UNIVERSITY OF MALTA

INSTITUTE OF AEROSPACE TECHNOLOGIES

DOCTOR OF PHILOSOPHY IN AIR TRAFFIC MANAGEMENT (PPHDATMFTR6)

PHD DISSERTATION

On Engineless Taxiing with Autonomous Electric Tow Trucks

by

Ing. Stefano Zaninotto

Supervisor: Dr Ing. Jason Gauci

Co-Supervisor: Dr Ing. Brian Zammit

April 2024

Abstract

The increase in air traffic during the last decades has had a significant negative impact on the environment in terms of noise pollution, air pollution, and fuel emissions. Historically, efforts to increase efficiency have mostly concentrated on the airborne phase of the flight mission; however, recently, the ground phase of the flight mission, or taxiing, has been getting more attention.

An aircraft's engines are optimised for cruise speed, hence using them for taxiing is rather inefficient. In addition, high traffic volumes and inefficient taxi operations – with frequent stop-and-go aircraft movements – lead to increased fuel consumption during taxiing. Thus, it is understood that one of the main sources of noise and pollution at airports is taxiing. This is one of the issues that the Single European Sky ATM Research Joint Undertaking (SESAR JU) initiative is tackling, and a number of options have been proposed as alternatives to conventional taxiing.

One of the solutions proposed by the aerospace industry is to introduce electric tow trucks to tow aircraft from the stand to the runway (or vice-versa). However, the introduction of tow trucks results in more surface traffic, which is undesirable from the perspective of an Air Traffic Controller (ATCO), as it leads to higher workload. There are numerous ways to compensate for this, including the use of automated planning and execution, but the majority of the solutions proposed in the literature have one or more of the following drawbacks: severe limitations in the deviations that can be introduced in terms of planned taxi routes and scheduled timings; inability to schedule and plan routes for multiple active runways; and no consideration for tow-truck battery state-of-charge during planning. In terms of performance testing of such solutions, only singular performance metrics (e.g., number of vehicle conflicts or average taxi time) have been considered in the literature, thus limiting the validity and applicability of these solutions.

To enhance ground operations and get around some of the drawbacks of current methods, this work details a novel algorithm for taxi operations using autonomous tow trucks. The algorithm identifies conflict-free solutions that limit taxi-related delays and fuel consumption while taking full advantage of the use of tow trucks for taxi operations. It can cater for multiple active runways and accounts for tow truck battery state-of-charge, as well as limits in the number of tow trucks, tow truck depots and charging stations. The algorithm operates at a strategic level (i.e. prior to the start of taxi operations) and uses a centralised approach (i.e. the algorithm is executed on a single computer and may be used as an Air Traffic Control (ATC) tool for ensuring adequate traffic separation at all times).

The algorithm can use one of two approaches: a *Time-Wise Approach* – which prioritises taxi delays over fuel consumption – and a *Fuel-Wise Approach* – which prioritises fuel consumption over taxi delays. Additionally, when assigning tow trucks to arriving or departing

aircraft, the algorithm can either use *Static Allocation*, in which case a tow truck must be parked in a depot in order to be assigned to an aircraft, or *Dynamic Allocation*, in which case tow trucks can be anywhere on the airfield when they are assigned to an aircraft.

The proposed algorithm was tested for different airports with different numbers of active runways, with various levels of traffic and different quantities of tow trucks. A large number of performance metrics were defined to evaluate the performance of the algorithm. A real-time simulator was developed in order to test the updated schedules as provided by the algorithm, both in a deterministic environment (referred to as the *Deterministic Model*) – to ensure correct algorithm operation in terms of conflict-free routes – and in a probabilistic environment (referred to as the *Probabilistic Model*) – to test the robustness of the solutions when the updated schedule is executed under real-life conditions including uncertainty which results in deviations from the assumed parameters.

When evaluated with the *Deterministic Model*, the results demonstrate that the algorithm is capable of using tow trucks for aircraft taxiing without causing any traffic conflicts; to achieve conflict-free routes, nearly 70% of the flights required modest delays of up to 3 minutes to ensure that adequate traffic separation was maintained at all times. These delays were mostly caused by waiting at the stand before taxiing (for departures) or near the runway (for arrivals), and to a lesser extent caused by deviations from an ideal (i.e. shortest) taxi route. The algorithm is capable of assigning tow trucks to more than 90% of the flights, even with a tow truck fleet as small as 30% of the airfield's hourly rate of traffic, with small differences observed for different algorithm settings.

In the *Probabilistic Model*, uncertainties in the vehicle velocities and in the aircraft start time are introduced. When tested with this model, the algorithm shows higher robustness with *Static Allocation* than with *Dynamic Allocation*, in particular in terms of delays and number of tow trucks which fail to attach to the assigned aircraft, therefore indicating that the *Dynamic Allocation* settings do not always have a favourable impact on the algorithm, especially when there are large margins of uncertainty.

The algorithm was also tested for different battery performance characteristics. For relatively small variations in the velocity of discharge rates, significant variations in tow truck performance and average fuel savings were observed. This emphasises the impact of battery characteristics on tow truck performance, and on the size of the tow truck fleet that would be required to cope with the ground movements at a particular airfield.

Acknowledgements

Questo lavoro è dedicato a tutta la mia famiglia, che con tanto impegno e immenso amore mi ha cresciuto e sostenuto lungo la mia strada. Il ringraziamento più grande va a tre persone che hanno speso per me infinite ore ed energie delle loro vite: mamma Agnese, zia Teresina e zio Ervino "Madi".

Un ringraziamento speciale va ai miei supervisori, Dr Ing. Jason Gauci e Dr Ing. Brian Zammit, per aver seguito i progressi del mio dottorato in questi anni in modo puntuale ed estremamente professionale, migliorando costantemente la qualità del mio lavoro con frequenti incontri, molti consigli e dettagliate correzioni.

This work is dedicated to my entire family, who has raised and supported me along my journey with great effort and immense love. The greatest gratitude goes to three people who have devoted to me countless hours and energies of their lives: my mother Agnese, my aunt Teresina and my uncle Ervino "Madi".

A special thanks goes to my supervisors, Dr Ing. Jason Gauci and Dr Ing. Brian Zammit, for carefully following the developments of my PhD over these years in a timely and exceptionally professional manner, consistently improving the quality of my work through regular meetings, numerous suggestions and thorough revisions.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
ALDT	Actual Landing Time
AOBT	Actual Off-Block Time
A-SMGCS	Advanced Surface Movement Guidance and Control Systems
ATC	Air Traffic Control
ATCO	Air Traffic Controller
ATM	Air Traffic Management
CTL	Candidate Tugs List
CTOT	Calculated Take-Off Time
DASC	Digital Avionics Systems Conference
DFW	IATA Airport Code of Dallas Fort Worth International Airport
DST	Depot Status Table
ECL	Edges in Conflict List
EA	Evolutionary Algorithm
EFB	Electronic Flight Bag
EGTS	Electric Green Taxiing System
ELDT	Estimated Landing Time
EOBT	Estimated Off-Block Time
ETA	Estimated Time of Arrival
FDR	Flight Data Recorder
GA	Genetic Algorithm
GAv	General Aviation
GOT	Global Occupation Table
GUI	Graphical User Interface
HMI	Human Machine Interface
HNN	Hopfield Neural Network
HSD	High Standard Deviation
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
ICAS	International Council of the Aeronautical Sciences
ID	Identifier
LEP	Landing End Point
LSD	Low Standard Deviation
MAS	Multi-Agent System
MATS	Malta Air Traffic Services
MILP	Mixed Integer Linear Programming

MIP	Mixed-Integer Programming
MLA	IATA Airport Code of Malta International Airport
MLAT	Multilateration
MOA	Mathematical Optimisation Algorithm
NMOC	Network Manager Operations Centre
NPD	Normal Probability Distribution
PDF	Probability Density Function
ReaM	Reaching Mission
RetM	Returning Mission
RTP	Real-Time Performance
RTRM	Reaching-Towing-Returning Mission
SAF	Sustainable Aviation Fuel
SESAR JU	Single European Sky ATM Research Joint Undertaking
SLDT	Scheduled Landing Time
SMR	Surface Movement Radar
SOBT	Scheduled Off-Block Time
SPO	Shortest Path Optimiser
SQL	Structured Query Language
TLDT	Target Landing Time
TLS	IATA Airport Code of Toulouse–Blagnac Airport
TLV	IATA Airport Code of Ben Gurion International Airport
TOBT	Target Off- Block Time
ToP	Take-off Point
TowM	Towing Mission
TST	Tug Status Table
VOT	Vehicle Occupation Table

Symbols and Notations

Q	Set of depots
b_b	b-th depot
B	Total number of depots
S_b	Parking Slots Number
R	Total number of tow trucks
n_i	i-th node
$n_{ID.i}$	ID of n_i
x_i	Value of n_i along the x-axis of the graph
y_i	Value of n_i along the y-axis of the graph
e_m	m-th edge
$e_{ID.m}$	ID of e_m
$n_{ID.j}$	ID of the first node connected to e_m
$n_{ID.k}$	ID of the second node connected to e_m
$p_{ID.m}$	ID of airport road type corresponding to e_m
w_m	Weight (or cost) associated with edge e_m
U	Set of aircraft
a_a	a-th aircraft
A	Total number of aircraft
V	Set of tow trucks
r_r	r-th tow truck of the set
G_{nodes}	Airport Graph Nodes Set
G_{edges}	Airport Graph Edges Set
n_{st}	Start Node
n_{end}	End Node
n_{st}^a	Aircraft Start Node
n_{end}^a	Aircraft End Node
n_{st}^r	Tow Truck Start Node
n_{end}^r	Tow Truck End Node
P_{nodes}	Path Nodes Set
P_{times}	Path Times Set
t_n	Time when the vehicle passes through n_n
P_{edges}	Path Edges Set
$P_{times.min}$	Minimum Path Times Set
$t_{min.e}$	Time when the vehicle enters e_e
$P_{times.max}$	Maximum Path Times Set
$t_{max.e}$	time when the vehicle leaves e_e
$P_{edges.l}$	Path Edges Length Set
l_e	Length of e_e

$P_{st.dist}$	Path Start Distances Set
d_n	Distance of n_n from n_1
t_{is}	Ideal Start Time
t_{as}	Actual Start Time
Δt_{ds}	Start Delay
Δt_{it}	Ideal Taxi Time
Δt_{at}	Actual Taxi Time
t_{spe}	Shortest Path End Time
t_{ae}	Actual End Time
Δt_{dt}	Taxi Delay
Δt_{dtot}	Total Delay
Δt_{att}	Attachment Time
Δt_{det}	Detachment Time
t_{dl}	Depot Leaving Time
t_{ar}	Aircraft Reaching Time
Δt_{rea}	Reaching Time
Δt_{ream}	Reaching Mission Time
t_{al}	Aircraft Leaving Time
t_{dr}	Depot Returning Time
Δt_{ret}	Returning Time
Δt_{ream}	Returning Mission Time
Δt_{towm}	Towing Mission Time
Δt_{ru}	Tow Truck Utilisation Time
F_{cmin}	Minimum Fuel Consumption
F_{ca}	Actual Fuel Consumption
ΔF_c	Extra Fuel Consumption
t_w	Time Window
t_{ws}	Time Window Start Time
t_{we}	Time Window End Time
Δt_w	Time Window Duration
b_{min}	Minimum Level of Battery
b_{max}	Maximum Level of Battery
Δt_b	Buffer Time
$P_{times.min.b}$	Set of times when a vehicle starts to occupy an edge
$P_{times.max.b}$	Set of times when a vehicle finishes occupying an edge
c_{tow}^r	Towing Cost
b_r	Battery Level
c_{bat}^r	Battery Cost
$c_{tow.max}$	Maximum c_{tow}^r for all the tow trucks in the *CTL*
r_{bdh}	Higher Battery Discharging Rate
r_{bdl}	Lower Battery Discharging Rate

r_{bc}	Battery Charging Rate
y	Density of probability in a PDF of an NPD
x	Value of the variable being examined in a PDF of an NPD
μ	Mean of a PDF in an NPD
σ	Standard deviation in a PDF of an NPD
f	PDF of an NPD
v_{ea}^{a}	Speed of aircraft a in the *Probabilistic Model*
v_{av}	Average aircraft speed
$\sigma_{v_{aa}}$	Standard deviation of the aircraft speed
v_{er}^{r}	Speed of tow truck r in the *Probabilistic Model*
v_{av}	Average tow truck speed
$\sigma_{v_{ar}}$	Standard deviation of the tow truck speed
$t_{as.eff}^{a}$	New time of aircraft a to start taxi
t_{as}	Mean time to start taxiing
$\sigma_{t_{as}}$	Standard deviation of the mean time to start taxiing
C	Number of Conflicts
M	Number of Missed Missions
$\Delta t_{dtot.act}$	Actual Total Delay
Δt_{ds}^{avg}	Average Start Delay
Δt_{dt}^{avg}	Average Taxi Delay
Δt_{dtot}^{avg}	Average Total Delay
DA	Delayed Aircraft
TA	Towed Aircraft
Δt_{tow}	Towing Time
Δt_{ru}^{avg}	Average Tow Trucks Utilisation Time
S	Simulation time
ΔF_{c}^{avg}	Average Extra Fuel Consumption
ΔF_{s}^{avg}	Average Fuel Savings
C_{avg}	Average Number of Conflicts
Z	Total number of simulations run
C_{σ}	Standard Deviation of Number of Conflicts
M_{avg}	Average Number of Missed Missions
M_{σ}	Standard Deviation of Number of Missed Missions
$\Delta t_{dtot.act}^{avg}$	Average Actual Total Delay
$\Delta t_{dtot.act}^{\sigma}$	Standard Deviation of Actual Total Delay

Units of Measurement

kg	Kilograms
m	Meters
m/s	Meters per second
Hz	Hertz
s	Seconds
%/min	Percent per minute

1 Introduction

Taxiing is the process of moving an aircraft from one place in an airport to another, typically using dedicated taxiways, which are roads specifically built to carry out these operations, but in some cases also using the runways. For departing flights, taxiing (referred as to the *taxi-out phase*) begins from the aircraft parking slot on an apron and ends on the designated runway and then is followed by take-off. On the other hand, for arriving flights, taxiing (referred as to the *taxi-in phase*) begins on the runway following the landing and ends in the designated parking slot on an apron.

Currently, taxiing operations are mostly performed using the aircraft's engines with the pilot controlling the aircraft's speed and direction. The engines are usually started as soon as the aircraft is pushed back from the stand and remain running until the aircraft reaches the designated parking slot at the destination airport. On average, the taxi-out phase accounted for 10.2% of the duration of the intra-European flights in 2022, while the taxi-in phase for 5.1% [1]. Therefore, the taxiing times can be long, in particular at large airports, as shown in Fig. 1, and this results in high fuel consumption and emissions.

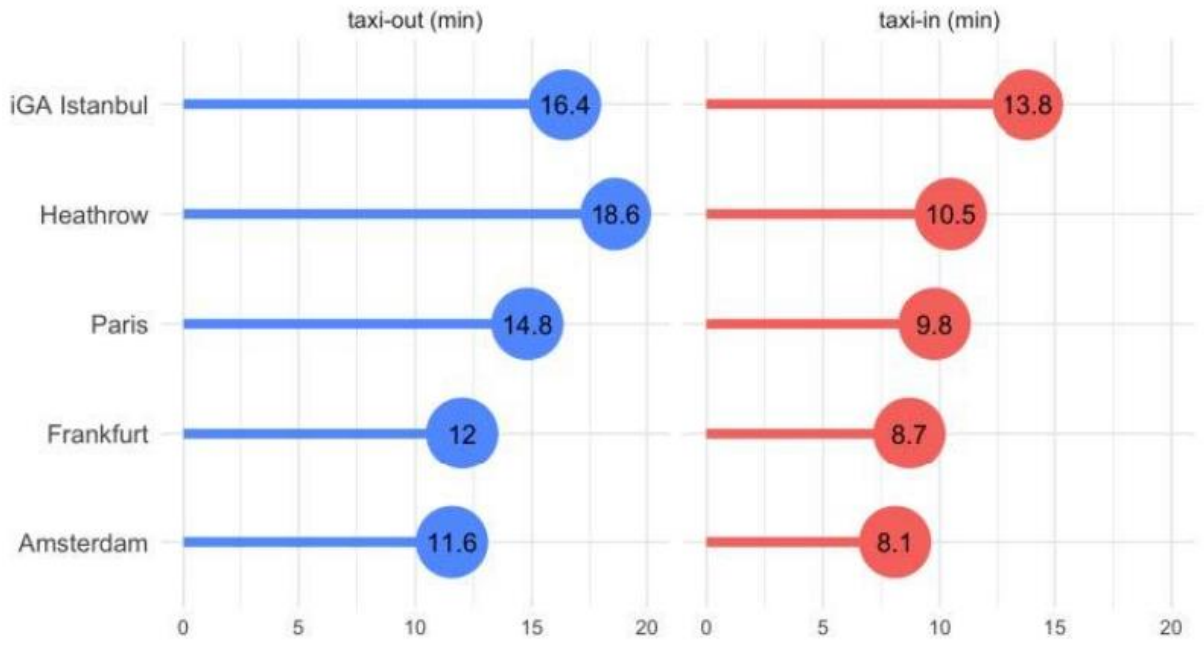


Fig. 1. Taxi-out and taxi-in time durations at the top five European airports in 2022 [1].

Around 2% of all emissions from human activities come from civil aviation [2], and, specifically during aircraft taxiing, every year five million tonnes of fuel are globally consumed, which represents 6% of all fuel consumed by aircraft on short-haul flights [3]. Long taxi times have a considerable detrimental impact on the environment and the economy (as explained in Section 1.1.2), and new technologies and techniques, which are described in Section 1.1.4, are being introduced to mitigate these negative effects.

1.1 Background

Historically, any effort related to improving the efficiency of aircraft operations has mainly focused on the airborne phase, for instance by optimising the flight trajectories [4] and by optimising the aircraft engines for cruise speed [5]; however nowadays the ground phase of the flight mission (i.e. taxiing) is also receiving attention. Currently, most taxiing operations are conducted in an inefficient manner, leading to environmental and economic impacts.

1.1.1 Role of Air Traffic Controllers in Taxiing Operations

During taxiing operations, Air Traffic Controllers (ATCOs) play a critical role in ensuring the safe and efficient movement of aircraft. They are responsible for coordinating with pilots and ground personnel to ensure that aircraft are guided along the designated taxi routes and that they are kept at a safe distance from other aircraft, vehicles, and objects on the ground.

Specifically, ATCOs are responsible for:

- Determining a taxi route for the aircraft and providing taxi instructions to pilots: before an aircraft can begin taxiing, ATCOs provide the pilot with taxi instructions, including the route to follow and any relevant information about the taxiway or runway conditions;
- Monitoring aircraft movement: ATCOs visually monitor the movement of aircraft on the ground to ensure that they are following the designated taxi routes and that they are maintaining a safe distance from other vehicles. ATCOs may also have a display showing the position of aircraft on the aerodrome, where each aircraft is labelled;
- Issuing clearance to cross runways: ATCOs issue clearances to the pilot in various situations during taxiing in order to ensure that the aircraft does not incur conflicts with other aircraft or other vehicles. This happens, for instance, before the aircraft starts the taxiing operations, or when it enters a taxiway from an apron, it enters the designated runway, or it crosses another runway;
- Communicating with ground personnel: ATCOs work closely with ground personnel, including aircraft maintenance crews and airport operations staff, to ensure that they are aware of the location and movement of aircraft on the ground;
- Monitoring weather conditions: ATCOs monitor weather conditions, including visibility, wind direction and speed, and precipitation, to ensure that aircraft can safely navigate the taxi routes.

1.1.2 State-of-the-Art of Taxiing Assistance Solutions

In recent years many tools have been developed to assist pilots and ATCOs to manage the complex dynamics of aircraft taxiing with enhanced safety and efficiency.

Surface Movement Radar (SMR) [6] represents a pivotal technology in enhancing the safety and efficiency of airport surface operations. This radar system is designed to monitor and track the movement of aircraft and ground vehicles on the airport tarmac, significantly reducing the risk of runway incursions and ensuring smooth traffic flow. SMR employs high-resolution radar imagery to provide ATCOs with a detailed, real-time view of all surface activities, even in adverse weather conditions or low visibility scenarios. This technology supports the critical need for continuous surveillance, offering a robust foundation for decision-making processes related to taxiing and ground handling operations.

Advanced Surface Movement Guidance and Control System (A-SMGCS) [7] is a system which incorporates multiple cooperative and non-cooperative sensors (including SMR and Multilateration (MLAT)) and advanced processing algorithms. It delivers an integrated suite of surveillance, control, and guidance functionalities and ensures the effective oversight of ground traffic by providing ATCOs with a detailed, real-time overview of all movements on the airport's surface, including aircraft and vehicular traffic. This integration of technologies transforms how airports manage and secure taxiing operations, promoting a safer and more streamlined flow of ground traffic.

Electronic Flight Bag (EFB) [8] for Taxiing Assistance, as a portable electronic device, has become paramount to flight crews for managing a wide range of flight management tasks. Having a high-resolution displays and interactive airport maps, EFBs serve a critical role in supporting taxiing operations. They furnish pilots with immediate access to dynamic airport layouts, designated taxi routes, and potential obstructions, enhancing situational awareness. The deployment of EFBs optimizes the exchange of information between pilots and ATCOs, ensuring the meticulous adherence to taxiing directives and contributing to the overall efficiency and safety of airport operations.

These tools collectively represent the advancement in technologies designed to assist ATCOs and pilots in managing airport surface movements, highlighting the industry's move towards increasing automation and enhanced operational efficiency.

1.1.3 Environmental and Economic Impact of Taxiing

As aircraft engines are operational during taxiing, this phase of flight results in a high environmental impact due to the emission of greenhouse gases and other pollutants that contribute to air pollution, such as nitrogen oxides and particulate matter. These emissions contribute to climate change and have a negative impact on local air quality, constituting a hazard for public health and wildlife. Also, taxiing contributes to noise pollution; the noise generated by aircraft engines during taxiing can be loud and disruptive, particularly for surrounding communities and local fauna living next to the airports.

The problem is amplified by the fact that, since the aircraft's engines are optimised for cruise speed, they are highly inefficient when used for taxiing purposes. In addition, a situation of high traffic level inside the airport, coupled with an inefficient way to manage the taxi

operations, could lead to congestion on an airport's taxiways as well as to repeated stop-and-go movements of the aircraft and long engine idle times, which, for example, in Dallas/Fort Worth International Airport accounted for approximately 18% of the fuel consumed in ground operations in the months of October-November 2008 [9]. Therefore, taxiing is considered one of the biggest contributors to pollution and noise at airports [10]. For instance, more than 56% of the total NOx in 2002 at Heathrow airport was generated during taxi operations [11]. Table 1 shows the fuel consumption of various aircraft during 20 minutes of taxiing, while Table 2 displays the fuel consumption and pollutant emissions of a number of aircraft on a taxiway in Shanghai Pudong International Airport (China) during different taxiing states, including the impact of stoppings and accelerations states (i.e. stop-and-go movements) on the total fuel consumption and emissions.

Table 1. Fuel consumption of different aircraft for a taxiing time of 20 minutes [12].

Type	Reference type	20 minutes fuel consumption benchmark (kg)
310-300/319/320/321	A320	230
330-200/330-300/AB6	A330-200	500
340-300/340-500/340-600	A340-300	500
72Y/733/734/735/736/737 738/739/73G/73Y/BBJ/	B737-800	227
744/74E/74L/74U/74X/74Y	B747-400	908

These effects have been acknowledged by the European Commission and strict targets have been identified for the future through initiatives such as "Flight Path 2050" [13] and the European Green Deal [14]. The first document sets ambitious targets, such as a 70% reduction in CO2, a 90% reduction in NOx, and a reduction in noise emissions in comparison to levels in the year 2000. While the second document targets emission reductions of 55% by 2030 in comparison to levels in the year 2019. In addition to these requirements, all taxiing procedures will be required to be carbon neutral by 2050.

Table 2. Fuel consumption and pollutant emissions of A320, A340, B738, and B747 during different taxiing states [3].

Indexes	Taxiing State	All Flights		Flight Arrival		Flight Departure	
		Value (kg)	%	Value (kg)	%	Value (kg)	%
Fuel consumption	Stopping	17,323	14	5571	9	11,752	20
	Acceleration	4115	3	1622	3	2493	4
	Uniform velocity	92,300	76	50,700	82	41,600	71
	Turning	7045	6	4050	7	2995	5
	Total	120,783	–	61,943	–	58,840	–
HC emission	Stopping	48	16	12	9	36	22
	Acceleration	9	3	3	2	6	4
	Uniform velocity	227	76	110	83	117	70
	Turning	15	5	8	6	7	4
	Total	299	-	133	–	166	–
CO emission	Stopping	1771	17	457	10	1314	24
	Acceleration	228	2	71	2	157	3
	Uniform velocity	7684	76	3890	83	3794	69
	Turning	489	5	255	5	234	4
	Total	10,172	–	4673	–	5499	–
NOx emission	Stopping	120	14	35	8	85	20
	Acceleration	31	4	13	3	18	4
	Uniform velocity	640	76	346	81	294	71
	Turning	52	6	32	8	20	5
	Total	843	–	426	–	417	–

Apart from an environmental impact, taxiing also has a strong economic impact. Fuel costs account for a considerable part of an airline's total operating costs (generally variable between 15% and 35% [15-17]), making it one of the largest expenses for the industry. Beside this, inefficient management of taxi operations, especially during peak travel periods, causes delays and reflects on the air traffic efficiency, creating further economic costs for the airlines and the airport.

Reducing emissions throughout the taxi phase is one of the challenges that is being addressed by the Single European Sky ATM Research Joint Undertaking (SESAR JU) programme [18] and a number of solutions are being proposed by the industry. For instance, some airlines have started implementing single-engine taxiing [19-20], where only one engine is used to taxi the aircraft instead of both engines. Also, several airlines and airports have already begun to explore the use of biofuels in their operations, including the use of Sustainable Aviation Fuels (SAFs) [21-23]. These solutions can help reduce fuel consumption and emissions while maintaining safety standards, however the improvements are only partial.

Arguably, the most promising solution to reduce emissions drastically during the taxi operations is engineless taxiing [24], as explained in the following section.

1.1.4 State-of-the-Art of Engineless Taxiing

Engineless taxiing, also known as electric taxiing or taxiing with power off, is a state-of-the-art technology that enables aircraft to taxi on the ground without using their engines. This technology involves the use of electric motors or tow trucks to move aircraft around airports,

reducing fuel consumption, emissions, and noise pollution. Two main engineless taxiing solutions have been proposed by the aerospace industry.

The first technological solution, found in systems such as WheelTug [25] (shown in Fig. 2) and the Electric Green Taxiing System (EGTS) [26-27], relies on the use of electric motors that are installed in the main (or nose) landing gear of an aircraft, which are powered by an onboard battery pack. With this solution, an aircraft can move around the airport and taxiways using electric power alone, drastically reducing the need for the aircraft's main engines and therefore saving fuel. Additionally, the electric motors are quieter than traditional jet engines, thus reducing noise pollution as well. One of the main advantages of WheelTug and EGTS is improved safety. In fact, by using electric motors to manoeuvre an aircraft, this system eliminates the need for other vehicles to tow the aircraft, thus reducing the risk of accidents and ground collisions. However, this solution requires modifications to the aircraft's landing gear and associated systems, as well as the addition of cockpit controls to allow pilots to taxi the aircraft using the electric motors. Furthermore, the electric motor would add extra weight to the aircraft, increasing the fuel consumption during the airborne phase.

Fig. 2. WheelTug technological solution [28].

With the second solution, manned or unmanned electric tow trucks are used to tow aircraft from the gate to the runway (or vice-versa) [29-30]. For instance, the TaxiBot solution [31], developed by the Israeli Aerospace Industry, uses a semi-robotic, pilot-controlled hybrid electric tow truck that attaches to the aircraft's nose wheel and provides the necessary propulsion for ground movement. TaxiBot has been successfully tested – and is currently in operation – at Frankfurt Airport [32] (as shown in Fig. 3). As for WheelTug and EGTS, TaxiBot also guarantees a significant reduction of emissions and noise pollution, thanks to the relatively quiet electric engines. Furthermore, this solution does not require modifications to the aircraft and does not add to the aircraft's weight. However, the use of tow trucks for taxi operations will increase surface traffic in the airport's movement area, potentially increasing the workload of ATCOs and creating congestion.

Both of these engineless taxiing systems present clear advantages in reducing fuel consumption [33]. However, this work focuses on the tow truck-based technology and recognises the need to avoid the possibility of an increase in ATCO workload caused by the introduction of the tow trucks. This can be accomplished by the design of a system capable of automating some of the tasks of ATCOs, as explained in the rest of this dissertation.

Fig. 3. TaxiBot system in operation at Frankfurt airport [34].

1.1.5 Challenges of aircraft towing solutions

The challenges associated with towing aircraft to or from the runway, as opposed to conventional aircraft taxiing, are rooted in several practical considerations:

- **Engine Warm-up and Stabilization**: the operation of jet engines, particularly those used in commercial aircraft, demands high efficiency under extreme conditions. A vital aspect of achieving this is the engine warm-up period, allowing engines to reach and maintain a stable operating temperature. This phase ensures adequate lubrication of mechanical parts and their expansion to operational tolerances. Bypassing this crucial step by towing could introduce risks of mechanical failures or diminished engine performance during flight;
- **Fuel Contamination Detection**: an essential part of the taxi-out phase is monitoring engine performance to identify any irregularities, such as fuel contamination. This pre-emptive check facilitates timely maintenance interventions. Without the engines running during taxi to the runway, identifying such issues promptly would be considerably more difficult;

- **Hydraulic Systems and Control Surfaces Verification**: modern aircraft depend on engine-generated power to activate hydraulic systems that manage various flight control surfaces. Taxiing provides an opportunity for pilots to conduct checks on these systems, ensuring their readiness and functionality prior to take-off;
- **Preparedness for Immediate Take-Off**: In the event of a situation that requires a quick departure, having the engines already running and systems checked ensures that the aircraft can take off without delay. This readiness can be crucial in scenarios including medical emergencies, sudden changes in weather, or the approach of another aircraft on its final landing path;
- **Operational and Logistical Implications**: implementing a system where aircraft are towed to runways instead of taxiing on their own necessitates modifications to airport design and workflow. Such a shift could introduce logistical bottlenecks, particularly at high-traffic airports, where the swift turnaround of planes is crucial for maintaining schedules and operational efficiency;
- **Cost-Benefit Analysis**: Introducing a towing system might save fuel, but this advantage needs to be weighed against the implementation's costs and logistical complexities. Therefore, an economic analysis is essential to determine whether the potential fuel savings justify the implementation of a tow truck system, in order to evaluate whether the economic benefits of reduced fuel consumption outweigh the efforts and investments necessary to adapt airport systems and procedures.

1.2 Research Aim and Objectives

The aim of this research was to design an algorithm for engineless taxi operations which automates and optimises the taxi phase by assigning tow trucks to aircraft and generating conflict-free routes to satisfy a pre-defined flight schedule. The algorithm is meant to fully automate taxi operations, meaning that it does not depend on ATCO interventions during the taxi phase of flight. Therefore, it is meant to find feasible routes for aircraft and tow trucks while avoiding conflicts between them and issuing automatic taxi clearances without human intervention.

In order to accomplish this, the objectives of this work were to:

- Identify the challenges that this work seeks to address and define a methodology to tackle them;
- Conduct an in-depth literature review of existing solutions for the optimisation of taxi operations and of techniques which are used in related fields of research;

- Design and develop an algorithm that identifies taxi routes for each vehicle (i.e. aircraft and tow trucks) while satisfying a number of constraints and optimisation criteria;
- Design and develop the capability to allocate tow trucks to specific aircraft in an efficient manner in order to maximise the benefits of their introduction for taxiing operations;
- Define metrics to evaluate the performance of the algorithm in terms of taxi delay, tow truck utilisation, fuel consumption, vehicle conflicts, etc.;
- Develop a simulation environment to test the algorithm, carry out analysis for a number of performance metrics and extract statistical data from a large number of scenarios;
- Test the algorithm based on the defined performance metrics for a wide range of scenarios by varying levels of traffic, tow truck availability, airport size and complexity, and environmental uncertainty.

1.3 Methodology

The methodology adopted in the course of this work was the following:

- Literature review and meetings with ATCOs: a review and analysis of existing solutions was carried out for tow truck taxiing operations, as well as for traditional taxi operations, and made it possible to identify a research gap. Algorithms used in different (but related) fields, such as robotics and logistics, were also reviewed in order to have a wide range of solutions. Work was carried out to understand the current working environment and working practices of Air Traffic Control (ATC). In this regard, a number of meetings were held with personnel from Malta Air Traffic Services (MATS) [35], who provided first-hand knowledge on current operations and best practices that that were taken into account during algorithm design;
- Definition of the real-world problem: the introduction of tow trucks on the airfield for the purpose of aircraft taxiing would result in an increase in vehicle traffic and potentially lead to greater ATCO workload, therefore the objective of this task was to understand the practical implications and challenges of using tow trucks for aircraft taxiing purposes;
- Definition of the scientific challenges: in order to tackle the issues associated with the real-world problem, the research focused on a series of scientific challenges, such as the creation of conflict-free routes, the identification of a suitable number of tow trucks for taxiing, the robustness of the algorithm against uncertainties and

unforeseen events, the definition of meaningful performance metrics and the scalability of the solutions;

- Modelling of the airport environment: the movement area of an airport was represented using a directed graph (network), with airport roads represented by edges, and intersections represented by nodes. Four existing airports of various geometries and sizes were modelled and used for simulation testing purposes, and a database was created to store the airports' data, as well as other data used to represent the taxi environment;
- Design and development of an optimisation algorithm: an algorithm was designed and developed in order to pursue the research aim, to address the scientific challenges with an innovative approach and to give a novel contribution to the real-world problem;
- Creation of an aircraft taxi simulator: a simulator was implemented to simulate the solutions provided by the proposed algorithm and to assess their robustness against uncertainties using Monte Carlo simulations;
- Definition of performance metrics: the definition of performance metrics was carried out on the basis of the aim and the objectives of this work. Several metrics were defined, including metrics to quantify delays, fuel consumption, fuel savings, tow truck utilisation and number of conflicts in uncertain conditions (i.e. when disturbances are introduced in the system);
- Assessment of the performance of the proposed algorithm: the selected algorithm was tested and its performance was assessed by a large number of scenarios defined by configurable (user-defined) parameters, such as airport size and geometry, level of traffic per hour and ratio of the number of tow trucks available to the number of aircraft per hour, in order to analyse the performance of the algorithm on the basis of the defined performance metrics, quantify the impact that the tow trucks bring to the taxiing environment, calculate the number of tow trucks necessary for different scenarios to optimise the solutions and determine the impact of the introduction of uncertainties in the system.

1.4 Publications

A number of papers were published during the PhD studies.

In June 2019, a paper called "Design of a Human-in-the-Loop Aircraft Taxi Optimisation System Using Autonomous Tow Trucks" [36], was presented at the American Institute of Aeronautics and Astronautics (AIAA) Conference held in Dallas, Texas, USA. The paper focused on an early version of the proposed algorithm, which was tested in different scenarios for a number of performance metrics, and showed some preliminary results.

