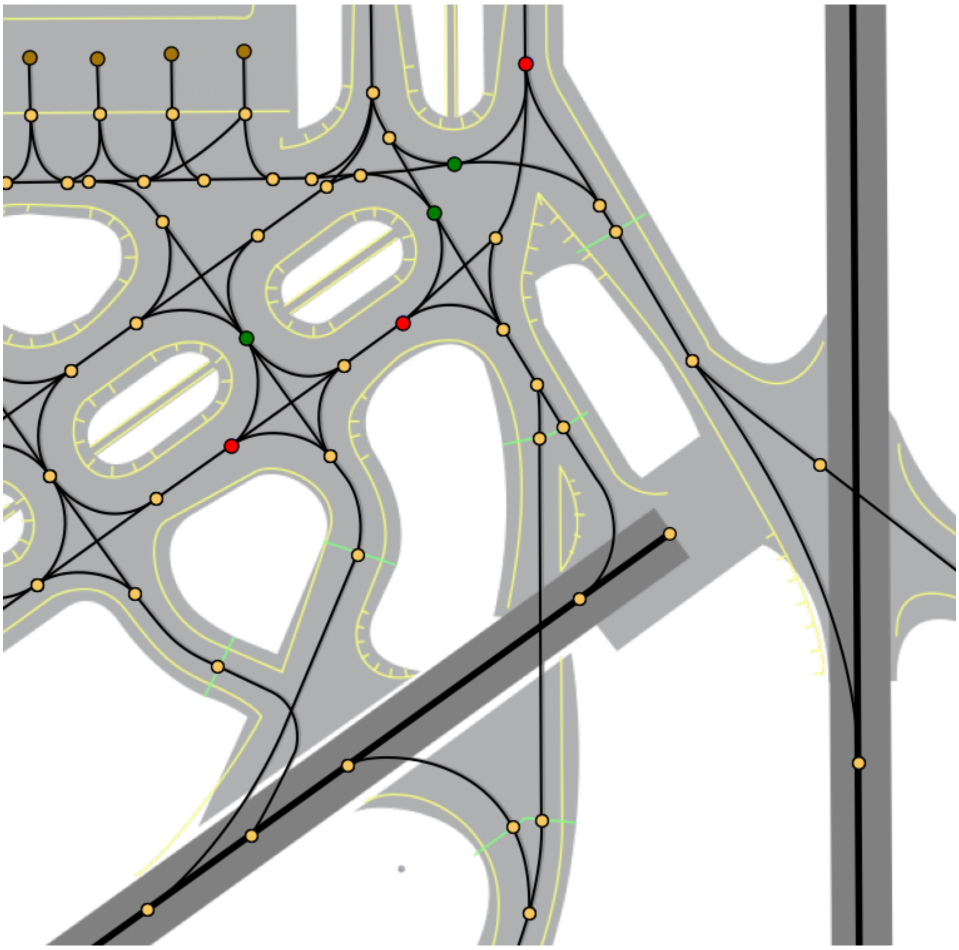


Figure A.5: The graph representing the layout of AAS at runway 24. Uncoupling nodes from left to right: B(A6-A7)(north-east), B(A7, A8)(north-east), B(A9, A8)(south).

A.3. Taxibot Service Roads

Taxibots must return to the parking facility after uncoupling. We selected one parking facility, located near the H-4 platform. Taxibots are allowed to drive on both Taxiways and several selected service roads. The service roads selected for this research are discussed in this section.

A circular service road network is located in the center of AAS and connected to the parking facility. The roads are visualised by the blue line in Figure A.6. The roads have enough capacity to facilitate the movement of all Taxibots for outbound flights. The central network is accessible through entries located at taxiway Alpha. To reach these locations, Taxibots are directed over taxiway Alpha, Bravo and some service roads in the outer area of the airport.

The service roads used by Taxibots after Taxiway uncoupling at 18L are visualised by the blue lines in Figure A.7. The small section on which the all-clear point of E6-I is located is non existing in reality. The section on which the all-clear point of E6-I is located does exist.

The service roads used by Taxibots after uncoupling at 36L are visualised by the blue lines in Figure A.8. The small section on which the all-clear point of V4-II is located is non existing in reality. The section on which the all-clear point of V4-I is located does exist.

The service roads used by Taxibots after holding platform uncoupling at 36C are visualised by the blue lines in Figure A.9. All infrastructure already exist. The service road heading north is assumed wide enough for Taxibot movements.

The service road used by Taxibots after taxiway uncoupling at 36C is visualised by the blue line in Figure A.10. All infrastructure already exist. The small service road is located on the taxiway and allow Taxibots to make a turn invalid for aircraft. This turn is used after uncoupling at uncoupling point B(A25-A26).

The service roads used by Taxibots after holding platform uncoupling at 36C or holding platform and taxiway uncoupling at 36L are visualised by the blue lines in Figure A.10. All infrastructure already exist. The roads are assumed wide enough for Taxibot movements.

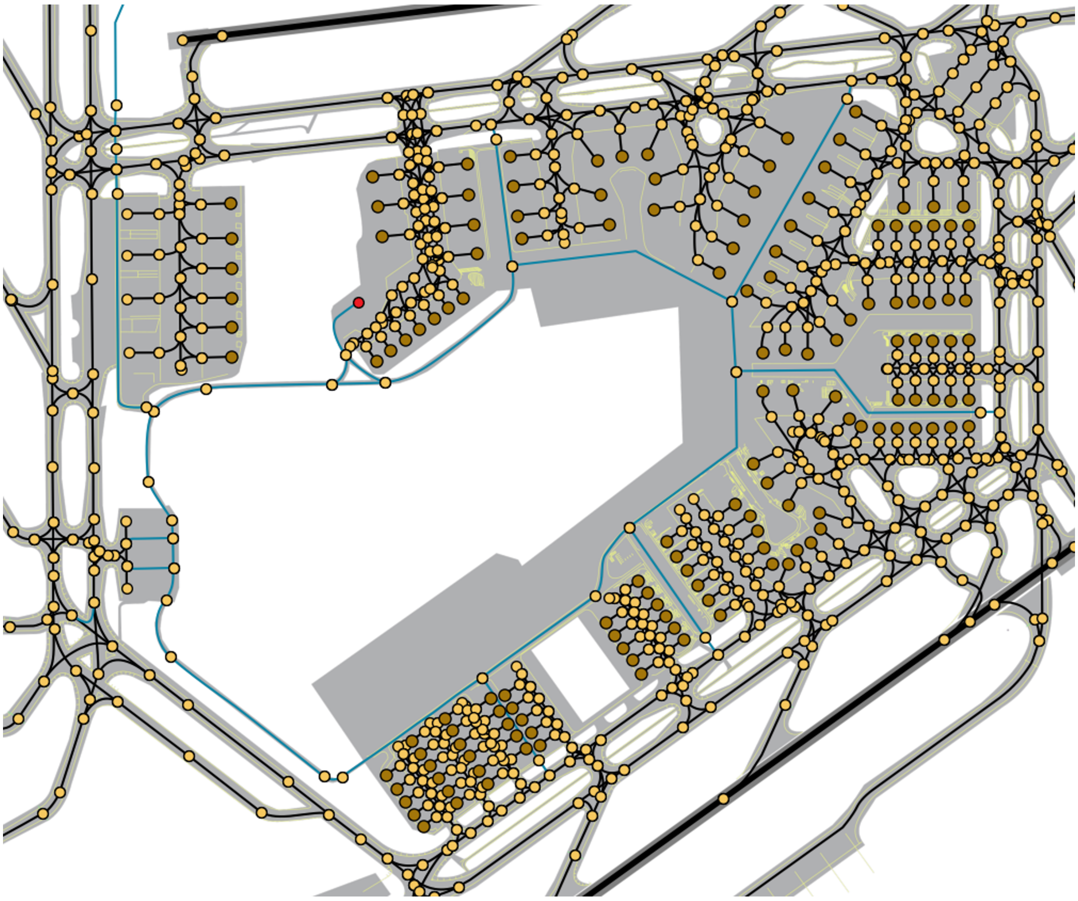


Figure A.6: Central circular service road. The parking facility is indicated in red.

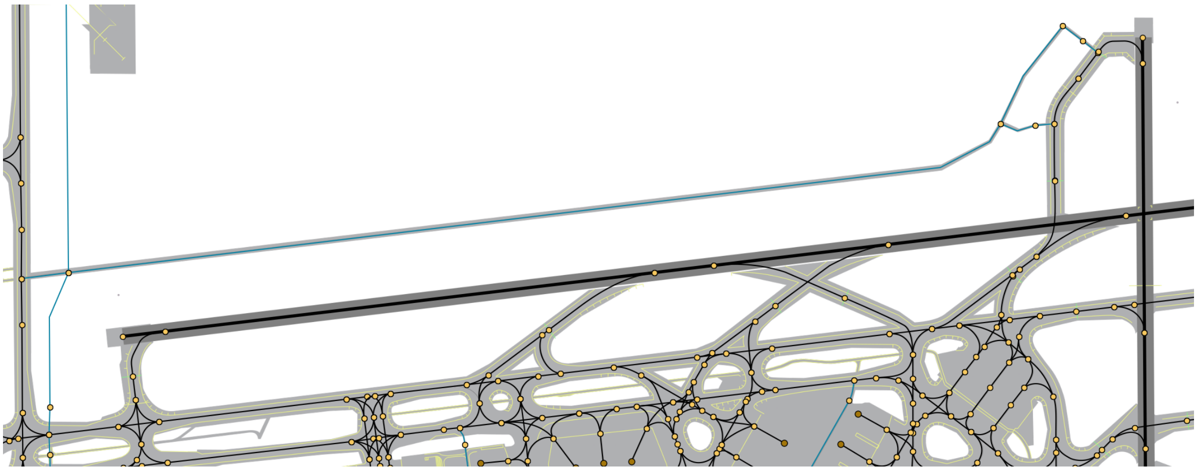


Figure A.7: Service road used for Taxiway uncoupling at 18L.

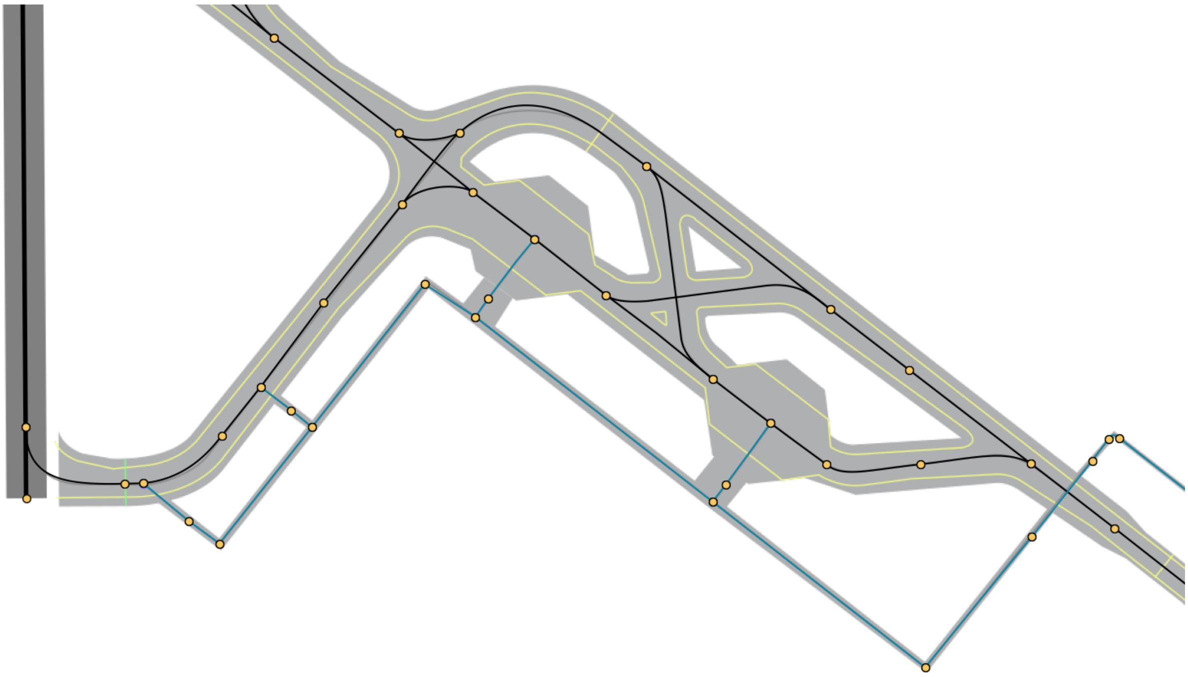


Figure A.8: Service road used for Taxiway and holding platform uncoupling at 36L.

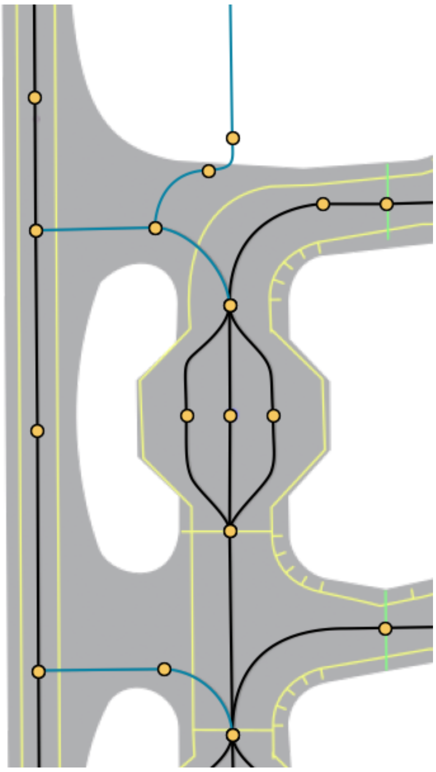


Figure A.9: Service road used for holding platform uncoupling at 36C.

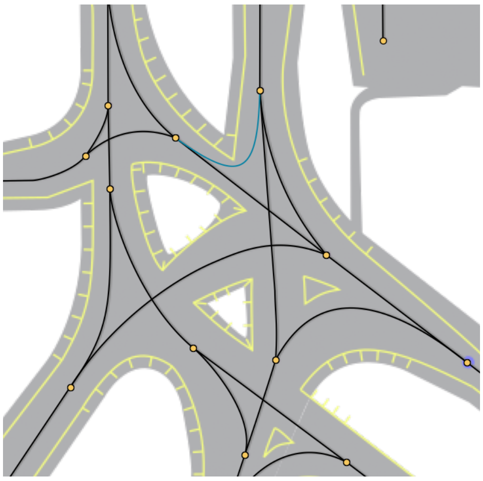


Figure A.10: Service road used for taxiway uncoupling at 36C.