In October 2021, another paper titled “A Testbed for Performance Analysis of Algorithms for Engineless Taxiing with Autonomous Tow Trucks” [37] was presented online at the Digital Avionics Systems Conference (DASC) in San Antonio, Texas, USA. The paper focused on the preliminary results of the PhD work and received the award of the best paper of the session [38]. This paper proposes an updated version of the proposed algorithm, which fully automates the taxi phase. The proposed algorithm is tested along with other algorithms in a simulation environment that was developed for the testing of engine-less taxi solutions and a performance analysis is carried out by defining a number of performance metrics and extracting statistical data from a large number of test scenarios.

Finally, in September 2022, the paper “An Engineless Taxi Operations System Using Battery-Operated Autonomous Tow Trucks” [39] was presented at the 2022 International Council of the Aeronautical Sciences (ICAS) in Stockholm, Sweden. The paper focuses on a mature version of the proposed algorithm and defines several metrics to evaluate its performance. The paper also includes extensive test results.

1.5 Thesis Outline

The rest of the work is organised as follows.

Chapter 2 explores previous work in order to identify the advantages and disadvantages of solutions that are proposed in various contexts. Some works that are not strictly related to aviation, but that could be applied to the problem of engineless taxiing with tow trucks, are also taken into consideration. The research gap is outlined at the end of this chapter.

The goal of Chapter 3 is to define the problem, which can be broken down into two main sub-problems: the first is to guarantee safe taxi operations for all vehicles by identifying the best routes that avoid conflicts, and the second is to assign a tow truck to each aircraft (when possible).

Chapter 4 gives a high-level description of how the proposed algorithm addresses the issue of assigning tow trucks to aircraft and explains how the challenge of providing conflict-free taxiing routes to all vehicles is handled.

The design and implementation of the algorithm is thoroughly discussed in Chapter 5. The data store and the modules of the system are presented and the issues of assigning conflict-free routes, detecting conflicts, and assigning tow trucks are addressed in detail.

Chapter 6 defines the methods and strategies used to assess the algorithm's performance, and defines the performance metrics and test scenarios used for the testing phase.

Chapter 7 presents test results in order to evaluate the overall performance of the system and validate its efficacy.

Finally, in Chapter 8, the main conclusions of this work are presented and the scientific contributions are discussed. This chapter also outlines potential avenues for further research.

2 Literature Review

This chapter reviews previous work in order to highlight strengths and weaknesses of the proposed solutions in different contexts. Apart from papers related directly to tow truck-based taxiing, the literature review also looks at papers in other related areas of research.

The use of tow trucks during taxi operations is an emerging field, which is still in development; therefore, the literature offers a limited number of papers which treat the subject matter directly. However, one can find several works which tackle similar problems. Therefore, the literature review is divided into in five sections, with the first four corresponding to four different topics and the fifth highlighting the research gap.

Section 2.1 focuses on papers which treat the topic of taxiing using tow trucks. The literature provides solutions for path allocation, conflict detection, conflict resolution and tug allocation. These papers are the most relevant for this work, however they are limited in number.

Section 2.2 focuses on the problem of conflict-free path planning. In this section, a number of papers from various fields, including aviation and robotics, are outlined. These works provide solutions which consider a long time horizon; therefore, they follow a strategic approach.

In Section 2.3, the problem of conflict detection and conflict resolution is addressed, and some papers, which address this issue in various fields, are presented. The time horizon defined in these papers is short, as solutions must be provided for sudden events, therefore they follow a tactical approach.

Section 2.4 presents papers which tackle both the issue of conflict-free path planning and the issue of conflict detection and resolution and propose a hybrid/mixed approach (strategic and tactical).

Lastly, Section 2.5 identifies and discusses the research gap which was exploited in this work.

2.1 Tow Truck Taxiing Solutions

This section presents a number of papers related to tow truck taxiing solutions.

The MoTa project [40-43] proposed an intelligent air traffic ground control system which can accommodate conventional taxiing as well as taxiing with autonomous tow trucks. However, with this solution, autonomous tow trucks are only used for departing aircraft. This work is based on a decentralized multi-agent system that can suggest routing solutions via a Human Machine Interface (HMI), as shown in Fig. 4, to dynamically optimize routing and scheduling tasks while taking into account human suggestions in real time. The solution includes a Multi-Agent System (MAS), made up of a reservation planner for the taxiways and a priority system for the agents (i.e., the vehicles) is established [44]. The MAS is incorporated into the taxiing

operations while taking into account taxiway separation and traffic flow restrictions. Based on the Dijkstra algorithm, the agents dynamically evaluate each possible trajectory, reserve the taxiways corresponding to the chosen trajectory and update their priority. Human intervention is taken into account dynamically: the ATCOs can suggest or impose new routes to the agents and modify the priorities. A database saves human choices and the MAS uses it to adapt its next suggestions. System testing however highlighted some drawbacks for ATC, including an increase in perceived controller workload and a sense of mistrust of the tow trucks. Furthermore, the system has only been tested in one part of Charles de Gaulle Airport in Paris with standard traffic flow and therefore the performance capabilities have not been completely characterised and evaluated.

Fig. 4. Prototype of MoTa ground controller interface, as in use for the South ground sector at Charles De Gaulle Airport in Paris [44].

The SAFETug project [45-46], carried out by NASA, proposed a fully autonomous taxiing system – including a surface scheduler, an automated route planning system and an HMI (see Fig. 5) – which can assist ATC, pilots and ground crew during tow truck-based taxi operations, by making tactical decisions to ensure safe and efficient procedures. The system is based on the Floyd-Warshall All-Pairs Shortest Path Optimiser (SPO) and has been tested at Dallas/Fort Worth International Airport. The project, as opposed to the other works discussed in this section, focuses almost completely on technical aspects such as the appearance of the HMI and the state-of-the-art of the tow trucks and of the obstacle detection technology. While the work addresses issues such as the logistical challenges associated with autonomous engines-off taxiing, the precision of navigation of the autonomous tow trucks and the situation awareness of ATCOs, it does not offer solutions for the reduction of the delays caused by the introduction of the tow trucks, or for an efficient route planning of the vehicles in high traffic scenarios.

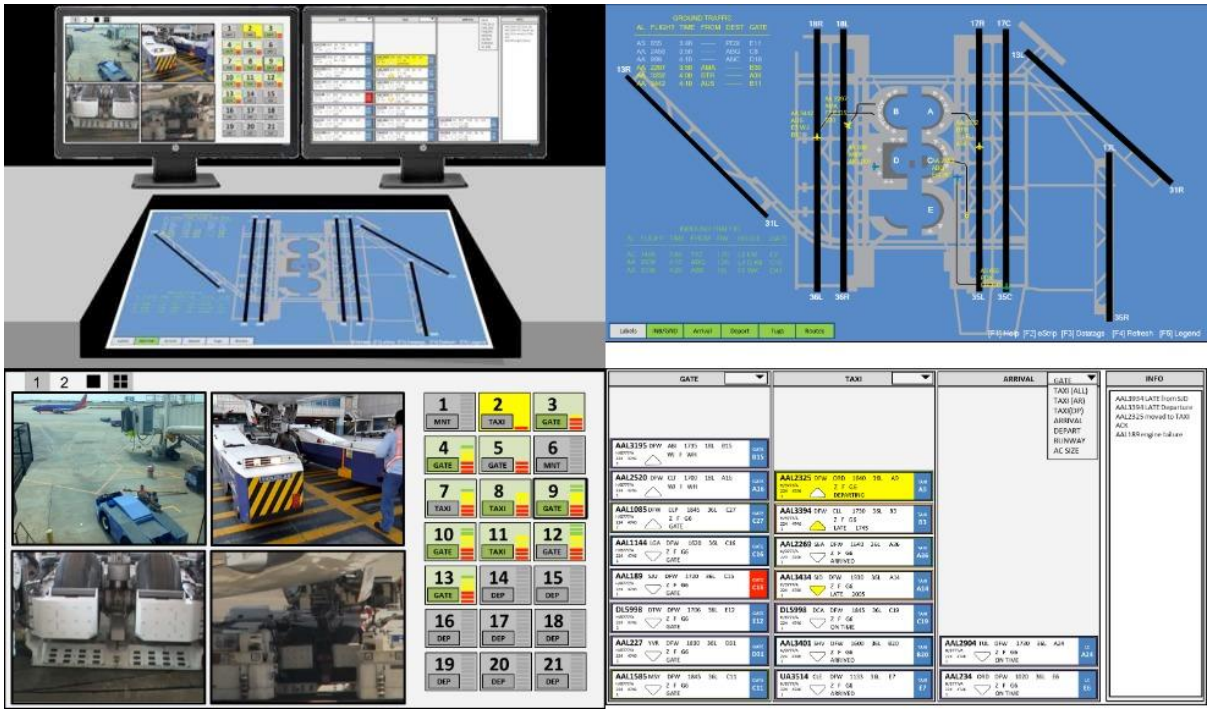


Fig. 5. Four HMI representations related to the SAFETug project. From the top left clockwise: a) general HMI conceptual mock-up; b) the tower surveillance display; c) the flight display; and d) the tow truck display [45].

Sirigu et al. [47-49] propose a system which includes tow truck selection logic, path planning and mission event sequencing algorithms. The solution is based on a centralised full-authority control system and a strategic approach is followed. Four different algorithms are implemented and compared: two based on graph theory (Dijkstra and A*) and two based on neural networks (standard Hopfield Neural Network (HNN) and a modified HNN). The algorithms determine the shortest route while accounting for dynamic obstacles, preferential directions, and prohibited airport zones. Each mission is divided into three phases: tow truck reaching aircraft, engine-off taxiing towing and tow truck returning. The results indicate that the four algorithms generate similar routes; however the two graph-theory-based algorithms are several orders of magnitude more efficient, requiring an almost negligible computational time and thus making them more attractive for practical use. The results of this work focus mainly on the computational time of the algorithms in order to find the most attractive techniques for possible real-time applications. Furthermore, the solutions offered by the algorithms are only based on route selections and do not consider, for instance, the possibility to postpone the departures of a flight in order to minimise the taxi delays. Finally, the system was only tested for one airport (Sandro Pertini Airport of Turin, shown in Fig. 6) and with low traffic flow (8 aircraft per hour).

Fig. 6. Example of a tow truck route in Sandro Pertini Airport of Turin from a tow truck depot to a departing aircraft. This route was calculated using the standard HNN algorithm [48].

2.2 Conflict-Free Path Planning

Each aircraft needs to follow a specific path in order to accomplish its mission, namely to go from its parking stand to a runway (in case of departures) or to go from the runway to a specific parking stand (in case of arrival). In addition, when a tow truck is assigned to an aircraft, it has to follow a specific route to reach the aircraft from its current position and, once the mission is concluded, it follows another path to return to a depot or to start another mission.

In order to prevent vehicle conflicts and collisions during taxi operations, accurate planning of the aircraft's route as well as minor adjustments of the flight schedule are required. To tackle this problem, a number of works available in the literature and presented in this section propose solutions for conflict-free path planning. Most of these works are based on SPOs, while others use Evolutionary Algorithms (EAs) and Mathematical Optimisation Algorithms (MOAs).

One of the assumptions of the works described in this section is that the trajectory of moving obstacles is known *a priori*, and that obstacles move at a constant speed. This limitation does not fit well with an airport environment, where vehicles might not have a constant speed, and unforeseen events may disrupt conflict-free routes.

2.2.1 Conflict-Free Path Planning Based on SPOs

SPOs, such as Dijkstra and A*, are used very frequently to identify conflict-free routes. Since they were developed to calculate shortest paths in a static environment, in many papers they have been modified to include the time dimension, making them suitable for environments with other moving objects.

The solutions proposed in the literature follow two main optimisation strategies: centralised and decentralised [50]. With centralized optimization, a central 'authority' coordinates and makes decisions for all ground movements. This strategy would find the global optimum; however, it would require the central authority to have knowledge of all the vehicles on the airport's surface [51-52]. In contrast, in a decentralised (multi-agent) optimisation method, such as the one implemented by Udluft [53], the decision-making authority is distributed among several agents, i.e. individual aircraft, tow trucks and taxiway or runway intersections (as shown in Fig. 7). Each intersection agent continuously computes the shortest path to the destination of each vehicle under its control using the Dijkstra algorithm and it dynamically responds to changes in the traffic situation. With this approach, only local optima are found; however, this solution scales well with the number of ground movements and is able to react more quickly and robustly to external disturbances (such as taxi delays or unscheduled events).

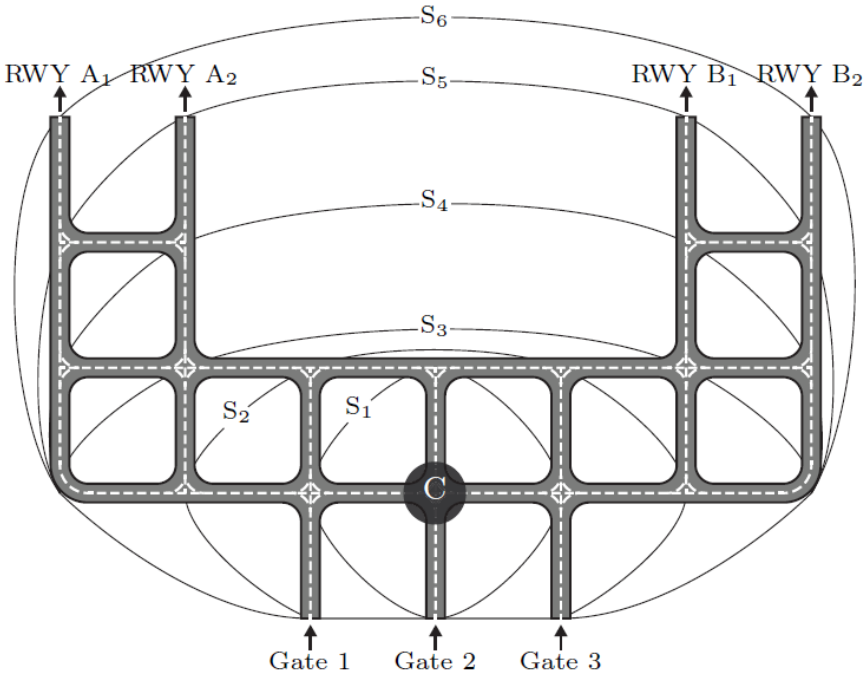


Fig. 7. Taxiway layout in a fictitious airport for the simulation and scope of information for one example controller *C* with different maximum distances S_i at which the controller can observe and make decisions about the traffic situation [53].

Vemula et al. [54] consider the problem of path planning in the presence of dynamic obstacles and use the idea of adaptive dimensionality to speed up the calculation time. Specifically, their approach considers the time dimension only in those regions of the environment where

a potential collision may occur (see Fig. 8), and plans in a low-dimensional state-space elsewhere. The method is validated for the problem of 2D vehicle navigation (x, y, heading) in a dynamic environment, which can be easily adapted to aircraft taxi operations in an airport environment. The solution is based on the A* SPO. The results show that the approach returns feasible, collision-free paths in dynamic environments. In addition, the approach achieves substantial speedups in planning time compared to a traditional 4D heuristic-based A* algorithm, however the solution space is limited (only changes of path are proposed, while postponed departures are not considered).

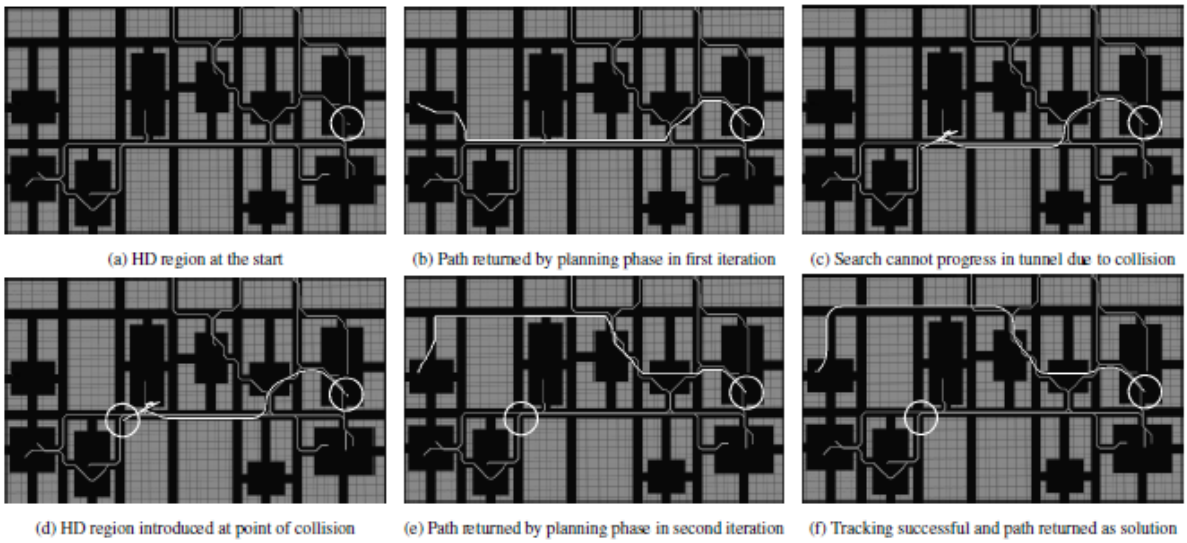


Fig. 8. Example run of the algorithm on a sample map. High Dimensional regions (i.e. regions which takes into account the time dimension) are indicated by white circles, paths of dynamic obstacles by grey lines and final path by white lines [54].

Gawrilow et al. [55] built a routing algorithm based on Dijkstra for Automated Guided Vehicles (AGVs) in an automated logistics system. The key feature of this algorithm is that it avoids collisions, deadlocks and livelocks at the time of route computation (i.e., it produces conflict-free routes) instead of dealing with these problems only during the execution of these routes. The algorithm consists of two parts. The first part is a pre-processing step, where the conflicts' sets are determined in a real-time route computation; during the second part, all the missions which encountered a conflict are rerouted iteratively in a new real-time route computation block, and for each computed route the time-windows of the affected edges are adjusted. In addition, since the AGVs might deviate from the computed routes in time and they might be affected by technical problems while traveling through the network, two different safety tubes – a distance-dependent one and a time-dependent one – are implemented in the calculations to allow space and time separation between vehicles. The algorithm is able to create conflict-free routes and to provide fast results, however it is not resilient to major disruptions of the system, such as big delays or major congestion.

In subsequent work, Gawrilow et al. [56] expanded the work, including an additional conflict management system that, at execution time of the routes, guarantees that no deadlocks

occur. This is done by iteratively allocating to a vehicle the next part of its route and blocking it for all other vehicles until it has been traversed (see Fig. 9). This approach proved to be very efficient in scenarios with a high traffic density.

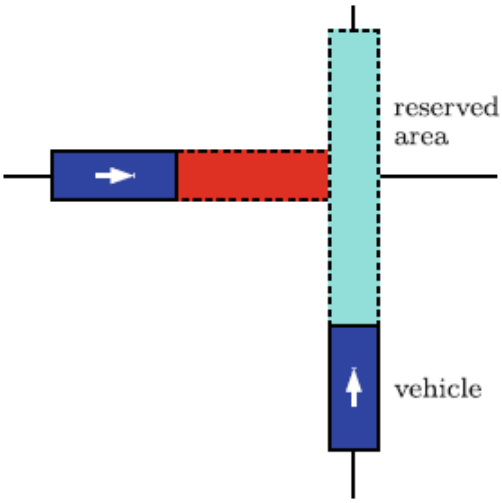


Fig. 9. Reservation procedure. Each vehicle reserves the next part of its route (reserved area). Mutually exclusive reservations guarantee a collision-free execution of computed (static) paths [56].

2.2.2 Conflict-Free Path Planning Based on EAs

Zhang et al. [57] propose a multi-objective optimization method for aircraft taxiing on an airport surface that considers both the airport's environment constraints and the aircraft conflicts. The method is based on multi-objective Genetic Algorithms (GAs) and aims to achieve a Pareto-optimized taxiing scheme in terms of taxiing time, fuel consumption, and pollutant emissions. In this regard, the algorithm was designed with two different taxiing schemes: aimed at prioritising time savings and another aimed at prioritising fuel savings.

The algorithm is developed as follows: first, the aircraft taxi routes are calculated based on unimpeded trajectories; then, a multi-objective optimization scheme under the presence of dynamic obstacles (i.e. other aircraft on the airport's surface) is established; on this basis, a multi-objective optimization scheme for aircraft taxiing under different operating conditions is established and a multi-objective optimization of the taxiing time and fuel consumption of different aircraft types is realized by acquiring their parameters and fuel consumption indices.

The algorithm was tested and calibrated for Shanghai Pudong International Airport (International Civil Aviation Organization (ICAO) code: ZSPD), Lhasa Gonggar International Airport (ICAO code: ZULS) and Kunming Changshui International Airport (ICAO code: ZSPD), which have different airport geometries. Fig. 10 shows results of the optimised fuel consumption for the three airports. The solution presented in this paper is Pareto-optimal, in the sense that it wouldn't be possible to reduce taxi delays further without increasing fuel emissions (and vice-versa); however, the optimisation procedure is based on the reduction of

waiting time of the aircraft while taxiing when a potential conflict is detected (i.e. when two or more aircraft need to cross the same taxiway or crossing) by rerouting the affected aircraft, while no other solutions (such as postponement of the aircraft's departure) are considered.

Aircraft	Taxiing Scheme	Fuel Consumption		
		ZULS	ZPPP	ZSPD
B738	Most time-saving	165.36	150.63	116.89
	Most fuel-saving	119.52	109.03	84.46
A320	Most time-saving	147.79	134.00	104.07
	Most fuel-saving	107.16	97.02	75.48
B777	Most time-saving	446.12	404.80	314.05
	Most fuel-saving	321.30	291.65	226.21
A310	Most time-saving	284.55	258.43	200.47
	Most fuel-saving	203.23	185.43	143.77
B747	Most time-saving	861.22	782.33	606.47
	Most fuel-saving	627.64	570.15	441.98
A340	Most time-saving	364.92	331.66	257.53
	Most fuel-saving	263.70	239.69	186.00

Fig. 10. Comparison of the optimised fuel consumption (Unit: kg) of various aircraft for the airports ZULS, ZPPP and ZSPD under the two taxiing schemes [57].

Rosa et al. [58] also apply GAs to aircraft taxiing. First, a database of 32 routes of the airport is created. Each route contains the initial and final node of each taxiway, its length and the speed limits along the paths. Then a table containing flight data is added: this contains the flight Identifier (ID) of the flight, aircraft type, scheduled time, whether the aircraft is arriving or departing and the initial and final points of its route. Without taking into consideration the different types of aircraft, the minimum distance between aircraft is fixed to 50 meters, in order to avoid crossing conflicts, and the maximum time in a crossing is fixed to 10 seconds. For the simulations, the Congonhas Airport in Sao Paulo was used. The simulations showed a reduction of up to 15% in the total taxi time when compared to the existing data provided by Congonhas airport. With simple modifications to the routes and to the flight plans, the proposed method can be adapted to other airports.

2.2.3 Conflict-Free Path Planning Based on Linear Programming

Smeltink et al. [59] suggest an optimisation model for airport taxi scheduling based on three different variations of a Mixed-Integer Programming (MIP) formulation to represent the movement of aircraft on the surface of an airport. A central 'authority' is implemented to coordinate and make decisions for all ground movements. The algorithm was tested at Amsterdam Airport Schiphol and the simulations involved 406 aircraft over a period of 5.5 hours. This strategy, based on a centralized approach, can handle up to 20 aircraft at a time and finds global optima; however, the reliability of this approach could be compromised by several factors, such as its scalability limit of handling only 20 aircraft simultaneously, reliance

on precise information about all airport surface vehicles, and vulnerability to a single point of failure. These weaknesses highlight potential areas for failure, particularly under high traffic conditions or in scenarios requiring rapid adaptation to unforeseen events.

Roling et al. [60] propose a Mixed Integer Linear Programming (MILP) technique for optimizing the planning and scheduling of aircraft movements on the airport surface. The goal of the model is to minimize the total delay caused by conflicting aircraft movements, while also taking into account a number of operational and safety constraints (e.g. maintaining the departure sequence of the flight schedule and a minimum separation between aircraft). The MILP technique includes an objective function, constraints and decision variables: the objective function is to minimize the total delays and to reduce the number of stops caused by conflicting aircraft movements, the constraints include runway capacity, aircraft weight, and take-off and landing times, while the decision variables include aircraft routes and speeds. The model was tested in a hypothetical airport (shown in Fig. 11) in order to find the optimal aircraft routes and speeds for Departing or arriving aircraft of different types. The results show that the MILP model can significantly reduce the total delays and number of stops caused by conflicting aircraft movements, however other objectives, e.g. the minimisation of the fuel consumption, are not taken into consideration. Furthermore, the model offers solutions to mitigate the effect of conflicts between aircraft, but not to prevent them completely.

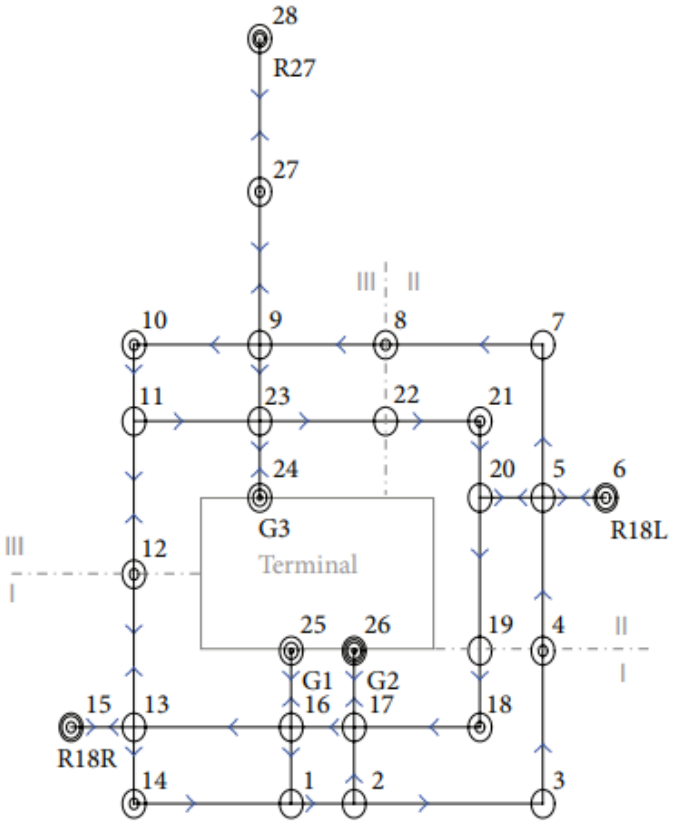


Fig. 11. Graphical representation of a hypothetical airport [60].

Montoya et al. [61] present a MILP model for solving a multiple route taxi scheduling problem. The proposed MILP model is formulated to minimise the total operational cost of the taxi operations with regards to taxi times and delays and is subjected to a set of constraints, including the maximum aircraft speed, the time windows for pick-up and drop-off, and other operational requirements such as the maximum number of allowed vehicles on a route and the minimum separation between aircraft. The model was tested in a section of the Dallas/ Fort Worth Airport, where the taxi routes of arriving and departing aircraft often intersect with each other. The model can determine when arrivals should use the perimeter taxiway (as shown in Fig. 12) to avoid conflicts with departing aircraft and therefore decrease their taxi time and delays. However, no other parameters are considered in the paper, such as the fuel consumption, which can increase when an aircraft uses the perimeter taxiway instead of a central taxiway (for instance the taxiway between the runways 17R and 17C in the airport in Fig. 12), which provides a shorter taxi route. Also, the algorithm was only tested in a section of a single airport.

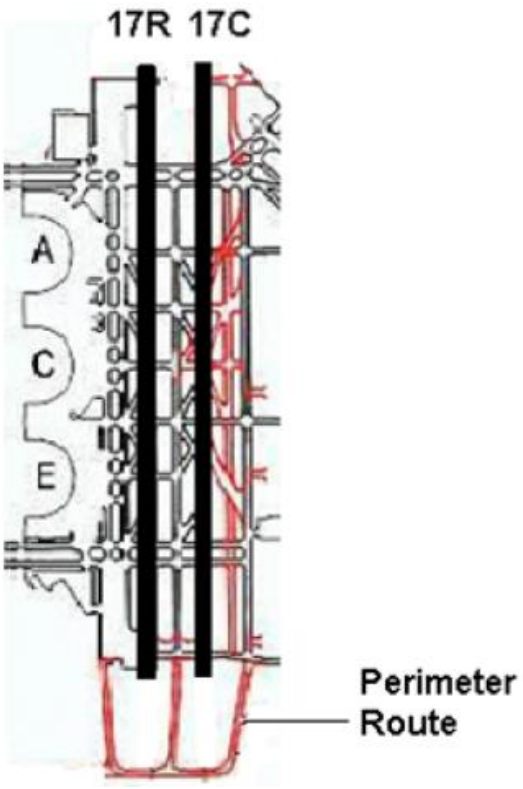


Fig. 12. The section of Dallas/Fort Worth airport tested with the indication of the perimeter route [61].

Clare et al. [62] present an automated tool based on a MILP model which offers solutions for the coupled issues of airport taxiway routing and runway scheduling, using a receding horizon formulation and an iterative method to allow the implementation in continuous time and the scalability of the algorithm. The proposed method combines discrete decisions – choosing among predetermined taxiways – with continuous decisions, related to the timing of the planned movement. Tests were conducted at Heathrow Airport in London using 240 aircraft for ground movement operations, and, according to the results, typical taxi waiting time can be shortened by 55% compared to a first-come-first-serve strategy. The primary benefit can be observed in the departure aircraft flow, however, no significant improvement in taxi time was observed for arrivals. Also, the algorithm needs to pre-determine a number of aircraft routes, which requires a lengthy process for the implementation of any new airport model in the system.

2.3 Conflict Detection and Resolution

As taxi operations are continuously affected by delays and unforeseen events, a strategic approach which plans routes and off-block times ahead of time might partially fail in some situations, when, for example, an unforeseen event occurs and delays are added to one or more vehicles. Consequently, the delays might disrupt the strategic plan resulting in unplanned traffic conflicts.

For this reason, some tactical solutions are required to constantly monitor the events, detecting possible conflicts between vehicles, and resolving conflicts in a timely manner.

Various solutions have been explored in the literature to cater for this issue and are mostly based on the use of SPOs.

Adacher et al. [63] propose an autonomous multi-agent approach to solve air traffic congestion in real-time. Air traffic is modelled with a graph and is partitioned in different sectors. Each sector has its own decision agent, which controls the traffic within the sector and imposes a real time aircraft schedule to respect timetables and capacity constraints. When a congestion is predicted, the aircraft scheduling is recalculated and verified until the capacity constraints are satisfied and the congestion is resolved (see Fig. 13). This is done by delaying aircraft on the ground and/or rerouting aircraft and/or postponing the congestion. Two different SPOs that calculate K feasible paths – Generalized Dijkstra and Bidirectional Search – are implemented for each aircraft involved in the congestion. The algorithms were tested in North Italian air-space and the results prove that the approach is capable of solving air traffic congestion. This approach could be adapted to the two-dimensional problem of conflict resolution during taxiing. However, a limitation of this work consists in the fact once a conflict occurs, both aircraft are already in motion, making it impossible to delay an aircraft and sometimes even impossible to reroute it, due to the lack of alternative paths; this could create knock-on effects on the surrounding environment, further increasing the number of conflicts.

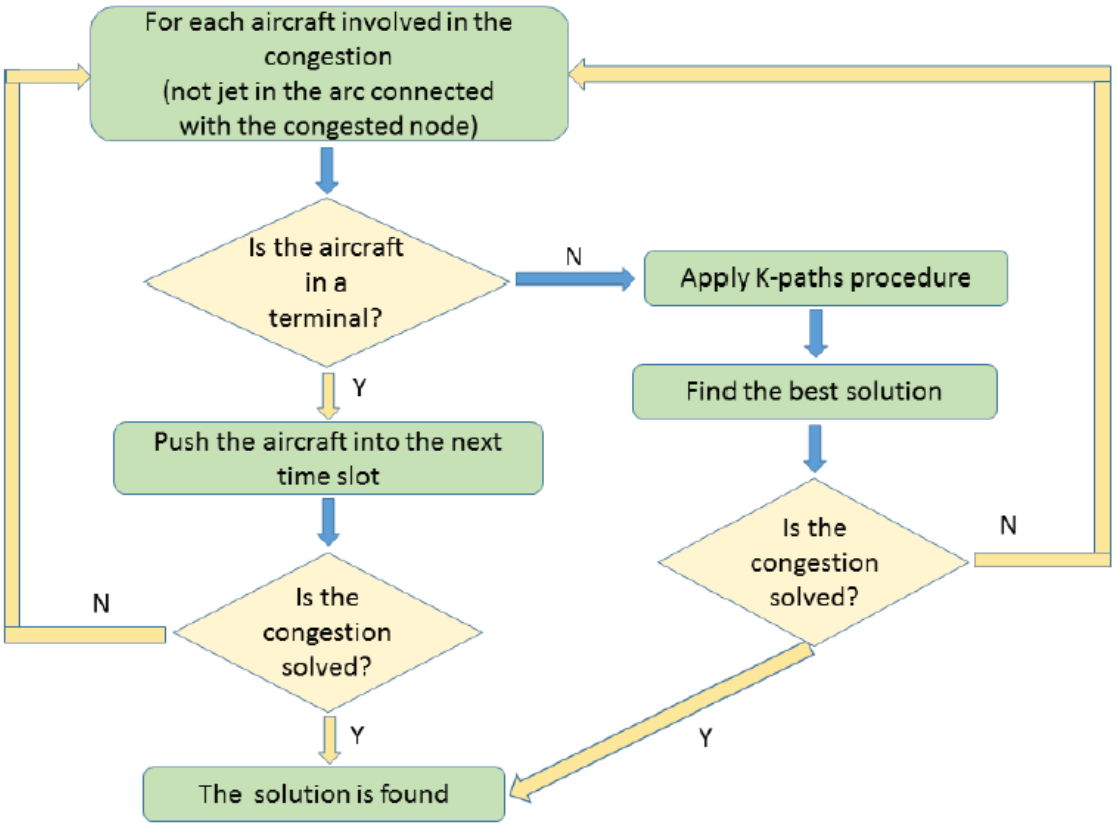


Fig. 13. Scheme of the approach to solve congestion [63].

Phillips et al. [64] suggest a solution for robotic path planning in the presence of dynamic obstacles. With traditional approaches, planning in the presence of dynamic obstacles is computationally expensive because it requires adding the time dimension to the space explored by the planner. In order to avoid an increase in the dimensionality of the planning problem, other real-time approaches to path planning treat dynamic obstacles as static and constantly re-plan as dynamic obstacles move (see Fig. 14). With this solution, a planner is built on the basis of safe time intervals. A safe time interval is a contiguous period of time for a configuration, during which there is no collision, whereas a collision occurs (or can occur) one time step prior and one time step after the period. The planner exploits this observation and constructs a search-space with states defined by the configuration of the robot (i.e. position, heading and velocity) and by safe time intervals. The planner, based on the A* SPO, provides the same completeness of the one guaranteed by planning with time as an additional dimension, and with a reduced calculation time. Simulation tests were carried out with up to 200 dynamic obstacles, making the planner feasible for use in real-time on robots operating in large dynamic environments. This approach could provide a solution for potential conflicts during taxiing, but has to be implemented very carefully in order to prevent deadlocks.