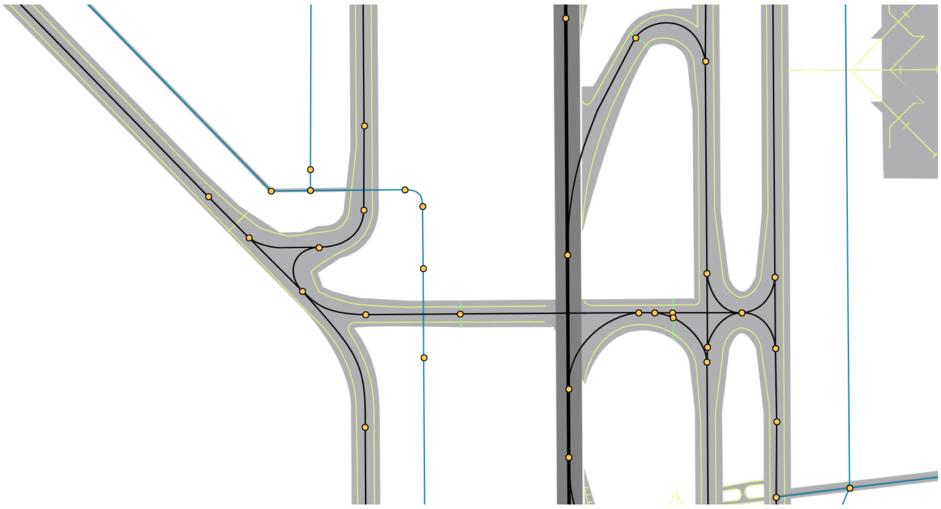


Figure A.11: Service road used for holding platform uncoupling at 36C and holding platform and taxiway uncoupling at 36L.

A.4. Operational Assumptions

This section first elaborated on some general operational assumptions. Then, the four simulated operations and their associated assumptions are given.

A.4.1 General

1. **Regular Aircraft flights and Taxibots are simulated.** No other support vehicles are considered. Aircraft relocation, holding, deicing and other supporting operations are neglected.

2. **Only commercial and passenger aircraft are simulated.** This is controlled by means of gates. All aircraft that used gates located on the K, R, S, and U platform were not taken into account in simulation. The number of flights using these gates on 17 and 18 July 2019 was 92 (5.5 % of all aircraft). These additional aircraft will likely result in additional delays. However, they are expected to be minor as private and general aviation often use runway 04, and cargo aircraft do not depart more during outbound peaks (those moments when increased delays occur).

3. **All outbound aircraft are towed by Taxibot.** As described in previous work, there are several operational and environmental reasons to only consider outbound towing [11] in this research. The modular

setup of the path planning algorithm does however allow for easy implementation of inbound towing.

4. **Flights are simulated based on real-world flight data.** Real-world data has been obtained from To70 aviation. This data includes the gate, runway, flight direction, ALDT, TOBT, and the aircraft type. The aircraft type is matched to one of the ICAO aircraft size categories, as given in Table B.6.

5. **Separation is required on taxiways.** Separation between aircraft and aircraft and Taxibots is maintained by the conflict avoidance mechanism. A safety distance of 20 meters is used. For all aircraft, an additional safety offset in the planning radius of 5 meters was used. See Appendix B.3 for more information.

6. **Separation is required on runways.** Separation on runways is maintained by the WTC separation minima set by RECAT-EU. Refer to Appendix B.4 for more information.

7. **All operations are performed according to a set of associated kinematics.** Kinematics are used in the path planning algorithm to plan routes. They include finite acceleration/deceleration rates and maximum velocities for both straight and curved segments. Refer to Appendix B.2 for more information.

8. **All operations are modelled as two-dimensional shape.** All agents are modelled as a 2-Dimensional shape. Refer to Appendix B.3 for more information.

9. **All tasks are executed perfectly.** The paths and associated operations obtained determined by the augmented path planning algorithm are assumed to be executed perfectly. This means that all external factors like human behaviour and meteorological considers are neglected.

A.4.2 Taxibotting

1. **Taxibot coupling is not considered.** Aircraft are assumed attached to Taxibots at the moment of departure. Aircraft are therefore never delayed by the unavailability of Taxibots.

2. **Taxibot movements from the parking facility to the gate is not considered.** All gates are connected to the central service road network visualised in Figure A.6. The Taxibot movements from the parking facility to the gates is not simulated. Given the assumption that Taxibots are attached at the moment of departure, these movements will not influence aircraft ground delays, and are therefore assumed redundant for this research.

3. **Sufficient Taxibots are available.** Sufficient Taxibots are available to meet all demand. All Taxibot moreover have sufficient fuel to execute all tasks perfectly.

4. **Physical shape is equal to the shape of the aircraft only.** The combined shape of aircraft and Taxibot is in reality larger than the aircraft only, as the Taxibot sticks out in front of the plane. This is however neglected in the simulation for simplicity.

A.4.3 Uncoupling

1. **Uncoupling locations.** The uncoupling locations and associated assumptions were discussed in Appendix A.2. No other assumptions apply besides those.

2. **The Taxibot movement from the uncoupling point to the all-clear point is not considered.** This movement is small and therefore not simulated. One of the assumptions made in Appendix A.2 states that this small path is always free of obstacles and other movements.

3. **All engines are warmed-up at the end of uncoupling.** It is assumed that all engines are warmed-up during Taxibotting and uncoupling. Engine warm-up during Taxibotting is currently not allowed for several aircraft types (for example the Boeing 737 series).

4. **The length of the taxibotting operation is always exactly t_u.** The uncoupling duration (t_u) is varied in this research to create multiple uncoupling scenarios. The uncoupling duration is always exactly t_u. It is however possible that aircraft are not able to leave the uncoupling location when uncoupling is finished due to the presence of other prioritised aircraft. In that case, Aircraft can hold longer at the

uncoupling point to give way to the prioritised aircraft. Taxibots still depart from the all-clear point at the moment uncoupling is finished.=.

5. **Start and final velocity is zero.** Upon arrival and departure, the velocity of the aircraft must be zero. This means that the goal velocity of Taxibotting, and the initial velocity of MET for outbound aircraft is set to zero.

6. **Physical shape is equal to the shape of the aircraft only.** During uncoupling, the Taxibot is moving around the aircraft. In the conflict avoidance mechanism, the shape of the aircraft and Taxibot is however assumed to remain simular to that of the aircraft. One of the assumptions made in Appendix A.2 states that the Taxibot movement area between the uncoupling location and the all-clear point is free of obstacles.

A.4.4 Multi-engine Taxiing

1. **Gates are always available.** The occupancy of gates is not taken into account. Aircraft are always directed to the gate stated in the flight schedule. Arriving aircraft are never delayed by the unavailability of their gate.

2. **Start velocity is zero for outbound aircraft.** For outbound aircraft departing at the gate, the velocity is set to zero. This means that the initial velocity of the Taxibotting operation is set to zero.

3. **Start velocity is v_{max} for inbound aircraft.** For inbound aircraft leaving the runway, the velocity is set to v_{max}. In reality, arriving aircraft can have a higher taxi speed than v_{max} on rapid exit taxiways. This is however neglected in simulation.

4. **Final velocity is zero for inbound aircraft.** For inbound aircraft arriving at the gate, a velocity of zero is required. This means that the goal velocity of the multi-engine taxiing operation is set to zero.

5. **Final velocity for outbound aircraft can be anything.** For outbound aircraft arriving at the runway, any velocity is allowed (as long it respects individual kinematic constraints). The path planning algorithm thus has the freedom to select any final velocity.

A.4.5 Taxibot Return Movement

1. **The parking facility has enough capacity.** Unlimited Taxibots are assumed available in the parking facility with unlimited capacity. All other operations inside the parking facility (refuelling, charging etc.) are not considered.

2. **Some service roads can be used.** Several, but not all, service roads are included in the simulation. Refer to Appendix A.3 for more information.

3. **Velocity of zero at the parking facility.** A velocity of zero is required in the parking facility. This means that the goal velocity of the Taxibot return movement operation is set to zero.

A.5. Constraint Settings for Conflict Avoidance Mechanism

This section elaborates on the constrains set in the classification systems (Table 3, 4, and 5 of the MSc Paper). Refer back to Figure 2, 3, and 4 of the MSc Paper for edge types and traversal timepoints.

The following USI are set on edges:

1 The USI in the same direction prevents a deprioritised agent from simultaneously entering the same edge. The first constraint in the same direction prevents overtaking of a deprioritised agent in front of the prioritised agent. The second constraint in the same direction prevents overtaking of a deprioritised agent behind the prioritised agent. The USI in the opposite direction prevent a head on conflict with a deprioritised agent.

2 Not constrained. The edge is constrained as type 1 by the two adjacent edges on path.

3 Not constrained. The edge is constrained as type 5 by the previous edge on the path.

4 The USI in the same direction prevents a node collision with a deprioritised agent at the start node. The first constraint in the same direction prevents an edge collision from the side with a leading deprioritised agent. In this case, T_x is set equal to the absolute time at which the prioritised agent has traversed the length of edge T4 plus its planning radius. The constraint in the opposite direction prevents an edge collision with a deprioritised agent.

5 The constraint in the same and opposite direction prevents an edge and node collision with a deprioritised agent.

6 Not constrained. The edge is constrained as type 4 by the next edge on the path.

7 The constraint in the same and opposite direction prevents an edge and node collision with a deprioritised agent.

8 The outward USI prevent prioritised agents entering outward edges, meaning that no agents are able to cross the waiting node. No inward constraints are set to allow for queuing.

9 Similar constraints are set in both directions to prevent a collision with prioritised agents during waiting.

Runways The USI ensures that there is no deprioritised agent present on any runway edge a certain time before and after the agent crosses the stopbar node with the front of its shape.

The following regular constraints are set on edges:

1, right A deprioritised agent that has entered the edge after τ_2 must leave after τ_4. As a result, the prioritised agent cannot be overtaken by deprioritised agent from the back.

1, left A deprioritised agent that has entered the edge before τ_1 must have left the same edge before τ_3. As a result, the prioritised agent cannot be overtaken by deprioritised agent from the front.

4, left A deprioritised agent that has entered the type 4 edge after τ_2 must leave after η_4. η_4 represent the time point in which the prioritised agent travelled the length of the type 4 edge (till the back of the shape) on the type 1 edge. As a result, the deprioritised agent can never hit the prioritised agent from the back/side.

4, right A deprioritised agent that has entered the type 4 edge before τ_1 must leave before η_3. η_3 represent the time point in which the prioritised agent travelled the length of the type 4 edge (till the front of the shape) on the type 1 edge. As a result, the prioritised agent can never hit the deprioritised agent from the front/side.

The following USI are set on nodes:

N_{close} When v>0, vicinity nodes are blocked for waiting during the entire edge traversal. When v=0, vicinity nodes are blocked during the entire waiting action.

N_{wait} This constraint ensures that no agent waits at the waiting node if a prioritised agent waits at the node.

A.6. SIPP Anticipation Example

The anticipation method used in SIPP is more elaborately explained by means of 4 steps. Each step has been visualized to assist in the understanding of the concept. A simplified graph visualized in each step in Figure A.12 to Figure A.15 is used as example. In these figures, circles and lines represent nodes and edges. The scenario contains seven nodes (A,B,C,D,E,F,G) and seven edges (AB, BD, AC, CD, DE, EF, GF). Node D is a goal node. Yellow boxes on edges indicate safe intervals. Yellow, red and green boxes contain information of states. Yellow states are states from which the motion is considered. Green states require a feasible motion from the yellow state, and are named feasible states. Red states require an unfeasible motion from the yellow state, and are named infeasible states.

The basis safe interval $[-\inf, \inf]$ is set on each edge in both directions. Only ED contains two safe intervals: $[-\inf, 4]$, $[24, \inf]$, also in both directions. A maximum velocity of 15 m/s is sued. If edges are perpendicular, a turn must be made. Maximum turn velocity is 5 m/s. The velocity at a goal node is set to 0 m/s. The activity number is not included in the state independent variables, as the same activity is considered for the entire example. Finally, note that variables for edge lengths, deceleration and acceleration is neglected for simplicity.

Step 1. This step is visualized in Figure A.12. A state at node F is opened and explored. The next node is E, the velocity of the state is 15 m/s, and the interval number (defined on FE) is one. This state is called the *explored state*. All possible states in the next node (E) are explored. Two new states are found, one for each safe interval on ED. The state located in the first safe interval is marked green, as the motion to this state is feasible. The state in the second safe interval is marked red, as the motion towards that state requires a too high deceleration to arrive in the interval, making it infeasible. This state can be discarded as it would have been generated by anticipation during exploration of earlier states. The feasible state with the highest velocity is called the *prime motion*, and the state is named the *prime state*. This state is explored for anticipation.

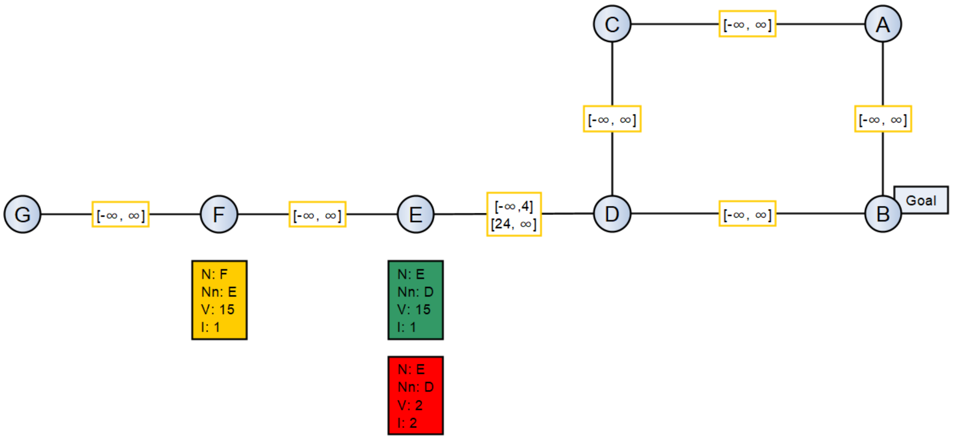


Figure A.12: Explore start state. Identify prime state.

Step 2. This step is visualized in Figure A.13. The prime state is explored for anticipation. All nodes within the braking distance of the prime state are gathered and their corresponding states are found. In node D, two states are generated: one to make a turn, and one to go straight. In node B, also two nodes are generated: one to finish the search, and one to make a turn. In node C, one state is generated to make a turn. It is now determined which of these five states require an infeasible. These unfeasible motion states are labelled in red. The states that require a feasible motion are marked in green. These states can be discarded as they will be generated by exploration of future states.

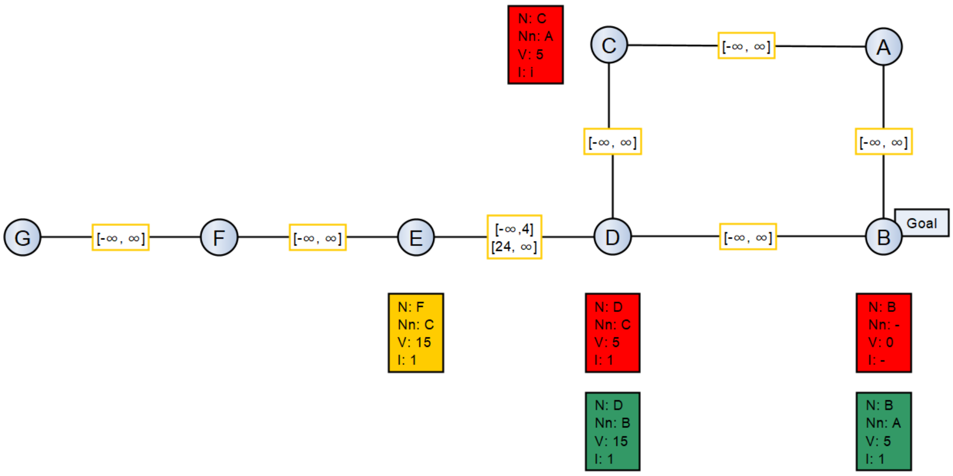


Figure A.13: Explore prime state. Identify unfeasible states.

Step 3. This step is visualized in Figure A.14. The states found to be infeasible from the prime state are tested to be feasible from the explored state. States can remain unfeasible if the required deceleration is still too high, velocity limits are violated at intermediate nodes, or safe intervals are violated at intermediate edges. In the example, two of these states are found to be feasible, marked in green. One state requires an unfeasible motion, as the turn speed in turn EDC is violated. This state can be discarded as it will be generated by exploration of future states, if possible.

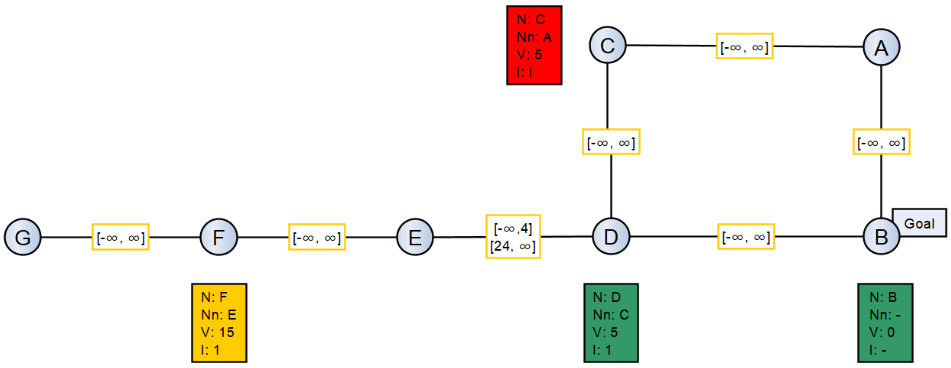


Figure A.14: Explore motions from start state to unfeasible states.

Step 4. This step is visualized in Figure A.15. All states reachable in a feasible motion from the *explored state* are generated. This includes the prime state found in step 1, and all state unfeasible from the prime state but feasible from the explored state, as found in step 3.

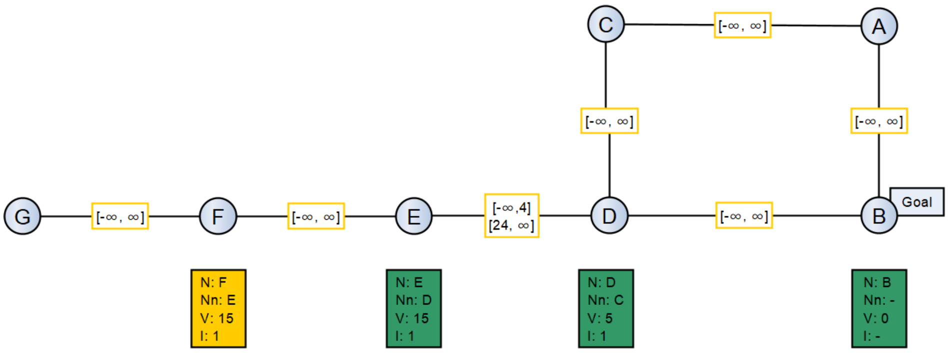


Figure A.15: Generate all feasible states.

B
Scenario Specifications

This chapters provides all scenario settings used in this research. The two days of operational data used for this research contained the aircraft type. These types were translated to one of the six ICAO aerodrome reference size categories as provided by ICAO Annex 14 [6]. An overview of all activities and associated objective coefficients is given in Appendix B.1. Each ICAO size category and the single Taxibot type is given specific kinematic and dimensions, as given in Appendix B.2 and Appendix B.3, respectively. The WTC constraints used during take-off are given in Appendix B.4. The RMO schedule of 17 and 18 July 2019 used for simulation is given in Appendix B.5. Finally, the mapping of aircraft types extracted from the flight data to one of the six aircraft categories is given in Appendix B.6.

B.1. Activity Objectives

For all four activities simulated in this research, three objective coefficients exist. Two of them represent the weight of traversing a path in terms of time (c_t) and distance (c_d). The other is the individual weight coefficients (c_a), used to express the relative importance of activities. The cost of traversing a distance d within time t during a certain activity is thus given by Equation (B.1). The objective coefficients used in this research are given in Table B.1. Note that the uncoupling activity does not include movement and therefore does not contain a c_d value.

$$cost = c_a \cdot (c_t \cdot t + c_d \cdot d) \tag{B.1}$$

Table B.1: Simulated operations and associated kinematics, dimensions, and objective coefficients.

			Objective		
Simulated operation	**Kinematics**	**Dimensions**	c_t	c_d	c_a
Taxibotting	Table B.2	Table B.3	1	0	1
Multi-engine taxiing			1	0	1.1
Taxibot return movements			0	1	0
Uncoupling	n/a		1	n/a	1

B.2. Kinematic Values

Vehicles move according to a specific set of kinematics. The kinematics that apply for each vehicle are given in Table B.2. v_{max} indicates the maximum possible velocity, anywhere on the airport. Maximum acceleration and deceleration (negative acceleration) apply. r_{turn} is the minimal radius of a turn at which the vehicle must deceleration to turn speed (or the maximum radius at which turn velocity is not required). v_{turn} is the associated turn speed. d_t is the distance needed for take-off. This distance is used to select a set of possible runway entrances. The data is based on AAS interview, Taxibot manuals, and Taxibot trail [4, 7**?**].

Table B.2: Kinematic data for all vehicles in every. When Taxibotting, the aircraft type is irrelevant. Abbreviations: TB, Taxibotting; UN, uncoupling; MET, multi-engine taxiing; TRM, Taxibot return movement; acc, acceleration; dec, deceleration.

Operation	**Vehicle(s)**	v_{max},$[m/s]$	a (acc),$[m/s^2]$	a (dec),$[m/s^2]$	r_{turn},$[deg]$	v_{turn},$[m/s]$	d_t,$[m]$
TB	*Aircraft + Taxibot*	11,8	0,2	-0,6	100	5	n/a
MET	*ICAO-A*	15	0,25	-0,75	100	5	1200
MET	*ICAO-B*	15	0,25	-0,75	100	5	1700
MET	*ICAO-C*	15	0,25	-0,75	100	5	2500
MET	*ICAO-D*	15	0,25	-0,75	100	5	2500
MET	*ICAO-E*	15	0,25	-0,75	100	5	3100
MET	*ICAO-F*	15	0,25	-0,75	100	5	3100
TRM	*Taxibot*	8,33	0,3	-0,9	50	5	n/a

B.3. Geometric Shapes

The dimensions of all vehicles used for simulation are given in Table B.3. A safety offset, d_s, of 20 meters was used for all vehicles. Data retrieved from ICAO Annex 14 [6]. An additional 5 meters safety distance is included to account for the generalisation of vehicles.

Table B.3: Dimensions of all vehicles in various operations used in simulation. Abbreviations: TB, Taxibotting; UN, uncoupling; MET, multi-engine taxiing; TRM, Taxibot return movement

Operation(s)	**Vehicle**	**Length (l)**	**Span (b)**	**Shape Radius (d_s)**
TB/MET/UN	*ICAO-A*	10	12	5
TB/MET/UN	*ICAO-B*	25	22	5
TB/MET/UN	*ICAO-C*	40	35,8	5
TB/MET/UN	*ICAO-D*	54	47,6	5
TB/MET/UN	*ICAO-E*	72	64,8	5
TB/MET/UN	*ICAO-F*	72,7	79,8	5
TRM	*Taxibot*	5	12	5

B.4. Wake Turbulence Separation Minima

Departing aircraft must adhere to WTC separation minima when crossing the stop bar. We chose to include the re-categorised (RECAT-EU) separation minima, in contrast to the ICAO WTC separation criteria currently applied by AAS [4, 10]. These separation minima - established by EUROCONTROL, the Federal Aviation Administration, research facilities and the aviation industry - are time-based separations optimised for runway capacity that well match the various aircraft types in the simulator. All six ICAO types used in simulation were therefore mapped to weight categories established by RECAT-EU. The WTC separation minima are given in Table B.4.

Table B.4: Time based WTC separation minima given in seconds. Separation minima established by RECAT-EU. Note: the ICAO aircraft types used in simulation are presented, not the size types of RECAT-EU (RECAT-EU uses A (largest) to F (smallest) - ICAO does the exact opposite).

Leader\Follower	*ICAO-A*	*ICAO-B*	*ICAO-C*	*ICAO-D*	*ICAO-E*	*ICAO-F*
ICAO-A	80	60	60	60	60	60
ICAO-B	100	60	60	60	60	60
ICAO-C	120	60	60	60	60	60
ICAO-D	120	100	80	60	60	60
ICAO-E	140	120	100	60	60	60
ICAO-F	180	160	140	120	100	60

B.5. RMO Schedule

The focus of this research was set to the two most commonly used RMO. RMO North was active on July 17. In RMO North, runway 36L is used as main departing runway and runway 36C is opened during outbound peaks. RMO South was active on July 18. In RMO South, runway 24 is used as main departing runway and runway 18L is opened during outbound peaks. The schedule is given in Table B.5.

Table B.5: RMO schedule of 17 and 18 July, 2019.

Day	Runway takeoff	Runway landing	Effective from:
17-7-2019	36L	06	00:00:00
17-7-2019	36L, 36C	06, 36R	06:45:00
17-7-2019	36L	06, 36R	07:45:00
17-7-2019	36L, 36C	06, 36R	09:15:00
17-7-2019	36L	06, 36R	11:15:00
17-7-2019	36L, 36C	06, 36R	11:45:00
17-7-2019	36L	06, 36R	14:00:00
17-7-2019	36L, 36C	06, 36R	14:15:00
17-7-2019	36L	06, 36R	15:30:00
17-7-2019	36L, 36C	06, 36R	16:15:00
17-7-2019	36L	06, 36R	17:45:00
17-7-2019	36L, 36C	06, 36R	20:30:00
17-7-2019	36L	06, 36R	22:00:00
18-7-2019	36C	18R	00:30:00
18-7-2019	24	18R	01:50:00
18-7-2019	24, 18L	18R, 18C	06:55:00
18-7-2019	24	18R, 18C	07:45:00
18-7-2019	24, 18L	18R, 18C	09:10:00
18-7-2019	24, 18L	18R	09:35:00
18-7-2019	24, 18L	18R, 18C	10:10:00
18-7-2019	24, 18L	18R	11:45:00
18-7-2019	24, 18L	18R, 18C	12:35:00
18-7-2019	24, 18L	18R	14:50:00
18-7-2019	24, 18L	18R, 18C	13:45:00
18-7-2019	24	18R, 18C	15:40:00
18-7-2019	24, 18L	18R, 18C	16:00:00
18-7-2019	24, 18L	18R	16:15:00
18-7-2019	24	18R	17:45:00
18-7-2019	24	18R, 18C	18:25:00
18-7-2019	24, 18L	18R	20:30:00
18-7-2019	24	18R	22:10:00

B.6. Mapping Aircraft Types

The real-life data obtained by To70 aviation contained the aircraft type for each flight. These aircraft types were mapped to one of the six ICAO size categories used in simulation according to Table B.6.

Table B.6: Aircraft types and ICAO size categories used in simulation.

Manufacturer	Name	Vehicle
Cirrus	Cirrus Vision SF50	ICAO-A
Bombardier	Learjet all types (largest aircraft)	ICAO-B
Embraer	EMB-120 Brasilia Adv.	ICAO-B
Canadair	Canadair CRJ-100/200ER	ICAO-B
Cessna	All types (largest dimensions)	ICAO-B
Dassault	Falcon all types (largest aircraft)	ICAO-C
Bombardier	Q400	ICAO-C
Embraer	Embraer 170/190	ICAO-C
Boeing	B737-300	ICAO-C
Gulfstream	All types (largest aircraft)	ICAO-C
Boeing	B737-300 + winglets	ICAO-C
Airbus	A318-100 (largest 318)	ICAO-C
Boeing	B737-900	ICAO-C
Boeing	B737-600	ICAO-C
Boeing	B737-700	ICAO-C
McDonnell Douglas	McDonnell Douglas MD-80	ICAO-C
Boeing	B737-800	ICAO-C
Bombardier	CS-300 Bombardier	ICAO-C
Boeing	B737-800 NG Winglets	ICAO-C
Airbus	A320-200 Sharklets (largest 320)	ICAO-C
Boeing	B737-700 NG Winglets	ICAO-C
Airbus	A319-100/LR/CJ (largest 319)	ICAO-C
Airbus	A321-200 (largest 321)	ICAO-C
Airbus	A320-200	ICAO-C
Airbus	A310-200/300 Freighter	ICAO-D
Airbus	A300F Freighter	ICAO-D
Airbus	A300F Freighter	ICAO-D
Boeing	B767-300	ICAO-D
Airbus	A330-300	ICAO-E
Airbus	A330-200	ICAO-E
Airbus	A340-300E	ICAO-E
Boeing	B747-400 + 400ER	ICAO-E
Boeing	B747-400 Combi	ICAO-E
Airbus	A350-900	ICAO-E
Airbus	A350-1000	ICAO-E
Boeing	B777F	ICAO-E
Boeing	B747-800F	ICAO-F
Airbus	A380	ICAO-F
Boeing	B777	ICAO-E
Boeing	B787	ICAO-E

C
Model Verification

This sections provides examples of steps taken during verification of the developed model. Appendix C.1 provides an explanation of the uncoupling and take-off schedules used for high-level verification of the uncoupling operations. In Appendix C.2, an example of a search tree visualisation is given, used to verify the optimisation of the PBS search. Appendix C.3 provides the simplified layout used during testing. Appendix C.5 presents an example of the velocity profiles found during the SIPP search and used during execution. The performance of the initial path planning algorithm was tested in the midterm of this thesis and is provided in Appendix C.4. *The visualisation interface for execution is provided in Appendix C.6. Heat maps to track large number of routes are presented in Appendix C.7.