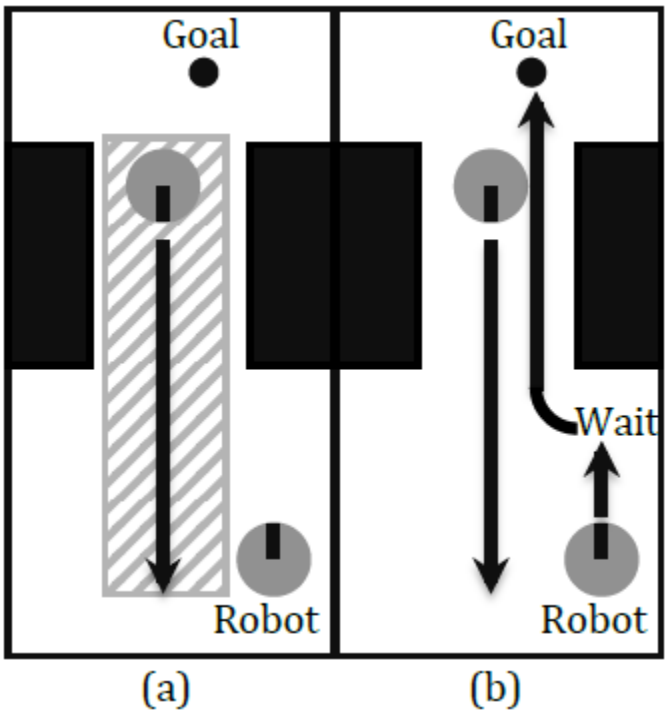


Fig. 14. Comparison of two approaches: (a) treating the obstacle as static often results in no solutions; (b) planning with safe intervals of time finds a solution by waiting for the obstacle to pass and then proceeding [64].

Fransen et al. [65] propose real-time path planning for a grid-based AGV system to sort parcels. AGVs move from pick-up to drop-off locations and vice-versa to transfer parcels to their designated sorting destinations. Each AGV is either unloaded or loaded with one parcel at every moment in time. As soon as a parcel is dispatched to an idle AGV, that AGV starts moving towards the pick-up location where the parcel is located, to load it. From there, it moves to a designated drop-off location. The AGV unloads the parcel once it has reached this location and then continues its journey to another pick-up location to load the next dispatched parcel. The parcels entering to the system are not known *a priori*, but they appear in an online manner; therefore, the solution is based on a tactical path planning approach, meaning that part of the planned paths might be changed based on real-time information.

The grid is divided into buffer zones and the centre of each of them is represented by a weighted vertex; the weight of each vertex is initialised to zero and it is updated over time via exponential smoothing according to the time spent by AGVs waiting idly. The paths are planned and re-planned through an extended version of A* SPO, which takes into account the weights of the vertices, penalties for turning and the possibility to select one of a set of multiple possible destinations. Furthermore, a deadlock avoidance strategy is implemented in the model; however, it cannot avoid their occurrence in all situations. In this case, deadlock can be resolved by path re-planning, as long as an alternative path to the same destination exists for at least one of the AGVs waiting in deadlock.

This work has many characteristics in common with tow truck taxiing: for example, the parcel sorting using AGVs could be associated with the aircraft towing using tow trucks, and both systems need to be updated with information. However, a limitation of this work is that the path planning approach only uses information from the past to update the vertex weights, while to improve the process it should incorporate present conditions as well. In addition, the scalability of the approach should be better investigated, for instance by performing simulations for bigger layouts. Also, a difference between the taxiing environment and the AGV environment described in this work is that the order in which the aircraft depart and land is known *a priori* (a predefined schedule exists), whereas the order in which parcels enter the system is completely unknown beforehand.

2.4 Hybrid Strategies

This section presents solutions which combine conflict-free path planning with conflict detection and resolution. The approach followed is both strategic and tactical, and this is achieved through an initial path planning before the beginning of the simulations, and a number of re-planning iterations during the simulation. The solutions in the literature following this approach are based on SPOs and EAs.

2.4.1 Hybrid Strategies Based on SPOs

Van den Berg et al. [66] present an efficient anytime method for robotic path planning in dynamic environments. Traditional approaches to planning in dynamic environments either assume that the environment is static and re-plan when changes are observed, or assume that the dynamics of the environment are perfectly known *a priori*. Instead, this approach takes into account all prior information about both the static and dynamic elements of the environment, and efficiently updates the solution when changes to either are observed. First, a roadmap is built representing the static portion of the planning space. Then, an initial trajectory is planned over this roadmap taking into account any known dynamic obstacles. This trajectory is continually improved until the time reserved for this strategic phase is exhausted. Furthermore, while the robot, which represents the agent, executes its mission, its trajectory continues to be improved, so, when changes occur to the dynamic elements of the environment, the current trajectory is repaired to reflect these changes. This is done in an anytime fashion, meaning that a solution can be extracted at any time. The approach is based on two modified versions of the D* algorithm and the results prove that this solution is capable of solving large instances of the navigation problem in dynamic environments with incomplete information about the behaviour of the obstacles. This approach could be extremely helpful in the taxiing environment, where the positions of the dynamic obstacles (i.e. of the other vehicles) is generally known *a priori* over time, but might undergo sudden changes due to unforeseen events. Indeed, while resolving a conflict or deadlock situation among robots might be relatively straightforward, the same cannot be said for aircraft. The

challenge with aircraft is significantly greater, as they cannot easily reverse or manoeuvre in tight spaces, making any deadlock potentially problematic. Consequently, although this approach holds promise for enhancing taxiing operations, it cannot be directly applied without meticulous adaptations and considerations to accommodate the constraints and safety requirements of airport environments.

As waiting and rerouting techniques are frequently utilised to avoid aircraft surface taxiing conflicts, a model is proposed by Zhao et al. [67] where the taxiing paths are initially built, the taxiing area is divided into subregions and time is discretised into time windows. Based on an analysis of the taxiing activity in each subregion, conflict detection is carried out and, by comparing aircraft priorities, two different solutions – waiting or rerouting – are selected to solve conflicts, assigning new paths and time windows to the aircraft using an improved A* algorithm. The method was tested at Xi'an Xianyang International Airport and the study's findings indicate that the use of time windows for conflict detection and resolution can further reduce the total taxiing time of aircraft, leading to a significant decrease in the number of aircraft conflicts and ensuring the safety of airport operations. The algorithm, although very precise in calculating the taxiing routes and times, has a long execution time and is not efficient in case of large fleets (more than 30 aircraft). Also, the need to divide the airport into subregions indicates that it is not straightforward to adapt the algorithm to other airports.

Li et al. [68] discretise the ground structure of a fictitious airport in a directed graph model and introduce a multi-factor constrained aircraft taxiing path optimization scheme based on Dijkstra algorithm. A number of factors – such as the change of the runway, conflicts between aircraft or engine failure during taxi – are taken into account and the taxi path is calculated through specific analysis of their influence on the path selection, which determines the weight of the graph's edges. Fig. 15 shows a flow chart of the algorithm used to determine the routes of the aircraft. The algorithm, however, ignores factors such as taxi time, fuel consumption or taxi delay when optimising the routes, making it more suitable for real-time (tactical) decisions than strategic decisions.

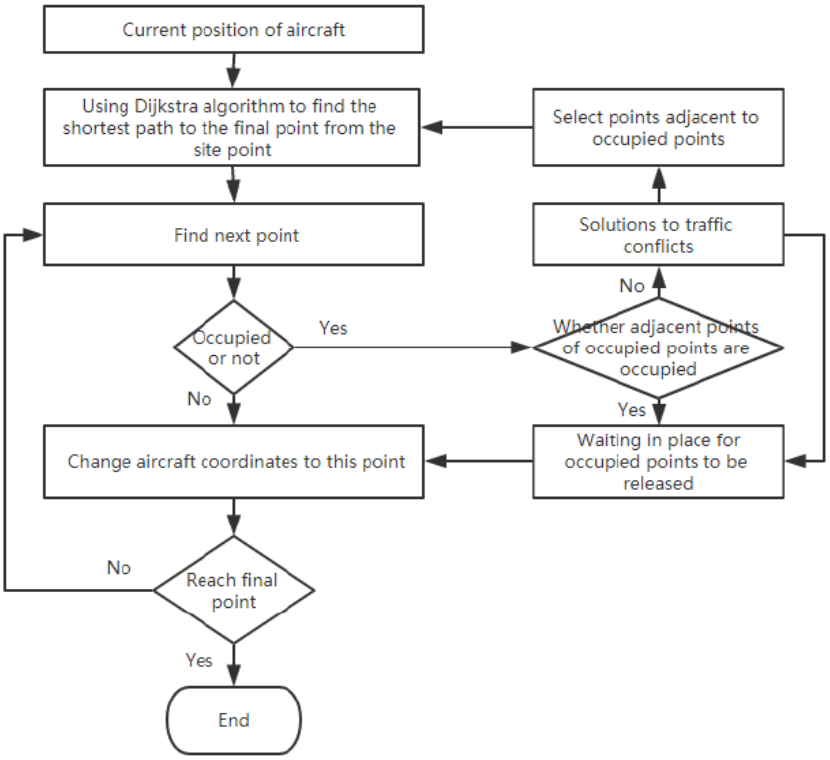


Fig. 15. Flow chart of the Dijkstra algorithm to find the shortest path [68].

2.4.2 Hybrid Strategies Based on EAs

Ibrahim et al. [69] propose an algorithm for robot path planning in a dynamic environment using a GA. Each gene in a chromosome of this system is encoded with information to identify the next coordinates or location that a robot should move to and the total number of genes in a chromosome is determined based on the minimum steps taken to reach the final point. The proposed algorithm is able to find an optimum path for robots and avoid static and dynamic obstacles. One of the important elements of the robot path planning algorithm is its ability to respond to changes in the environment. Initially, the robot path is predetermined based on static obstacles occurring in any possible path. However, when a robot moves after a certain moment, the surrounding environment may change due to the existence of dynamic obstacles in the predetermined path. The algorithm should be able to immediately identify a new path before the robot collides with dynamic obstacles that suddenly obstruct the current path, as shown in Fig. 16. The results of the work are very promising in terms of obstacle avoidance; however, a limitation of this study is that aircraft and tow trucks are constrained to move along taxiways/runways, whereas the robots in this paper can move in any direction. This might be difficult to apply within an airport environment, where the direction of movement of aircraft and tow trucks is constrained by the network of taxiways and runways. Other limitations of this method are due to the stochastic (non-deterministic) nature of GAs

and EAs in general, which does not guarantee repeatability of the results under identical initial conditions.

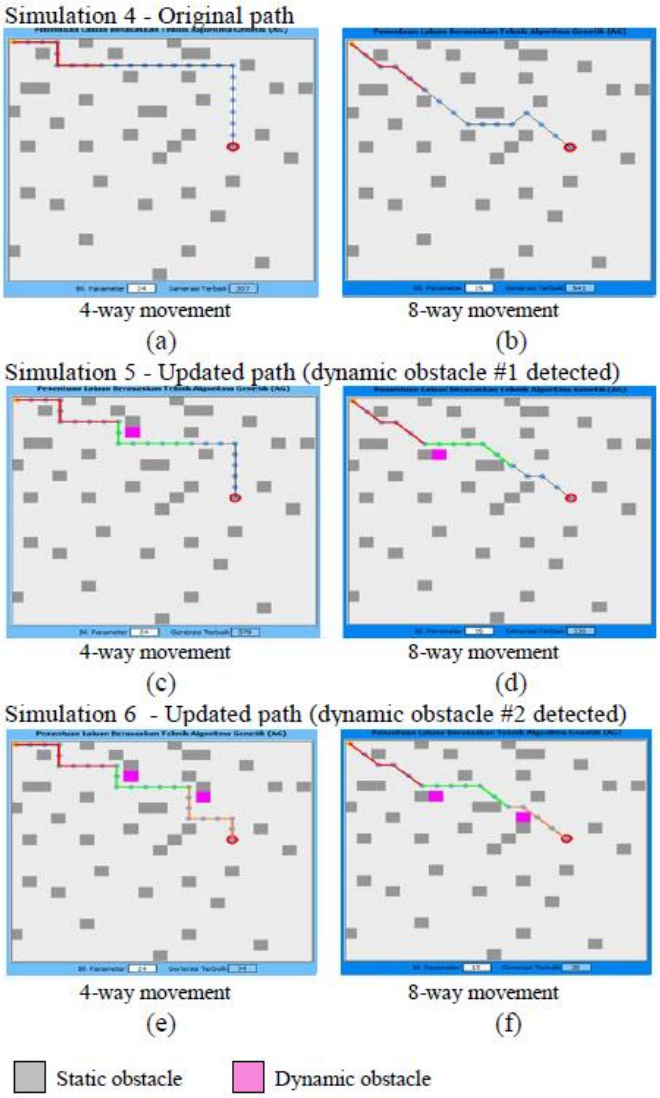


Fig. 16. Simulation results in environment with static and dynamic obstacles [69].

2.5 Research Gap

After analysing in detail the solutions described in this chapter with regards to the taxiing problem being addressed, a number of limitations were identified as follows:

- Most of the algorithms proposed in the literature for automating tow truck movements are only partially automated and require the intervention of ATCOs to issue clearances and/or to solve potential conflicts which might occur between vehicles during taxiing operations, adding significant ATCO workload;

- In order to achieve results in shorter timeframes, some works in the literature allow solutions with potential vehicle conflicts during the strategic planning phase. These are then resolved during the tactical phase or through real-time ATCOs interventions;
- Most of the works related to tow truck-based taxiing use algorithms designed to identify conflict-free routes, however little has been done to allocate tow trucks in an efficient manner (e.g. minimising the taxi delays or maximising the fuel savings);
- The metrics chosen for assessing the performance of the proposed solutions in the literature are usually limited to taxi delays and number of potential vehicle conflicts. Additional metrics – such as fuel consumptions and fuel savings – should be incorporated, as these would highlight the expected environmental benefits of using tow trucks;
- The majority of works that adopt a strategic approach assume an entirely deterministic environment and do not introduce any degree of uncertainty in any parameter being used. This approach is theoretically valid but cannot quantify the robustness of solutions in practical environments;
- The taxiing solutions proposed in the literature are tested on a single airport layout. This is acceptable when tuning the system to a particular airfield, but does not guarantee that the proposed solution can be applied to airports with different sizes and geometries. For instance, if an algorithm performs well when applied to in a large airport with an extensive network of taxiways, it is not guaranteed that similar results will be achieved for a much smaller airport, where taxiway bottlenecks are common and the possibility of having deadlocks is high. Similarly, if an algorithm is only tested in a small or medium-size airport, the scalability of the solutions is not guaranteed;
- Most of the performance evaluations of the taxiing algorithms proposed in literature are conducted using a single (medium) level of traffic. This approach, although valid for initial testing purposes, is not sufficient to fully extract the robustness of the system to varying levels of traffic;
- Performance evaluations are carried out with a number of tow trucks that does not change during simulation testing. Thus, it is not clear how the proposed algorithms would behave with different numbers of tow trucks. Furthermore, no studies have been carried out to determine the ideal number of tow trucks (i.e. the size of the fleet of tow trucks) that would be required to cope with different levels of aircraft traffic at an airport;
- Performance analyses are carried out without taking into account the state-of-charge of tow truck batteries during scheduling. This approach skews the performance of the tow trucks and underestimates the number of tow trucks that might be needed to carry out the taxi operations.

3 Problem Definition

The problem definition of the work consists of two main parts. The first consists of guaranteeing safe taxi operations for all vehicles by finding optimal conflict-free taxi routes for all departing and landing aircraft and while optimizing a number of aircraft performance metrics, such as taxi delays and fuel consumption. The second part of the problem consists in allocating a tow truck to each aircraft (when possible), optimizing a number of tow truck performance metrics such as their utilisation, while guaranteeing that aircraft taxi performance is not significantly affected by the introduction of the tow trucks. A key assumption of this work is that no ATC clearance is required to taxi anywhere on the airport. Additionally, it assumes that aircraft and tow trucks are the sole vehicles operating on the airfield.

The purpose of the rest of this chapter is to define the problem. First it discusses the airport environment (with regards to taxi operations), the tow truck depots and the airport modelling. Then, the mission of the tow trucks is explained; some of the variables used throughout the rest of the dissertation – to determine the vehicles' paths, taxi times and fuel consumption – are defined; the flight schedule is introduced; and the concept of time windows is explained. Finally, vehicle conflicts and conflict-free path planning are described.

3.1 Airport Environment

This section describes the elements of an airport environment which are relevant to taxi operations.

3.1.1 Aprons and Aircraft Stands

The aprons are the areas of an airport where the aircraft are parked, refuelled, boarded and maintained; therefore they are located in close proximity to gates, hangars and other airport facilities. Each apron is divided into a number of aircraft stands, each able to host a single aircraft. An aircraft stand is assigned to each arriving aircraft before landing or during taxi operations, while each departing aircraft starts taxiing from an aircraft stand. In this work it is assumed that pushback operations are not required (i.e. the departing aircraft are ready to taxi at their assigned scheduled time) and the tow trucks are only used for taxi operations. It is also assumed that if a tow truck is assigned to an aircraft, the attachment (in case of a departure) or detachment (in case of an arrival) operations are carried out at the assigned aircraft stand.

3.1.2 Airport Roads

The airport's roads can be divided into three main categories: runways, taxiways and service roads. Runways are reserved for take-offs and landings and cannot be used for taxi operations by aircraft or tow trucks unless there is no alternative route and the runways are unoccupied. This work assumes that all taxiways and runways at an airport are uniformly capable of supporting all aircraft types. Taxiways connect runways with other sections of the airport (including aprons, hangars and other facilities). It is assumed that all taxiways at an airport are uniformly capable of supporting all aircraft types and that they can also be used by tow trucks. Service roads are intended for small vehicles and connect various parts of the airport e.g. an aircraft hangar and a flying school. They can be used by vehicles, such as cars and unloaded tow trucks, but are not suitable for aircraft movements. No ATC clearance is required to enter a service road.

3.1.3 Take-Off and Landing Points

The take-off and/or landing direction of a runway in use is defined by ATCOs according to a number of factors such as the current wind speed and direction. While in real-world operations an aircraft can enter a runway at different points (depending on the number of runway holding points), or vacate the runway from different points, in this work it is assumed that aircraft will require most of the length of the runway for take-off and landing. Therefore, once the take-off and/or landing direction is defined, a point towards the beginning of the runway (according to the runway direction) is defined as the **Take-off Point** (*ToP*), where it is expected that a departing aircraft concludes its taxi operations and begins its take-off roll, while a point towards the end of the runway (according to the runway direction) is defined as the **Landing End Point** (*LEP*), where it is expected that an arriving aircraft concludes its landing phase and begins taxiing.

In this work the following procedure is introduced for tow truck taxi operations: when a tow truck is assigned to a landing aircraft, the attachment operations are carried out in a taxiway adjacent to or near the runway *LEP* (which is close to a runway holding point), as the aircraft vacates the runway using its own engines before being attached to a tow truck. This work assumes that no engine warm-up would be required for departing aircraft; therefore, when a tow truck tows a departing aircraft, the detachment operations are carried out in a taxiway adjacent to or near the runway *ToP* (which is close to a runway holding point), and then the aircraft enters the runway using its own engines.

3.1.4 Tow Truck Depots

The use of autonomous electric tow trucks for taxi operations requires the implementation in the aerodrome of a number of depots, which can be used by the tow trucks for parking and battery recharging purposes. Since most airports do not currently make use of electric tow truck-based aircraft taxiing operations – and hence do not have any tow truck depots –, the

locations of the depots were defined manually for each airport that was considered in this work. The location of the tow truck depots was based on the following criteria: absence of other facilities in the location; ease of connection with nearby taxiways and service roads; and proximity to one or more aprons. Each tow truck depot is connected to the rest of the airport's road network via one or (more) short connections, which were defined manually and classified as service roads.

The set of tow truck depots in an airport is defined as:

$$Q = \{b_1, \dots, b_b, \dots, b_B\} \quad (1)$$

b_1 is the first depot,
b_b is the bth depot,
b_B is the last depot, and
B is the total number of depots in the airfield.

The number of parking slots available per depot is a design choice and can vary from one airfield to another. To ensure an adequate number of free charging points, the total number of parking slots – defined as **Parking Slots Number** (S_b) – at each depot b is given by:

$$S_b = \lceil 1.5 \times (R/B) \rceil \quad (2)$$

where:
R is the total number of tow trucks.
Using this equation, the total number of charging points is always greater than the number of tow trucks and scales in proportion to the size of the tow truck fleet. This surplus of charging points is particularly critical during peak periods of taxiing activity, which may be concentrated in specific airport areas (for instance, when only some of the runways are active). Such strategic distribution guarantees that most tow trucks will always have the opportunity to recharge at a conveniently located depot, thereby maintaining efficient ground operations.

3.1.5 Airport Model

The position of a point on the surface of an airport is expressed in terms of x-y coordinates. For each airport, the origin of the coordinate system is manually defined at the south-west of the airport, with the x-coordinate increasing from West to East, and the y-coordinate increasing from South to North.

The airport is modelled as a directed graph linking the airport's roads, stands and tow truck depots. The graph consists of two main components: nodes and edges. The nodes represent all of the relevant points of the airport, including the aircraft stands, take-off points, landing points, tow truck depots, and intersections between runways, taxiways and service roads. The ith node n_i is represented as follows:

$$n_i = (n_{ID.i}, x_i, y_i) \quad (3)$$

where:
$n_{ID.i}$ is the ID of n_i,
x_i is the x-coordinate of n_i (with 1 m precision), and
y_i is the y-coordinate of n_i (with 1 m precision).

On the other hand, the edges, which connect pairs of nodes, represent the airport's roads. An edge e_m exists between nodes n_j and n_k if the nodes n_j and n_k are physically connected by a road (e.g. a taxiway). In this case e_m is represented as follows:

$$e_m = \left(e_{ID.m}, n_{ID.j}, n_{ID.k}, p_{ID.m}, w_m\right) \quad (4)$$

where:
$e_{ID.m}$ is the ID of e_m,
$n_{ID.j}$ is the ID of the first node connected to e_m,
$n_{ID.k}$ is the ID of the second node connected to e_m,
$p_{ID.m}$ is the ID of the airport road corresponding to e_m (as explained in Section 3.1.2), and
w_m is the weight (or cost) associated with edge e_m. This is proportional to the physical distance between the nodes n_j and n_k, and it is dynamically adjusted, as explained in Section 5.5.

Table 3 shows the relation between $p_{ID.m}$ and the airport road type.

Table 3. Airport road types.

$p_{ID.m}$	Airport Road Type
1	Runway
2	Taxiway
3	Service road

Fig. 17-Fig. 20 show a graphical representation of the airports analysed in this work together with the corresponding directed graph.

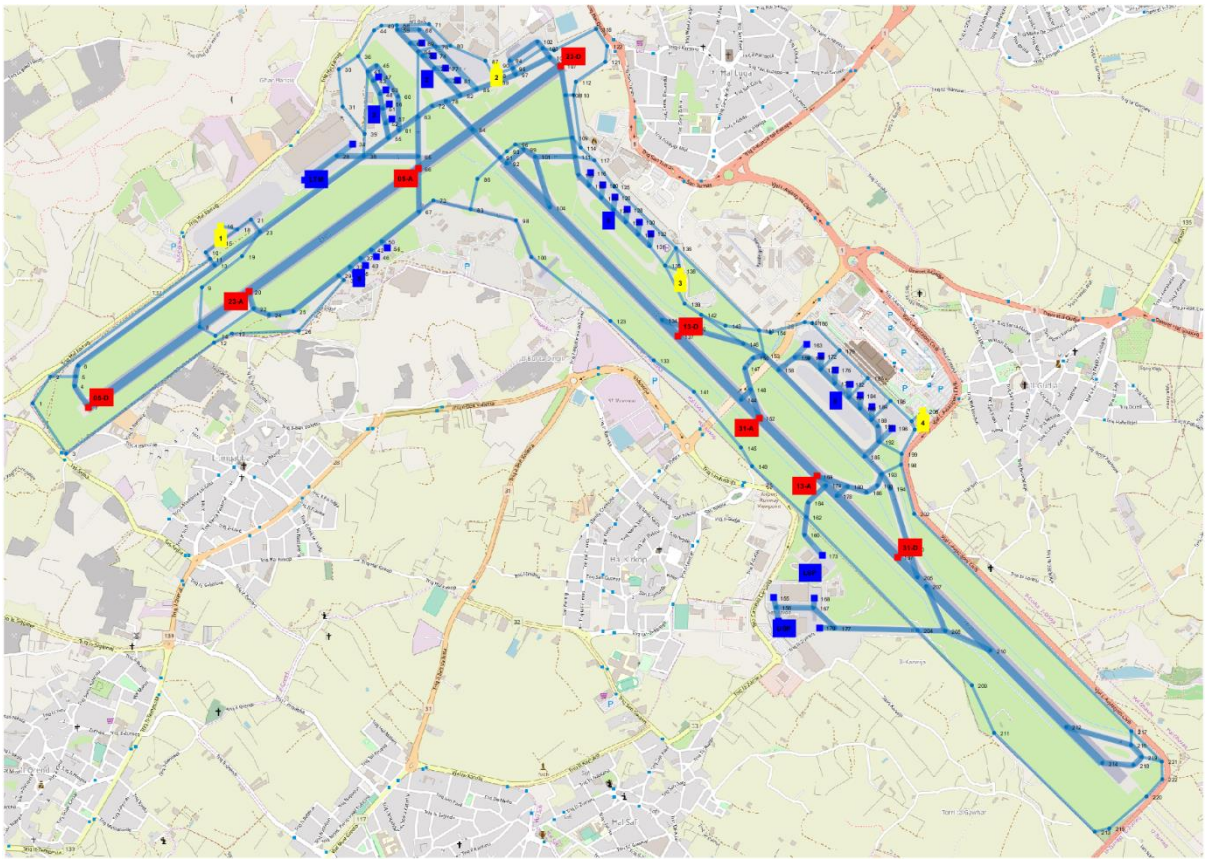


Fig. 17. Directed graph for Malta International Airport (International Air Transport Association (IATA) Airport Code: MLA). The thickest lines, the medium thickness lines and the thinnest lines represent the runways, taxiways and service roads respectively. The blue nodes represent the aircraft stands, the red nodes mark the ToPs/LEPs and the yellow nodes show the position of the tow truck depots.

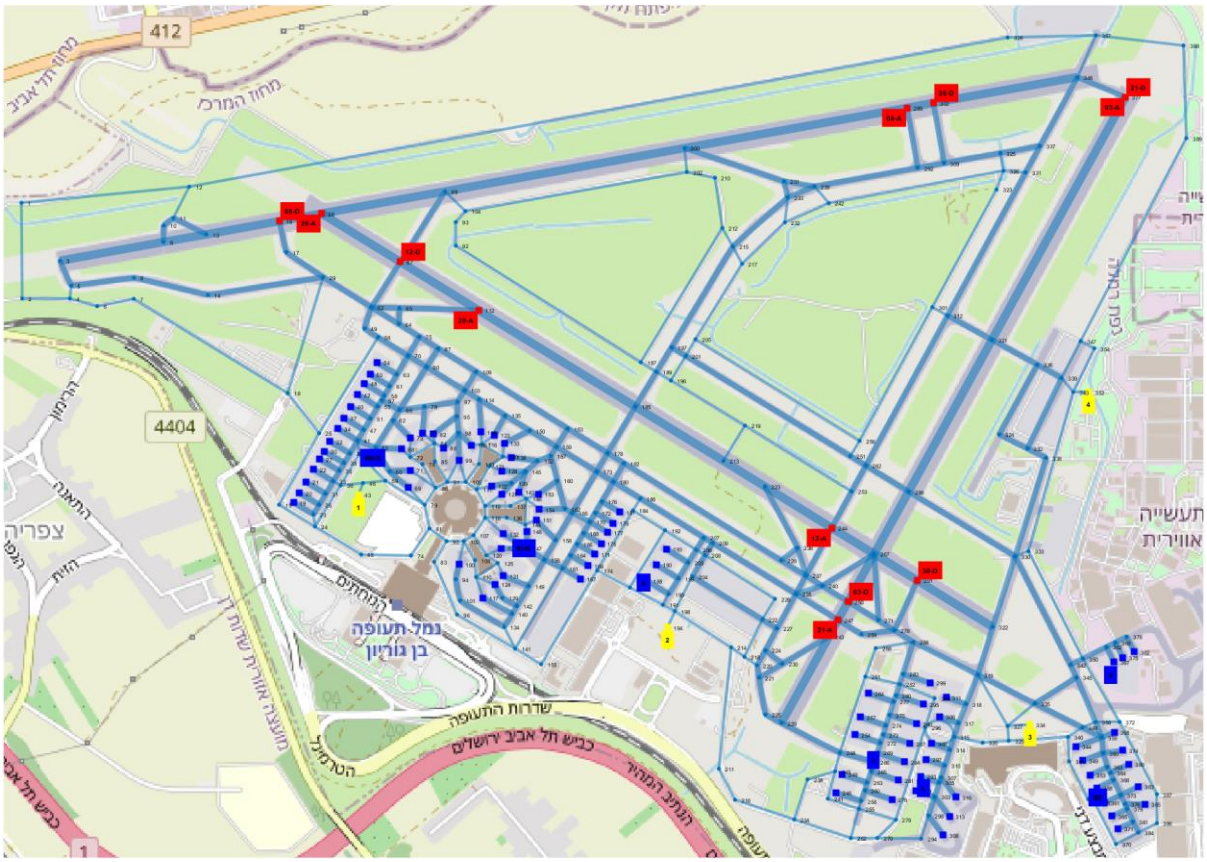


Fig. 18. Directed graph for Ben Gurion Airport (IATA Airport Code: TLV). The thickest lines, the medium thickness lines and the thinnest lines represent the runways, taxiways and service roads respectively. The blue nodes represent the aircraft stands, the red nodes mark the ToPs/LEPs and the yellow nodes show the position of the tow truck depots.

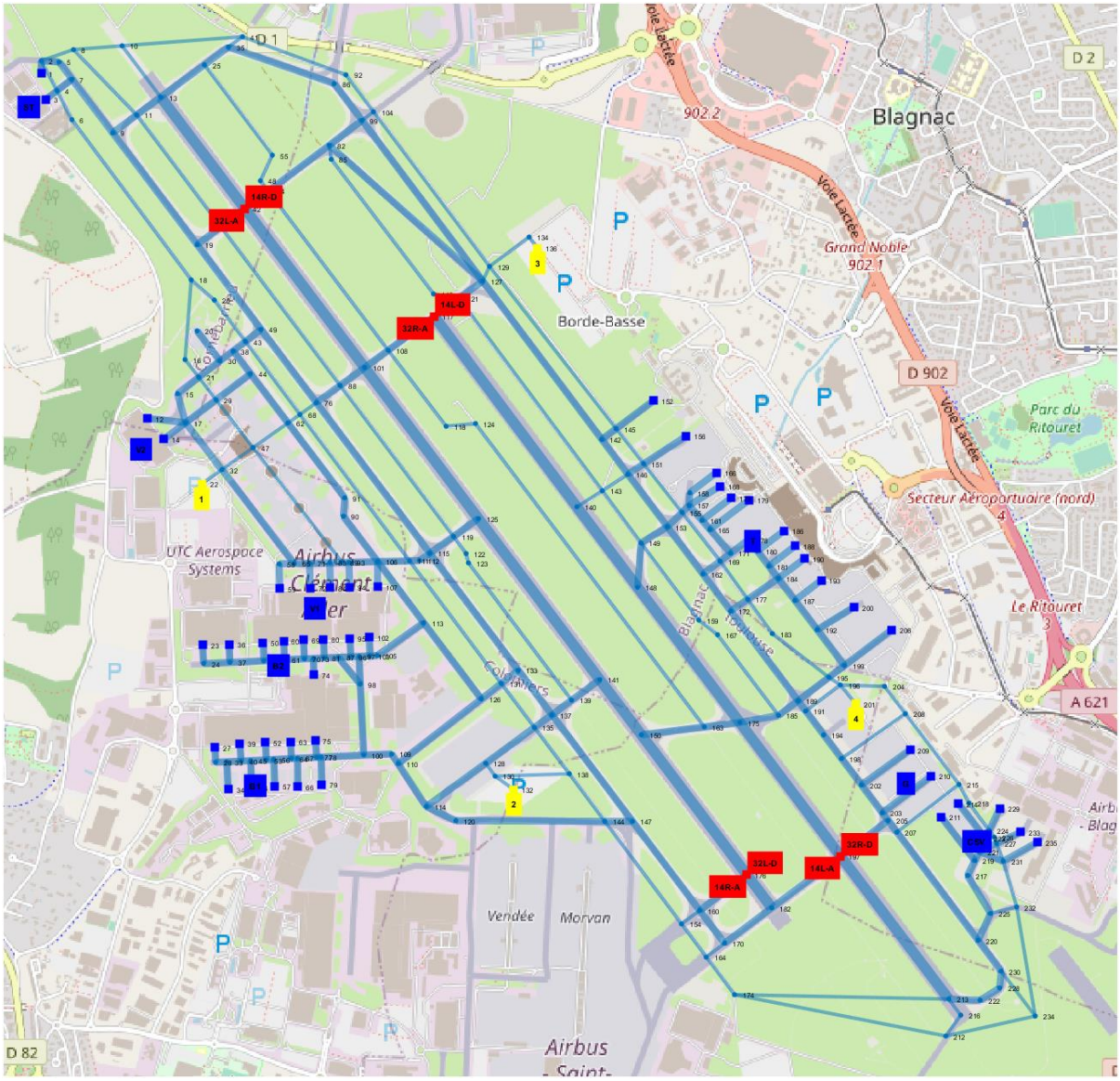


Fig. 19. Directed graph for Toulouse–Blagnac Airport (IATA Airport Code: TLS). The thickest lines, the medium thickness lines and the thinnest lines represent the runways, taxiways and service roads respectively. The blue nodes represent the aircraft stands, the red nodes mark the ToPs/LEPs and the yellow nodes show the position of the tow truck depots.