C.1. Uncoupling and Take-off Schedules

The uncoupling and take-off schedules were created to verify the uncoupling operations on a high-level. An example can be found for the mixed uncoupling strategy of runway 36L in Figure C.1. One hour of uncoupling operations is plotted. This mixed configuration has 4 uncoupling locations. Every uncoupling location has one horizontal timeline. The yellow bars represent an uncoupling operation at the location associated with the timeline. The horizontal dashed line to the right of the yellow bar indicate the multi-engine taxi phase. The vertical dashed line at the end of this line indicates take-off. The blue arrows on top of the schedule visualises take-off separation to verify that WTC constraints are met.

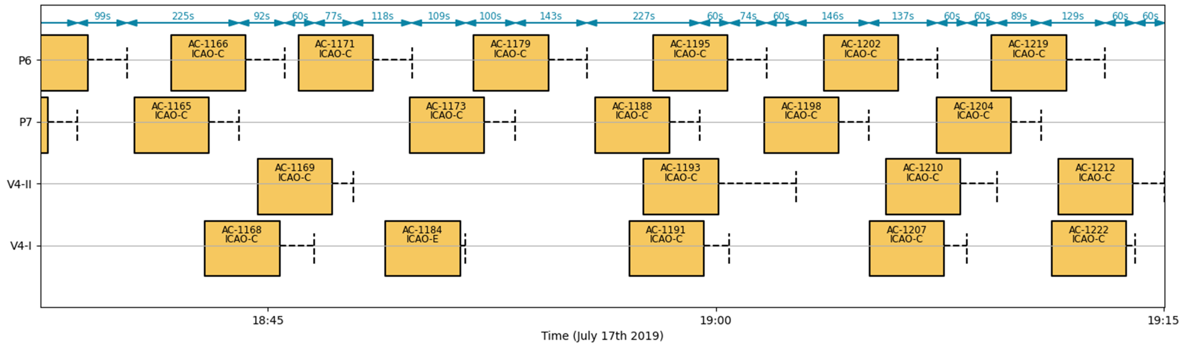
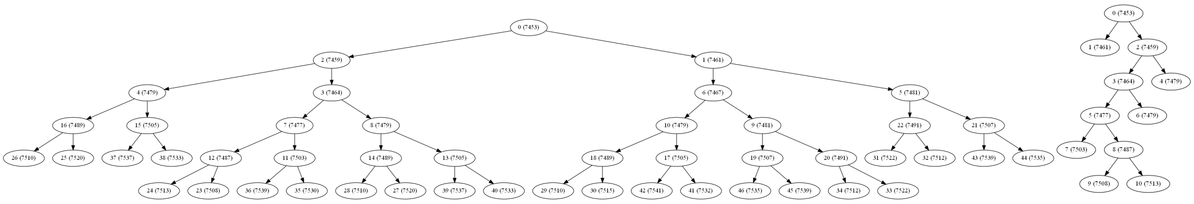


Figure C.1: Example of uncoupling schedule. Example given for mixed uncoupling strategy at runway 36L.

C.2. PBS Search Tree Visualisation

PBS search tree exploration is tested throughout different stages of development by visualising the search tree in terms of solution cost and exploration order. An example can be found in Figure C.2 for the best- and depth-first search. In each node, it's identification number is plotted followed by its cost in parentheses. A node identification number is only assigned when nodes are generated. Hence, one could follow the behaviour of the search tree for verification. In this case, the outcome of the best-first search is identical to



(a) Best-first search.

(b) Depth-first search.

Figure C.2: PBS priority tree visualisation of two developed search techniques. The depth-first search is adopted for the results presented in this research.

the outcome of the depth-first search. For the results presented in the research, the depth-first search was adopted.

C.3. Simplified Initial Layout

The simplified layout used for verification through different stages of development is given in Figure C.3. The layout represents taxiway A,B,C,D and Q. Red nodes indicate gates, and pink nodes runway exit/entries. The layout was obtained by previous work presented by Benda, Fines and Noortman [3, 5, 8].

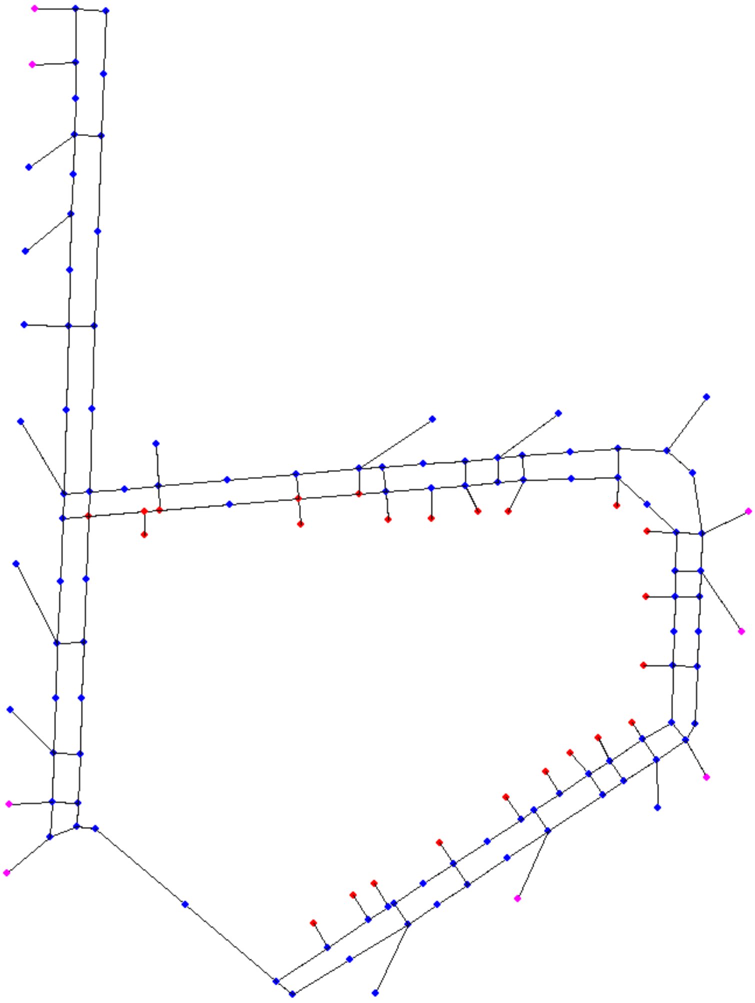


Figure C.3: Simplified layout used in initial stages of model development. Layout based on previous work by B. Benda [3].

C.4. Path Planning Performance

The performance of the path planning algorithm was tested throughout multiple stages of development. An example of measurements in the average cost per vehicle, algorithmic runtime and success-rate were previously presented in [12] for four different search methods and three different search windows. These measurements were done on the simplified layout, given in Figure C.3. The main results of these analyses are shown in Figure C.4.

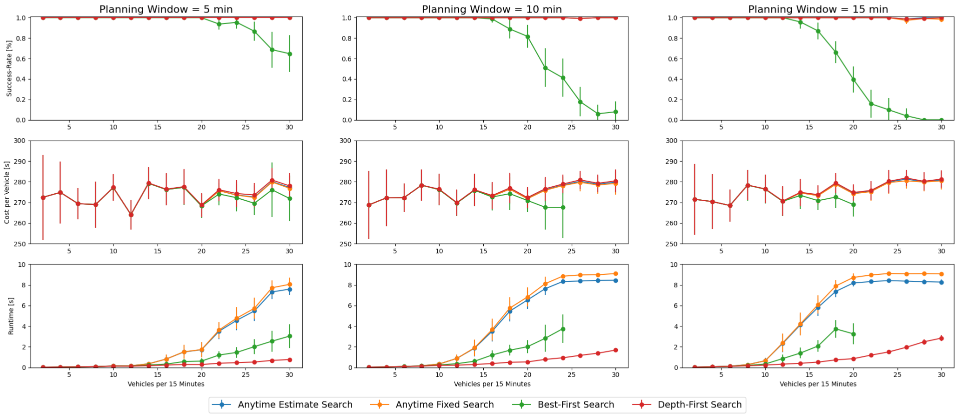


Figure C.4: PBS + SIPP fastest path ($c_t = 1$, $c_d = 0$) performance in different search windows and search strategies (c_v stabilisation of 5% for 50 runs, 95% confidence intervals).

C.5. Individual Aircraft Routes

Individual routes were verified by velocity maps, and distance and velocity plots. [1] An example is given for aircraft 586 (ICAO-C) in Figure C.5 and Figure C.6. Kinematic properties of the aircraft can be found in Table B.2. It can be observed that the uncoupling process happens between around 1530 and 1650 (t_u = 2:00, normal duration) seconds (Figure C.6) at P7 (Figure C.5). Before uncoupling, Taxibotting occurs, meaning that the kinematic properties in the first row of Table B.2 apply. After uncoupling, multi-engine taxiing occurs, meaning that the kinematic properties in the fourth row (ICAO-C) of Table B.2 apply. Differences between the operations can indeed by observed by a different maximum velocity and acceleration rate. Turn velocities can be verified on Taxiway Alpha and Quebec for Taxibotting at around 940, 1000, and 1100 seconds.

[1] The figures of this section are based on collaborative work with equal contribution by three students from Delft University of Technology. Team members: Malte von den Burg (PhD candidate), Jorick Kamphof (MSc) and Joost Soomers (MSc).

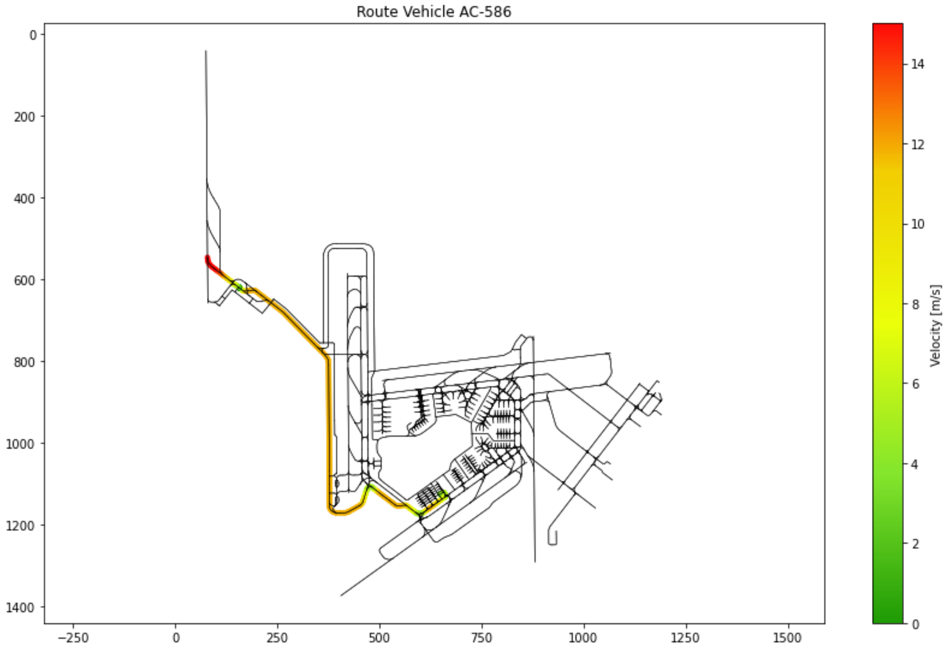


Figure C.5: Velocity map of aircraft 586 (ICAO-C).