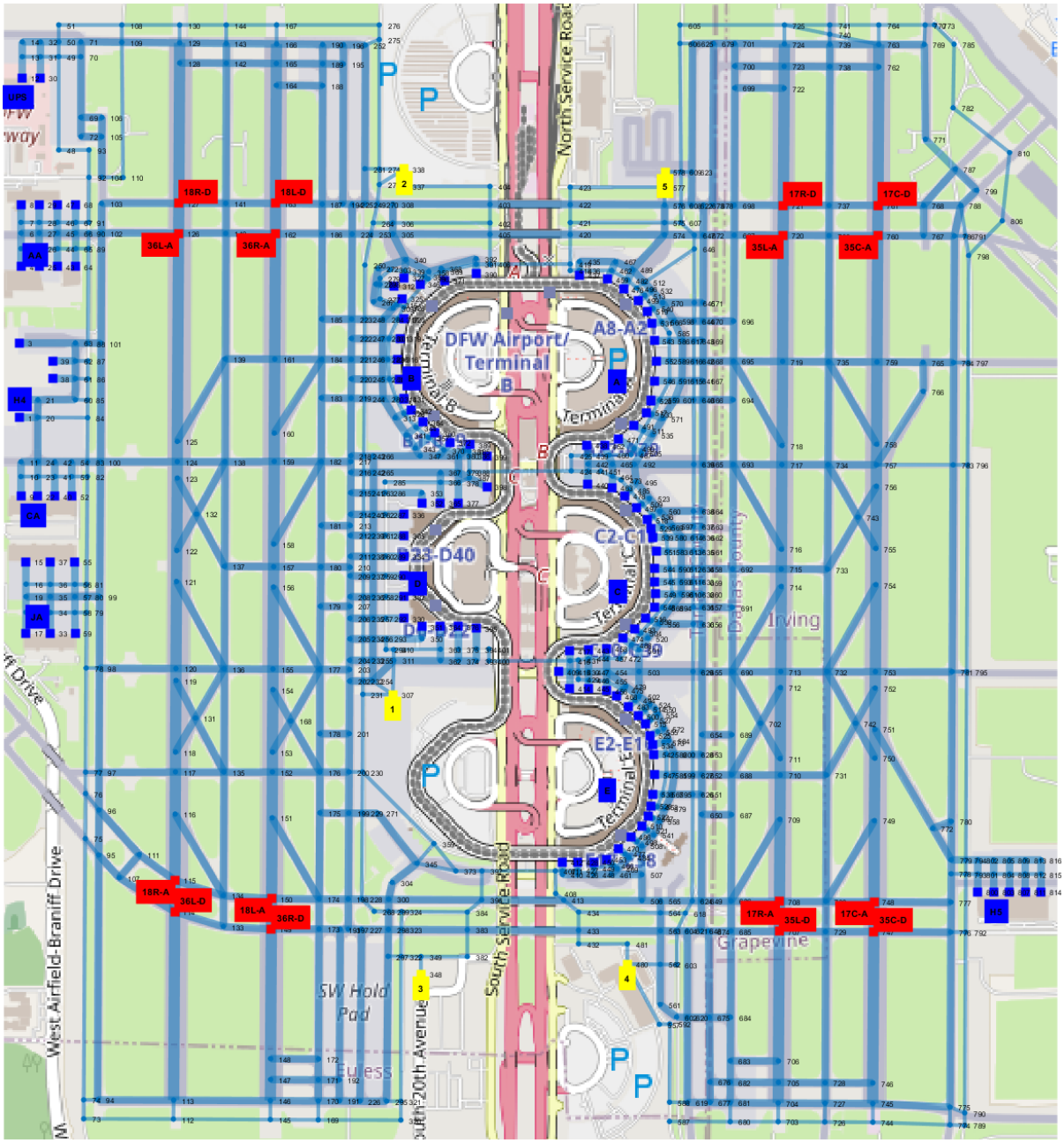


Fig. 20. Directed graph for Dallas Fort Worth International Airport (IATA Airport Code: DFW). The thickest lines, the medium thickness lines and the thinnest lines represent the runways, taxiways and service roads respectively. The blue nodes represent the aircraft stands, the red nodes mark the ToPs/LEPs and the yellow nodes show the position of the tow truck depots.

3.2 Tow Truck Mission

A tow truck is allocated, whenever possible, to each departing or arriving aircraft. In this work it is assumed that a tow truck is able to tow all types of aircraft. For a departing aircraft, the mission of a tow truck consists of three phases: (a) travel from a tow truck depot (or another position in the airport) to the aircraft stand and attach to the aircraft; (b) tow the aircraft to

its *ToP*; and (c) detach from the aircraft and return to one of the tow truck depots (or travel directly to another aircraft). Similarly, for an arriving aircraft, the mission of a tow truck consists of three tasks: (a) travel from a tow truck depot (or another position in the airport) to the *LEP* of the aircraft and attach to the aircraft; (b) tow the aircraft to its allocated stand; and (c) detach from the aircraft and return to one of the tow truck depots (or travel directly to another aircraft).

If a tow truck is assigned to a depot during the phase (c), this could be either the same tow truck depot of the phase (a) (i.e. the depot which the tow truck left at the beginning of its mission) or a different tow truck depot.

The three phases of a mission are defined in the rest of the work as follows:

- The *Reaching Mission Phase* (**ReaM**), when the tow truck travels from its position to the initial position of the aircraft and then attaches to the aircraft;
- The *Towing Mission Phase* (**TowM**), when the tow truck tows the aircraft from its initial position to its final position;
- The *Returning Mission Phase* (**RetM**), when the tow truck detaches from the aircraft and then returns from the aircraft's final position to a depot (this phase could be partially cancelled if the tow truck is assigned to another aircraft before reaching its assigned depot).

These three phases together form a complete tow truck mission, defined as *Reaching-Towing-Returning Mission* (**RTRM**).

3.3 Definition of Variables

This section defines the variables which are used in the rest of this work. The precision of the distance-related variables is 1 m, while the precision of the time-related variables is 1 s.

The set of aircraft in the airport is defined as:

$$U = \{a_1, \dots, a_a, \dots, a_A\} \tag{5}$$

a_1 is the first aircraft,
a_a is the ath aircraft,
a_A is the last aircraft, and
A is the total number of aircraft in the flight schedule (as explained in 3.4).

On the other hand, the set of tow trucks is defined as:

$$V = \{r_1, \dots, r_r, \dots, r_R\} \tag{6}$$

r_1 is the first tow truck,

r_r is the r[th] tow truck,
r_R is the last tow truck, and
R is the total number of tow trucks.

3.3.1 Definition of Path-Related Variables

The complete set of nodes of an airport graph is defined as **Airport Graph Nodes Set** (G_{nodes}), while its complete set of edges is defined as **Airport Graph Edges Set** (G_{edges}). In the airport graph, the initial position of a generic vehicle is defined as the **Start Node** (n_{st}), while its final position is defined as the **End Node** (n_{end}).

The initial position of an aircraft is defined as **Aircraft Start Node** (n_{st}^{a}) and its final position is defined as the **Aircraft End Node** (n_{end}^{a}). On the other hand, the initial position of a tow truck is defined as **Tow Truck Start Node** (n_{st}^{r}) and its final position is defined as the **Tow Truck End Node** (n_{end}^{r}).

The path which a vehicle follows from n_{st} to n_{end} consists of a set of nodes and is defined as the **Path Nodes Set** (P_{nodes}):

$$P_{nodes} = \{n_1, \dots, n_n, \dots, n_N\} \tag{7}$$

where:
n_1 is the first node and is equal to n_{st},
n_n is the n[th] node,
n_N is the last node and is equal to n_{end}, and
N is the total number of nodes in the path.

The set of times when a vehicle passes through the assigned P_{nodes} is defined as the **Path Times Set** (P_{times}):

$$P_{times} = \{t_1, \dots, t_n, \dots, t_N\} \tag{8}$$

where:
t_1 is the time when the vehicle passes through n_1,
t_n is the time when the vehicle passes through n_n, and
t_N is the time when the vehicle passes though n_N.

The edges that a vehicle follows in order to pass through the assigned P_{nodes} is defined as the **Path Edges Set** (P_{edges}):

$$P_{edges} = \{e_1, \dots, e_e, \dots, e_E\} \tag{9}$$

where:
e_1 is the first edge and passes between n_1 and n_2,
e_e is the e[th] edge and passes between n_n and n_{n+1},
e_E is the last edge and passes between n_{N-1} and n_N, and

E is the total number of edges and is equal to $N - 1$.

The **Minimum Path Times Set** ($P_{times.min}$) represents the time when the vehicle enters each edge and is defined as follows:

$$P_{times.min} = \{t_{min.1}, \dots, t_{min.e}, \dots, t_{min.E}\} \quad (10)$$

where:
$t_{min.1}$ is the time when the vehicle enters e_1, and is equal to t_1,
$t_{min.e}$ is the time when the vehicle enters e_e, and is equal to t_n, and
$t_{min.E}$ is the time when the vehicle enters e_E, and is equal to t_{N-1}, where
t_{N-1} is the time when the vehicle passes though n_{N-1}, and
n_{N-1} is the node before n_N.

The **Maximum Path Times Set** ($P_{times.max}$) represents the time when the vehicle leaves each edge and is defined as follows:

$$P_{times.max} = \{t_{max.1}, \dots, t_{max.e}, \dots, t_{max.E}\} \quad (11)$$

where:
$t_{max.1}$ is the time when the vehicle leaves e_1, and is equal to t_2,
$t_{max.e}$ is the time when the vehicle leaves e_e, and is equal to t_{n+1}, and
$t_{max.E}$ is the time when the vehicle leaves e_E, and is equal to t_N.

The **Path Edges Length Set** ($P_{edges.l}$) represents the geometrical lengths of the edges of the path and is defined as follows:

$$P_{edges.l} = \{l_1, \dots, l_e, \dots, l_E\} \quad (12)$$

where:
l_1 is the length of e_1,
l_e is the length of e_e, and
l_E is the length of e_E.

Finally, the **Path Start Distances Set** ($P_{st.dist}$) represents the distance of each node in the path from n_1 and is defined as follows:

$$P_{st.dist} = \{d_1, \dots, d_n, \dots, d_N\} \quad (13)$$

where:
d_1 is equal to 0,
d_n is the distance of n_n from n_1 and is equal to $\sum_{i=1}^{n-1} l_i$, and
d_N is the distance of n_N from n_1 (i.e. the total length of the path) and is equal to $\sum_{i=1}^{E} l_i$.

3.3.2 Definition of Aircraft Time-Related Variables

Each departing flight is estimated to leave the stand and taxi towards a runway at a specific time, commonly referred to in aviation as the **Estimated Off-Block Time** (*EOBT*) [70]. Also, in aviation the **Scheduled Off-Block Time** (*SOBT*) [71] is defined as the time when a departing aircraft is scheduled to depart from its parking position. Finally, the **Target Off- Block Time** (*TOBT*) [71] is the time that an aircraft operator estimates that an aircraft will be ready to startup or pushback upon reception of clearance from ATC. In this work it is assumed that the *EOBT*, *SOBT* and *TOBT* are all equal to each other.

On the other hand, the time when an arriving flight is estimated to land, commonly referred to in aviation as **Estimated Landing Time** (*ELDT*) [71] or **Estimated Time of Arrival** (*ETA*) [72], corresponds to the time when the aircraft is expected to touchdown on the runway. The **Scheduled Landing Time** (*SLDT*) [71] is the time when an aircraft is scheduled to land, while the **Target Landing Time** (*TLDT*) [71] is the time when an aircraft is expected to land taking into account the queue of arriving aircraft. For the purpose of this work, the *ETA* of an aircraft is assumed to be the time when an aircraft is estimated to start its taxi operations after landing, and the *ELDT* and the *TLDT* are equal to the *ETA*. When an aircraft lands, it is expected to vacate the runway without delay in order not to interfere with subsequent arrivals and departures; therefore, an aircraft generally vacates the runway and waits at a runway holding point for taxi clearance.

The *EOBT* and *ELDT* (or *ETA*), which both correspond to the time when an aircraft is expected to start its taxi operations, are collectively defined in this work as **Ideal Start Time** (t_{is}).

In aviation the time that an aircraft actually vacates the parking position is defined as **Actual Off-Block Time** (*AOBT*) [71], while the actual time that an aircraft lands on a runway is defined as **Actual Landing Time** (*ALDT*) [71]. For the purpose of this study, the *ALDT* of an aircraft is assumed to be the time when an aircraft actually starts its taxi operations after landing.

The *AOBT* and *ALDT*, which both correspond to the time when an aircraft actually starts its taxi operations, are collectively defined in this work as **Actual Start Time** (t_{as}). The delay experienced by an aircraft while waiting next to the runway (for arrivals) or at the stand (for departures) is defined as **Start Delay** (Δt_{ds}) and is given by:

$$\Delta t_{ds} = t_{as} - t_{is} \tag{14}$$

In busy airspace, a departing aircraft is usually assigned an Air Traffic Management (ATM) slot within which it should be airborne, called **Calculated Take-Off Time** (*CTOT*) [71, 73-74]. In Europe, the CTOT is calculated by Eurocontrol's Network Manager Operations Centre (NMOC) [75-76] and takes into account the aircraft's *EOBT* and the average taxi time at an airport. A departing aircraft should take off within 5 minutes before the *CTOT* or within 10 minutes after the *CTOT*, and, if it cannot achieve this, it has to reapply for another time slot [77]. In this work it is assumed that the aircraft are always ready to taxi at the time defined in the flight

schedule, called referred as **Flight Schedule Time** (t_{fs}), but never earlier than t_{fs}, and a degree of flexibility to delay the t_{fs} is given not just to the departures but also to the arrivals. Finally, the 10 minutes of delay allowed in the *CTOT* are set in this work as the maximum delay which is permitted to start the taxi operations and this applies both for departures and arrivals. Therefore:

$$0 \leq \Delta t_{ds} \leq 600\ s \tag{15}$$

The time taken by an aircraft to taxi along an ideal (shortest) route without interference along its path is defined as the **Ideal Taxi Time** (Δt_{it}), while the actual amount of time taken by an aircraft to taxi along its assigned path is defined as the **Actual Taxi Time** (Δt_{at}). The time taken by an aircraft to complete an ideal (shortest) route without interference along its path is defined as the **Shortest Path End Time** (t_{spe}), while the actual time taken when an aircraft completes its assigned path is defined as the **Actual End Time** (t_{ae}). The delay accumulated by an aircraft while taxiing is referred to as **Taxi Delay** (Δt_{dt}) and is given by:

$$\Delta t_{dt} = t_{ae} - t_{spe} \tag{16}$$

The total delay experienced by an aircraft during taxi operations is defined as the **Total Delay** (Δt_{dtot}) and is given by:

$$\Delta t_{dtot} = \Delta t_{ds} + \Delta t_{dt} \tag{17}$$

Fig. 21 shows a graphical representation of the variables introduced in Eqs. 15-17.

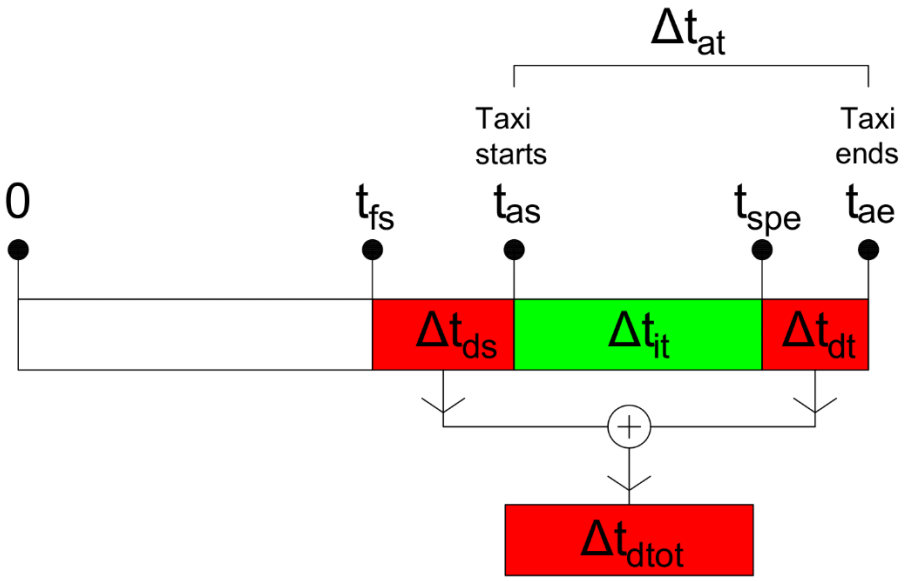


Fig. 21. Graphical representation of the time-related variables associated with taxiing.

3.3.3 Definition of Tow Truck Time-Related Variables

The time required for a tow truck to attach to an aircraft is defined as **Attachment Time** (Δt_{att}), while the time required to detach from it is defined as **Detachment Time** (Δt_{det}). Both Δt_{att} and Δt_{det} are set equal to 30 s.

The time when a tow truck is set to leave the depot is defined as **Depot Leaving Time** (t_{dl}), while the time when a tow truck is planned to reach the assigned aircraft is defined as **Aircraft Reaching Time** (t_{ar}). The time required for a tow truck to reach its assigned aircraft after leaving from the depot is called **Reaching Time** (Δt_{rea}) and is given by:

$$\Delta t_{rea} = t_{ar} - t_{dl} \tag{18}$$

The time required for a tow truck to reach its assigned aircraft after leaving from the depot and to attach to the aircraft is called **Reaching Mission Time** (Δt_{ream}) and is given by:

$$\Delta t_{ream} = \Delta t_{rea} + \Delta t_{att} \tag{19}$$

The time when a tow truck is set to leave the assigned aircraft is defined as **Aircraft Leaving Time** (t_{al}), while the time when a tow truck returns to the depot is defined as **Depot Returning Time** (t_{dr}). The time required for a tow truck to return to the depot after detaching from its assigned aircraft is called **Returning Time** (Δt_{ret}) and is given by:

$$\Delta t_{ret} = t_{dr} - t_{al} \tag{20}$$

The time required for a tow truck to detach from an aircraft and to reach its assigned tow truck depot is called **Returning Mission Time** (Δt_{retm}) and is given by:

$$\Delta t_{retm} = \Delta t_{det} + \Delta t_{ret} \tag{21}$$

The time required for a tow truck to tow the assigned aircraft is called **Towing Mission Time** (Δt_{towm}) and is equal to the Δt_{at} of the assigned aircraft.

Fig. 22 shows a graphical representation of the variables introduced in Eqs. 18-21.

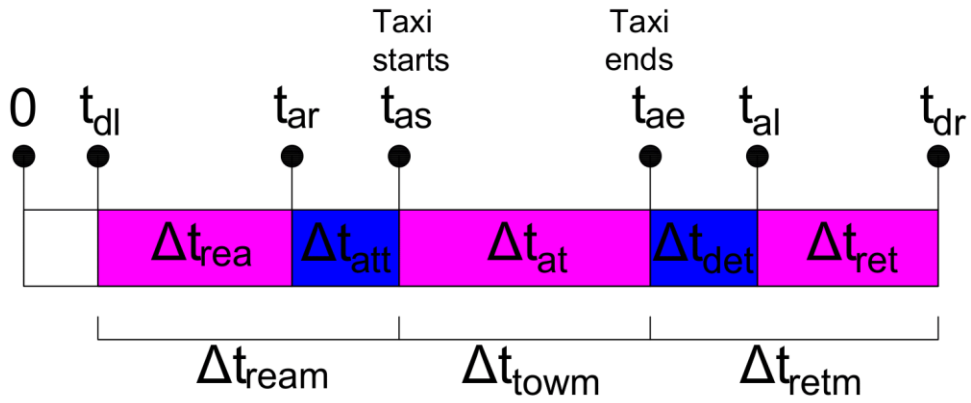


Fig. 22. Graphical representation of the time-related variables of a tow truck.

Finally, the total time during which a tow truck is utilised is defined as **Tow Truck Utilisation Time** (Δt_{ru}) and corresponds to the total time during which the tow truck is not parked in a depot.

3.3.4 Definition of Fuel-Related Variables

The fuel consumed by an aircraft when taxiing using its own engines along an ideal (shortest) route without interference along its path is defined as **Minimum Fuel Consumption** (F_{cmin}), while the actual fuel that an aircraft consumes when taxiing using its own engines along its assigned path is defined as **Actual Fuel Consumption** (F_{ca}). The difference between F_{ca} and F_{cmin} is defined as **Extra Fuel Consumption** (ΔF_c) and is given by:

$$\Delta F_c = F_{ca} - F_{cmin} \tag{22}$$

F_{ca} and F_{cmin} are calculated using a model developed by Khadilkar et al. [78], which estimates the on-ground fuel consumption of an aircraft, given its surface trajectory. The taxi fuel burn is modelled as a linear function of several factors including taxi time, number of stops, number of turns and number of acceleration events, and the parameters are estimated using Flight Data Recorder (FDR) archives provided by ICAO. The results provided in [78] show that the number of stops and acceleration events is a significant factor in determining taxi fuel consumption, indicating that choosing no-stop taxi solutions leads to significant fuel savings. Also, the geometry of the airport (i.e. the number and the sharpness of curves along taxiways) has a significant impact on the fuel consumption. The model is calibrated for taxi-out operations; however, in this work, it is extended to taxi-in operations as well.

3.4 Definition of Flight Schedule

An airport's taxi operations are dependent on the airport's flight schedule, which determines the time frame of the arrivals and departures. Table 4 shows an extract of a flight schedule generated and used in this work.

The values of the flight schedule's variables are assigned randomly to each flight; however, some constraints are applied. In particular, both departures and arrivals represent 50% of the total flights. The *EOBT* and *ETA* are assigned in ascending order and the values are rounded to 1 minute. The parking stands (and the corresponding aprons) are assigned sequentially from the first flight, and, once assigned, become unavailable to the next flight for 20 minutes; this time was arbitrarily selected to avoid the possibility of two or more aircraft occupying the same parking stand at the same time. The aircraft types are assigned completely randomly from the set of values shown in Table 5, which correspond to the types modelled in Khadilkar et al. [78] and influence the fuel consumption of the taxi operations as estimated in the model.

Table 4. Example of flight schedule generated for Ben Gurion Airport in Tel Aviv.

Flight	*Aircraft Type*	*D/A* [a]	*EOBT/ETA*	*Apron*	*Stand*
1	A330-202	D	10:00:00	APR J	257
2	A319	D	10:05:00	APR J	276
3	B767	A	10:05:00	APR WHS	71
4	A330-202	A	10:07:00	APR L	313
5	ARJ85	A	10:10:00	APR WHS	73
6	A340-500	D	10:10:00	APR EHS	138
7	A319	A	10:12:00	APR WHS	99
8	ARJ85	A	10:14:00	APR WHS	115
9	A330-202	D	10:15:00	APR WHS	19
10	A321	A	10:20:00	APR BE	374
11	A330-234	A	10:22:00	APR EHS	179
12	ARJ85	D	10:25:00	APR EHS	167
13	B767	D	10:25:00	APR EHS	181
14	A340-500	D	10:30:00	APR EHS	126
15	A319	A	10:30:00	APR EHS	127
16	B757	A	10:32:00	APR BE	385
17	A321	D	10:40:00	APR EHS	169
18	B767	D	10:40:00	APR L	300
19	ARJ85	A	10:45:00	APR L	289
20	ARJ85	A	10:47:00	APR L	316
21	A330-202	D	10:50:00	APR EHS	100
22	A319	D	10:55:00	APR WHS	106

[a.] Departure/Arrival

Table 5. Type of aircraft in the flight schedule.

Aircraft Types
A319
A320
A321
A330-202
A330-234
A340-500
ARJ85
B757
B767
B777

3.5 Time Discretisation

In this work, time is discretised into equal time blocks, defined as **Time Windows** (t_w). Each t_w has an associated **Time Window Start Time** (t_{ws}) and a **Time Window End Time** (t_{we}), representing the time when t_w starts and ends, respectively. t_w– defined as **Time Window Duration** (Δt_w) – has a duration arbitrarily selected equal to 10 s. The values of these three variables are organised in a table defined as **Time Windows Table** (*TWT*). Table 6 shows the initial part of a *TWT* with the first 6 t_w, with t_w^i, t_{ws}^i and t_{we}^i representing the i-th t_w, t_{ws} and t_{we}, respectively.

Table 6. Initial part of a *TWT*.

t^i_w	t^i_{ws} [mm:ss]	t^i_{we} [mm:ss]
1	00:00	00:10
2	00:10	00:20
3	00:20	00:30
4	00:30	00:40
5	00:40	00:50
6	00:50	01:00
…	…	…

3.6 Vehicle Conflicts

As explained in Section 1.1.5, a fundamental objective of the work is to create conflict-free paths for the vehicles (both aircraft and tow trucks); therefore, the occurrence of conflicts between vehicles must be avoided.

In order to minimise such conflicts, taxiways and runways do not allow bi-directional traffic of aircraft. Also, an aircraft moving in close proximity to another vehicle poses a safety hazard

and a minimum distance separation between two taxiing aircraft following each other must be maintained according to ICAO regulations. The distance must not be less than 50 meters due to aircraft jet blasts and ideally not less than a distance between 100 meters and 300 meters according to the type of aircraft [79], as shown in Table 7.

Table 7. Minimum separation rule for taxiing aircraft (units: meters) [79].

Type of leading aircraft	Type of trailing aircraft		
	Light (L)	Medium (M)	Heavy (H)
Light (L)	100	100	100
Medium (M)	200	200	200
Heavy (H)	300	300	300

In this work, a conflict is considered to occur when the geometrical centres of two vehicles – of which at least one is an aircraft or a tow truck towing an aircraft – come closer than 200 m. In order to implement this, for each vehicle a circular area (defined as **Buffer Area**, A_b) with a radius of 100 m (defined as **Buffer Distance**, d_b) and centred on the vehicle's geometrical centre is defined. A conflict is considered to occur when the circular areas of two vehicles intersect.

If both vehicles are unloaded tow trucks (i.e. tow trucks which are not towing aircraft), the minimum required distance between them is considered to be negligible and it is assumed that they can even cross each other's or travel side by side in all the types of airport roads (including service roads). Therefore, it is assumed that no conflicts arise between tow trucks in any case and A_b is consequently not applicable.

Fig. 23 represents all the possible scenarios of potential conflicts between vehicles and, when applicable, A_b around vehicles. These scenarios are applicable not only when the vehicles are moving in opposite directions of one other, as shown in Fig. 23, but also when they are moving in the same direction, and one vehicle is trailing another.

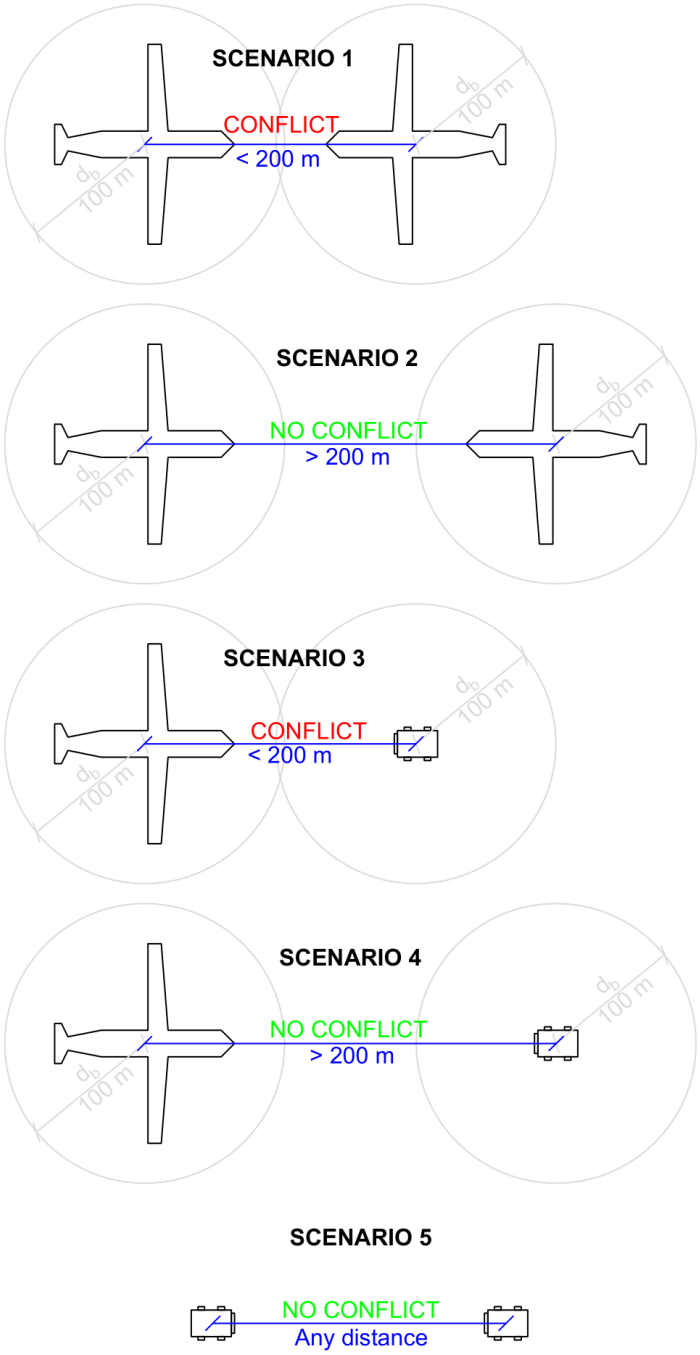


Fig. 23. Graphical representation of 5 potential conflict scenarios between vehicles. Scenarios 1-2 show potential conflicts between aircraft (or tow trucks loaded with aircraft); scenarios 3-4 represent potential conflicts between aircraft (or tow trucks loaded with aircraft) and unloaded tow trucks; scenario 5 represents potential conflicts between unloaded tow trucks. A_b around vehicles is represented when required (Scenarios 1-4).

4 System Operation and Design Choices

Based on the airport environment, the flight schedule and the tow truck fleet described in Chapter 3, this chapter gives a high-level description of how the system handles the problem of providing conflict-free taxi routes to all vehicles and assigning tow trucks to aircraft, whenever possible.

This chapter defines the overall objectives of the system and the design approach, discusses the choice of the core algorithm to define the vehicles' routes, objectives and settings of the two system's main parts (the aircraft scheduling and the tow truck allocation), and presents the vehicle motion model and the concept of buffer time between vehicles.

4.1 System Objectives and Design Approach

The purpose of the algorithm is to identify conflict-free taxi routes that allow as many aircraft as possible to taxi using electric tow trucks, thereby minimising fuel consumption. In addition, with regards to the aircraft, the algorithm aims to minimise their taxi delays, while with regards to the tow trucks, it attempts to maximize the utilisation of the available fleet.

In order to achieve this, the algorithm is divided in two main parts. In the first part, each aircraft is assigned a conflict-free route and its schedule can be adjusted within certain limits if no route is identified. In the second part, the algorithm allocates tow trucks to the aircraft, identifies conflict-free tow truck routes and creates a tow truck schedule.

The algorithm operates at a strategic level, i.e. all the routes, the adjusted aircraft schedule and the tow truck schedule are set up before the start of taxi operations. This approach was selected for the following reasons:

- A tactical approach, operating only when a potential conflict arises, might create situations of high traffic knock-on effects on other vehicles, increasing delays and creating further potential conflicts;
- As taxiways and runways do not allow bi-directional traffic and the aircraft are not allowed to backtrack (refer to Section 4.5), a tactical approach cannot avoid the creation of deadlocks where one or more vehicles end up blocking each other's path with no alternative route to reach their destination;
- As the number of elements of the airport environment (i.e. the number of nodes and edges) is limited to a few hundreds, and the number of vehicles moving at the same time around the airport is also limited to possibly a few tens, the execution time associated with a strategic approach is expected to be manageable.

Furthermore, the algorithm is based on a centralised approach, i.e. the algorithm is executed on a single computer (a 'central authority') which identifies conflict free routes for all the movements in the airfield. This choice is based on the following reasons:

- A centralised approach is already in use in real-world airport operations, where a central authority (i.e. ATC) is responsible for assigning taxi routes and preventing vehicle conflicts, and pilots and airport operators communicate with this central authority. The algorithm – acting as a central 'authority' – will be equivalent to ATC;
- In this work, the central authority has knowledge of the entire situation i.e. it is aware of the state and intent of all ground vehicles. Therefore, a centralised approach would be suitable;
- Since an airport is a restricted and controlled environment with a limited number of vehicles (depending on the airport's capacity), it is practical and safer to have a central authority that monitors all ground movements and vehicle operations.

4.2 Path Planning

As discussed in Chapter 2, various path planning techniques and algorithms were identified in the literature. The selection of an appropriate technique for this work was based on the following considerations and requirements:

- The problem is more manageable if aircraft routes are planned sequentially (i.e. one at a time) rather than all at once. While the SPOs could analyse one flights one at a time, EAs and MOAs analyse the problem as a whole and some adaptations should be made to tackle continuously updated requests [80-82];
- As the system should respond to practical needs to manage the airport traffic, the algorithm should be easy to tune in different airports, e.g. with regards to the system's constraints. SPOs are relatively simple to implement, as they involve traversing a graph and maintaining a list of the shortest path distances from a source node to all other nodes and the algorithm terminates when the destination node is reached or when all reachable nodes have been visited. On the other hand, MOAs involve optimizing an objective function subject to multiple constraints. In addition, MOAs often require more sophisticated data structures and numerical techniques to solve large-scale problems efficiently, meaning that implementing MILP algorithms can be challenging and might require significant computational resources [83-85];
- The algorithm should be able to cope with different types of airports, irrespective of airport size, layout, number of runways, etc. Tuning EAs and MOAs for operation in different environments is considered challenging [85-86]. A SPO based algorithm is much simpler to adapt;

- Given that the initial and final vehicle positions are different for each flight, and the geometry of the environment changes radically from one airport to another airport, it is important that the algorithm automatically finds the shortest route for each flight. In case of SPOs and EAs, these are able to identify the shortest path without having an initial guess, whereas for MOAs, providing a valid initial condition (i.e. a valid route) is critical and will improve the convergence speed [87]. In fact, if the initial condition is close to the optimal solution, then the algorithm will likely converge faster. On the other hand, if the initial conditions are not adequate, the algorithm may either take longer to converge, get stuck in local optima or keeps on executing indefinitely without converging [88];
- The results provided by the algorithm should be repeatable and deterministic, in order to facilitate system testing and validation in different conditions. SPOs and MOAs are repeatable given the same initial conditions. On the other hand, EAs are stochastic in nature since genes are evolved randomly during mutations and crossovers. Therefore repeatability of the solutions is not warranted [85, 89];
- The algorithm should have a short execution time in order to be able to provide in solutions within an acceptable time when applied to a real scenario. SPO algorithms and EAs have a relatively low execution time, while in some cases MOAs are unable to find a solution within a reasonable amount of time [90-91].

Table 8 summarises key features of SPOs, MOAs and EAs. Based on this table and the considerations above, an SPO based algorithm was selected for this work.

Table 8. Comparison of SPOs, MOAs and EAs.

Feature	SPOs	MOAs	EAs
Possibility of sequentiality	YES	NO	YES
Ease of implementability	HIGH	LOW	MEDIUM
Scalability	HIGH	LOW	LOW
Sensitivity to initial conditions	LOW	HIGH	LOW
Repeatability	YES	YES	NO
Speed of execution	FAST	SLOW	FAST

Three different SPOs were considered for implementation: Dijkstra [92-93], A* [94] and Floyd-Warshall [95]. These were tested and compared for the same airport [36]. The test results show that the performance of the three techniques is identical in terms of identification of the shortest path and execution time [49]. Such results were expected given that an airport's graph has a very limited number of nodes and edges, which makes the search for the shortest path very straightforward and fast. Since the performances of the three algorithms were equivalent, a Dijkstra-based SPO was used.

4.3 Aircraft Scheduling

The aircraft scheduler uses the SPO selected in Section 4.2 to find a conflict-free route for each aircraft in the flight schedule. For each aircraft, the scheduler assigns a path with corresponding nodes P_{nodes} and edges P_{edges} connecting n_{st}^{a} to n_{end}^{a} (which are defined in Section 3.3.1) together with a taxiing start time (which corresponds to t_{as} as defined in Section 3.3.2).