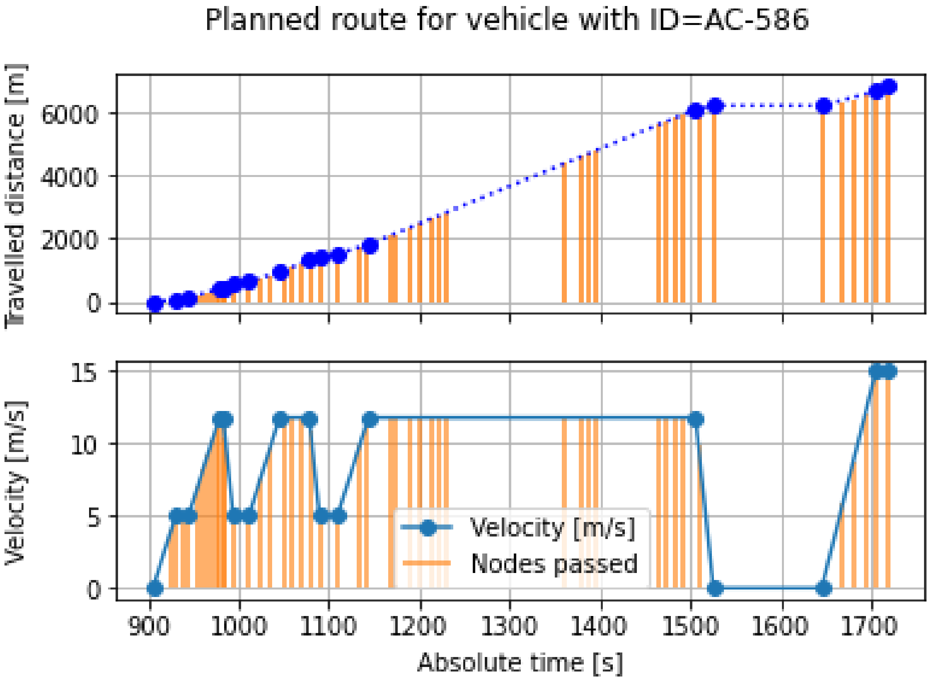
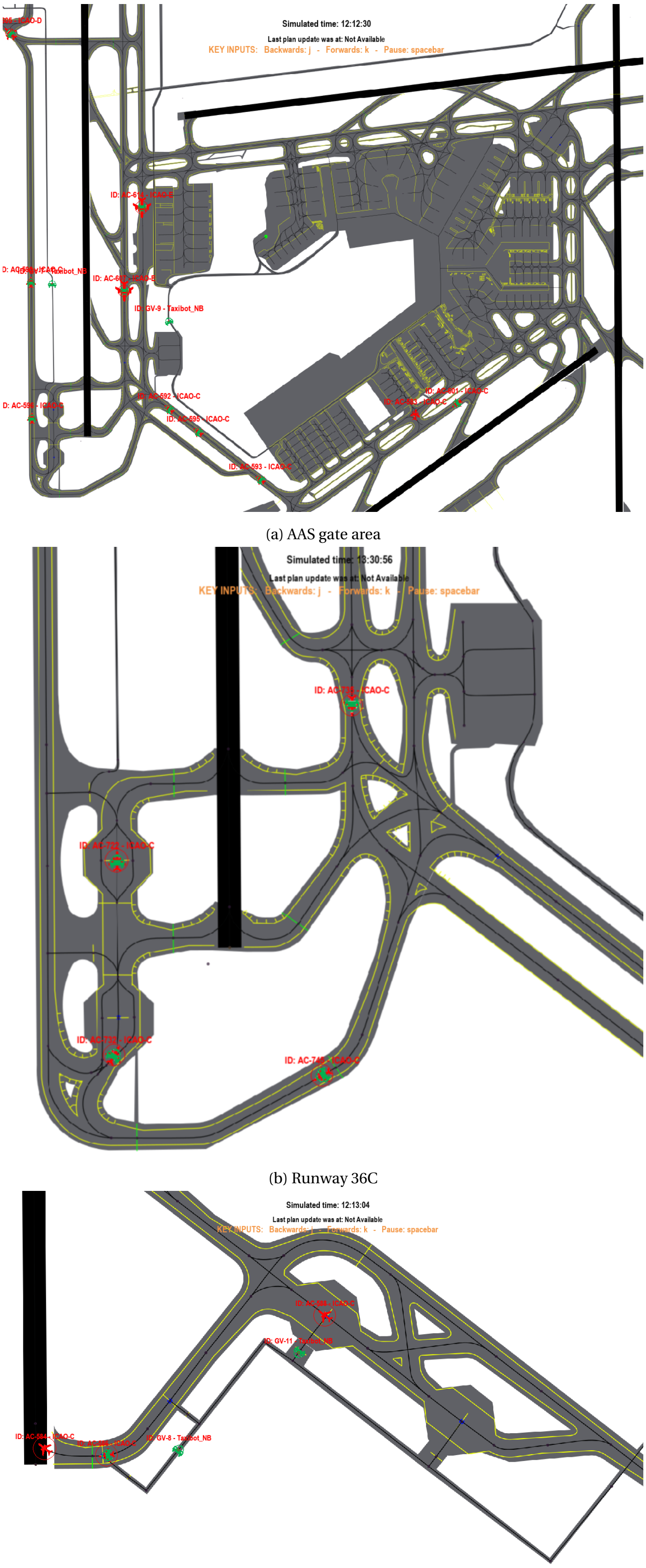


Figure C.6: Distance and velocity plot of aircraft 586 (ICAO-C).

C.6. MAS Execution

A screenshot of the simulator used to verify all operations is given in Figure C.7. [2] One can observe uncoupling operations at V4-I and P7 for runway 36L, and at P5 and B(A25-A26) for runway 36C.

[2] The figures of this section are based on collaborative work with equal contribution by three students from Delft University of Technology. Team members: Malte von den Burg (PhD), Jorick Kamphof (MSc) and Joost Soomers (MSc).



Simulated time: 12:12:30

Last plan update was at: Not Available

KEY INPUTS: Backwards: j - Forwards: k - Pause: spacebar

(a) AAS gate area

Simulated time: 13:30:56

Last plan update was at: Not Available

KEY INPUTS: Backwards: j - Forwards: k - Pause: spacebar

(b) Runway 36C

Simulated time: 12:13:04

Last plan update was at: Not Available

KEY INPUTS: Backwards: j - Forwards: k - Pause: spacebar

(c) Runway 36L

Figure C.7: Screenshots of simulator.

C.7. Airport Heatmaps

Aircraft heatmaps were deployed to visualise the behaviour of aircraft groups and detect ground bottlenecks. The heatmaps present the number of aircraft that traversed the edges. An example heatmap of the entire airport is given in Figure C.8. Two hours of the mixed uncoupling strategy with normal uncoupling duration is presented at 17 July. On this day, RMO North was active. High traffic volumes can be observed at the single Quebec lane and the taxiway at the south side of 36C. The runway crossing at 36C towards 36L is not open as 36C is active in the entire two hours of simulation. Figure C.9 presents the entrance of 36C for a mixed, taxiway and holding platform uncoupling strategy. Aircraft are most nicely distributed in the mixed uncoupling strategy. In the taxiway uncoupling strategy, aircraft enter 36C only from the east. vice versa, in the holding platform uncoupling strategy, aircraft enter 36C only from the west. The holding platform strategy leads to high traversals of the taxiway south of 36C. Figure C.10 presents the entrance of 36L for a mixed, taxiway and holding platform uncoupling strategy.

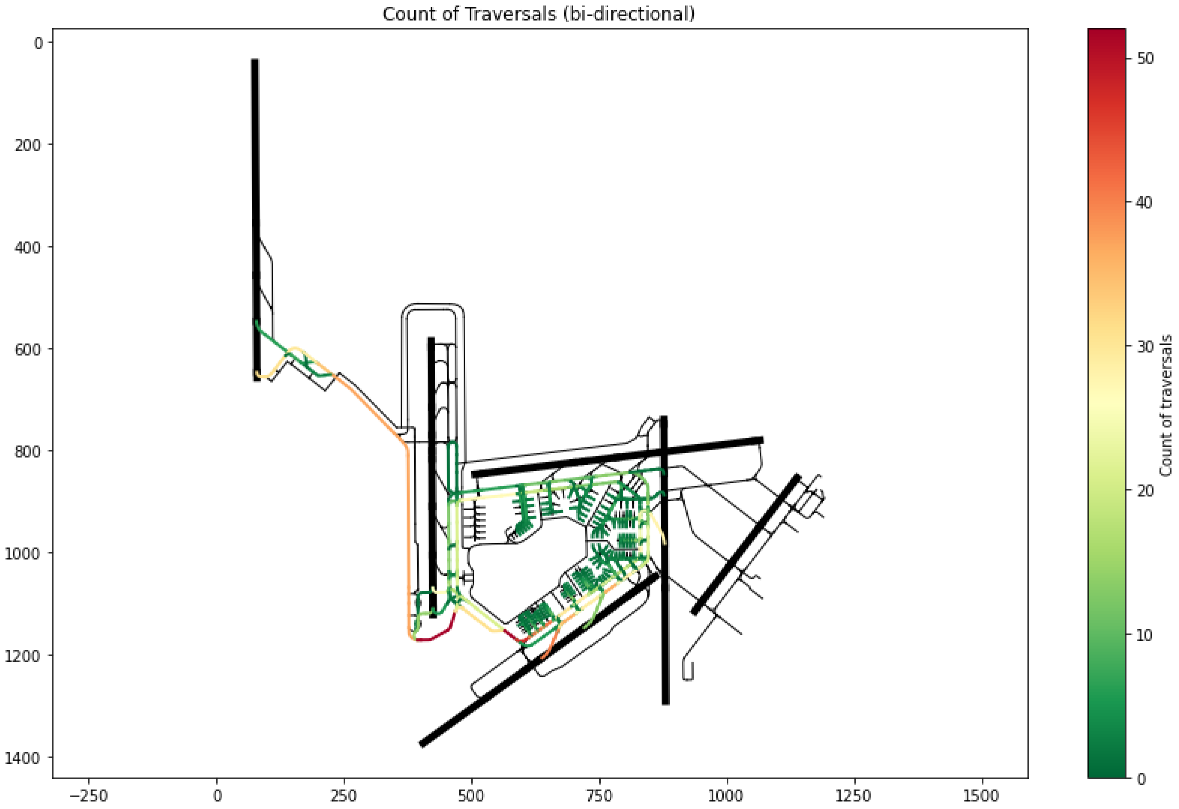
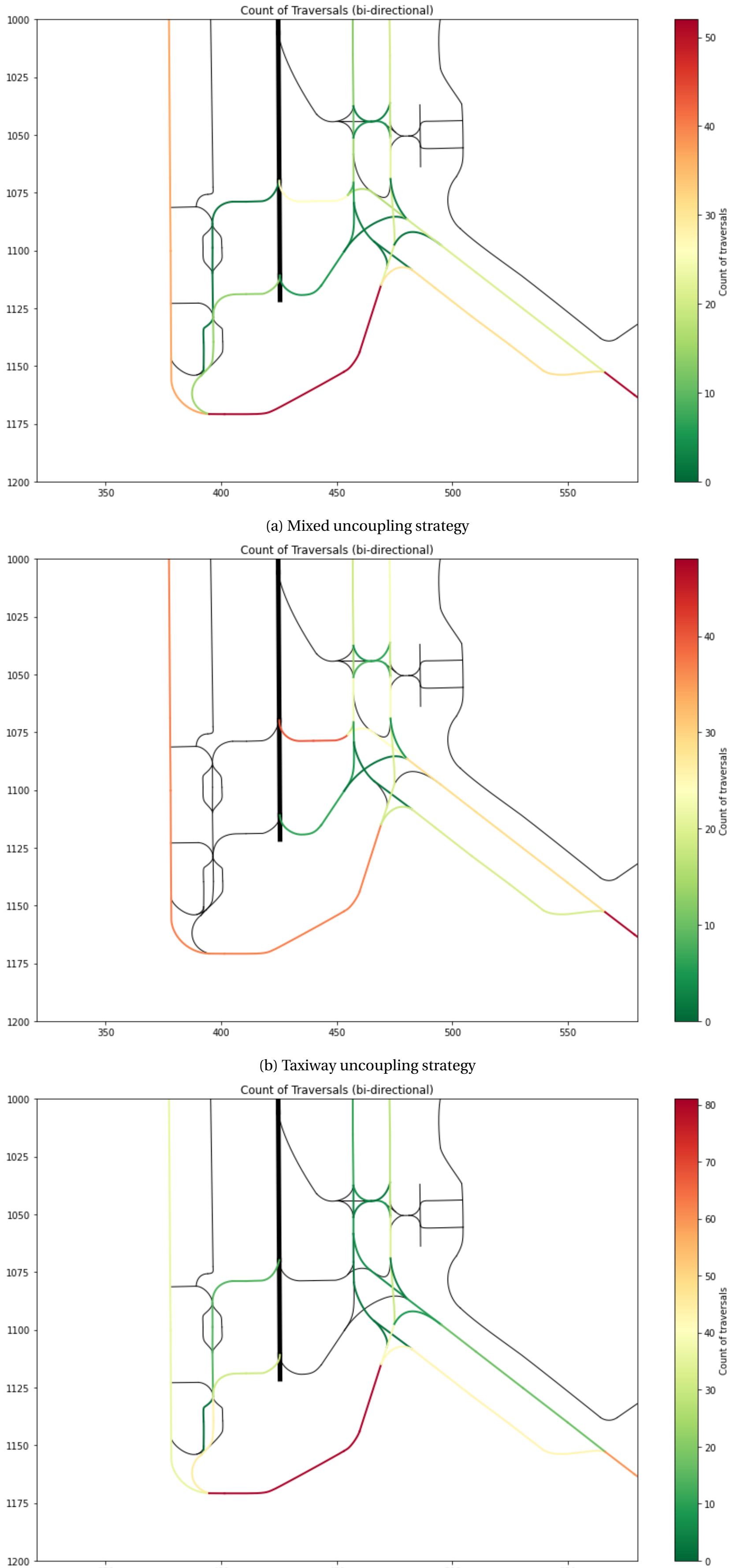


Figure C.8: Airport heatmap of aircraft movements between 12:00 and 14:00 at 17 July. Scenario: mixed uncoupling strategy with normal uncoupling duration.

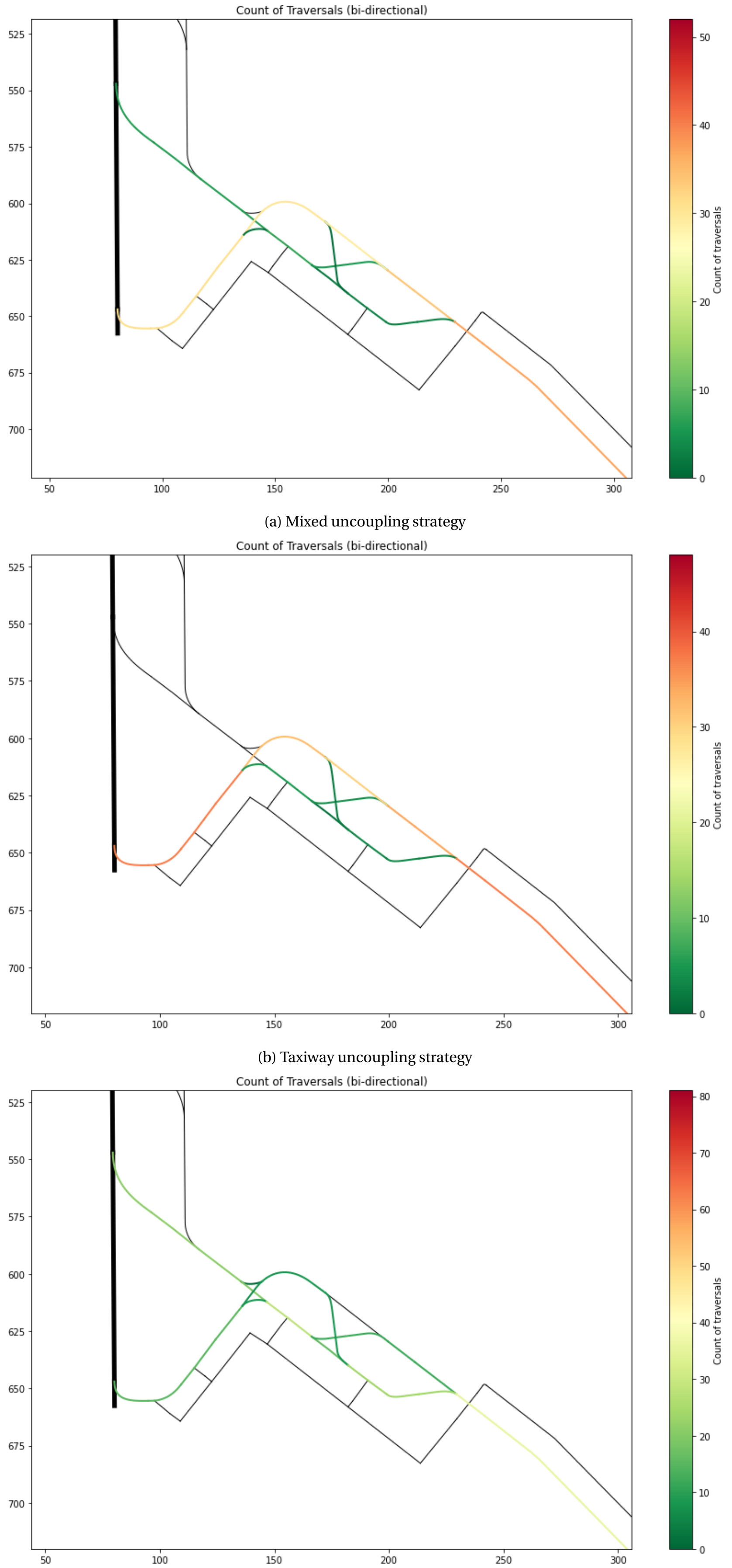


(a) Mixed uncoupling strategy

(b) Taxiway uncoupling strategy

(c) Holding platform uncoupling strategy

Figure C.9: heatmaps of aircraft movements between 12:00 and 14:00 at 17 July for the three uncoupling strategies at 36C. A normal uncoupling duration (t_u = 120 seconds) is used. Note: colour axis differ per plot.



(a) Mixed uncoupling strategy

(b) Taxiway uncoupling strategy

(c) Holding platform uncoupling strategy

Figure C.10: heatmaps of aircraft movements between 12:00 and 14:00 at 17 July for the three uncoupling strategies at 36L. A normal uncoupling duration (t_u = 120 seconds) is used. Note: colour axis differ per plot.

D Simulation Results

This section provides all obtained results. Statistical analyses of the results are given in Appendix E. The taxi times for inbound, outbound, or all traffic is given in Appendix D.1. The taxi times for aircraft departing from similar runways are given in Appendix D.2. Runway delays and runway throughput throughout the two simulated days are provided for all runways in Appendix D.3. The average duration of all simulated runway-specific outbound operations are provided in Appendix D.4. Finally, three uncoupling schedules of the sensitivity analysis for delay minimisation are presented in Appendix D.5.

D.1. Airport Delays

Table D.1 presents the mean and standard deviation of taxi times in the conventional scenario and the nine Taxibot scenarios. Aircraft are grouped by flight direction.

Table D.1: Taxi time simulation results of the conventional scenario and all nine Taxibot scenarios. Times given in minutes for all aircraft (Total), arriving aircraft (Arr), and departing (Dep) aircraft. Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

			Taxibot scenarios								
			t_u = 90 seconds			t_u = 120 seconds			t_u = 150 seconds		
Samples		**Conventional**	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
Total	μ	7.23	8.49	8.58	7.99	9.11	9.38	8.32	9.46	10.33	8.67
	σ	3.76	5.10	5.25	4.40	6.01	6.49	4.62	6.28	7.93	4.88
Arr	μ	5.98	6.00	5.99	5.99	6.02	6.00	5.99	6.01	6.05	6.00
	σ	3.38	3.38	3.38	3.39	3.42	3.39	3.39	3.39	3.41	3.39
Dep	μ	8.47	10.99	11.18	9.99	12.21	12.77	10.65	12.91	14.61	11.35
	σ	3.71	5.31	5.51	4.38	6.43	7.06	4.51	6.61	8.80	4.67

D.2. Runway Delays

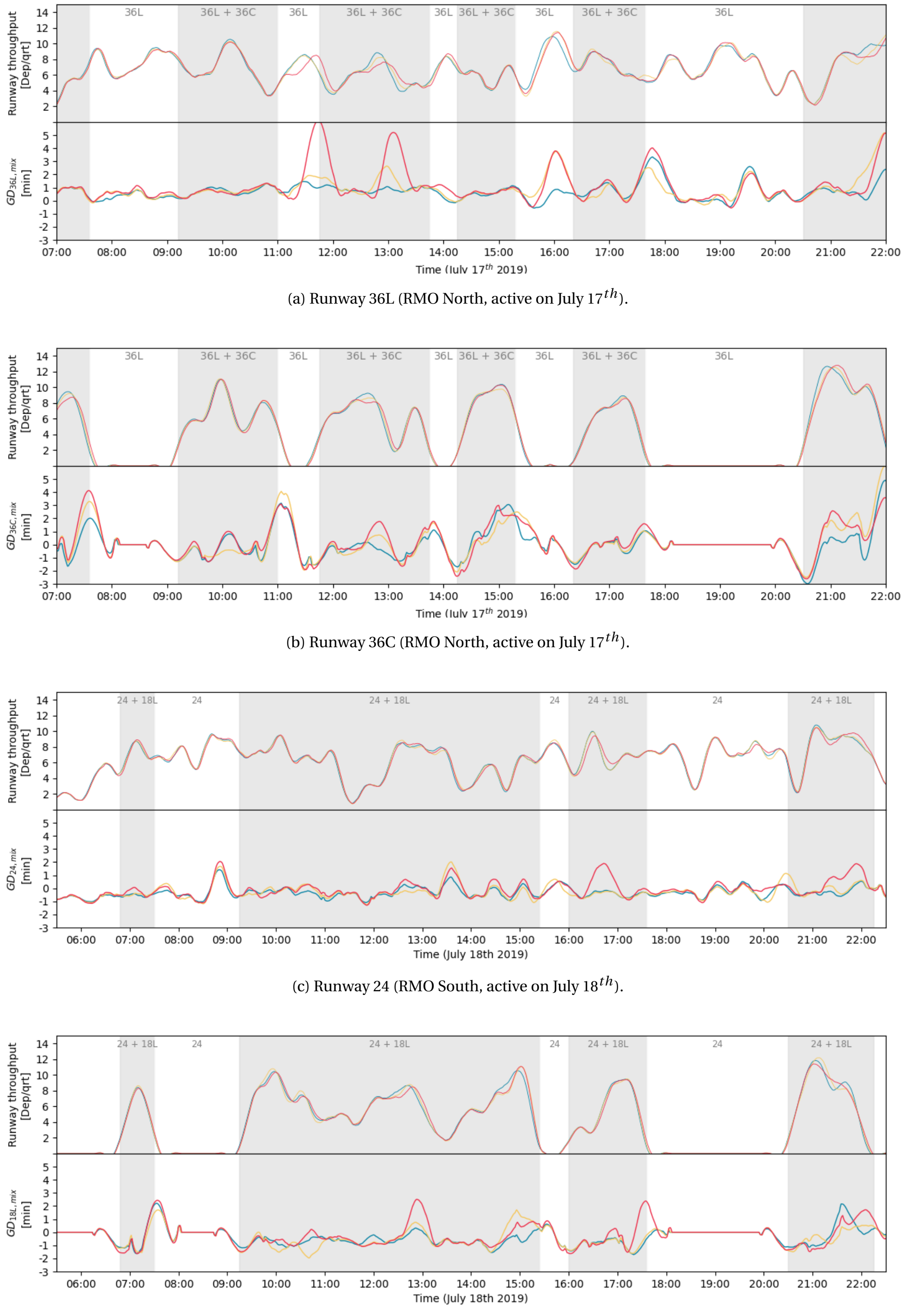
Table D.2 presents the mean and standard deviation of taxi times from departing aircraft in the conventional scenario and the nine Taxibot scenarios. Aircraft are grouped by runway.

Table D.2: Taxi time simulation results of the conventional scenario and all nine Taxibot scenarios. Times given in minutes for aircraft departing from similar runways. Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

			Taxibot scenarios								
			t_u = 90 seconds			t_u = 120 seconds			t_u = 150 seconds		
Samples		**Conventional**	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
18L	μ	7.47	7.15	10.27	6.90	7.64	13.22	6.95	7.62	15.27	7.13
	σ	2.78	2.85	5.15	3.04	4.00	8.06	2.78	3.93	9.52	3.16
36C	μ	7.78	11.52	7.97	7.82	12.57	8.12	8.02	13.91	9.30	8.26
	σ	2.21	5.76	3.69	3.60	6.85	3.20	3.17	8.03	5.02	3.38
36L	μ	12.50	14.00	14.72	13.14	14.78	16.29	13.45	15.15	18.56	13.6
	σ	2.30	3.69	4.26	2.64	5.29	6.16	3.18	8.21	8.42	3.53
24	μ	5.33	n/a	5.17	n/a	n/a	5.23	n/a	n/a	5.31	n/a
	σ	1.88	n/a	2.50	n/a	n/a	2.56	n/a	n/a	2.66	n/a

D.3. Runway Delays Through Time

Figure D.1 presents traffic demand and ground delays for a short, normal and long uncoupling duration throughout the 17^{th} of July. A mixed uncoupling strategy is presented.



(a) Runway 36L (RMO North, active on July 17^{th}).

(b) Runway 36C (RMO North, active on July 17^{th}).

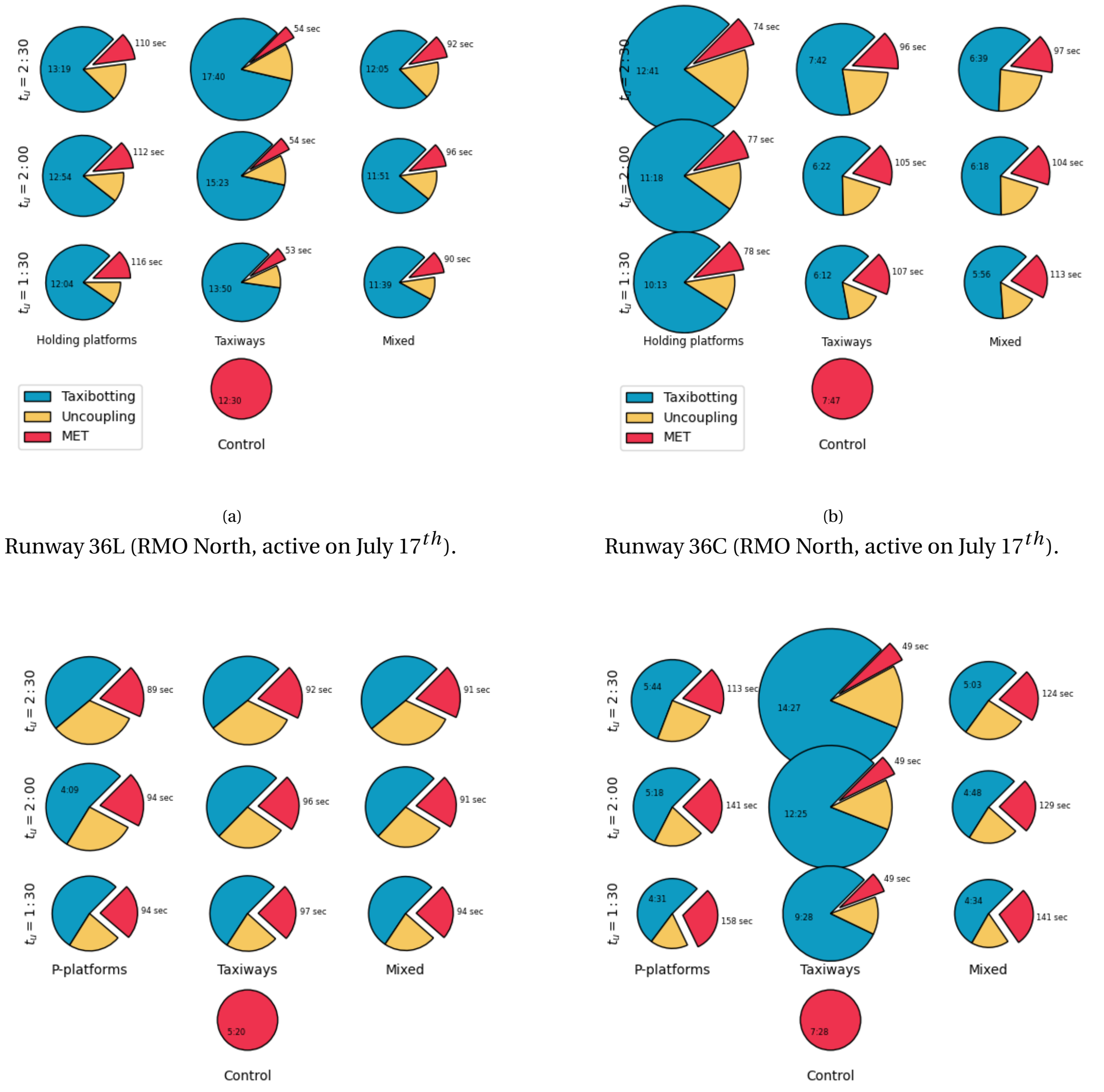
(c) Runway 24 (RMO South, active on July 18^{th}).

(d) Runway 18L (RMO South, active on July 18^{th}).

Figure D.1: Ground delays and runway throughput for a short, normal and long uncoupling duration. Mixed uncoupling strategy is used in all scenarios. Top figure presents the runway throughput per fifteen minutes. The bottom plot presents the average ground delays in a time window of fifteen minutes. Grey areas indicate airport departure peaks, including the active RMO. A fourth degree polynomial Savitsky-Golay filter was used with a window length of one hour to smooth the data. [2].

D.4. Runway Delays per Operation

The average duration of all simulated runway-specific outbound operations is given in Figure D.2.



(a)

Runway 36L (RMO North, active on July 17^{th}).

(b)

Runway 36C (RMO North, active on July 17^{th}).

(c)

Runway 24 (RMO South, active on July 18^{th}).