For a departing aircraft, n_{st}^{a} corresponds to the aircraft stand assigned in the flight schedule (as shown in Table 4), while n_{end}^{a} coincides with one of the *TOPs*. For an arriving aircraft, n_{st}^{a} corresponds to one of the *LEPs*, while n_{end}^{a} coincides with the aircraft stand assigned in the flight schedule (as shown in Table 4).

To ensure conflict-free paths, the scheduler checks each path for traffic conflicts and alters the path if a conflict is predicted. For this work, two strategies have been implemented to solve predicted traffic conflicts. With the first strategy, the conflict is resolved by altering the taxi path of the aircraft, while, with the second strategy, the taxiing start time is postponed. Both strategies can resolve any identified conflicts at the cost of delaying the arrival time at the target end point. Furthermore, a combination of the two strategies is also possible and has been implemented as well.

4.4 Tow Truck Allocation

Once conflict-free taxi routes have been identified, the system needs to assign a tow truck to each aircraft. Once again, a conflict-free route is required for the tow trucks. Therefore, the algorithm checks for conflicts along the path of each tow truck and, in the event that a conflict is detected between the tow truck and one or more aircraft, it re-plans the tow truck's path (and re-checks for conflicts) until a conflict-free route is identified and assigned to the tow truck.

4.4.1 Static and Dynamic Allocation of Tow Trucks

The algorithm assigns tow trucks to aircraft using two different types of allocation: *Static Allocation* and *Dynamic Allocation*.

With ***Static Allocation***, in order to be assigned to an aircraft, a tow truck must be parked in a depot and, once its mission is concluded, it must return to the same (or a different) depot. Therefore, in the case of *Static Allocation*:

- In the *ReaM* phase, the initial position of tow truck r, n_{st}^{r}, matches the location of a depot, while its final position, n_{end}^{r}, coincides with the initial position of the assigned aircraft a, n_{st}^{a};

- In the *TowM* phase, the initial position of tow truck r, n_{st}^{r}, is equal to the initial position of the assigned aircraft a, n_{st}^{a}, while its final position, n_{end}^{r}, is equal to the final position of a, n_{end}^{a};
- In the *RetM* phase, the initial position of tow truck r, n_{st}^{r}, is the same as the final position of the assigned aircraft a, n_{end}^{a}, while its final position, n_{end}^{r}, coincides with the location of a depot.

On the other hand, with ***Dynamic Allocation***, a tow truck can be assigned to an aircraft from any location of the airport, and, once the towing operation is concluded, it either returns to a tow truck depot or it is re-assigned to a new mission. Re-assignment can also occur while the tow truck is on its way to a depot.

Therefore, in the case of Dynamic Allocation:

- In the *ReaM*, the initial position of the tow truck r, n_{st}^{r}, of the tow truck could be any location in the airport, while its final position, n_{end}^{r}, coincides with the initial position of the assigned aircraft a, n_{st}^{a};
- In the *TowM*, the initial position of the tow truck r, n_{st}^{r}, is equal to the initial position of the assigned aircraft a, n_{st}^{a}, while its final position, n_{end}^{r}, is equal to the final position of a, n_{end}^{a};
- In the *RetM*, the initial position of the tow truck r, n_{st}^{r}, matches with the final position of the assigned aircraft a, n_{end}^{a}, while its final position, n_{end}^{r}, coincides with the location of a tow truck depot or with the initial position of another aircraft. If it coincides with the location of a depot, the final position of the tow truck could still change dynamically during the *RetM*, matching with the initial position of another aircraft.

4.4.2 Tow Truck Allocation Criteria

In order to decide which tow truck should be assigned to an aircraft, the system first excludes the unavailable tow trucks i.e. tow trucks that are already assigned to another mission or that do not have enough battery charge (i.e. the level of their battery charge is less than b_{min}). If there are no trucks with enough battery charge, the aircraft is ordered to taxi using its own engines (i.e., in a conventional manner). On the other hand, if there is at least a tow truck with sufficient battery charge, the algorithm selects one of them based on the following three criteria (sorted in order of priority):

- Availability of a conflict-free route from the tow truck's location to the aircraft's n_{st} that allows the tow truck to arrive at the aircraft exactly at the t_{as} of the aircraft;
- If the first criterion is satisfied for more than one tow truck, the tow truck with the lowest associated cost (as defined in Section 5.8) is selected;

- If more than one tow truck has the lowest associated cost defined in the second criterion, the tow truck with the least utilisation time is selected to ensure a fair distribution of missions between the tow trucks.

Finally, if the three criteria are satisfied for multiple tow trucks, no further criteria are considered and an arbitrary tow truck is assigned to the aircraft.

Table 9 shows an example of tow truck allocation between 6 tow trucks. The tow trucks which are not available or for which no conflict-free route is available are excluded (Tow Trucks 2, 3 and 6). Among the others, the tow trucks with the lowest associated cost are selected (Tow Trucks 4 and 5). Finally, among these tow trucks, the one with the lowest utilisation time is assigned to the aircraft (Tow Truck 4).

Table 9. Example of tow truck allocation.

Tow Truck	Available	Conflict-Free Route Available	Cost	Utilisation Time [s]
1	TRUE	TRUE	190	-
2	TRUE	FALSE	-	-
3	FALSE	TRUE	-	-
4	TRUE	TRUE	120	850
5	TRUE	TRUE	120	1320
6	FALSE	FALSE	-	-

The possible minimum and maximum levels of tow truck battery charge are defined as follows:

- The **Minimum Level of Battery** (b_{min}) is the minimum charge allowable for a tow truck to complete a towing mission and is equal to:

$$b_{min} = 20\% \tag{23}$$

- The **Maximum Level of Battery** (b_{max}) is the full charged level and is equal to:

$$b_{max} = 100\% \tag{24}$$

In order to be able to be assigned to an aircraft, a tow truck must have the level of its battery charge at least equal to b_{min}.

4.4.3 Time-Wise and Fuel-Wise Approach

In case there are no tow trucks which satisfy the first criterion of Section 4.4.2 (i.e. availability of a conflict-free route for any tow truck), the algorithm uses one of two approaches to proceed to the next step: the *Time-Wise Approach* and the *Fuel-Wise Approach*.

If the ***Time-Wise Approach*** is adopted (i.e. the algorithm prioritises taxi delays over fuel savings), no tow trucks are assigned to the flight and the aircraft is ordered to taxi with its own engines.

Instead, if the ***Fuel-Wise Approach*** is adopted (i.e. the algorithm prioritises fuel savings over taxi delays), the algorithm attempts to postpone the t_{as} of the aircraft (while adhering to the time restriction indicated in Eq. 15) and to recalculate the aircraft schedule in order to check if, in these conditions (a) the aircraft route is still feasible and (b) at least one tow truck is available to satisfy the first criterion. If both conditions are satisfied, the algorithm updates the aircraft schedule and assigns it a tow truck, whereas, if either of the two conditions is not satisfied, the aircraft is ordered to taxi with its own engines.

4.4.4 Depot Allocation Criteria

After the towing phase, the algorithm assigns a destination depot to a tow truck based on three criteria:

- A conflict-free route is available from the final position of the assigned aircraft, n_{end}^{a}, to the depot under consideration; and
- The depot under consideration has at least one available parking slot at t_{dr}; and
- The time needed to return to the depot under consideration is less than the time needed to return to any other depot.

This process is illustrated in Fig. 24.

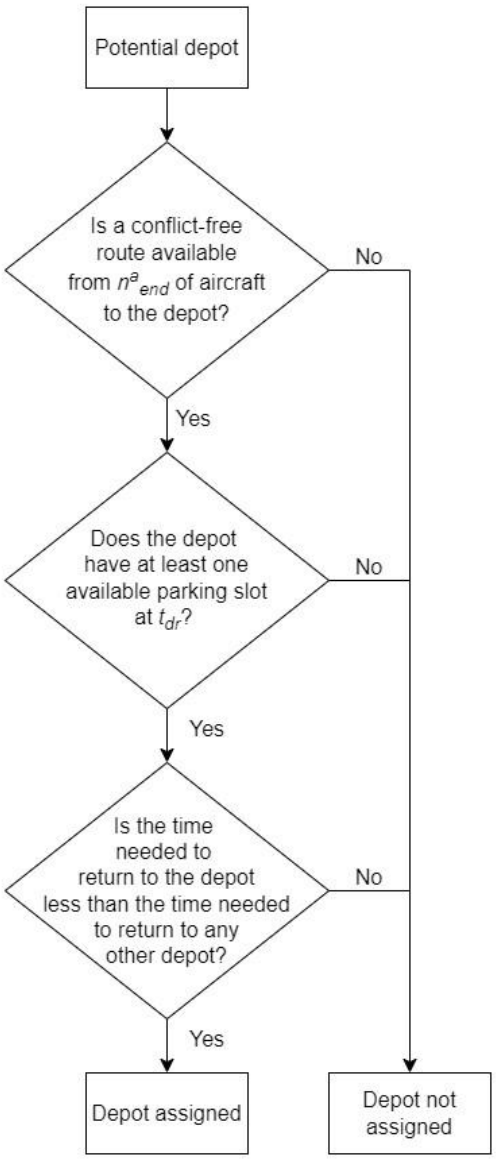


Fig. 24. Flowchart of the depot allocation process.

If there are no depots which satisfy the first two criteria, no tow truck is assigned to the flight and the aircraft is ordered to taxi with its own engines (and any potential update of the aircraft schedule with the *Fuel-Wise Approach* is cancelled).

This method of depot allocation is used both in the case of *Static Allocation* and *Dynamic Allocation*. However, in the case of the latter, during its *RetM*, the tow truck is already available for a new mission; therefore, its route to the depot might be subsequently modified to head towards another aircraft.

Table 10 shows an example of depot allocation between 4 depots. The depots for which no conflict-free routes are available, or which do not have at least one empty parking slot are excluded (Depots 1 and 3); of the remaining depots, the depot which can be reached first is assigned to the tow truck (Depot 2).

Table 10. Example of depot allocation.

Depot	Conflict-Free Route Available	Parking Slots	Time To Reach The Depot [s]
1	FALSE	2	-
2	TRUE	1	50
3	TRUE	0	-
4	TRUE	3	75

4.5 Vehicle Motion Model

In this work, no acceleration or deceleration is modelled for the vehicle's motion and it is assumed that the vehicles (both aircraft and tow trucks) are either idle (i.e. with a velocity of 0 m/s) or move at a constant speed, defined as **Average Vehicle Velocity** (v_{av}) and equal to 10 m/s (19.4 Knots). This value was selected as it is in the range of typical aircraft taxi speeds [96].

A vehicle is idle before the start of a mission, after the mission is over, during the attachment phase, during the detachment phase, or while waiting for the assigned tow truck/aircraft. In all other cases, a vehicle moves at v_{av}, as it is not allowed to stop along its route. This constraint was introduced in order to limit the solution space available for the algorithm and, in case the vehicle is an aircraft which is taxiing using its own engines, also to improve its fuel savings, as start-stop manoeuvres have an impact on fuel consumption during taxi [3, 9, 78].

It is also assumed in this work that a vehicle is always aligned with the heading of the edge it moves along and that its heading changes instantly when it transitions from one edge to another.

Finally, it is assumed that an aircraft cannot backtrack; therefore, when a departing aircraft reaches its *ToP* or an arriving aircraft starts taxiing from its *LEP*, it always has to move along the runway direction.

4.6 Buffer Time

A constant interval of time, defined as **Buffer Time** (Δt_b), determines how long a position (e.g. an edge or a node) should not be occupied by other vehicles before or after a vehicle passes through that position (unless all the vehicles are tow trucks, as explained in Section 3.6). Moreover, when a departing aircraft enters the runway, or when an arriving aircraft is set to land, the entire runway is restricted from use by other vehicles for a period of time equal to Δt_b.

Δt_b is calculated as:

$$\Delta t_b = \frac{d_b}{v_{av}} = 10\ s \tag{25}$$

Since the vehicles proceed at a constant speed along their route, it is possible to calculate the times when a vehicle occupies an edge. In particular, set of times when a vehicle begins to occupy an edge, $P_{times.min.b}$, is defined as follows:

$$P_{times.min.b} = P_{times.min} - \Delta t_b \tag{26}$$

On the other hand, the set of times when a vehicle stops occupying an edge, $P_{times.max.b}$, is defined as follows:

$$P_{times.max.b} = P_{times.max} + \Delta t_b \tag{27}$$

It is assumed that Δt_b not only ensures a safe distance between aircraft along the taxiing route but also guarantees a departing aircraft sufficient time to take off before another departing aircraft approaches the runway, thus avoiding a lineup of aircraft in proximity to the runway.

5 Algorithm Design

While Chapter 4 provided a high-level overview of the algorithm, this chapter explains the design and implementation of the algorithm in detail. In particular, the chapter explains the algorithm's control flow; provides details about the data store of the system and the modules that exchange data between the algorithm and the data store; shows how the system assigns routes and time to start taxiing to each aircraft of the flight schedule; explains the process of assigning routes to vehicles; illustrates the method of detecting conflicts between vehicles; describes the general module to manage the tow trucks in the system; details the method to assign the routes to the tow trucks; shows how the tow trucks are assigned to the aircraft; and illustrates how the statuses of the tow trucks and of the depots are updated over time.

5.1 Control Flow of the Algorithm

The algorithm was implemented in Matlab R2022a [97] and consists of nine modules. It takes input data from and returns output data to a Structured Query Language (SQL) database (called *Data Store*) implemented in SQLite 3.41.2 [98]. A control flow diagram of the algorithm is shown in Fig. 25.

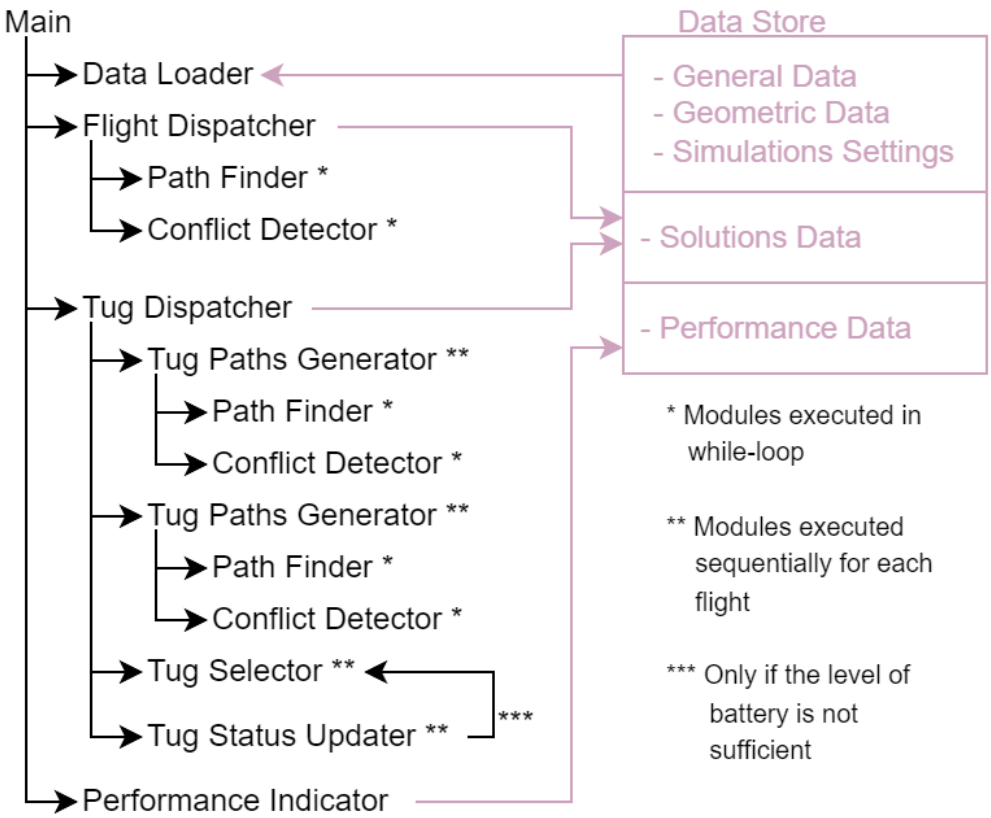


Fig. 25. Control flow of the algorithm.

5.2 Data Store, Data Loader and Performance Indicator

The ***Data Store*** contains the values of all the variables of the algorithm, layout information related to the selected airport, relevant simulation settings and data related to the performance of the algorithm. In particular, three types of data are input to the algorithm: *General Data*, *Geometric Data* and *Simulation Settings*; whilst, two types of data are generated by the algorithm, i.e. *Solutions Data* and *Performance Data*.

General Data includes all the fixed values of the variables defined in Chapters 3 and 4, i.e. path-related variables, time-related variables, fuel-related variables, vehicle motion variables and tow truck battery settings. It also contains other time-related data (such as the length of simulations), relevant aircraft features (e.g. aircraft types and fuel consumption, as detailed in Khadilkar model [78]) and graphical data (e.g. vehicle icons).

Geometric data consists of the nodes and edges of the selected airport graphs (as shown for example in Table 11 and Table 12), parking locations, *ToPs*, *LEPs* and tow truck depot positions (see Section 3.1.5). In order to replicate the geometry of the airports and create an airport model, a satellite image of each airport was imported in AutoCAD 2022 [99] and a series of segments was drawn over it. The segments coincide with the centrelines of the roads and

have different types of lines according to the type of airport road (i.e. runway, taxiway, service road). In addition, relevant locations – such as parking locations, *ToPs*, *LEPs* and tow truck depots – are indicated using different types of points. A custom-built Matlab function reads the AutoCAD files of each airport, imports all of the relevant airport information and creates an SQL file with a database for each airport. Each database contains tables with the nodes and edges of the corresponding airport graph, parking locations, *ToPs*, *LEPs* and tow truck depot positions.

Table 11. Example of the first 10 nodes in the *Data Store* for MLT.

$n_{ID.i}$	x_i	y_i
1	116	1836
2	187	1943
3	252	1633
4	285	1905
5	291	1944
6	299	1988
7	343	1820
8	791	2144
9	801	2304
10	816	2445
…	…	…

Table 12. Example of the first 10 edges in the *Data Store* for MLT. The column labelled 'w_m' contains NULL values (unknown values), as the edge weights are dynamically updated by the algorithm.

$e_{ID.m}$	$n_{ID.j}$	$n_{ID.k}$	$p_{ID.m}$	w_m
1	1	2	3	NULL
2	1	3	3	NULL
3	2	5	3	NULL
4	2	11	3	NULL
5	3	12	3	NULL
6	4	5	2	NULL
7	4	7	2	NULL
8	5	6	2	NULL
9	6	13	2	NULL
10	7	8	1	NULL
…	…	…	…	…

Simulations Settings includes all the data which defines the different scenarios (runways in use, level of traffic, ratio of tow trucks to aircraft, etc.) and a flight schedule associated with each scenario (which is generated as described in Section 3.4). The *Simulation Settings* are described more extensively in Section 6.6.

The ***Data Loader*** extracts variable data from *General Data*; airport layout information (related to the selected airport) from *Geometric Data*; and relevant simulation settings and flight schedules from *Simulation Settings*, thus creating an airport environment for each type of simulation. It then initialises the value of each w_m by setting it equal to the Euclidean distance between n_j and n_k of its corresponding edge e_m. In the subsequent modules, the values of w_m are dynamically modified by the algorithm, as explained in Section 5.4. Then, the algorithm uses the data provided by the *Data Loader* to calculate solutions for each flight. Once this process is completed, the algorithm stores the solutions (including the adjusted schedules, routes, allocation of tow trucks, etc.) in the *Data Store* as *Solutions Data*.

Finally, ***Performance Indicator*** computes various metrics (described in Section 6.4) to give an indication of the performance of the algorithm and stores the results in the *Data Store* as *Performance Data*. This data is generated at the end of each simulation and consists of test results, including results related to the performance metrics.

5.3 Flight Dispatcher

The *Flight Dispatcher* is responsible for identifying conflict-free routes for each aircraft between the landing runway(s) and the allocated stands for arrivals, or between the allocated stands and the departure runway(s) for departures, while minimising Δt_{dtot} for each aircraft. The module calculates a flight's Δt_{dtot} for all the *LEPs* (in case of arrival) or all the *ToPs* (in case of departures), in order to decide which runway the aircraft lands on or departs from.

The pseudocode for this module is shown in Fig. 26.

```
for aircraft = 1, A
        iteration =1
        for runway = 1, number_of_runways
                runway_iteration = 0
                Δt_ds = 0
                runway_node = find node (runway)
                parking_node = find node (parking)
                if flight is arrival
                        n^a_st = runway_node
                        n^a_end = parking_node
                else if flight is departure
                        n^a_st = parking_node
                        n^a_end = runway_node
                end
                run Path Finder
                        (inputs: G_nodes, G_edges, n^a_st, n^a_end)
                        (outputs: d_N.id)
                Δt_it= d_N.id / v_av
                while runway_iteration ≤ runway_iterations_cap
                        iteration = iteration + 1
                        runway_iteration = runway_iteration + 1
                        t_as = t_is + Δt_ds
                        set conflict_free_path as false
                        set ECL as empty
                        while conflict_free_path is false
                                run Path Finder
                                        (inputs: G_nodes, G_edges, n^a_st, n^a_end, ECL)
                                        (outputs: P_nodes, P_st.dist, d_N, path_feasibility)
                                if path_feasibility is false
                                        set Δt_dtot as infinite
                                        break loop
                                else if path_feasibility is true
                                        Δt_at = d_N / v_av
                                        Δt_dt = Δt_at − Δt_it
                                        P_times = (t_as + P_st.dist) / v_av
                                        run Conflict Detector
                                                (inputs: G_edges, ECL, P_nodes, P_times, GOT)
                                                (outputs: ECL, conflict_free_path, VOT)
                                end
                        end
                        Δt_dtot= Δt_ds+ Δt_dt
                        save [P_nodes, P_times, Δt_ds, conflict_free_path, Δt_dtot, VOT] as solution (iteration)
                        if Δt_ds (iteration) ≤ any (Δt_dtot (1, …, iteration))
                                Δt_ds= Δt_ds+ Δt_dsi
                        else
                                break loop
                        end
                end
        end
        if all (Δt_dtot (1, …, iteration) is infinite)
                set simulation_feasibility false
                break loop
        end
        selected_iteration = find iteration with minimum (Δt_dtot (1, …, iteration)))
        selected_solution = solution (selected_iteration)
        GOT = append (VOT (selected_iteration)) to GOT
end
```

Fig. 26. Pseudocode for the *Flight Dispatcher*.

Each aircraft $a \in U$ is analysed sequentially (i.e. according to its arrival or departure time in the flight schedule) and solutions are explored for each *LEP* (in case of an arrival) or each *ToP* (in case of a departure), as follows.

First, as shown in lines 8-14 of the pseudocode in Fig. 26, n_{st}^{a} and n_{end}^{a} are calculated (as explained in Section 4.3). These nodes are input to the *Path Finder* (see Section 5.4) which finds the ideal (i.e. shortest) taxi distance $d_{N.id}$. This distance is then divided by v_{av} to find Δt_{it}. Then, the module attempts to find a conflict-free solution. First, *Path Finder* is executed once again (lines 26-28) and P_{nodes}, $P_{st.dist}$ and d_N are generated; then, the feasibility of the path – indicating whether the module found a feasible path (i.e. a path made up of a set of edges with non-infinite edge weights w_m, as explained in Section 5.4) – is checked.

If the path is not feasible, the solution is discarded; otherwise, P_{times} is calculated (line 35) and P_{nodes} and P_{times} are processed by the *Conflict Detector* (described in Section 5.5). This module checks if the path is conflict-free; produces a **Vehicle Occupation Table** (*VOT*, see Section 5.5), which stores all the time windows during which the edges of the path are occupied by the vehicle; and, if potential conflicts are detected, stores them in the **Edges in Conflict List** (*ECL*, see Section 5.5) (lines 29-39). This process is repeated until a conflict-free path is identified or, as mentioned, until the path is flagged as not feasible. In case a conflict-free path is found, the module calculates and stores Δt_{dtot} of the current iteration (lines 41-42).

Then, Δt_{ds} is incremented by a time interval defined as Δt_{dsi} and equal to 10 s (i.e. with the same duration of a time window) and the process is repeated for a new iteration (lines 19-47). New solutions are calculated until the Δt_{ds} of a new solution is greater than or equal to the Δt_{dtot} of any solution, in which case the search for solutions is stopped for the analysed runway. The whole process (lines 3-49) is repeated for the next runway until all the *LEPs* (in case of an arrival) or all the *ToPs* (in case of a departure) are analysed. In case no conflict-free solutions are found (meaning that all of the solutions are discarded because the corresponding paths are considered to be unfeasible), the whole set of solutions is marked as unfeasible and the algorithm stops the calculations for the selected simulation (lines 50-53).

Finally, as shown in lines 54-56, the solution with the lowest Δt_{dtot} is selected and the *VOT* of the selected solution is appended to the **Global Occupation Table** (*GOT*, see Section 5.5), which represents the combination of all the *VOTs* of the selected solutions of the previously analysed aircraft; therefore, when the first aircraft is analysed, the *GOT* is empty.

5.4 Path Finder

The *Path Finder* computes the shortest path between two nodes using Dijkstra's algorithm [92-93]. Given G_{nodes}, G_{edges}, n_{st}^{a} and n_{end}^{a} as inputs, the module returns P_{nodes}, which represents the shortest path between n_{st}^{a} and n_{end}^{a}, together with the associated $P_{st.dist}$ and d_N.

When the *Path Finder* module is used for the first time in each iteration of the *Flight Dispatcher* (lines 15-17), *ECL* (a list of edges, see Section 5.5) is not input to the module, the length of each edge in G_{edges} is set equal to its geometric length, and P_{nodes}, $P_{st.dist}$ and d_N represent an ideal (shortest) path, where the aircraft can taxi unhindered (as no other vehicles are present along the airport roads).

Instead, if *ECL* is input to the module, all the edges listed in *ECL* are excluded from G_{edges} (i.e. their weight w_m is no longer equal to their geometric length, but is set equal to an infinite value), and Dijkstra's algorithm must find the shortest path between n_{st}^a and n_{end}^a that does not pass through these edges. If excluding these edges results in an unfeasible path, the *Path Finder* does not return a solution and indicates that that path is not feasible.

It is important to note that that each time the Path Finder algorithm seeks a route, it initially excludes the nodes and edges which are part of the active runways. If no feasible solutions are found, the algorithm then includes the elements of the active runways and checks if feasible solutions are available.

5.5 Conflict Detector

The purpose of the *Conflict Detector* is to create a *VOT*; to determine if the path provided by the *Path Finder* is conflict-free; and to store the edges where potential conflicts are detected in the *ECL*.

First, from P_{nodes} and P_{times}, the associated P_{edges}, $P_{times.min.b}$ and $P_{times.max.b}$ are determined (refer to Sections 3.3.1 and 4.6). Then, $P_{times.min.b}$ and $P_{times.max.b}$ are compared with the *TWT* (refer to Section 3.5). For each edge in the path of an aircraft, the *Conflict Detector* identifies all the time windows t_w^i which partially or completely overlap the interval between $P_{times.min.b}$ and $P_{times.max.b}$ values corresponding to the edge. All of the identified pairs (e, t_w) are then stored in the *VOT* associated with the aircraft.

In order to better understand how the *Conflict Detector* works, a simple graph is shown in Fig. 27. This graph shows nodes (n_n), edges (e_e) and a path (marked in red) between n_{st}^a and n_{end}^a. The time when the vehicle passes through each node is shown in brackets. P_{edges} are obtained from P_{nodes}, and $P_{times.min.b}$ and $P_{times.max.b}$ are calculated from P_{times} (as explained in Sections 3.3.1 and 4.6), as shown in Table 13. For each edge e_e, the interval between its $t_{min.e}$ and its $t_{max.e}$ is then compared with the time intervals defined in the *TWT* (Table 6). The overlapping time windows are stored in the *VOT*, as shown in Table 14.

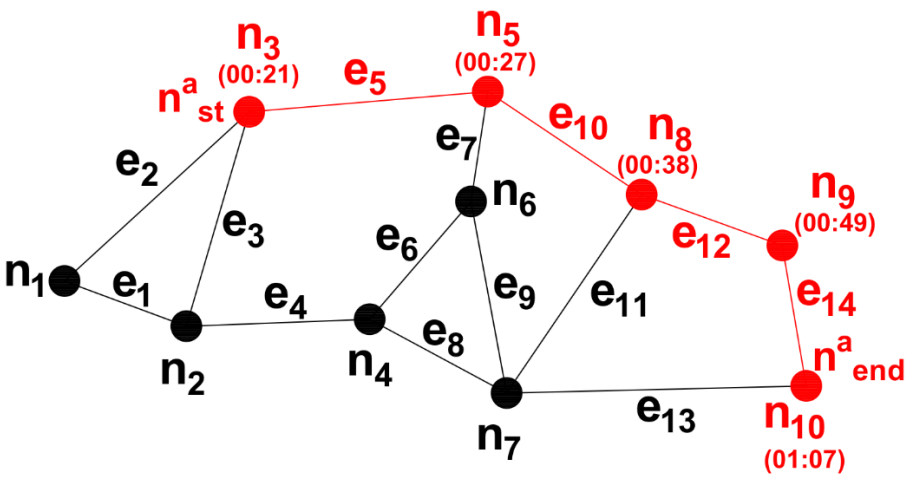


Fig. 27. Example of a path (marked in red) through a directed graph.

Table 13. Path parameters in the *Conflict Detector* module.

P_{nodes}	P_{times}	P_{edges}	$P_{time.min.b}$	$P_{time.max.b}$
3	00:21	5	00:11	00:37
5	00:27	10	00:17	00:48
8	00:38	12	00:28	00:59
9	00:49	14	00:39	01:17
10	01:07	-	-	-

Finally, the *Conflict Detector* compares the *VOT* with the *GOT* (which includes the *VOTs* of all the vehicles analysed previously) and looks for pairs of edges and time windows which appear in both tables. If no matches are found – meaning that the analysed vehicle does not occupy any of the edges of previously analysed vehicles at the same time – the module considers the path to be conflict-free. Otherwise, the path is considered to conflict with other paths, and the edges contained in the matching pairs are added to the *ECL* and, as explained in Section 5.4, they will be excluded from the airport graph by the *Path Finder* module when searching for a new shortest path. For instance, if both the *VOT* and *GOT* contain the pairs (e_{10}, t_w^2), (e_{10}, t_w^3) and (e_{12}, t_w^5), e_{10} and e_{12} will be included in the *ECL*.

Table 14. Example of a *VOT*.

$\mathbf{e_e}$	$\mathbf{t^i_w}$
e_5	t^2_w
e_5	t^3_w
e_5	t^4_w
e_{10}	t^2_w
e_{10}	t^3_w
e_{10}	t^4_w
e_{10}	t^5_w
e_{12}	t^3_w
e_{12}	t^4_w
e_{12}	t^5_w
e_{12}	t^6_w
e_{14}	t^4_w
e_{14}	t^5_w
e_{14}	t^6_w
e_{14}	t^7_w
e_{14}	t^8_w

5.6 Tug Dispatcher

If the *Flight Dispatcher* is able to assign a route to each flight, the *Tug Dispatcher* is activated. This module is responsible for: assigning a tow truck to each flight; finding a conflict-free route for each tow truck between its position and the start node of the aircraft, and between the end node of the aircraft and each tow truck depot; assigning a depot to a tow truck once its towing mission is completed; and updating the status of the assigned tow trucks and of their destination depots.

In order to accomplish this, the *Tug Dispatcher* analyses each aircraft $a \in U$ sequentially and, for each of them, uses the sub-modules *Tug Paths Generator* (to generate conflict-free routes, as explained in Section 5.7); *Tug Selector* (to assign a tow truck to a flight, as explained in Section 5.8); and *Tug Status Updater* (to update the status of the assigned tow trucks and of their destination depot, as explained in Section 5.9).

The objectives of the *Tug Dispatcher* are: to maximise the use of the tow trucks available (thus minimising the number of aircraft which need to taxi using their main engines i.e. in the conventional manner); to generate routes for the tow trucks which do not conflict with any aircraft route calculated by the *Flight Dispatcher*; and to balance the load between the

available tow trucks, allowing them to recharge in the depots when their battery level drops below a predefined threshold. As explained in Section 4.4.1, the *Tug Dispatcher* has two different allocation modes: *Static Allocation* and *Dynamic Allocation*.

5.7 Tug Paths Generator

The aim of the *Tug Paths Generator* is to generate conflict-free paths (each with a t_{dl}) which the tow trucks can follow to reach an aircraft on time (i.e. at the aircraft's t_{as}), or to return to a depot. To accomplish this, the module is executed twice for each flight: first, to search for feasible paths from each tow truck to the attachment node (i.e. n_{st}^{a}) of the analysed flight, and then to search for feasible paths from the detachment node (i.e. n_{end}^{a}) of the analysed flight to the depots. The *Tug Dispatcher* reports to the *Tug Paths Generator* the phase of the mission which has to be analysed (*ReaM* or *RetM*) and the module adapts accordingly.

The operation of the *Tug Paths Generator* is almost identical to that of the *Flight Dispatcher*; however, there are a number of differences which should be taken into account. First, it is assumed that two unloaded tow trucks can pass through an edge (e.g. a taxiway) side by side (as explained in Section 3.6). Therefore, if a conflict-free path is found by the *Tug Paths Generator*, the *VOT* is not added to the *GOT*, unlike what happens in the case of the *Flight Dispatcher*. This allows multiple tow trucks to use (i.e. occupy) the same paths (or parts of them) at the same time.

While the *Flight Dispatcher* attempts to find solutions between the active runway(s) and the designated stand (in case of an arrival), or between the stand and the active runway(s) (in case of a departure), the *Tug Paths Generator* looks for solutions between the position of the tow trucks and n_{st}^{a} (in case of *ReaM*), or between n_{end}^{a} and the depots (in case of *RetM*), as explained in Section 4.4.1. Also, while the *Flight Dispatcher* only selects one solution, the *Tug Paths Generator* computes one solution for each depot (and, in the case of *Dynamic Allocation*, also computes a solution from the position of each available tow truck which is outside the depots), in order to provide a potential conflict-free route for each tow truck to reach the aircraft.

In case of the *ReaM*, P_{times} is calculated in reverse, starting from the last node (equal to n_{st}^{a}) to the first node (equal to the tow truck's initial position). This is done because the tow truck must arrive at the attachment point not later than the predefined t_{ar} (see Section 3.3.3) in order not to disrupt the aircraft's route defined by the *Flight Dispatcher*. In the case of a departure, since the aircraft is waiting at a stand, the *Tug Paths Generator* explores more solutions, such as sending the tow truck to the stand earlier (it can reach the aircraft up to 10 minutes before the predefined t_{ar}) and comparing the Δt_{dtot} of the different iterations (as the *Flight Dispatcher* does). However, in the case of an arrival, this operation is not allowed in order to prevent the tow truck from waiting near a runway, as this might create traffic congestion.