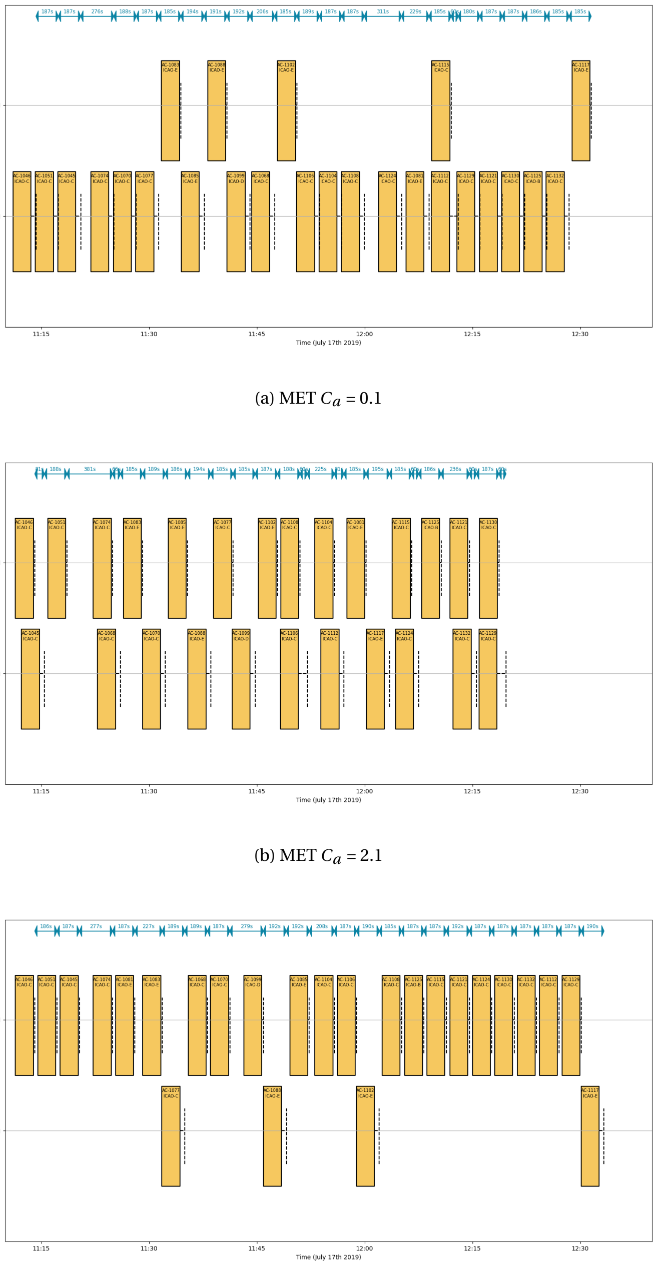
(d)

Runway 18L (RMO South, active on July 18^{th}).

Figure D.2:
Mean duration of all operations for outbound aircraft in all scenarios. Scenarios in the same row have identical uncoupling durations, as indicated at the left (note: uncoupling time given in minutes). Similarly, scenarios in the same column have identical uncoupling strategies, as indicated at the bottom. The radius of each chart is proportional to the taxi time increase with respect to the conventional scenario. Abbreviations: MET, multi-engine taxiing; sec, seconds.

D.5. Sensitivity Analysis for Delay Minimisation

Section 8 of the paper provided a sensitivity analysis of the multi-engine taxiing weight coefficient to minimise delays. The taxiway uncoupling strategy was analysed at a peak outbound hour between 11:00 and 12:00 on July 17^{th} at runway 36L. A normal uncoupling duration was used. Figure D.3 provides the uncoupling schedules extracted from this sensitivity study at $c_a = 0.1$, $c_a = 2.1$, and $c_a = 6$. The shift between the selected uncoupling points is visible. For a low c_a, the furthest uncoupling location from the runway, V4-II, is largely selected. Vice versa, for a high c_a, the closest uncoupling location to the runway, V4-I, is largely selected. It can be observed that at $c_a = 2.1$, delays are minimal as all aircraft have departed at around 12:20, rather than 12:30.



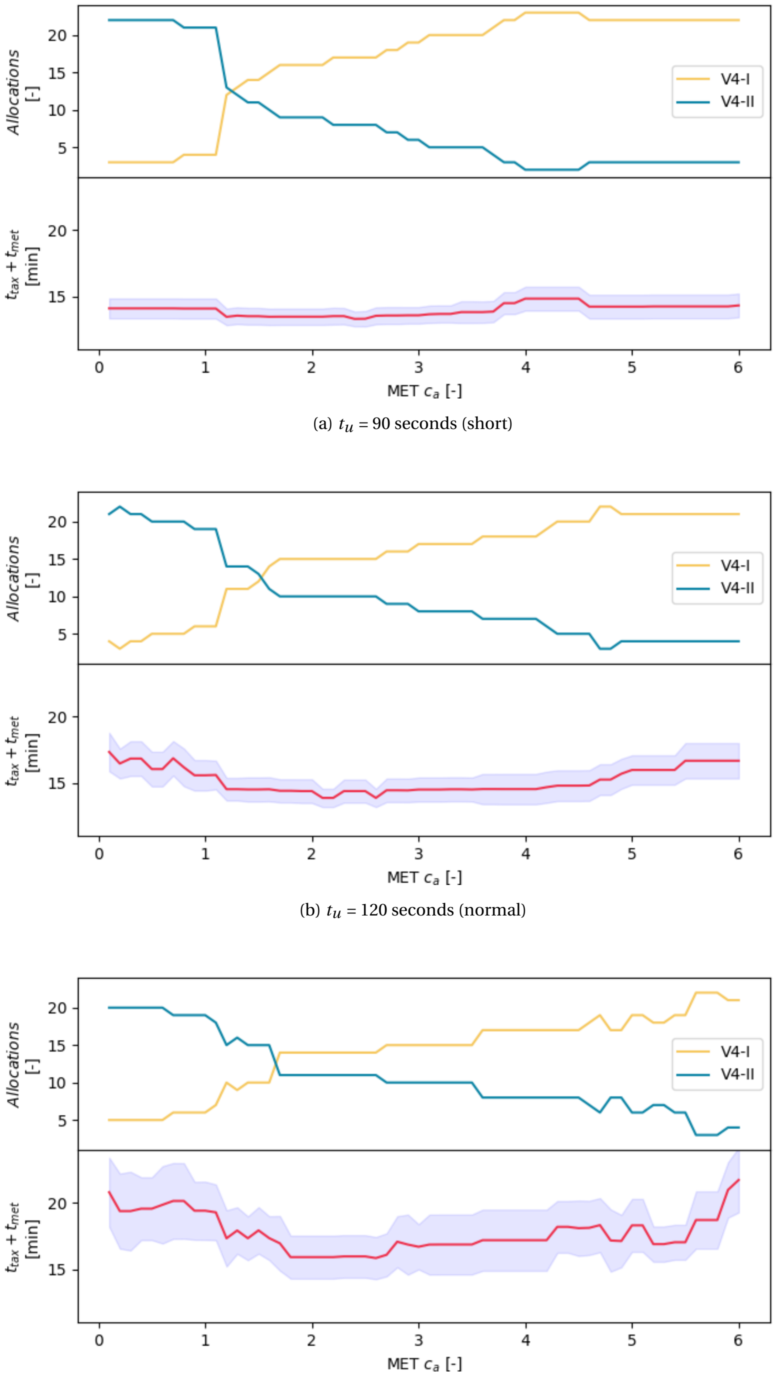
(a) MET $C_a = 0.1$

(b) MET $C_a = 2.1$

(c) MET $C_a = 6$

Figure D.3: Uncoupling schedules for taxiway uncoupling at 36L for all flights between 11:00 and 12:00 with normal uncoupling duration.

The sensitivity analyses was performed for a normal uncoupling duration. The results for a short and long uncoupling duration can be found in Figure D.4. The effects on taxi times in the scenario with a short uncoupling duration are little. For normal and large uncoupling duration, the consequences become larger. The scenarios are statistically evaluated in Appendix E.4.



(a) t_u = 90 seconds (short)

(b) t_u = 120 seconds (normal)

(c) t_u = 150 seconds (large)

Figure D.4: Consequences of varying the multi-engine taxi weight coefficient for aircraft departing at 36L between 11:00 and 12:00 at July 17^{th}. The Taxibotting weight coefficient remains one. Upper plot: number of aircraft allocations per uncoupling point. Lower plot: average aircraft taxi time including a 95% confidence interval in blue. Abbreviations: MET, multi-engine taxiing.

E
Statistical Elaboration

Hypotheses are accepted or rejected based on the outcome of two statistical test. The non-parametric wilcoxon signed-rank test is used to test for a statistically significant difference in taxi times. A Bonferroni-adjusted alpha level is used to correct for family-wise errors in data extracted from similar scenarios [9]. The alpha level (P-value cutoff for statistically significant difference) is determined by Equation (E.1). In this equation, n expresses the number of scenario-specific tests.

$$\alpha = \frac{0.05}{n} \tag{E.1}$$

In the event that a statistical difference is found, the - similarly non-parametric - Vargha-Delaney A-test is adopted to test if the statistical difference is relevant in size [13]. The A-test compares two samples for stochastic equality by returning a value between 0 and 1, indicating the probability that a randomly selected value from one sample is larger than a randomly selected value from the other. A value of 0.5 thereby indicate that the medians are equal. Values greater than 0.56, 0.64 and 0.71 indicate a small, medium and large difference. The same intervals apply below 0.5 if the statistical relation is reversed. All values greater than 0.56 or smaller than 0.44 are therefore considered as significant in size.

Note again that, when talking about delays, we refer to the comparison of taxi times in the Taxibot scenario and the conventional scenario. The statistical tests for general airport delays, and some inter-comparisons of Taxibot scenarios, are given in Appendix E.1. The statistical tests for the runway delays are provided in Appendix E.2. Finally, the statistical tests used in the sensitivity analyses are given in Appendix E.3. Insignificant delays found by the wilcoxon signed-rank test are indicated in blue. The statistically significant delays are subsequently A-tested to test if the differences are relevant in size. Differences indicating increased delays are coloured by light red (small), red (medium), and dark red (large). Differences indicating decreased delays coloured by light green (small), green (medium), and dark green (large). Finally, prove that taxi time data does not follow a normal distribution is given in Appendix E.4.

E.1. Airport Delays

The taxi times of all aircraft, arriving aircraft, and departing aircraft in the Taxibot scenarios are compared to the same groups in the conventional scenario. Results from the wilcoxon signed-rank test are given in Table E.1. Because three samples are compared per scenario, a Bonferroni-adjusted alpha level of $\alpha = 0.05/3 = 0.0166$ is used; meaning that P-values lower than 0.0166 indicate a statistically significant delay. Results from the Vargha-Delaney A-test are given in Table E.2.

Table E.1: P-values from wilcoxon signed-rank test for ground delays in the nine Taxibot scenarios. Compared groups: all aircraft (total), arriving aircraft (arr), and departing aircraft (dep). $\alpha = 0.0166$. Statistically insignificant differences indicated in blue. Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

	Taxibot scenarios								
	t_u = 90 seconds			t_u = 120 seconds			t_u = 150 seconds		
Samples	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
Total	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Arr	0.0006	0.0363	0.0135	0.0000	0.0237	0.0026	0.0000	0.0000	0.0005
Dep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table E.2: A-values from Vargha-Delaney A-test for ground delays in the nine Taxibot scenarios. Compared groups: all aircraft (total), arriving aircraft (arr), and departing aircraft (dep). Differences indicating increased delays coloured by light red (small), red (medium), and dark red (large). Differences indicating decreased delays coloured by light green (small). Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed; n/s, not statically significant. *Determined by wilcoxon signed-rank test in Table E.1

	Taxibot scenarios								
	$t_u = 1:30$ seconds			$t_u = 2:00$ seconds			$t_u = 2:30$ seconds		
Samples	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
Total	0.56	0.56	0.54	0.58	0.58	0.56	0.59	0.60	0.58
Arr	0.50	n/s*	0.50	0.50	n/s*	0.50	0.50	0.51	0.50
Dep	0.64	0.65	0.60	0.69	0.70	0.65	0.72	0.74	0.69

An intercomparison of all uncoupling strategies in the Taxibot scenarios was performed. The taxi times of all departing aircraft in every uncoupling strategy was intercompared for all uncoupling durations. Results from the wilcoxon signed-rank test are given in Table E.3. As only one samples is compared per scenario, a Bonferroni-adjusted alpha level of $\alpha = 0.05/1 = 0.05$ is used; meaning that P-values lower than 0.05 indicate a statistically significant difference in taxi times. Results from the Vargha-Delaney A-test are given in Table E.4.

Table E.3: P-values from wilcoxon signed-rank test obtained by outbound taxi time comparisons of the three different uncoupling strategies. Samples given as: treatment-control (values higher than 0.5 indicate increased delays in the first sample). $\alpha = 0.05$. Statistically insignificant differences indicated in blue. Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

	Taxibot scenarios		
Samples	$t_u = 1:30$ seconds	$t_u = 2:00$ seconds	$t_u = 2:30$ seconds
Hol-mix	0.0000	0.0000	0.0000
Tax-mix	0.0000	0.0000	0.0000
Tax-hol	0.0061	0.0000	0.0000

Table E.4: A-values from Vargha-Delaney A-test obtained by outbound taxi time comparisons of the three different uncoupling strategies. Samples given as: treatment-control (values higher than 0.5 indicate increased delays in the first sample). Differences indicating increased delays coloured by light red (small). Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed.

	Taxibot scenarios		
Samples	$t_u = 1:30$ seconds	$t_u = 2:00$ seconds	$t_u = 2:30$ seconds
Hol-mix	0.55	0.56	0.56
Tax-mix	0.55	0.57	0.59
Tax-hol	0.51	0.52	0.54

E.2. Runway Delays

All departing aircraft were grouped based on their take-off runway and their taxi times in the Taxibot scenarios were compared to those in the conventional scenario. Results from the wilcoxon signed-rank test are given in Table E.5. Because four samples are compared in the holding platform and mixed uncoupling strategy, a Bonferroni-adjusted alpha level of $\alpha = 0.05/4 = 0.0125$ is used. Similarly, as three samples are compared in the taxiway uncoupling strategy, a Bonferroni-adjusted alpha level of $\alpha = 0.05/3 = 0.0166$ is used. Results from the Vargha-Delaney A-test are given in Table E.6.

Table E.5: P-values from wilcoxon signed-rank test for runway delays in the Taxibot scenarios. Values obtained by comparing taxi times of aircraft departing from similar runways in Taxibot scenarios with the conventional scenario. $\alpha = 0.0166$ for the holding platform and mixed uncoupling strategy; $\alpha = 0.0125$ for the taxiway uncoupling strategy. Statistically insignificant differences indicated in blue. Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed; n/a, not applicable.