In case of the *RetM*, P_{times} is calculated from the first node (equal to n_{end}^{a}) to the last node (equal to the depot), with t_1 equal to t_{al} (see Section 3.3.3). In the case of an arrival, since the aircraft arrives at a stand, the *Tug Paths Generator* explores more solutions, such as postponing the departure of the tow truck from the stand and comparing the Δt_{dtot} (as the *Flight Dispatcher* does). However, in the case of a departure, this operation is not allowed in order to prevent the tow truck from waiting next to a runway, as this could also lead to traffic congestion. If, for both *ReaM* or *RetM*, no solutions are found for any depot, the algorithm uses one of two approaches to proceed to the next step: the *Time-Wise Approach* or the *Fuel-Wise Approach*, as detailed in Section 4.4.3.

If *Time-Wise Approach* is adopted, the analysed aircraft is not towed – meaning that it will have to taxi using its main engines – and the *Tug Dispatcher* moves onto the next flight in the schedule. For the *Fuel-Wise Approach*, the algorithm adopts another solution (with a higher Δt_{dtot}) calculated in the *Flight Dispatcher* for the analysed flight – updating its P_{nodes}, P_{times} and *VOT*, and thus updating the *GOT* as well – and the *Tug Paths Generator* attempts to find an available path once again. If all of the solutions calculated by the *Flight Dispatcher* are attempted and *Tug Paths Generator* cannot find any available path, the analysed aircraft is not towed (therefore it needs to taxi using its main engines) and the *Tug Dispatcher* moves onto the next flight in the schedule.

5.8 Tug Selector

The *Tug Selector* is responsible for assigning a tow truck to an aircraft and identifying the depot which the tow truck returns to after its taxi mission is completed. In order to do this, the module analyses a number of tow truck parameters and the availability of each depot. These parameters are stored in two sets of tables: a set consisting of a **Tug Status Table** (*TST*) for each tow truck $r \in V$, which shows the status of the tow truck in each time window (see example in Table 15), and another set consisting of a **Depot Status Table** (*DST*) for each depot $b \in Q$, which shows the status of the depot in each time window (see example in Table 16).

For each time window, the *TST* indicates the depot in which the tow truck is located; whether the tow truck is parked; whether it is attached to an aircraft; its battery level; whether sufficient charge is available to handle a towing mission (the minimum threshold is set as per Eq. 23); and the tow truck's availability.

Table 15. A section of a *TST*.

t^i_w	Depot ID	Parked	Loaded	Battery level	Battery sufficient	Available
…	…	…	…	…	…	…
t^{77}_w	2	TRUE	FALSE	20.3%	TRUE	TRUE
t^{78}_w	-	FALSE	FALSE	20.4%	TRUE	FALSE
t^{79}_w	-	FALSE	TRUE	20.3%	TRUE	FALSE
t^{80}_w	-	FALSE	TRUE	20.1%	TRUE	FALSE
t^{81}_w	-	FALSE	TRUE	19.9%	FALSE	FALSE
…	…	…	…	…	…	…

Table 16. A section of a *DST*.

t^i_w	Number of available tugs	Number of available parking slots
…	…	…
t^{61}_w	1	5
t^{62}_w	1	4
t^{63}_w	1	4
t^{64}_w	2	4
…	…	…

If the *Tug Dispatcher* uses *Static Allocation* to allocate the tow trucks, a tow truck is only considered to be available if it is parked and adequately charged; however, if *Dynamic Allocation* is used, a tow truck is also considered to be available when it is not parked in a depot. In both cases, the tow truck must be unloaded, with sufficient charge and without any mission assigned to it. The *DST* indicates the availability of a depot in each time window in terms of the number of available tow trucks and parking spaces (charging points). The number of parking spaces available per depot is determined using Eq. 2. The *Tug Selector* first excludes the parked tow trucks for which no conflict-free path to the analysed aircraft was found, as well as tow trucks that are unavailable between the time window during which they should leave their position (t_{dl}), as calculated by the *Tug Paths Generator*, and the time window during which they should detach from the aircraft (t_{al}). The list of remaining (non-excluded) tow trucks is defined as the **Candidate Tugs List** (*CTL*). Then, a destination depot is assigned to each of the tow trucks in the *CTL* based on the criteria detailed in Section 4.4.4. Finally, for each tow truck r, a **Towing Cost** (c^r_{tow}) is determined as follows:

$$c^r_{tow} = \Delta t^r_{ream} + \Delta t^r_{towm} + \Delta t^r_{retm} \tag{28}$$

where:

Δt^r_{ream} is the Δt_{ream} of tow truck r,

Δt^r_{towm} is the Δt_{towm} of tow truck r, and
Δt^r_{retm} is the Δt_{retm} of tow truck r.

In addition, for each tow truck r, the associated battery charge, defined as **Battery Level** (b_r), is imported from the *TST*, converted to a **Battery Cost** (c^r_{bat}) and scaled to the magnitude levels of the towing cost as follows:

$$c^r_{bat} = c_{tow.max} - \frac{c_{tow.max}}{(b_{max} - b_{min})} \times (b_r - b_{min}) \tag{29}$$

where:
$c_{tow.max}$ is the maximum c^r_{tow} for all the tow trucks in the *CTL*.

This equation makes it possible to add c^r_{bat} and c^r_{tow} (without restricting the maximum value of c^r_{bat} to 100%) and to assign a low cost to a tow truck with a high battery charge (so as to prioritise tow trucks with higher battery levels). In fact, the equation results in a zero-battery cost for fully charged tow trucks. For partial charges, the cost is linearly increased up to a maximum of $c_{tow.max}$. Fig. 28 shows c^r_{bat} for different levels of b_r and for different values of $c_{tow.max}$.

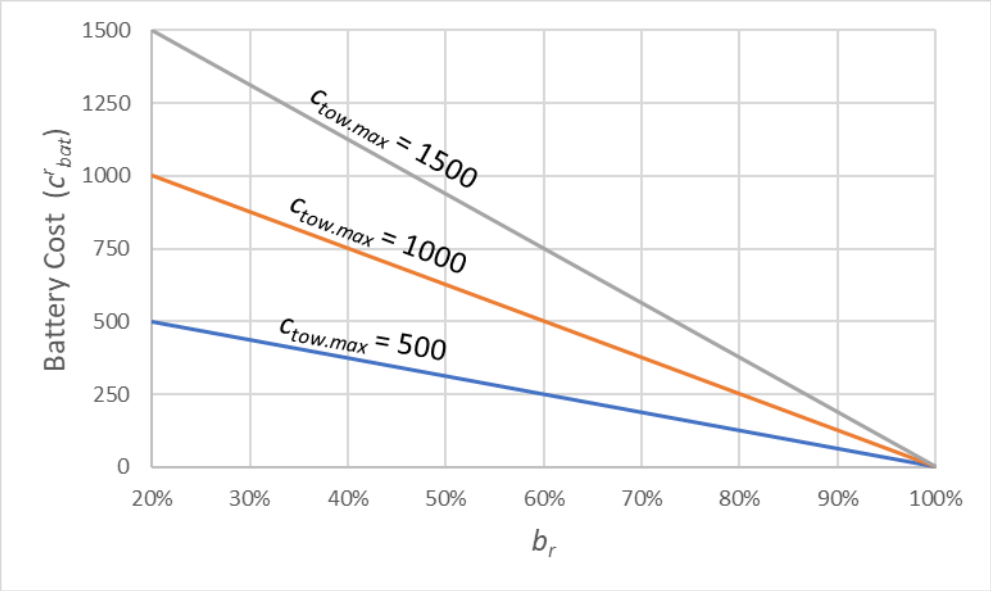


Fig. 28. A plot of c^r_{bat} for different levels of b_r and for three different values of $c_{tow.max}$.

Finally, the **Total Mission Cost** (c^r_{tot}) for each tow truck r in the *CTL*, is computed as follows:

$$c^r_{tot} = c^r_{tow} + c^r_{bat} \tag{30}$$

The tow truck r with the minimum c^r_{tot} is selected and assigned to the aircraft under consideration. If more than one tow truck has the same c^r_{tot}, the algorithm selects the tow truck with the lowest Δt^r_{ru} (which represents the Δt_{ru} of tow truck r). If the utilisation times are all equal, then one of the tow trucks is randomly selected, as explained in Section 4.4.2.

5.9 Tug Status Updater

Once a tow truck is assigned to an aircraft, the *Tug Status Updater* updates its *TST*. This is updated from the time window during which it should leave its position (t_{dl}) to the last time window of the simulation.

The battery levels of all the tow trucks are initially all equal to b_{max}. The rates of battery discharge and recharge are assumed to be constant and are updated on the basis of the following three parameters:

- The **Higher Battery Discharging Rate** (r_{bdh}) is the battery discharge rate applied when the tow truck is in motion and loaded and is set as:

$$r_{bdh} = 2\%/minute \tag{31}$$

- The **Lower Battery Discharging Rate** (r_{bdl}) is the battery charge rate applied when the tow truck is in motion and unloaded and is set as:

$$r_{bdl} = 1\%/minute \tag{32}$$

- The **Battery Charging Rate** (r_{bc}) is the battery charge rate applied when the tow truck is at a charging point in a depot and is set as:

$$r_{bc} = 2\%/minute \tag{33}$$

When a tow truck is not in motion (but not plugged into a charging point), it is assumed that its battery discharge rate is negligible.

The literature presents a diverse range of battery charging and discharging policies. Van Oosterom et al. [100] adjust charging times based on the battery's remaining state-of-charge. Zoutendijket et al. [101-102] developed a model for discharging that takes into account the mass of the tow truck, the mass of the aircraft, and their speed. Meanwhile, Salihu et al. [103] notes that charging and discharging times are influenced by the technologies used for charging and the battery itself. Given that battery modelling was not the primary focus of this work, a model that uses constant discharging and recharging rates with values that are considered reasonable was implemented. This approach was chosen to acknowledge the role of battery status within the system, albeit in a simplified manner. It is assumed that the energy for the batteries comes from sustainable sources such as wind or solar power. Future research may explore more detailed battery models (refer to Section 8.4).

Given these rates, a tow truck is assumed to have an endurance of between 30 and 60 minutes with a full battery and, since its speed is equal to v_{av}, it is therefore able to travel a distance of between 18 and 36 km on a single battery charge. The values of the rates were selected to ensure that, while a tow truck can complete a number of missions throughout a simulation (which is set equal to 1 hour, as detailed later in Section 6.6) it will occasionally need to recharge in a depot. If the updated battery level of a tow truck falls below 0% at any time

(meaning that the tow truck would not be able to complete the mission with its current battery charge), the tow truck is excluded and the *Tug Selector* is run again to search for another available tow truck.

Once the *TST* is updated, the *Tug Status Updater* updates all of the *DSTs* according to the new values of the *TST*. Then, the *Tug Dispatcher* moves onto the next flight in the schedule and repeats the whole process, as described in Sections 5.6 to 5.9.

6 Algorithm Testing

In this chapter, the methodology and approach used to assess the performance of the algorithm are presented. The primary objective of this chapter is to define the conditions to validate the effectiveness of the proposed algorithm and assess its overall performance. A detailed account of the testing process is provided, including the objectives of the evaluations, the test environment, the performance measures adopted and the dataset used for the algorithm's evaluation.

6.1 Objectives

The testing of this work should provide an indication of how effective the solutions calculated by the algorithm are for different airports and for various traffic conditions and how much the introduction of the tow trucks affects taxi operations. First of all, it is critical to ensure that, following the introduction of the tow trucks, the taxi operations still respect the timings dictated by the flight schedule, and therefore one of the main goals of testing is to quantify the resulting average taxi delay. Such delays can be introduced at the start of the taxiing operations (Δt_{ds}), during the taxi route (Δt_{dt}) or as a combination of both (Δt_{dtot}), as defined in Section 3.3.2. In order to understand the significance of aircraft delays in relation to the length of a route, it is also helpful to compare the delays accrued by the aircraft during their taxi route with their ideal (shortest path) taxiing time.

In the case of the of tow trucks, the testing should make it possible to predict the expected utilisation of the tow truck fleet at various airport traffic levels, and to determine how many tow trucks an airport would need to handle the expected levels of ground traffic. Therefore, it is important to quantify the percentage of aircraft that are towed and for how long the aircraft are towed rather than taxiing using their own engines. Additionally, in order to quantify the tow truck usage, the testing should also measure the length of time that the tow trucks are used throughout the simulation for both allocation strategies that were implemented, namely the *Static Allocation* and *Dynamic Allocation*.

The expected utilisation of the tow truck fleet and the length of time that the tow trucks are in use during the simulation can be very important pieces of information in the early design stages of the algorithm where the desired size of the tow truck fleet is still being identified. In this respect, simulations can be carried out at the airport being considered using the expected traffic levels and varying the tow truck fleet size until a target performance level, namely high tow truck utilisation and maximum fuel savings, is reached.

One of the main motivations for using electric tow trucks for taxiing purposes is to reduce fuel consumption and, as a result, fuel emissions. Hence, other crucial objectives of the testing phase include calculating the average fuel consumption of the aircraft; the added fuel consumption of the aircraft when following a route assigned by the algorithm (compared to

the fuel required for an ideal shortest path route); and the aircraft fuel savings brought about by the use of electric tow trucks.

In order to test the *Time-Wise* and *Fuel-Wise* approaches that were implemented in this work, the taxi delays, fuel consumption and fuel savings results obtained with the two approaches will be compared and analysed.

Furthermore, one of the test objectives is to understand how the algorithm behaves (a) when it has perfect knowledge of the system and (b) in the presence of different levels of uncertainty (for instance, in aircraft taxi speed), as explained in Section 6.2.

6.2 Deterministic and Probabilistic Vehicle Models

Various modules of the proposed algorithm need to predict the trajectory of vehicles (aircraft and tow trucks) on the airfield's surface. This prediction allows the system to ensure adequate physical separation between vehicles at all times, which is crucial for safe operation. The position of each vehicle is predicted at timestep intervals Δt_s equal to 1 s and therefore this results in an update frequency of 1 Hz. At each timestep, the algorithm updates the position of all the vehicles and check for any violation of physical separation. Such physical conflicts are detected using the method explained in Section 3.6.

It is worth noting that the trajectory prediction has to assume the value of certain parameters, such as vehicle speed. In practice, these parameters will differ from the predicted value and this will introduce errors in the predicted trajectory which might lead to unpredicted traffic conflicts. In view of this, two different models were adopted during simulation testing: a *Deterministic Model* and a *Probabilistic Model*.

With the ***Deterministic Model***, the simulation assumes no uncertainty in the parameters used for trajectory prediction and uses the values defined in the *Data Store*, *Flight Dispatcher* and *Tug Dispatcher*; in other words, the vehicles behave exactly as modelled by the algorithm. In particular, each vehicle is assumed to be moving at a constant speed v_{av}, each flight starts taxiing at the assigned t_{as} and each tow truck starts its mission at the assigned t_{dl}. The scope of the *Deterministic Model* is to prove that the set of solutions calculated by the algorithm is valid and does not result in any traffic conflicts when executed.

On the other hand, the ***Probabilistic Model*** was created to model uncertainties in vehicle speed and in aircraft start time. These uncertainties can occur, for instance, if the taxi speed of an aircraft or the speed of a tow truck is different from the predicted speed; if an aircraft is delayed at the gate due to late passenger boarding; or if an aircraft lands late due to late departure from the previous airport. For this model, a set of Monte Carlo simulations [104-105] were used to introduce bounded variations to the parameters that directly affect the trajectory of the vehicles. The *Probabilistic Model* challenges the robustness of solutions calculated by the algorithm as it can produce unplanned conflicts.

With the *Probabilistic Model*, the vehicle speed and the time to start taxiing are randomly selected from Normal Probability Distributions (NPD) [106] with pre-defined means and standard deviations. To further test the sensitivity of the system to uncertainty, two magnitudes of standard deviations were defined for each variable, namely a ***Low Standard Deviation*** (LSD) and ***High Standard Deviation*** (HSD).

The Probability Density Function (PDF) [107] of a normal probability distribution is defined as;

$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{34}$$

where:
y is density of probability,
x is the value of the variable being examined,
μ is the mean of the distribution, and
σ is the standard deviation of the distribution.

For simplicity, Eq. 34 is represented as follows:

$$y = f(\mu, \sigma) \tag{35}$$

where:
f is the PDF of the NPD.

In particular, for each aircraft $a \in U$, a random speed v_{ea}^{a} is selected at the beginning of the simulation – and kept constant throughout – from an NPD with the following PDF:

$$y = f\left(v_{av}, \sigma_{v_{aa}}\right) \tag{36}$$

where:
v_{av} is the average aircraft speed, and
$\sigma_{v_{aa}}$ is the standard deviation of the aircraft speed and is set equal to 2 m/s (3.9 knots) in case of LSD and 4 m/s (7.8 knots) in case of HSD.

These two values are set relatively high because the speed of the aircraft is not easy to predict and depends on multiple factors, depending on a series of factors (e.g. pilot behaviour and airline policies). However, the values of v_{ea}^{a} are bounded between 2 m/s (3.9 knots) and 18 m/s (35 knots).

On the other hand, for each tow truck $r \in V$, a random speed v_{er}^{r} is selected at the beginning of the simulation – and kept constant throughout – from an NPD with the following PDF:

$$y = f\left(v_{av}, \sigma_{v_{ar}}\right) \tag{37}$$

where:
v_{av} is the average tow truck speed, and

$\sigma_{v_{ar}}$ is the standard deviation of the tow truck speed and is equal to 1 m/s (1.9 knots) in case of LSD and 2 m/s (3.9 knots) in case of HSD.

These two values are set relatively low, as the tow trucks are assumed to be autonomous (and therefore equipped with a control system) and their velocity is quite easy to predict. The values of v_{er}^{r} are bounded between 2 m/s (3.9 knots) and 18 m/s (35 knots).

Finally, for each aircraft $a \in U$, a time to start taxiing $t_{as.eff}^{a}$ is randomly selected from an ND with the following PDF:

$$y = f\left(t_{as}, \sigma_{t_{as}}\right) \tag{38}$$

where:
t_{as} is the average time to start taxiing, and
$\sigma_{t_{as}}$ is the standard deviation of the mean time to start taxiing and is equal to 10 s in case of LSD and 30 s in case of HSD. These two values are set relatively small in order to be proportionate to the delays generated by the algorithm in the *Deterministic Model*.

The selection of standard deviation values in Eqs. 36-38 aims to capture the stochastic characteristics of the taxiing environment. These values do not have direct equivalents in the literature; instead, they were selected to generate realistic and reasonable variations in velocity and delays.

With the *Probabilistic Model*, when a tow truck is assigned to an aircraft, the two vehicles must wait for each other to be ready to start the attachment operations, leading to an increase in aircraft taxi delays.

In this study, each vehicle is assigned a fixed speed using the *Probabilistic Model*, and this speed remains constant throughout the simulation. In reality, the speed of a vehicle may vary during the taxi phase, thus leading to greater prediction uncertainty. Addressing the variability of vehicle speeds during their missions is recommended for future research, as discussed in Section 8.4.

6.3 Simulator

The *Simulator* module calculates – in real-time – the position and status of the vehicles and tests the solutions provided by the algorithm. This module is run several times, first using the *Deterministic Model* and then using the *Probabilistic Model*, with the values of the parameters v_{ea}^{a}, v_{er}^{r} and $t_{as.eff}^{a}$ varied according to the PDFs as described in Section 6.2. The sub-modules of the *Simulator* are shown in a control flow diagram in Fig. 29.

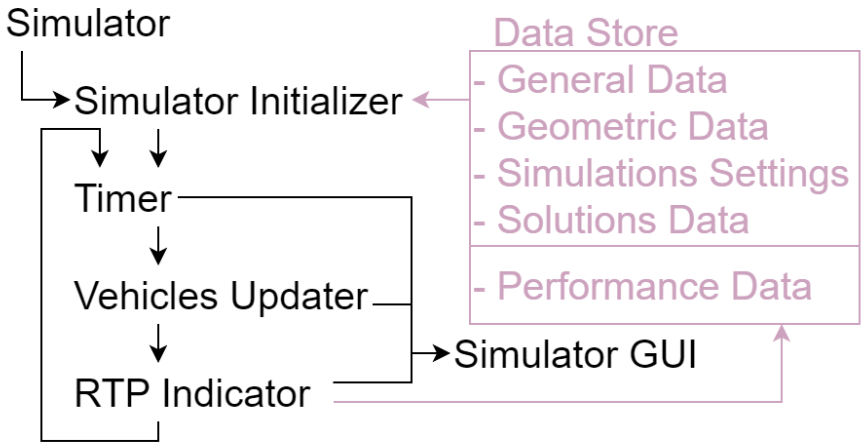


Fig. 29. Control flow diagram of the *Simulator* module.

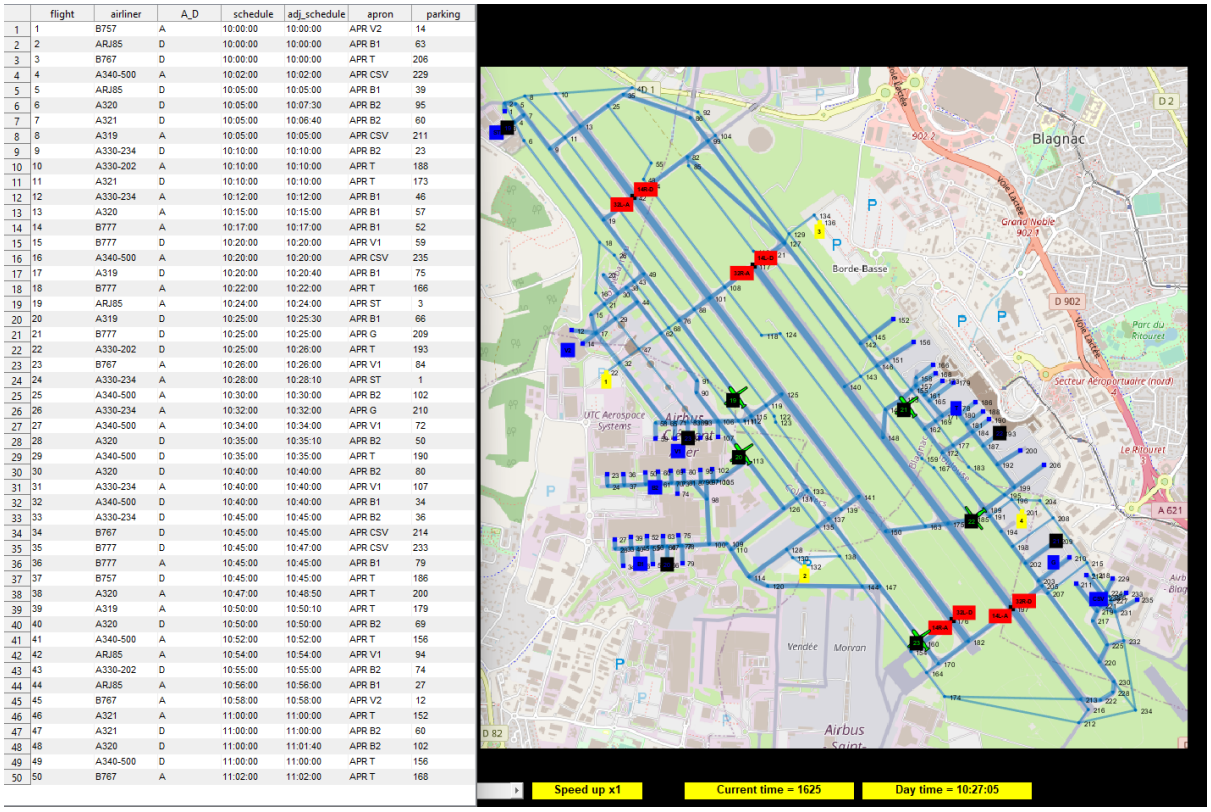
First, the *Simulation Initializer* module takes data (i.e. *General Data*, *Geometric Data*, *Simulations Settings* and *Solutions Data*, as defined in Section 5.2) from the *Data Store*, assigns to the aircraft their P_{nodes} as calculated by the *Flight Dispatcher* (and stored in *Solutions Data*) and to the tow trucks their P_{nodes} for their *ReaM*, *TowM* and *RetM* as calculated by the *Tug Dispatcher* (and stored in *Solutions Data*). Then, if the *Deterministic Model* is used, the module assigns to the aircraft their t_{as} and to the tow trucks their t_{dl}; on the other hand, if the *Probabilistic Model* is used, the module initializes for each aircraft a v_{ea}^{a} and a $t_{as.eff}^{a}$, and for each tow truck a v_{er}^{r}.

Then the *Timer*, *Vehicles Updater* and *Real-Time Performance Indicator* (*RTP Indicator*) modules are executed in a loop until the simulation time is over. The purpose of *Timer* is simply to increment the simulation time by Δt_s at each iteration, whereas *Vehicles Updater* updates the position of the vehicles at the frequency of 1 Hz according to the time assigned to them to start moving, their velocity and their path, according the motion model described in Section 4.5.

The *RTP Indicator* (or *Real-Time Performance Indicator*) calculates a number of performance metrics (*Performance Data*) in real-time and stores them in the *Data Store*. The algorithm creates an area A_b around the geometric centre of each vehicle and detects vehicle conflicts using the method detailed in Section 3.6. The module registers any conflicts and adds them to the **Number of Conflicts** (C), defined as the sum of all conflicts. When a conflict occurs, it is assumed (for the purpose of testing) that the vehicles continue along their path i.e. no collision or deadlock occurs. Also, when the *Probabilistic Model* is used, if a tow truck is not available at the time when it is assigned to start its operations, the assigned aircraft performs its taxi operations using its own engine. In this case, the module registers the event and adds it to the **Number of Missed Missions** (M), defined as the percentage of total tow trucks missions (assigned by the algorithm) which are not successfully completed by the tow trucks. When using the *Probabilistic Model*, the module also computes the **Actual Total Delay**

($\Delta t_{dtot.act}$), which is calculated in the same way as Δt_{dtot} when using the *Deterministic Model* (as detailed in Section 3.3.2).

The *Simulator GUI* module receives data from *Timer*, *Vehicles Updater* and *RTP Indicator* and displays it on a Graphical User Interface (GUI), which shows the position and heading of each vehicle and highlights conflicts between vehicles over time. This module is an optional feature of the *Simulator* and as such is not required during operation and can be turned off. Fig. 30 shows an image of the GUI.



	flight	airliner	A_D	schedule	adj_schedule	apron	parking
1	1	B757	A	10:00:00	10:00:00	APR V2	14
2	2	ARJ85	D	10:00:00	10:00:00	APR B1	63
3	3	B767	D	10:00:00	10:00:00	APR T	206
4	4	A340-500	A	10:02:00	10:02:00	APR CSV	229
5	5	ARJ85	D	10:05:00	10:05:00	APR B1	39
6	6	A320	D	10:05:00	10:07:30	APR B2	95
7	7	A321	D	10:05:00	10:06:40	APR B2	60
8	8	A319	A	10:05:00	10:05:00	APR CSV	211
9	9	A330-234	D	10:10:00	10:10:00	APR B2	23
10	10	A330-202	A	10:10:00	10:10:00	APR T	188
11	11	A321	D	10:10:00	10:10:00	APR T	173
12	12	A330-234	A	10:12:00	10:12:00	APR B1	46
13	13	A320	A	10:15:00	10:15:00	APR B1	57
14	14	B777	A	10:17:00	10:17:00	APR B1	52
15	15	B777	D	10:20:00	10:20:00	APR V1	59
16	16	A340-500	A	10:20:00	10:20:00	APR CSV	235
17	17	A319	D	10:20:00	10:20:40	APR B1	75
18	18	B777	A	10:22:00	10:22:00	APR T	166
19	19	ARJ85	A	10:24:00	10:24:00	APR ST	3
20	20	A319	D	10:25:00	10:25:30	APR B1	66
21	21	B777	D	10:25:00	10:25:00	APR G	209
22	22	A330-202	D	10:25:00	10:26:00	APR T	193
23	23	B767	A	10:26:00	10:26:00	APR V1	84
24	24	A330-234	A	10:28:00	10:28:10	APR ST	1
25	25	A340-500	A	10:30:00	10:30:00	APR B2	102
26	26	A330-234	A	10:32:00	10:32:00	APR G	210
27	27	A340-500	A	10:34:00	10:34:00	APR V1	72
28	28	A320	D	10:35:00	10:35:10	APR B2	50
29	29	A340-500	D	10:35:00	10:35:00	APR T	190
30	30	A320	D	10:40:00	10:40:00	APR B2	80
31	31	A330-234	A	10:40:00	10:40:00	APR V1	107
32	32	A340-500	D	10:40:00	10:40:00	APR B1	34
33	33	A330-234	D	10:45:00	10:45:00	APR B2	36
34	34	B767	D	10:45:00	10:45:00	APR CSV	214
35	35	B777	D	10:45:00	10:47:00	APR CSV	233
36	36	B777	A	10:45:00	10:45:00	APR B1	79
37	37	B757	D	10:45:00	10:45:00	APR T	186
38	38	A320	A	10:47:00	10:48:50	APR T	200
39	39	A319	A	10:50:00	10:50:10	APR T	179
40	40	A320	D	10:50:00	10:50:00	APR B2	69
41	41	A340-500	A	10:52:00	10:52:00	APR T	156
42	42	ARJ85	A	10:54:00	10:54:00	APR V1	94
43	43	A330-202	D	10:55:00	10:55:00	APR B2	74
44	44	ARJ85	A	10:56:00	10:56:00	APR B1	27
45	45	B767	A	10:58:00	10:58:00	APR V2	12
46	46	A321	A	11:00:00	11:00:00	APR T	152
47	47	A321	D	11:00:00	11:00:00	APR B2	60
48	48	A320	D	11:00:00	11:01:40	APR B2	102
49	49	A340-500	D	11:00:00	11:00:00	APR T	156
50	50	B767	A	11:02:00	11:02:00	APR T	168

Fig. 30. Example of GUI for Toulouse–Blagnac Airport. The flight schedule of the simulation is represented on the left, while the position of the vehicles on the airfield is shown on the right.

6.4 Performance Metrics

This section defines the metrics which were defined to characterise the performance of the algorithm. The selection and the definition of the performance metrics was carried out by considering the test objectives defined in Section 6.1.

6.4.1 Performance Metrics Related to Taxi Delays

The performance metrics related to the added delays are defined as follows:

- **Average Start Delay** (s), Δt_{ds}^{avg}: This is the average delay accumulated by all the aircraft while waiting next to the runway (for arrivals) or at the stand (for departures) and is given by:

$$\Delta t_{ds}^{avg} = \frac{\sum_{a=1}^{A} \Delta t_{ds}^{a}}{A} \tag{39}$$

where A is the total number of aircraft and Δt_{ds}^{a} is the start delay Δt_{ds} for aircraft a.

- **Average Taxi Delay** (s), Δt_{dt}^{avg}: This is the average delay accumulated by all the aircraft while taxiing (both when they are towed and when they taxi on their own) and is given by:

$$\Delta t_{dt}^{avg} = \frac{\sum_{a=1}^{A} \Delta t_{dt}^{a}}{\mathrm{A}} \tag{40}$$

where Δt_{dt}^{a} is Δt_{dt} for aircraft a.

- **Average Taxi Delay** (%), Δt_{dt}^{avg}: This is the average delay accumulated by all the aircraft while taxiing (both when they are towed and when they taxi on their own) compared to their ideal taxi time and is given by:

$$\Delta t_{dt}^{avg} = \frac{\sum_{a=1}^{A} \Delta t_{dt}^{a}}{\sum_{a=1}^{A} \Delta t_{it}^{a} \times A} \times 100 \tag{41}$$

where Δt_{it}^{a} is Δt_{it} for aircraft a.

- **Average Total Delay** (s), Δt_{dtot}^{avg}: This is the average total delay (sum of start delay and taxi delay) accumulated by all the aircraft (both when they are towed and when they taxi on their own) and is given by:

$$\Delta t_{dtot}^{avg} = \frac{\sum_{a=1}^{A} \Delta t_{dtot}^{a}}{\mathrm{A}} \tag{42}$$

where Δt_{dtot}^{a} is Δt_{dtot} for aircraft a.

- **Average Total Delay** (%), Δt_{dtot}^{avg}: This is the average total delay (sum of start delay and taxi delay) accumulated by all the aircraft (both when they are towed and when they taxi on their own) compared to their ideal taxi time and is given by:

$$\Delta t_{dtot}^{avg} = \frac{\sum_{a=1}^{A} \Delta t_{dtot}^{a}}{\sum_{a=1}^{A} \Delta t_{it}^{a} \times A} \times 100 \tag{43}$$

- **Delayed Aircraft** (%), DA: This is the percentage of aircraft whose start time is delayed (i.e. $\Delta t_{ds} > 0$) and is given by:

$$DA = \frac{\sum_{a=1}^{A} L_{del}^{a}}{A} \times 100 \tag{44}$$

where L_{del}^{a} is equal to 1 if $\Delta t_{ds}^{a} > 0$, and 0 otherwise.

6.4.2 Performance Metrics Related to Tow Truck Utilisation

The following metrics reflect the ability of the algorithm to maximise the number of aircraft towing operations and to use the available tow trucks. These are defined as follows:

- **Towed Aircraft** (%), TA: This is the percentage of towed aircraft and is given by:

$$TA = \frac{\sum_{a=1}^{A} L_{tow}^{a}}{A} \times 100 \tag{45}$$

  where L_{tow}^{a} is equal to 1 if aircraft a is towed, and 0 otherwise.

- **Towing Time** (%), Δt_{tow}: This is the percentage of total aircraft taxiing time during which the aircraft are towed and is given by:

$$\Delta t_{tow} = \frac{\sum_{a=1}^{A} (\Delta t_{at}^{a} \times L_{tow}^{a})}{\sum_{a=1}^{A} \Delta t_{at}^{a}} \times 100 \tag{46}$$

  where Δt_{at}^{a} is Δt_{at} for aircraft a.

- **Average Tow Trucks Utilisation Time** (%), Δt_{ru}^{avg}: This is the average time that each tow truck is used expressed as a percentage of the total simulation time and is given by:

$$\Delta t_{ru}^{avg} = \frac{\sum_{r=1}^{R} \Delta t_{ru}^{r}}{R \times S} \times 100 \tag{47}$$

  where Δt_{ru}^{r} is the Δt_{ru} for the tow truck r and S is the simulation time.

6.4.3 Performance Metrics Related to Fuel Consumption

As one of the main reasons for using tow trucks for taxiing purposes is to reduce fuel consumption and, as a result, fuel emissions, the following metrics were defined to measure the performance of the algorithm:

- **Average Extra Fuel Consumption** (kg), ΔF_c^{avg}: This is the average extra fuel consumption (as defined in Section 3.3.4) of all aircraft when taxiing with their own engines and is given by:

$$\Delta F_c^{avg} = \frac{\sum_{a=1}^{A} \Delta F_c^{a}}{A} \tag{48}$$

  where ΔF_c^{a} is the ΔF_c for aircraft a.

- **Average Extra Fuel Consumption** (%), ΔF_c^{avg}: This is the average extra fuel consumption of all aircraft when taxiing with their own engines expressed as a percentage of minimum fuel consumption (as defined in Section 3.3.4) and is given by:

$$\Delta F_c^{avg} = \frac{\sum_{a=1}^{A} \Delta F_c^a}{\sum_{a=1}^{A} F_{cmin}^a \times A} \times 100 \quad (49)$$

where F_{cmin}^a is F_{cmin} for aircraft a.

- **Average Fuel Savings** (kg), ΔF_s^{avg}: This is the average fuel saved per aircraft when tow truck-based taxiing is used and is given by:

$$\Delta F_s^{avg} = \frac{\sum_{a=1}^{A} (F_{ca}^a \times L_{tow}^a)}{A} \quad (50)$$

where F_{ca}^a is the F_{ca} for aircraft a (calculated assuming that the aircraft taxis with its own engines).

6.4.4 Performance Metrics Related to Algorithm Robustness

By design, the algorithm generates conflict-free routes for the vehicles. However, this is based on the assumption that the algorithm can perfectly predict the trajectory of each vehicle. The following metrics were defined to assess the sensitivity and robustness of the algorithm to uncertainties introduced by the *Probabilistic Model*:

- **Average Number of Conflicts**, C_{avg}: This is the average number of vehicle conflicts which occur to the aircraft during taxi operations when the system is tested with the *Probabilistic Model* and is given by:

$$C_{avg} = \frac{\sum_{s=1}^{S} C_s}{Z} \quad (51)$$

where C_s is the C at the end of simulation s and Z is the total number of simulations run with the *Probabilistic Model*.

- **Standard Deviation of Number of Conflicts**, C_σ: This is the standard deviation of the number of vehicle conflicts which occur during taxi operations when the system is tested with the *Probabilistic Model* and is given by:

$$C_\sigma = \sqrt{\frac{\sum_{s=1}^{S} \left(C_s - C_{avg}\right)^2}{Z}} \quad (52)$$

- **Average Number of Missed Missions** (%), M_{avg}: This is the average number of missions that the tow trucks fail to complete (i.e. when the tow trucks are unable to reach the aircraft on time) when the *Probabilistic Model* is used, expressed as a percentage of Z:

$$M_{avg} = \frac{\sum_{s=1}^{S} M_s}{Z} \times 100 \quad (53)$$

where M_s is the M at the end of simulation s.

- **Standard Deviation of Number of Missed Missions** (%), M_σ: This is the standard deviation of the number of missions that the tow trucks fail to complete when the system is tested with the *Probabilistic Model* and is given by:

$$M_\sigma = \sqrt{\frac{\sum_{s=1}^{S}\left(M_s - M_{avg}\right)^2}{Z}} \tag{54}$$

- **Average Actual Total Delay** (s), $\Delta t_{dtot.act}^{avg}$: This is the average total delay (sum of start delay and taxi delay) accumulated by an aircraft during taxi operations when the system is tested with the *Probabilistic Model* and is given by:

$$\Delta t_{dtot.act}^{avg} = \frac{\sum_{s=1}^{S} \Delta t_{dtot.act}^{mean.s}}{\mathrm{Z}} \tag{55}$$

where $\Delta t_{dtot.act}^{mean.s}$ is given by:

$$\Delta t_{dtot.act}^{mean.s} = \frac{\sum_{a=1}^{A} \Delta t_{dtot.act}^{a.s}}{\mathrm{A}} \tag{56}$$

where $\Delta t_{dtot.act}^{a.s}$ is the $\Delta t_{dtot.act}$ (detailed in Section 6.3) for aircraft a at the end of simulation s.

- **Standard Deviation of Actual Total Delay** (s), $\Delta t_{dtot.act}^{\sigma}$: This is the standard deviation of the total delay (sum of start delay and taxi delay) accumulated by the aircraft during taxi operations when the system is tested with the *Probabilistic Model* and is given by:

$$\Delta t_{dtot.act}^{\sigma} = \sqrt{\frac{\sum_{s=1}^{S}\left(\Delta t_{dtot.act}^{mean.s} - \Delta t_{dtot.act}^{avg}\right)^2}{Z}} \tag{57}$$

6.5 Airport Selection

An important aspect of the testing phase is to ensure that the algorithm can be tuned and successfully implemented at airports with different sizes and geometries. In reality, even if the algorithm is tested and yields satisfactory results for a large airport with a wide network of taxiways, it cannot be guaranteed that the system will yield comparable results in a small airport with, say, frequent bottlenecks and a high likelihood of conflicts even with low level of traffic. On the other hand, if the algorithm is only tested for small or medium airports, the scalability of the solutions will be unknown. For this reason, the algorithm was tested using four airports with different sizes and geometries as follows: Malta International Airport, Toulouse–Blagnac Airport, Ben Gurion Airport and Dallas/Fort Worth International Airport.

Malta International Airport (MLA), is situated in Luqa, Malta. The airport is relatively small, with one terminal and both runways can be in operation at the same time (one used for

General Aviation (GAv) traffic and the other for commercial traffic) that cross each other. The taxiway network is limited; thus, it might be challenging to avoid bottlenecks when defining taxiing routes. Besides this, MLA, which in July 2022 had 2,206 aircraft movements in one month [108], was selected from among the small airports because it had already been used in previous work [36] and a collaboration with the ATCOs of MATS helped to pinpoint various technical aspects of the taxiing operations at this airport, recommending the location of *ToPs* and *LEPs* and suggesting potential locations for the tow truck depots. A chart of the airport is shown in Fig. 31, while the airport model is displayed in Fig. 17.

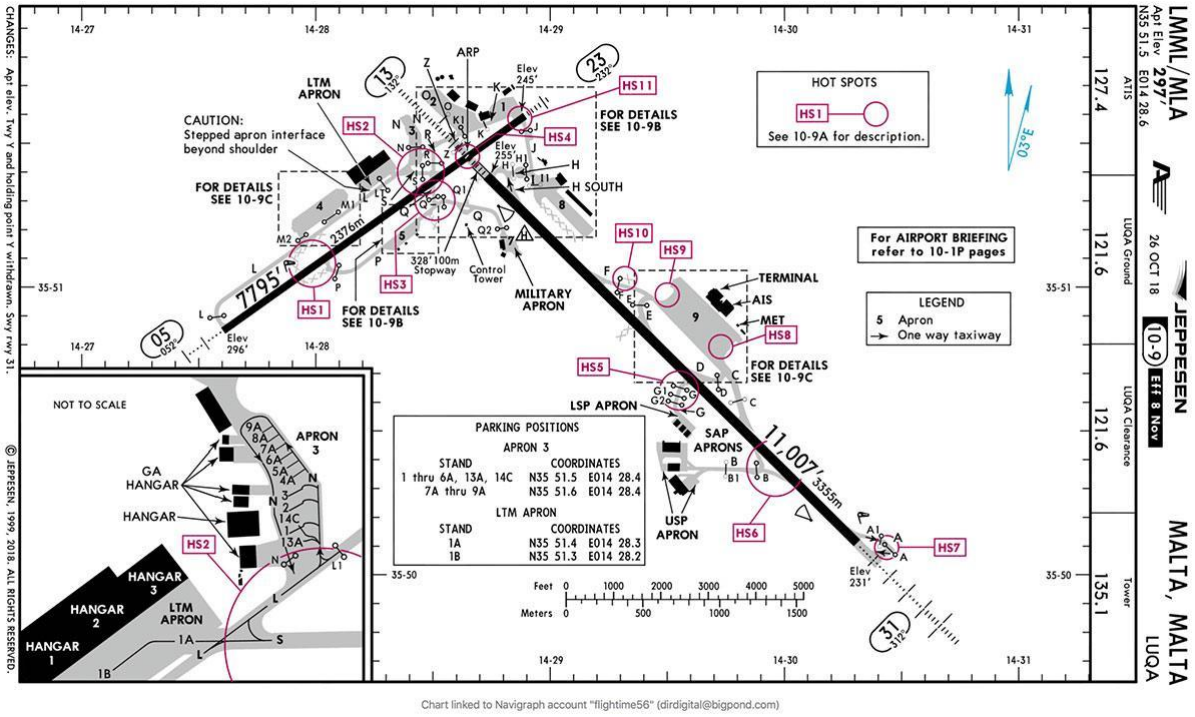


Fig. 31. Chart of Malta International Airport (MLA) [109].

Ben Gurion Airport (TLV) in located in Tel Aviv, Israel. It has medium dimensions with three terminals and three runways (which are used one at a time). In 2019, it had an average of 6,932 aircraft movements per month [110]. TLV was chosen due to the peculiar geometry of the airport's roads, which may represent an interesting case study. In fact, the runways have a triangular pattern and they all cross each other, while the taxiways are mostly located between the runways. Fig. 32 depicts the airport's chart, whereas Fig. 18 shows a model of the airport.

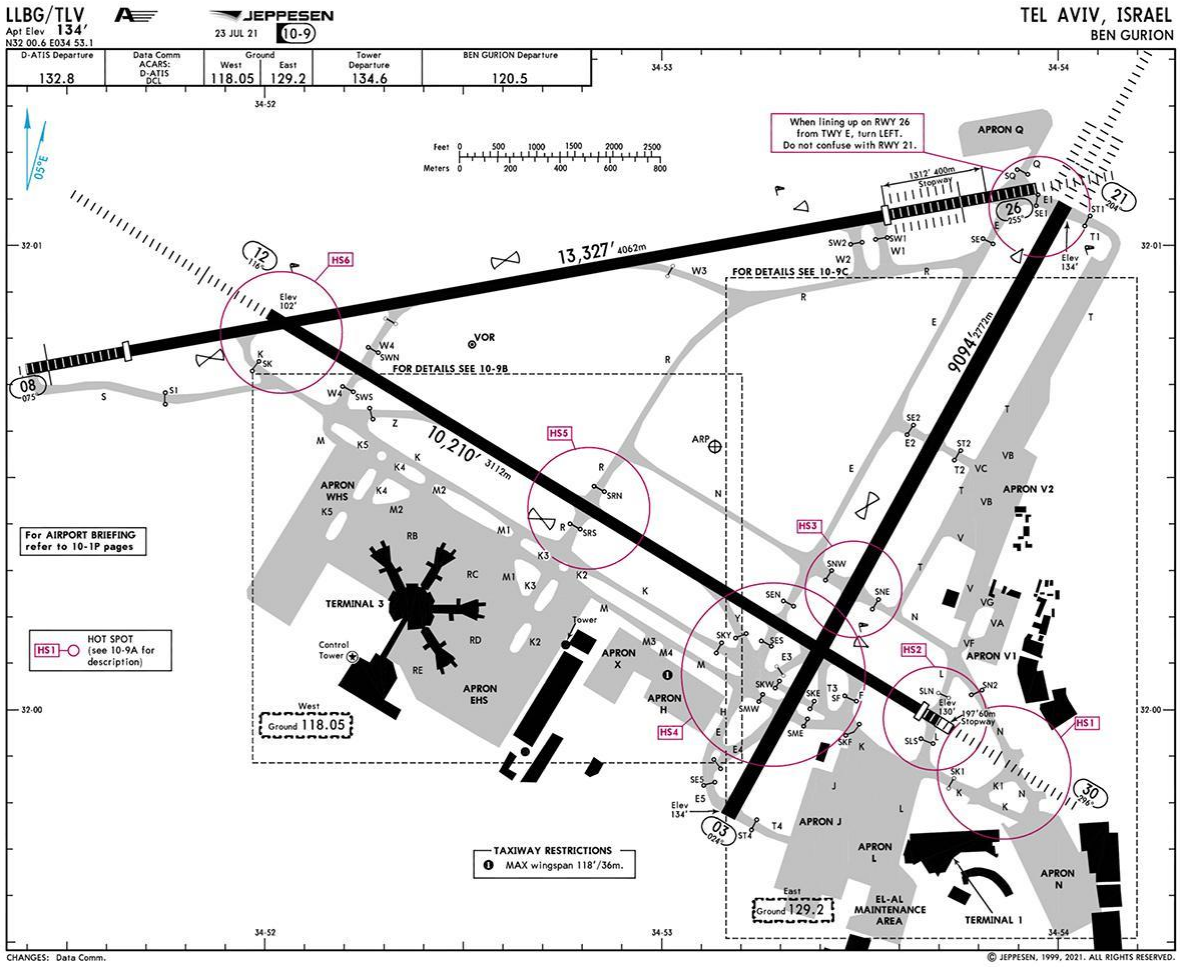


Fig. 32. Chart of Ben Gurion Airport (TLV) [111].

Toulouse–Blagnac Airport (TLS) is located in Toulouse, and partially in Blagnac, in France. The airport has a medium size with one terminal and two concurrently usable parallel runways and in July 2022 it counted 6,990 aircraft movements per month [112]. The airport was chosen for testing as it represents a ‘classical’ airport layout, with parallel runways and various parallel and perpendicular taxiways, which has many alternative taxi routes for the vehicles. A chart of the airport is shown in Fig. 33, while the airport model is displayed in Fig. 19.

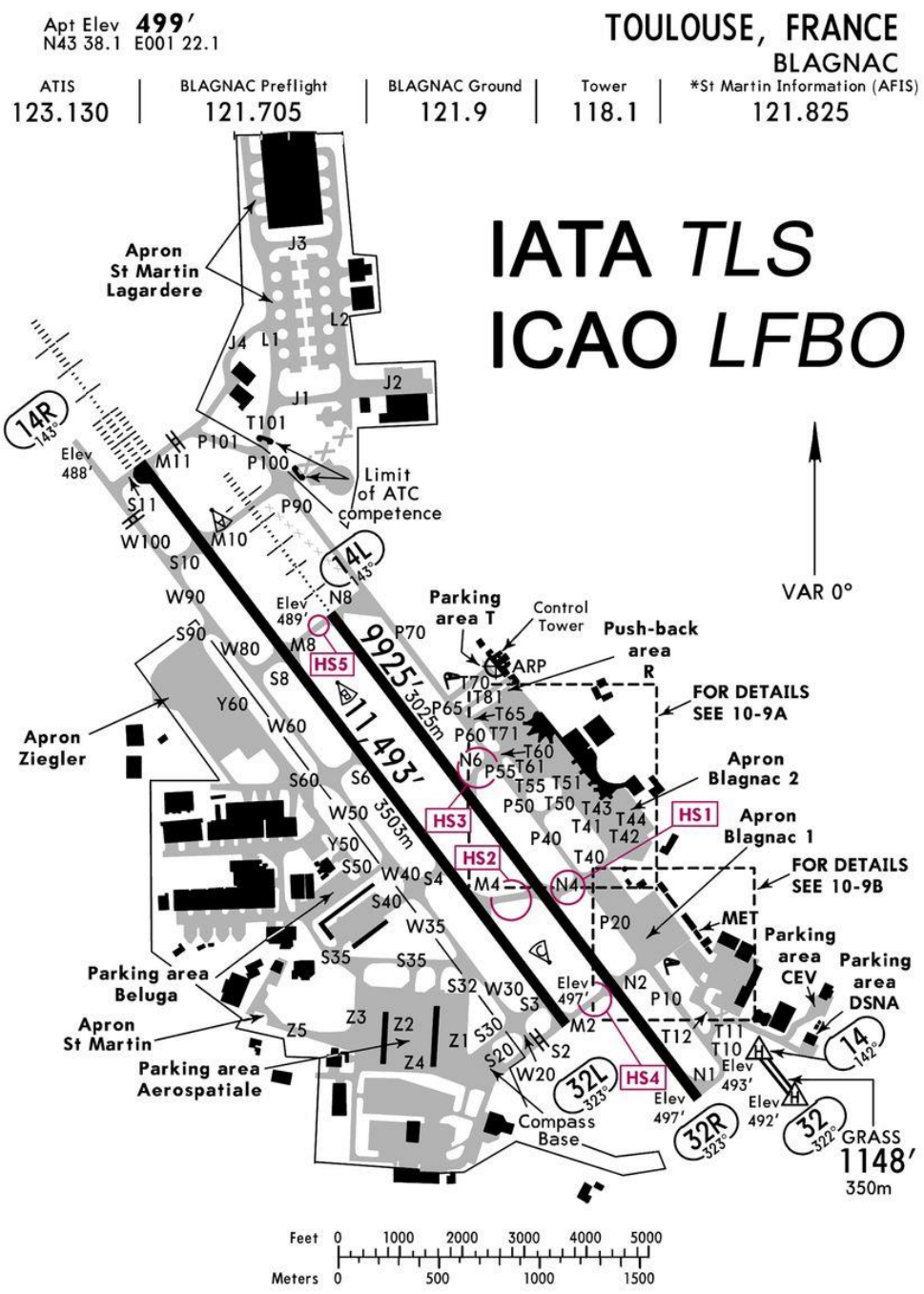


Fig. 33. Chart of Toulouse–Blagnac Airport (TLS) [113].

Dallas/Fort Worth International Airport (DFW) is situated between the cities of Dallas and Fort Worth in Texas, USA. The airport is vast, with five terminals and seven runways. The taxiway network is extensive and it can accommodate a high volume of traffic; in fact, DFW is one of the busiest airports in the world [114], with an average of 55,250 aircraft movements per month in 2022 [115]. For the sake of simplicity, only the central four runways were included in the airport model. These runways are all parallel and can be used at the same time. Fig. 34 depicts the airport's chart and Fig. 20 shows the airport model.

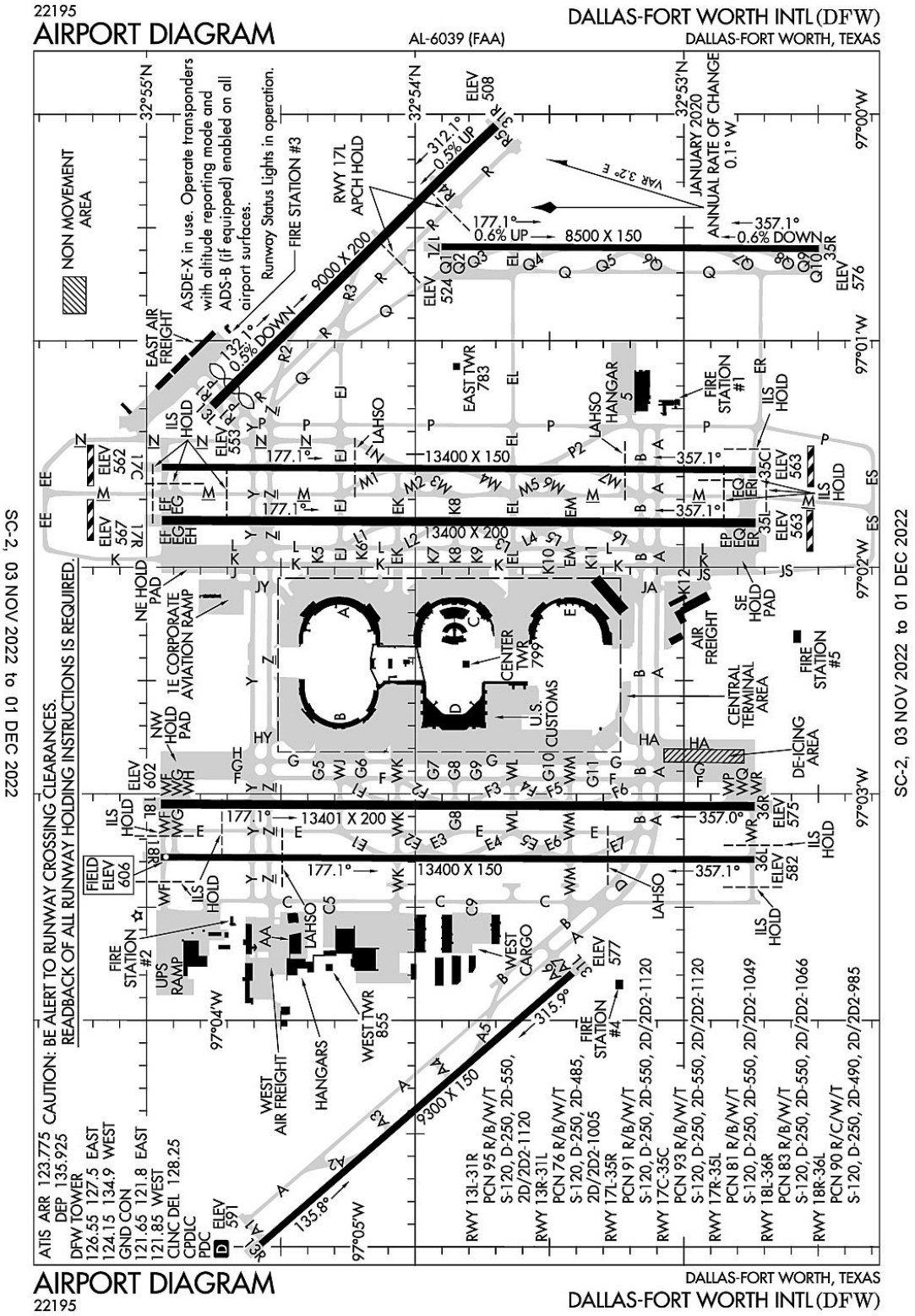


Fig. 34. Chart of Dallas/Fort Worth International Airport (DFW) [116].

6.6 Simulation Settings

A number of simulation settings were defined to allow the algorithm to be tested for different combinations of traffic volume, airport geometry and tow trucks fleet size. The settings were defined as follows:

- **Airport size and geometry**: Four airports with different sizes and geometries were chosen, as explained in Section 6.5;
- **Runways in use**: Tests were carried out for each of the runways in use at each airport. As each runway is bi-directional, the number of actual runways is double the number of physical runways defined at each airport. As in TLS two geometric runways and in DFW four geometric runways can be active at the same time (as detailed in Section 6.5), the simulations for these two airports were defined for two groups of actual runways: one group consisting of two actual runways in the case of TLS, and another consisting of four actual runways in the case of DFW. On the other hand, in MLA (which has 4 actual runways in total) and in TLV (which has 6 actual runways in total), the simulations were defined for every actual runway. Table 17 summarises the runways in use for each airport, including the number of groups of runways that each airport has. Runways are grouped together when they are parallel and share the same direction (they are typically selected based on wind direction and utilised concurrently). The runways belonging to the same group can be used at the same time during a simulation run. Table 18 lists the runways of each airport including their ID, name and ID of the runway group to which they belong. It should be noted that in the real world, the transition between active runways is a delicate operation. It is neither frequent nor abrupt, but rather, it undergoes meticulous planning by ATCOs. In this work, it is assumed that the same runway (or group of runways) remains active throughout all simulations.
- **Level of traffic**: This parameter is defined as the number of aircraft per hour (and thus also referred to as **Aircraft per hour**). The number of aircraft per hour is always even to keep equal numbers of departures and arrivals (as explained in Section 3.4). Once a combination of airport and level of traffic is selected, the algorithm generates a randomized flight schedule, as detailed in Section 3.4;
- **Ratio of tow trucks to aircraft**: This parameter (also referred to simply as **Percentage of tow trucks**) is defined as the ratio of the number of tow trucks available to the number of aircraft per hour. For instance, a ratio of tow trucks to aircraft equal to 0.5 (i.e. 50%) means that there is a tow truck for every two aircraft. The ratio can be adjusted in steps of 0.05 (i.e. 5%). The minimum ratio of tow trucks to aircraft in a given simulation run is 0, which corresponds to a situation where all aircraft taxi with their own engines (i.e. conventional taxiing).

Table 17. Details of runways at each airport.

Airport	Runways (geometric)	Runways (actual)	Actual runways in use per simulation	Groups of runways
MLT	2	4	1	4
TLV	3	6	1	6
TLS	2	4	2	2
DFW	4	8	4	2

Table 18. List of runways in the airports under test.

Airport	Runway ID	Runway name	Runway group ID
MLT	1	05	1
MLT	2	13	2
MLT	3	23	3
MLT	4	31	4
TLV	1	4	1
TLV	2	03	2
TLV	3	08	3
TLV	4	12	4
TLV	5	21	5
TLV	6	26	6
TLS	1	14L	1
TLS	2	14R	1
TLS	3	32L	2
TLS	4	32R	2
DFW	1	17C	1
DFW	2	17R	1
DFW	3	18L	1
DFW	4	18R	1
DFW	5	35C	2
DFW	6	35L	2
DFW	7	36L	2
DFW	8	36R	2

6.7 Test Scenarios

A set of seven scenarios, each of which includes a number of simulations defined by various combinations of simulation settings (detailed in Section 6.6), was used to assess the performance of the algorithm. The *Probabilistic Model* was used in some of these scenarios

together with Monte Carlo simulation. In such scenarios, simulations were repeated 200 times for each combination of simulation settings – 100 times with LSD and 100 times with HSD. In total, 914 simulation runs were conducted using the *Deterministic Model*, while 4,000 simulation runs were carried out using the *Probabilistic Model*, as detailed in this section.

Unless stated otherwise, the duration of each simulation was set equal to 1 hour (resulting in the number of aircraft in each simulation being equal to the number of aircraft movements per hour). This duration was chosen to guarantee a steady flow of tow trucks during testing (i.e. the tow trucks are able to complete various missions during the simulation), while at the same time not overextending the simulations, especially given the total number of simulations carried out.

Moreover, an assumption that was taken is that at the start of the simulation, the tow trucks are distributed equally between the depots of the airport.

The simulations were conducted on a PC equipped with an Intel Core i7 processor and 16 GB of RAM. To provide an indication of computation time, a simulation over a period of 1 hour at DFW, with traffic volumes of 60 aircraft per hour and a percentage of tow trucks equal to 20, typically ranged from 2 to 3 hours.

6.7.1 Test Scenario 1

Test Scenario 1 was designed to test the performance of the algorithm according to the time-related metrics Δt_{ds}^{avg}, Δt_{dt}^{avg}, Δt_{dtot}^{avg} and DA, and to the fuel-related metric ΔF_c^{avg}. Since these metrics are related to the *Flight Dispatcher* module – which is executed before and independently of the allocation of tow trucks – the percentage of tow trucks in *Test Scenario 1* was set equal to zero. While the minimum number of aircraft movements per hour was set equal for all airports, the maximum was set in proportion to the size and level of traffic of each airport [108, 110, 112, 115]. For each airport, simulations were repeated for all the runway groups of each airport (as listed in Table 18) and for levels of traffic ranging from the minimum number of aircraft per hour to the maximum number of aircraft per hour, in increments of 10 aircraft per hour. The parameters of *Test Scenario 1* which were used together with the *Deterministic Model* are shown in Table 19.

Table 19. Airfield parameters used for *Test Scenario 1* with the *Deterministic Model*.

Airport	Groups of runways	Aircraft per hour		Percentage of tow trucks (%)
		Minimum	Maximum	
MLA	4	20	40	0
TLV	6		60	
TLS	2		60	
DFW	2		80	

This scenario was also used to extract the metrics C_{avg} and $\Delta t_{dtot.act}^{avg}$ by using the *Probabilistic Model*. In this case, for each airport only one group of runways was used (the *Runway group ID* 1 of each airport, shown in Table 18) and only one level of traffic (corresponding to the average number of aircraft per hour used in the *Deterministic Model*) was selected. Table 20 shows the parameters of *Test Scenario 1* which were used together with the *Probabilistic Model*.

Table 20. Airfield parameters used for *Test Scenario 1* with the Probabilistic *Model*.

Airport	Groups of runways	Aircraft per hour	Percentage of tow trucks (%)	Standard deviation in use	Number of simulations
MLA	1	30	0	LSD	100
				HSD	100
TLV		40		LSD	100
				HSD	100
TLS		40		LSD	100
				HSD	100
DFW		50		LSD	100
				HSD	100

6.7.2 Test Scenarios 2-5

Test Scenarios 2-5 were designed to test the performance of the algorithm according to the tow truck-related metrics TA, Δt_{tow} and Δt_{ru}^{avg}, and the fuel-related metric ΔF_s^{avg}. For each airport, three different levels of traffic were used (25% more than the minimum level, average level and 25% less than the maximum level) in order to obtain results for low, medium and high levels of traffic, and simulations were repeated for different percentages of tow trucks (between 10% and 50%, in increments of 10%), where the percentage of tow trucks was set relative to the number of aircraft per hour. The difference between these scenarios is the following: *Test Scenario 2* uses *Static Allocation* with *Time-Wise Approach*, *Test Scenario 3* uses *Static Allocation* with *Fuel-Wise Approach*, *Test Scenario 4* uses *Dynamic Allocation* with *Time-Wise Approach*, while *Test Scenario 5* uses *Dynamic Allocation* with *Fuel-Wise Approach*. In *Test Scenario 3* and *Test Scenario 5*, the performance of the algorithm was also tested according to the metric Δt_{dtot}^{avg}, in order to measure the delays produced to maximise fuel savings. The parameters of *Test Scenarios 2-5*, which were used together with the *Deterministic Model* are shown in Table 21.

Table 21. Airfield parameters used for *Test Scenarios 2-5* with the *Deterministic Model*.

Airport	Groups of runways	Aircraft per hour	Percentage of tow trucks (%)	
			Minimum	Maximum
MLA	4	25, 30, 35	10	50
TLV	6	30, 40, 50		
TLS	2	30, 40, 50		
DFW	2	36, 50, 64		

Test Scenarios 2-5 also test the performance of the algorithm according to the metrics C_{avg}, M_{avg} and $\Delta t_{dtot.act}^{avg}$ using the *Probabilistic Model*. In this case, for each airport, only one group of runways was used (the *Runway group ID* 1 of each airport, shown in Table 18) and only the average level of traffic was selected, and the percentage of tow trucks was set equal to 30%. Two different levels standard deviation were assumed, as explained in Section 6.2. Table 22 shows the parameters of *Test Scenarios 2-5* which were used together with the *Probabilistic Model*.

Table 22. Airfield parameters used for *Test Scenarios 2-5* with the *Probabilistic Model*.

Airport	Groups of runways	Aircraft per hour	Percentage of tow trucks (%)	Standard deviation in use	Number of simulations
MLA	1	30	30	LSD	100
				HSD	100
TLV		40		LSD	100
				HSD	100
TLS		40		LSD	100
				HSD	100
DFW		50		LSD	100
				HSD	100

6.7.3 Test Scenario 6

Test Scenario 6 was designed to investigate how the performance of the tow trucks (represented by the metrics TA, Δt_{tow} and Δt_{ru}^{avg}) and the average fuel savings (represented by the metric ΔF_s^{avg}) vary with different battery characteristics. Only one airport (TLS) and only one group of runways were selected for this scenario. The tests were carried out using the *Deterministic Model* for a medium level of traffic (40 aircraft per hour), with a percentage of tow trucks equal to 20%, for a total simulation time of 6 hours. The test scenario used *Dynamic Allocation* with *Time-Wise Approach*. Table 23 shows the parameters of *Test Scenario 6* which were used together with the *Deterministic Model*.

Table 23. Airfield parameters used for *Test Scenario 6*.

Airport	Groups of runways	Aircraft per hour	Percentage of tow trucks (%)
TLS	1	40	20

The scope of *Test Scenario 6* was to assess how variations in tow truck performance relate to changes in their battery performance. For this reason, the two discharge rates defined in Section 5.9, r_{bdh} and r_{bdl}, were varied between -0.75%/min and +1.00%/min in steps of 0.25%/min. Table 24 displays the values of the discharge rates in the test scenario. *Test Scenario 6* was repeated for each discharge rate percentage variation shown in the table.

Table 24. Variation of the discharge rates r_{bdh} and r_{bdl} in *Test Scenario 6*.

Discharge rates percentage variation (%/min)	r_{bdh} (%/min)	r_{bdl} (%/min)
-0.75	1.25	0.25
-0.50	1.50	0.50
-0.25	1.75	0.75
0	2.00	1.00
+0.25	2.25	1.25
+0.50	2.50	1.50
+0.75	2.75	1.75
+1.00	3.00	2.00

6.7.4 Test Scenario 7

Test Scenario 7 mirrors the setup of *Test Scenario 6*, using identical settings and with battery discharge rates equal to the values defined in Section 5.9. The main difference between *Test Scenario 6* and *7* lies in the duration of the simulation: whereas *Test Scenario 6* spans 6 hours, *Test Scenario 7* covers 24 hours. The objective of *Test Scenario 7* is to examine the performance of the algorithm over a whole day of aircraft ground operations and to determine whether or not the results differ substantially from those of the other test scenarios, particularly *Test Scenario 6*. Table 25 displays the values of the discharge rates in the test scenario.

Table 25. Variation of the discharge rates r_{bdh} and r_{bdl} in *Test Scenario 7*.

Discharge rates percentage variation (%/min)	r_{bdh} (%/min)	r_{bdl} (%/min)
0	2.00	1.00

7 Results and Discussion

This chapter provides a comprehensive analysis and discussion of the results obtained from the testing process. The algorithm's performances are examined across seven different scenarios, and the results are presented and discussed. The algorithm was tested according to the metrics detailed in Section 6.4 and for the scenarios defined in Section 6.7.

7.1 Results of Test Scenario 1

The results obtained for *Test Scenario 1* are shown in Fig. 35-Fig. 44. Results for each level of traffic are presented as average values for ease of reporting.

Fig. 35 and Fig. 36 show Δt_{ds}^{avg} (s) and DA (%), respectively, for different levels of traffic in each of the airports under test using the *Deterministic Model*. The figures show similar trends: for each airport, Δt_{ds}^{avg} (s) and DA (%) increase with the level of traffic – first moderately, then sharply. The trend of each airport also depends on the airport's size and geometry. For instance, in the case of MLA, the small dimensions of the airport cause the values to increase significantly for levels of traffic exceeding 30 aircraft per hour. While TLV has a complicated geometry (e.g. many taxiways cross the runways) and only one active runway at a time, TLS features a simple geometry and two active runways; therefore, it has lower average start delays and a smaller percentage of delayed aircraft compared to TLV for similar levels of traffic. Generally, for all the levels of traffic in the four airports, Δt_{ds}^{avg} (s) never exceeds 125 s.

Fig. 37 and Fig. 38 show Δt_{dt}^{avg} (s) and Δt_{dt}^{avg} (%), respectively, for different levels of traffic for each airport under test using the *Deterministic Model*. In all cases, Δt_{dt}^{avg} (s) is relatively low and has a weak association with the volume of traffic. This result was expected since the traffic levels being simulated are within the handling capacity of the airfields and therefore no significant changes to the ideal (shortest) taxi route were required. It is interesting to note that, despite the different dimensions of the airports, the differences in values of Δt_{dt}^{avg} (s) between the airports are very small (generally less than 20 seconds), except for TLV, which has slightly higher values. This might be due to the airport's geometry; for instance, DFW is large, but most of the stands are situated in the middle of the aerodrome. In contrast, MLA is small, but a number of stands are distant from some of the runways. Also, DFW has an extensive taxiway network – which provides the opportunity to find several alternative routes to the ideal one – whereas, in MLA, the number of taxiways is very limited and aircraft might be forced to follow long detours to avoid potential conflicts with other vehicles. Δt_{dt}^{avg} (%) compares the taxi delays with ideal taxi routes, thereby giving an idea of their magnitude compared to the size of the airport (and to the airport's taxiway network). In this case, the impact of the taxi delay is mostly felt in MLA (15% at the peak), due to the airport's short

routes, while the taxi delays have an almost insignificant impact in DFW, increasing the ideal taxi route by a maximum of 5% only.