	Taxibot scenarios								
	$t_u = 1:30$ seconds			$t_u = 2:00$ seconds			$t_u = 2:30$ seconds		
Samples	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
18L	0.0004	0.0000	0.0000	0.0396	0.0000	0.0000	0.0785	0.0000	0.0002
36C	0.0000	0.0361	0.0005	0.0000	0.6185	0.1107	0.0000	0.0011	0.3616
36L	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	n/a	0.0000	n/a	n/a	0.0000	n/a	n/a	0.0000	n/a

Table E.6: A-values from Vargha-Delaney A-test for runway delays in the Taxibot scenarios. Values obtained by comparing taxi times of aircraft departing from similar runways in Taxibot scenarios with the conventional scenario. Differences indicating increased delays coloured by light red (small), red (medium), and dark red (large). Differences indicating decreased delays coloured by light green (small). Abbreviations: hol, holding platforms; tax, taxiways; mix, mixed; n/a, not applicable; n/s, not statically significant. *Determined by wilcoxon signed-rank test in Table E.5.

	Taxibot scenarios								
	$t_u = 1:30$ seconds			$t_u = 2:00$ seconds			$t_u = 2:30$ seconds		
Samples	Hol	Tax	Mix	Hol	Tax	Mix	Hol	Tax	Mix
18L	0.44	0.67	0.40	n/s*	0.73	0.42	n/s*	0.78	0.43
36C	0.76	n/s*	0.43	0.77	n/s*	n/s*	0.81	0.54	n/s*
36L	0.64	0.70	0.59	0.67	0.74	0.61	0.70	0.78	0.61
24	n/a	0.43	n/a	n/a	0.44	n/a	n/a	0.45	n/a

The consequences of changed uncoupling durations to runway delays were analysed by comparing the taxi times found in the scenarios with short and long uncoupling duration with taxi times from the scenarios with normal uncoupling duration. Results from the wilcoxon signed-rank test are given in Table E.7. Because four samples are compared in the holding platform and mixed uncoupling strategy, a Bonferroni-adjusted alpha level of $\alpha = 0.05/4 = 0.0125$ is used. Similarly, as three samples are compared in the taxiway uncoupling strategy, a Bonferroni-adjusted alpha level of $\alpha = 0.05/3 = 0.0166$ is used. Results from the Vargha-Delaney A-test are given in Table E.8.

Table E.7: P-values from wilcoxon signed-rank test obtained by comparing runway taxi times of the Taxibot scenarios with short ($t_u = 90$ seconds (short)) and long ($t_u = 150$ seconds (long)) uncoupling duration to the Taxibot scenario with normal uncoupling duration ($t_u = 120$ seconds (normal)). $\alpha = 0.0166$ for the holding platform and mixed uncoupling strategy; $\alpha = 0.0125$ for the taxiway uncoupling strategy. Statistically insignificant differences indicated in blue.

		Taxibot scenarios	
Samples		$t_u = 1:30$ seconds	$t_u = 2:30$ seconds
Holding platforms	*18L*	0.2220	0.7013
	36C	0.0048	0.0223
	36L	0.0000	0.0006
Taxiways	*18L*	0.0000	0.0000
	36C	0.0488	0.0000
	36L	0.0000	0.0000
	24	0.3690	0.2144
Mixed	*18L*	0.7033	0.2769
	36C	0.0269	0.0447
	36L	0.0006	0.1239

Table E.8: A-test values obtained by comparing runway taxi times of the Taxibot scenarios with short (t_u = 90 seconds (short)) and long (t_u = 150 seconds (long)) uncoupling duration to the Taxibot scenario with normal uncoupling duration (t_u = 120 seconds (normal)). Differences indicating increased delays coloured by light red (small). Differences indicating decreased delays coloured by light green (small). Abbreviations: n/s, not statically significant. *Determined by wilcoxon signed-rank test in Table E.7.

		Taxibot scenarios	
Samples		$t_u = 1:30$ seconds	$t_u = 2:30$ seconds
Holding platforms	*18L*	n/s*	n/s*
	36C	0.47	n/s*
	36L	0.47	0.54
Taxiways	*18L*	0.40	0.56
	36C	n/s*	0.55
	36L	0.44	0.57
	24	n/s*	n/s*
Mixed	*18L*	n/s*	n/s*
	36C	n/s*	n/s*
	36L	n/s*	n/s*

E.3. Sensitivity Analysis for Delay Minimisation

The statistical results of the sensitivity analysis is given in Table E.9. Six samples, each with different activity objective coefficients, are compared. The mean absolute difference, the Vargha-Delaney A-test values, and the Wilcoxon signed-rank P-values are given for all uncoupling durations. As only one samples is compared per scenario, a Bonferroni-adjusted alpha level of $\alpha = 0.05/1 = 0.05$ is used; meaning that P-values lower than 0.05 indicate a statistically significant difference in taxi times.

Table E.9: Statistical results of the sensitivity analyses in the multi-engine taxi activity coefficient c_a. For every uncoupling duration, the mean absolute difference (MAD), A-values from Vargha-Delaney A-test, and the P-values from wilcoxon signed-rank test are given. For the A-test, differences indicating increased delays coloured by light red (small), red (medium), and dark red (large). Differences indicating decreased delays coloured by light green (small), green (medium), and dark green (large). For the wilcoxon signed-rank test, Statistically insignificant differences indicated in blue. Samples given as: treatment-control.

	Taxibot scenarios								
	t_u = 1:30 seconds			t_u = 2:00 seconds			t_u = 2:30 seconds		
Samples	MAD	A-test	P-value	MAD	A-test	P-value	MAD	A-test	P-value
c_a= *0.1-1.1*	0.0134	0.50	0.3173	1.7393	0.62	0.1060	1.5188	0.53	0.6143
c_a= *0.1-2.1*	0.6290	0.55	0.0034	3.4709	0.74	0.0024	4.8971	0.70	0.0056
c_a= *0.1-6.0*	-0.2208	0.50	0.2996	0.6648	0.53	0.1645	-0.9236	0.46	0.3810
c_a= *1.1-2.1*	0.6156	0.55	0.0034	1.7315	0.62	0.0072	3.3783	0.66	0.0025
c_a= *1.1-6.0*	-0.2343	0.50	0.5424	-1.0745	0.41	0.3254	-2.4423	0.43	0.6721
c_a= *2.1-6.0*	-0.8499	0.44	0.0249	-2.8060	0.31	0.0024	-5.8206	0.29	0.0051

E.4. Distribution of Taxi Times

Taxi times where found to not follow a normal distribution, as determined by box plots in Figure E.1.

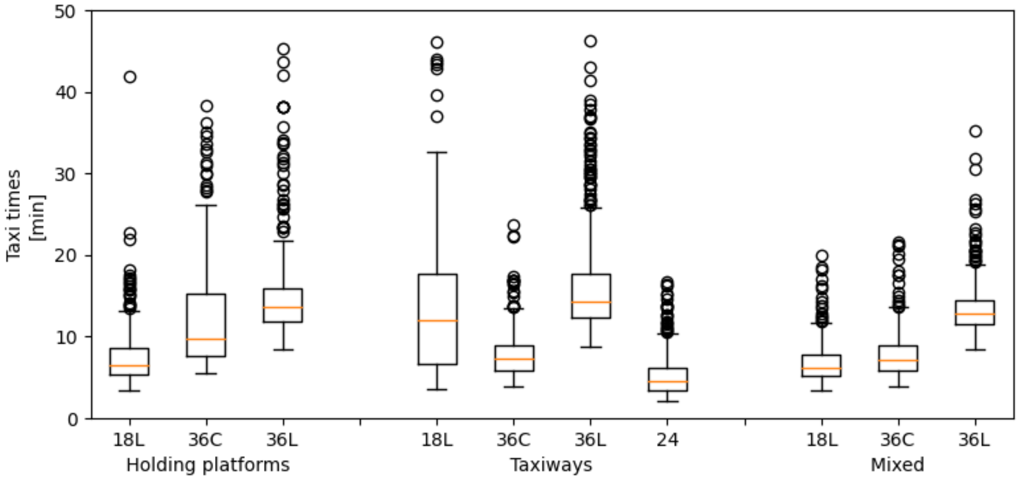


Figure E.1: Box plots of runway-specific taxi times for normal uncoupling duration (t_u = 120 seconds (normal)).

F
Recommendations for Future Work

Two recommendations to further minimise delays for specific uncoupling configurations:

Tune the search objective coefficients for each specific uncoupling configuration, run simulations again.
As a consequence of the PBS and the activity-based SIPP setup, delays were found to be reducible by tuning the ratio between the Taxibotting and multi-engine taxiing weight coefficients. This is especially true for uncoupling points arranged in series. It is therefore recommended to tune the activity weight coefficients for every uncoupling configuration used in simulation. The coefficients could subsequently be made aircraft-specific to ensure that delays at all uncoupling configurations are minimised to the full extent.

Determine the maximum capacity of uncoupling configurations.
For every uncoupling configuration, a maximum capacity determined by the runway traffic demand and uncoupling time exists at which queues start to form and delays increase. It is recommended to determine this maximum aircraft throughout for every uncoupling configuration and expected uncoupling time. One could subsequently consider adjusting flight schedules or changing uncoupling locations to prevent that the capacity of uncoupling configurations is exceeded.

Recommendations for future work are the following:

Run simulations using the online multi-agent simulator.
The results were obtained by an offline multi-agent path planning algorithm. This algorithm was used with search windows of one hour. At the moment all operations in the hour were optimised, the search stopped and a new hour was optimised. This approach, however, means that queues at the end of the hour are not considered in the next hour. As a consequence, delays are not able to stack up between hours. The consequences of the search window cutoff are likely noticeable for scenarios with large delays. For scenarios with medium to little delays, only minor differences are expected as no large queues were formed. Fortunately, the path planning algorithm considered in this research is part of an online multi-agent simulator comprising more agents and functionalities for execution. [1] It is therefore recommended to rerun the scenarios with the online simulator to reduce the consequences of the search window cutoff.

Determine if same principles apply on other airports.
Some general conclusions were made based on departures at four runways in eleven uncoupling configurations. These are however not extremely large numbers. It is therefore recommended to investigate if similar results can be found at other airports and other uncoupling configurations. The modular based setup of the simulator does allow for such adaptation. It is advised to obtain such maps from the Xplane simulator [1].

[1] Multi-agent simulator developed by collaborative work with equal contribution of three students from Delft University of Technology. Team members: Malte von den Burg (PhD candidate), Jorick Kamphof (MSc) and Joost Soomers (MSc).

Improve the alignment of the uncoupling operations with conventional airport processes.
Several important operations connected to Taxibot uncoupling are not considered. These operations might, however, influence the uncoupling operations and thereby resulting delays. Such operation include aircraft holding, gate occupancy, gate pushback, Standard Instrument Departure (SID) routes, runway capacities, and Calculated Take-Off Time (CTOT) slots. It is advised to include these operations in the online multi-agent simulator to model more accurate and realistic scenarios.

Base towing operations on environmental benefit and the availability of Taxibots.
All outbound aircraft are towed in this research. However, some flights are, from an environmental perspective, more attractive to tow than others. Moreover, in reality, no unlimited Taxibots are available. One could therefore include a system that estimates the environmental benefits and analyses the available number of Taxibots. An agent could be included that uses this information to perform a cost-benefit analysis for every flight. As such, environmental benefits can be maximised while reducing the consequences for investment costs (Taxibots, operators, etc.) and ground delays.

Consider self-separation to account for underconstrained situations in the conflict avoidance system.
The augmented path planning algorithm is not entirely conflict-free. The high complexity of the graph required a difficult conflict avoidance system that causes both over- and underconstrained situations for several locations on the graph. No overtaking or point-based conflicts were found, but vehicle shapes do overlap in several occasions as a result of underconstrained situations. These situations are, however, quite rare. It is therefore estimated that such conflicts can be solved by execution. Self-separation in the multi-agent simulator can be considered as a last safety feature to prevent conflict in the simulator.

Determine the consequences of occupied holding platforms.
This research considers holding platform as uncoupling points. These platforms are, however, used in conventional operations for inbound and outbound holding. Uncoupling operations on holding platforms are therefore expected to disturb these regular processes. It is recommended to include aircraft holding in the simulator to investigate the interaction of the two processes.

Bibliography

[1] X-Plane 2022. Worldeditor (wed). [Online]. Available: `https://developer.x-plane.com/tools/worldeditor/`, 11 2021. Accessed: 02-11-2021.

[2] A. Savitzky, J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964.

[3] B. Benda. Agent-based modelling and analysis of non-autonomous airport ground surface operations. [Online]. Available: `https://repository.tudelft.nl/`, Nov 2020. Accessed: 03-03-2021.

[4] D. Bresser and S. Prent. Outcome Taxibot Feasibility Study Schiphol. [Online]. Personal conversation, Jan 2021. Interview data: 20-01-2021.

[5] K. Fines. Agent-based distributed planning and coordination for resilient airport surface movement operations. [Online]. Available: `https://repository.tudelft.nl/`, Nov 2019. Accessed: 03-03-2021.

[6] Annex 14. *Aerodromes Design and Operations*. International Civil Aviation Organization (ICAO), 999 University Street Montréal, Québec Canada, 8th edition, July 2018.

[7] Vincent Metz and Nicolas Girard. Interview on taxibot uncoupling operations. [Online]. Personal conversation, Nov 2020. Interview data: 13-11-2020.

[8] T. Noortman. Agent-based modelling of an airports ground surface movement operation: Understanding the principles and mechanisms of decentralised control. [Online]. Available: `https://repository.tudelft.nl/`, May 2018. Accessed: 03-03-2021.

[9] P. Sedgwick. Multiple significance tests: the bonferroni correction. *BMJ (online)*, 344:509–509, Jan 2012. doi: 10.1136/bmj.e509.

[10] RECAT-EU. *European Wake Turbulence Categorisation and Separation Minima on Approach and Departure*. EUROCONTROL, 1.1 edition, July 2015.

[11] J. Soomers. A delay analysis in airport engine-off towing operations using hierarchical multi-agent path planning. [Online]. Available: `https://repository.tudelft.nl/`, June 2022. Accessible from: 03-06-2022.

[12] J.A.L. Soomers. Agent-based delay management in autonomous engine-off taxi systems (mid term). Available upon request, November 2021.

[13] A. Vargha and H. D. Delaney. A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, July 2000.