Fig. 39 and Fig. 40 show Δt_{dtot}^{avg} (s) and Δt_{dtot}^{avg} (%), respectively, for different levels of traffic for each airport under test using the *Deterministic Model*. It can be noted that, for each airport, there is a gradual increase of Δt_{dtot}^{avg}, that eventually climbs dramatically as traffic volume rises. Similar to the trend for Fig. 35 for Δt_{ds}^{avg} (s) (which is the most relevant component of Δt_{dtot}^{avg} (s)), the trend of each airport likewise depends on the size and geometry of the airport. Due to MLA's confined size, values significantly increase when traffic volume exceeds 30 aircraft per hour. TLS has two active runways and a simple geometry in contrast to TLV, which only has one active runway at a time and complex geometry. This results in shorter average delays in the case of TLS and a smaller percentage of delayed aircraft for the same volume of traffic. The values of Δt_{dtot}^{avg} (%) are displayed in percentage form and are compared to the ideal taxi route, demonstrating an increase in the impact of delays for MLA while, in the case of DFW, Δt_{dtot}^{avg} (%) is negligible up to 50 aircraft an hour, and then starts increasing up to a maximum of 20%.

Fig. 41 and Fig. 42 show ΔF_c^{avg} (kg) and ΔF_c^{avg} (%), respectively, for different levels of traffic for each airport under test using the *Deterministic Model*. ΔF_c^{avg} (kg) is the amount of fuel used when compared to an ideal scenario in which the aircraft can taxi undisturbed. As the fuel consumption is not only dependent on the length of the route, but also on the geometry of the airport (as explained in Section 3.3.4), the trend of ΔF_c^{avg} (kg) differs from, for example, Δt_{dtot}^{avg} (s) in Fig. 39. In fact, the trends of ΔF_c^{avg} (kg) for all the airports are comparable, with the exception of TLV, which has larger values of average extra fuel consumption, likely due to its complex shape. The tendency for TLV, however, is more restrained when the fuel consumption is expressed as ΔF_c^{avg} (%), whereas MLA has the largest percentage of average extra fuel consumption. This is because equivalent fuel consumptions have larger effects on shorter routes and, as a result, on smaller airports.

Finally, Fig. 43 and Fig. 44 show C_{avg} and $\Delta t_{dtot.act}^{avg}$ (s), respectively, for different levels of traffic for each airport under test using the *Probabilistic Model* with two different standard deviations (LSD and HSD). Unsurprisingly, the trend of C_{avg} with LSD rises steadily with increasing levels of traffic, while, with HSD, C_{avg} increases more abruptly for high volumes of traffic. It is interesting to note that C_{avg} computed for DFW with HSD is rather high (but only when the level of traffic exceeds 60 aircraft per hour), suggesting that significant margins of uncertainty could cause issues with traffic management for a high volume of aircraft per hour. On the other hand, the trends of $\Delta t_{dtot.act}^{avg}$ (s) in Fig. 44 are relatively similar to what is observed for Δt_{dtot}^{avg} (s) in Fig. 39, with a slight increase in delays when the metric is computed using LSD and a bigger increase when it is computed with HSD.

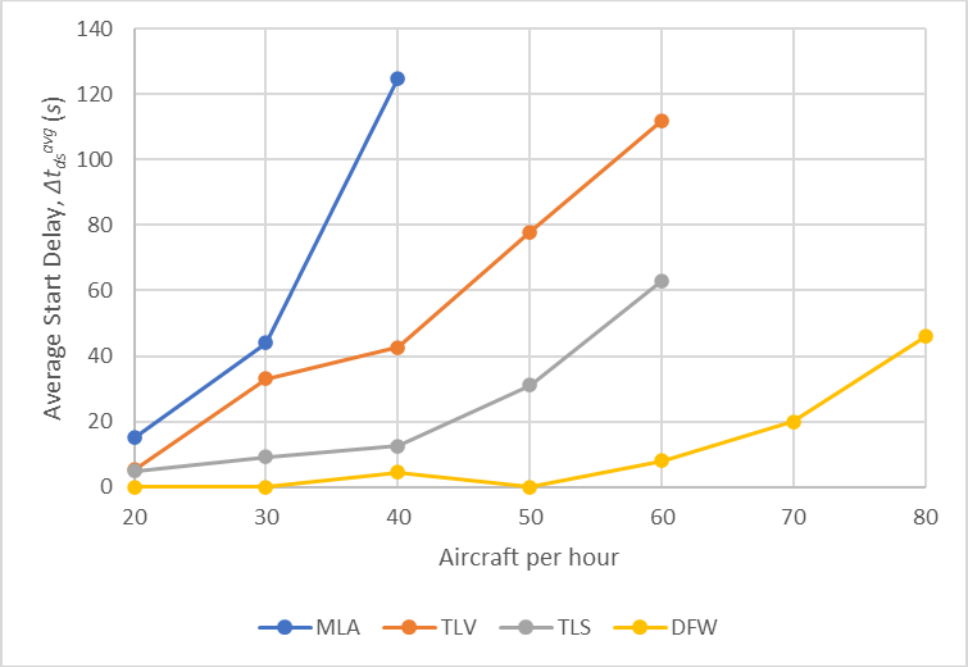


Fig. 35. Average Start Delay (s), Δt_{ds}^{avg} in *Test Scenario 1* using the *Deterministic Model*.

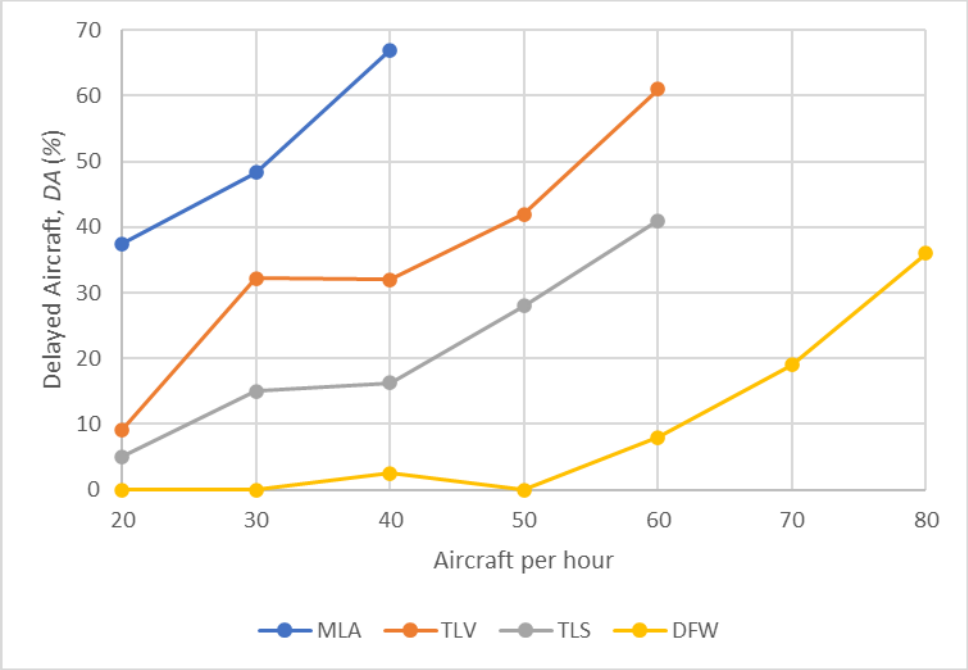


Fig. 36. Delayed Aircraft (%), *DA* in *Test Scenario 1* using the *Deterministic Model*.

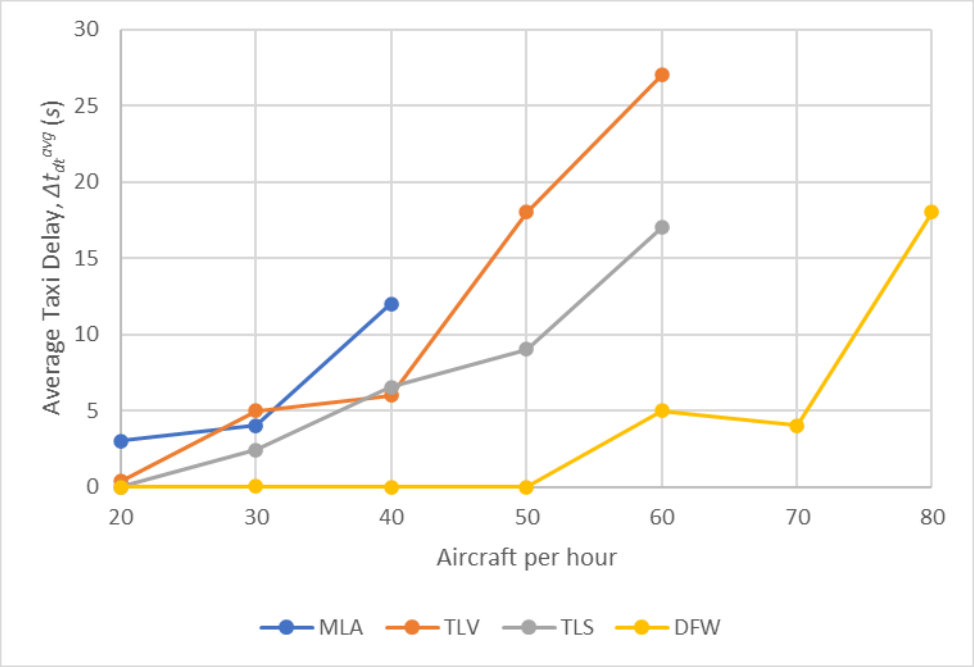


Fig. 37. Average Taxi Delay (s), Δt_{dt}^{avg} in *Test Scenario 1* using the *Deterministic Model*.

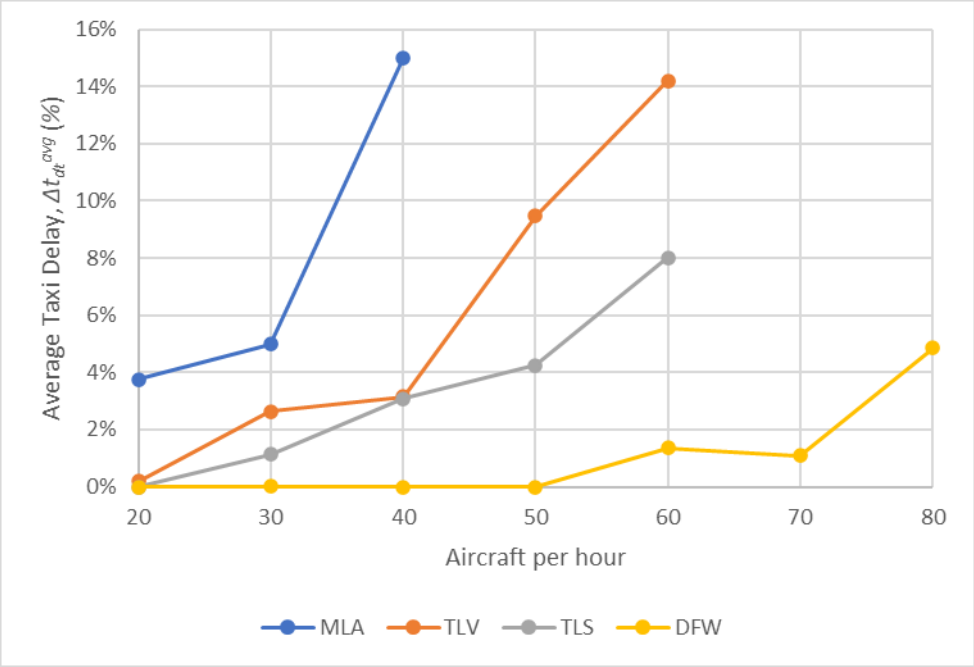


Fig. 38. Average Taxi Delay (%), Δt_{dt}^{avg} in *Test Scenario 1* using the *Deterministic Model*.

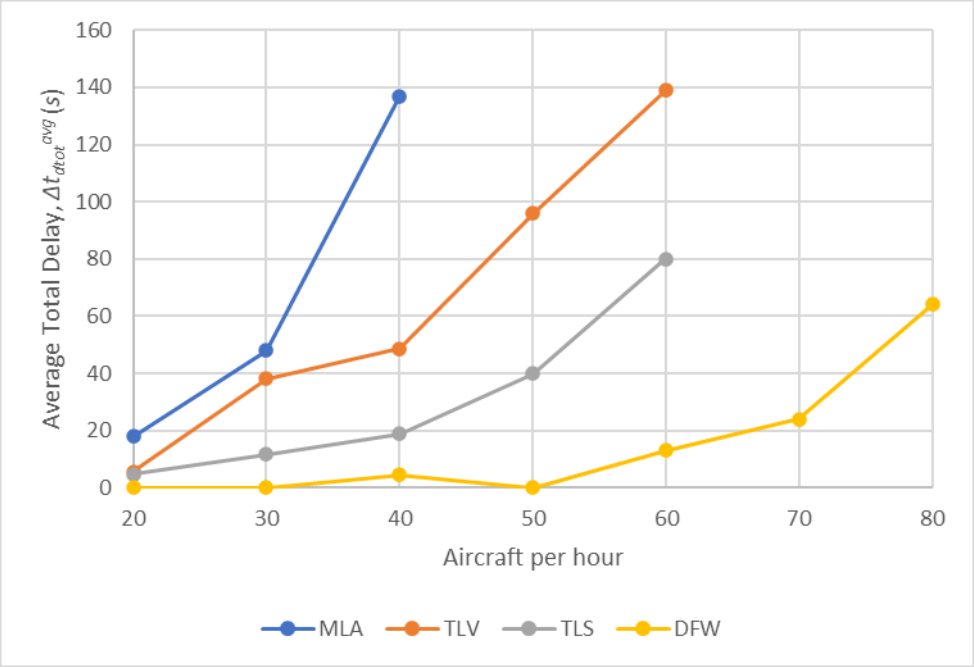


Fig. 39. Average Total Delay (s), Δt_{dtot}^{avg} in *Test Scenario 1* using the *Deterministic Model*.

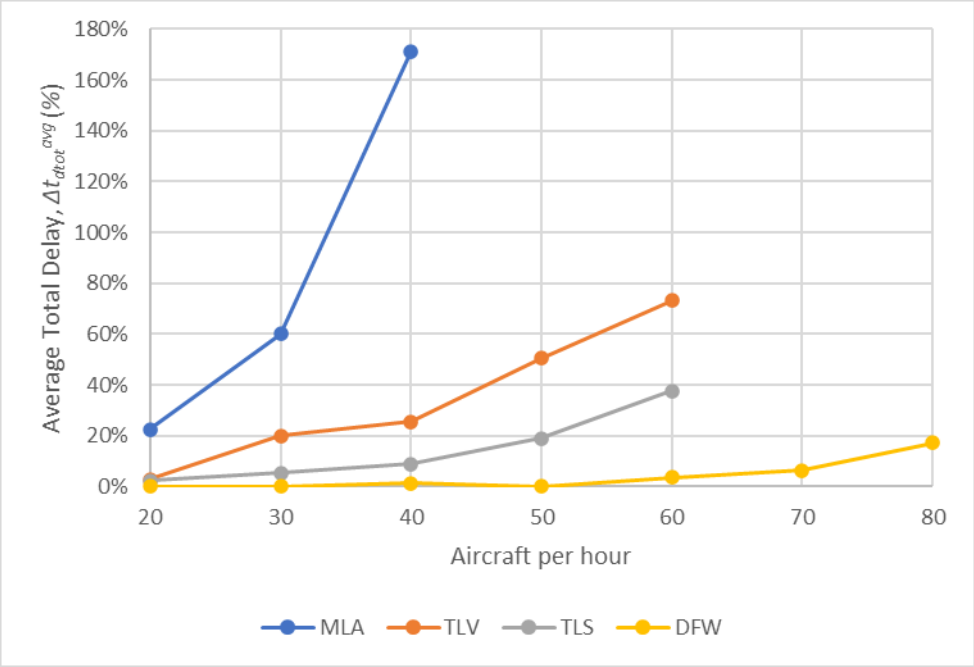


Fig. 40. Average Total Delay (%), Δt_{dtot}^{avg} in *Test Scenario 1* using the *Deterministic Model*.

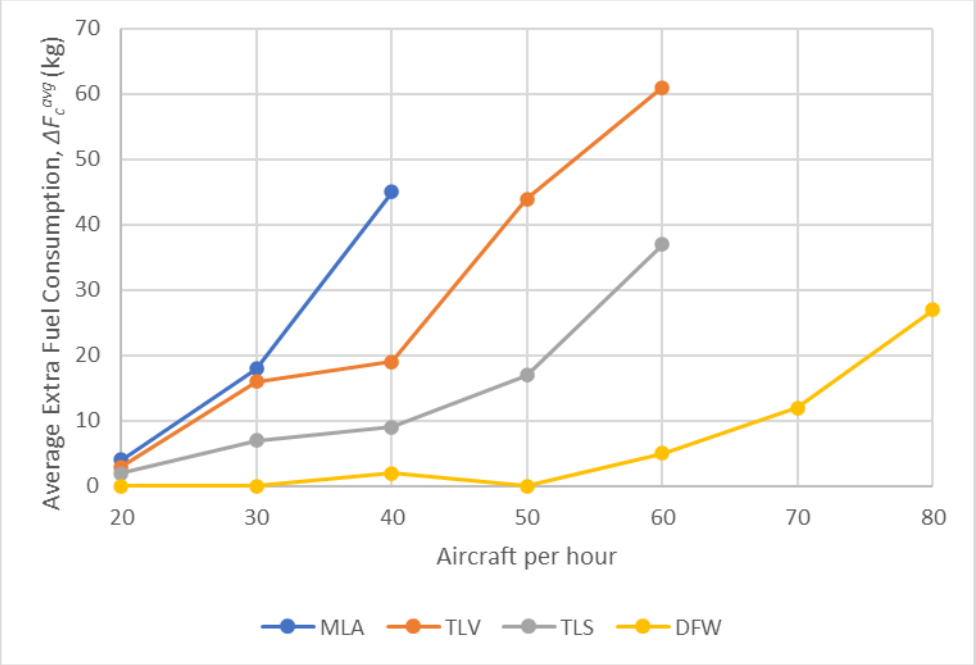


Fig. 41. Average Extra Fuel Consumption (kg), ΔF_c^{avg} in *Test Scenario 1* using the *Deterministic Model*.

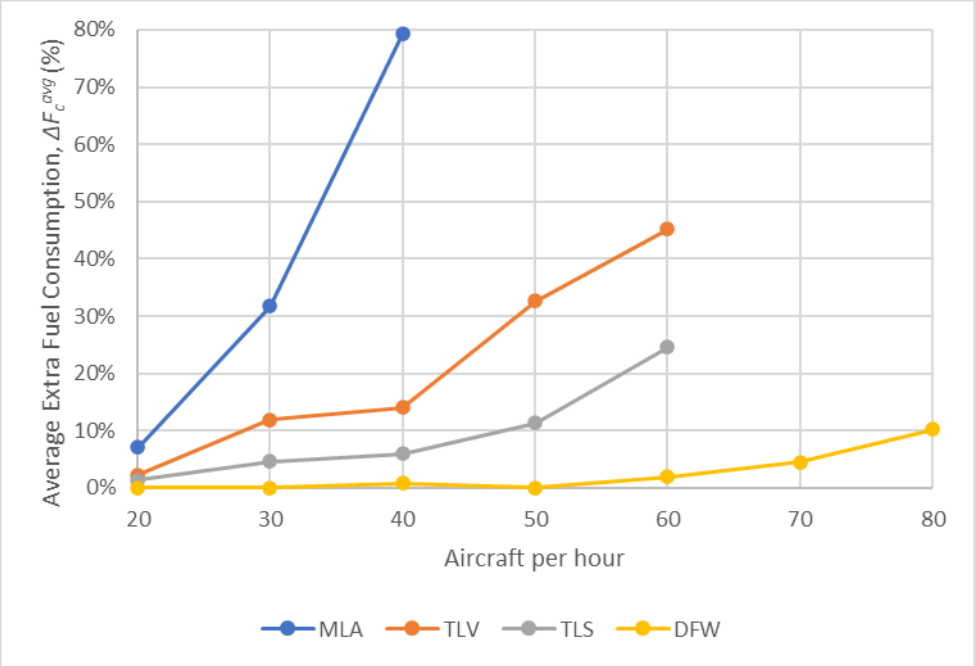


Fig. 42. Average Extra Fuel Consumption (%), ΔF_c^{avg} in *Test Scenario 1* using the *Deterministic Model*.

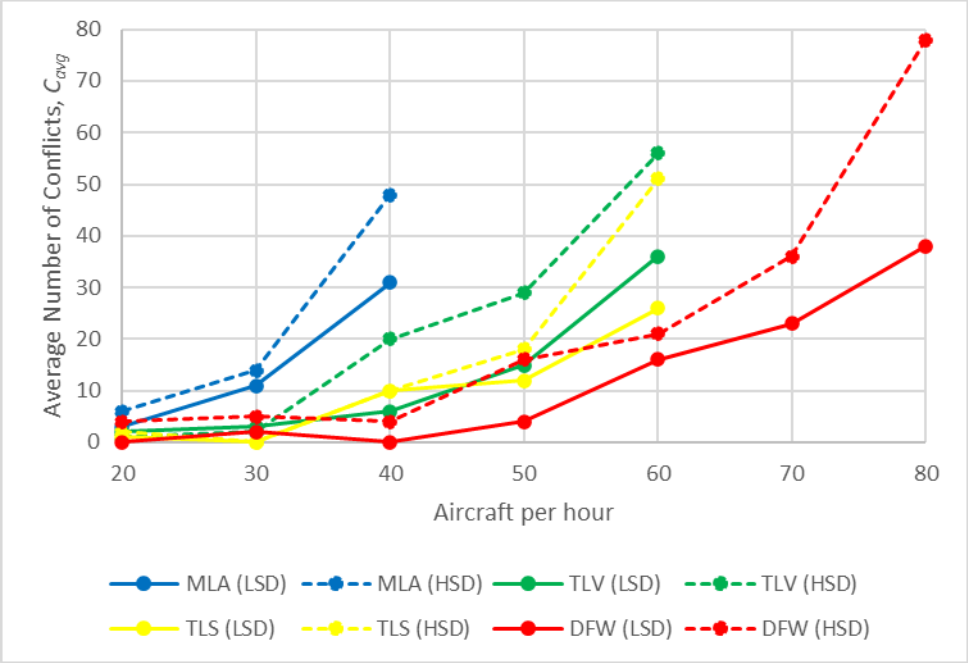


Fig. 43. Average Number of Conflicts, C_{avg} in *Test Scenario 1* using the *Probabilistic Model*.

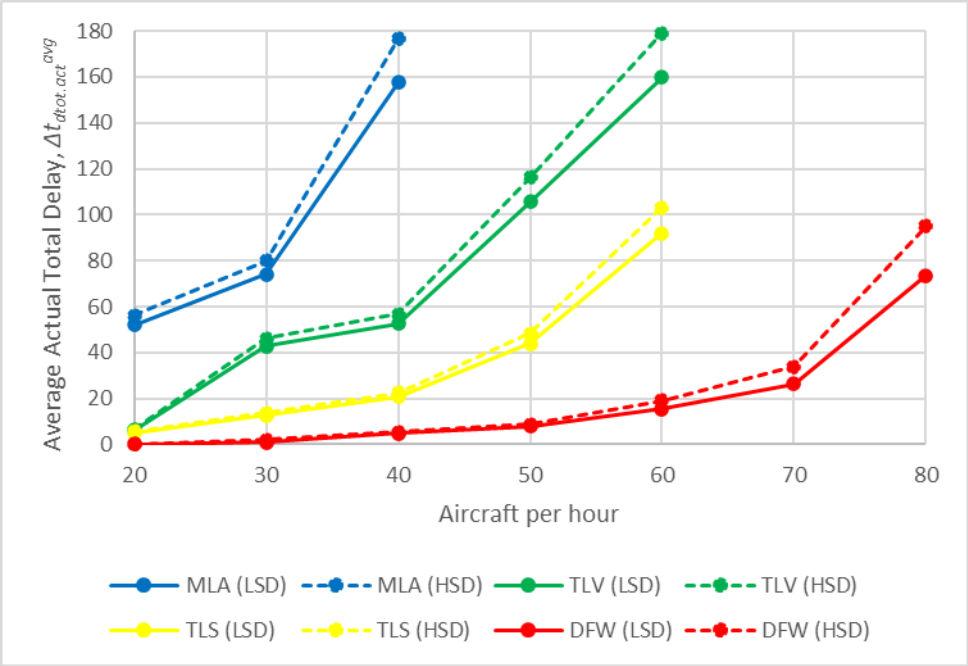


Fig. 44. Average Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ in *Test Scenario 1* using the *Probabilistic Model*.

7.2 Results of Test Scenario 2

The results obtained for *Test Scenario 2* are shown in Fig. 45-Fig. 51 show the results of *Test Scenario 2*, calculated using *Static Allocation* with a *Time-Wise Approach*. Results for each level of traffic and for each ratio of tow trucks to aircraft are presented as average values for ease of reporting.

Fig. 45 and Fig. 46 show TA (%) and Δt_{tow} (%), respectively, for different percentages of tow trucks for each airport using the *Deterministic Model*. Both percentages initially increase with an increasing percentage of tow trucks but then flatten out. From Fig. 45 it can be seen that, in both cases, when the percentage of tow trucks exceeds approximately 30%, 90% (or more) of the traffic is handled by the tow trucks. This means that only 10% (or less) of the aircraft have to taxi using their main engines and there is no significant improvement when the percentage of tow trucks is increased beyond 30%. The trend of ΔF_s^{avg} (kg), shown in Fig. 47, is strongly correlated with Δt_{tow} (%); in fact, when the percentage of tow trucks exceeds approximately 30%, there is no significant improvement in fuel savings. The fuel saved in MLA is significantly lower than the other airports, probably due to the limited length of its taxiway infrastructure; this is expected because fuel savings are proportional to route length. This indicates that this type of taxiing is mostly beneficial at large airports.

Fig. 48 shows Δt_{ru}^{avg} (%) for different percentages of tow trucks for each airport using the *Deterministic Model* where, as expected, Δt_{ru}^{avg} (%) steadily decreases for all airports as the percentage of tow trucks is increased. It is also interesting to note that the results are relatively similar for all airports; this could indicate that the results for this metric are not affected by different airport geometries and sizes. It is important to exercise caution when determining the appropriate number of tow trucks required to efficiently tow aircraft: a sufficient number of tow trucks must be available to tow as many aircraft as possible, while avoiding an excessive number of tow trucks is also crucial to ensure that they are not left parked and that their usage is maximized. Interestingly, Δt_{ru}^{avg} (%) never exceeds 50% for any airport; one of the reasons for this could be the fact that the tow trucks have to occasionally recharge their battery (and the tow truck utilisation time is calculated as a percentage of the total simulation time; thus, if a tow truck spends a lot of time recharging its battery, the value of the metric decreases). This clearly shows that battery performance is a crucial factor in tow truck-based electric taxi operations. Apart from using fast charging tow trucks, this utilisation value can be improved by using *Dynamic Allocation* (which is tested in *Test Scenario 4*), which assigns tow trucks not only when these are parked in a depot, but also whilst returning to a depot after a previous mission.

Fig. 49-Fig. 51 show C_{avg}, M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s), respectively, for different levels of traffic for each airport under test using the *Probabilistic Model* with two different standard deviations (LSD and HSD). While the trend of C_{avg} steadily rises with increasing percentages of tow trucks (with noticeably higher values for HSD than for LSD) and stabilises for

percentages higher than 30%, M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s) first increase between 10% and 30% of tow trucks; then they reach a peak for 30% of tow trucks and finally slightly decrease for higher percentages. Also, C_{avg} and $\Delta t_{dtot.act}^{avg}$ (s) are considerably higher in this test scenario when compared to the values in *Test Scenario 1* for the same level of traffic. This could be due to the fact that percentages of tow trucks higher than 30% indicate an excess of tow trucks (as can also be noticed in Fig. 45) and, since the algorithm favours allocating tow trucks with lower utilisation times, a tow truck that just completed a mission and returned to the depot is unlikely to be given a new assignment. With the *Probabilistic Model* this means that it is less likely that a tow truck will fail to reach the aircraft or encounter major delays in getting there. These results demonstrate the robustness of the algorithm when using *Static Allocation* with a *Time-Wise Approach* by showing an overall low sensitivity to uncertainty (especially for LSD).

Fig. 52 and Fig. 53 show examples of the histograms obtained whilst computing the number of conflicts C. These plots were obtained for 100 simulation runs in TLS and for a percentage of tow trucks equal to 30% using the *Probabilistic Model* with LSD and HSD, respectively. Additionally, Table 28 in Annex 1 shows C_σ, M_σ and $\Delta t_{dtot.act}^{\sigma}$ obtained with the *Probabilistic Model* (both with LSD and HSD) for all the airports and for different percentages of tow trucks, in *Test Scenario 2*.

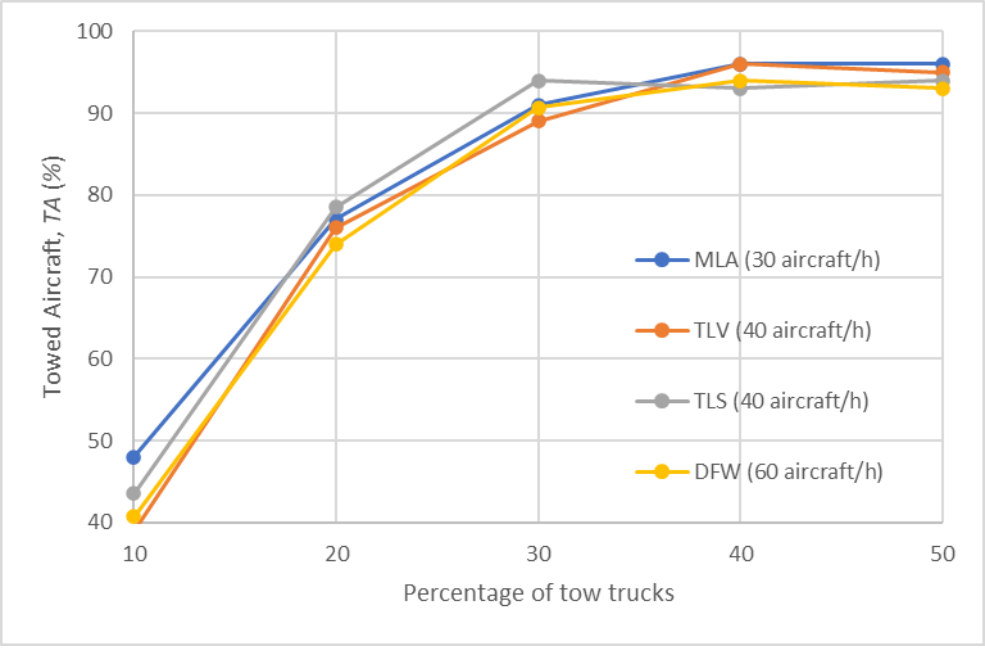


Fig. 45. Towed Aircraft (%), *TA* in *Test Scenario 2* using the *Deterministic Model*.

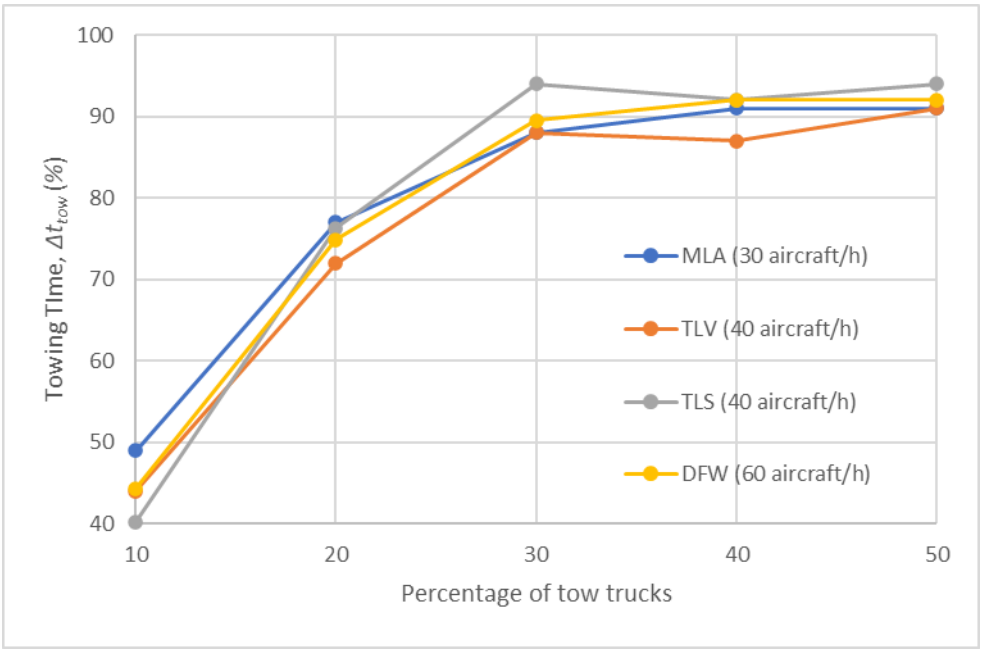


Fig. 46. Towing Time (%), Δt_{tow} in *Test Scenario 2* using the *Deterministic Model*.

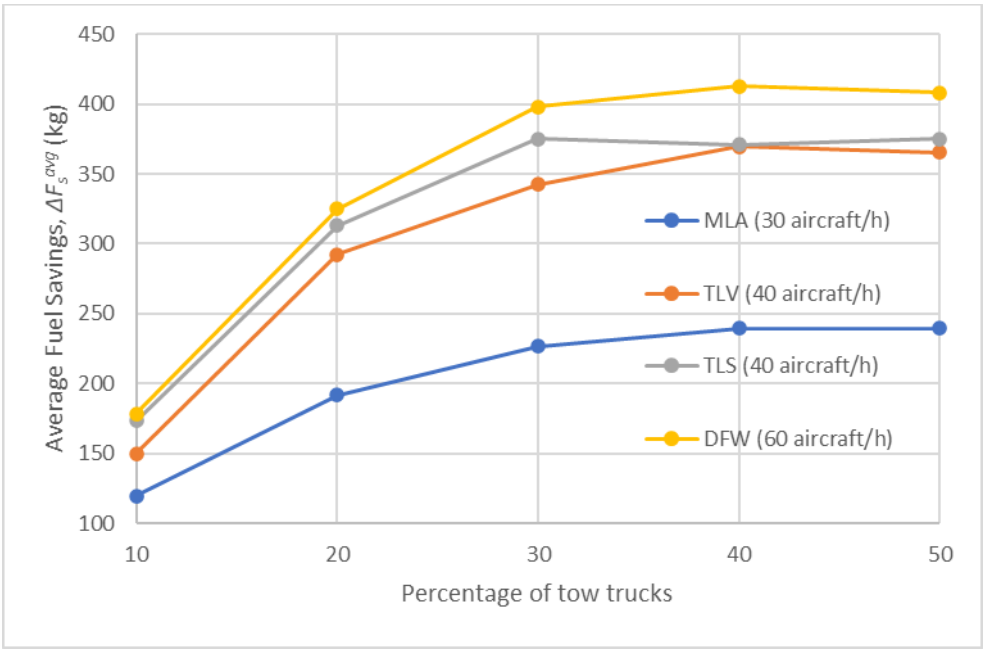


Fig. 47. Average Fuel Savings (kg), ΔF_s^{avg} in *Test Scenario 2* using the *Deterministic Model*.

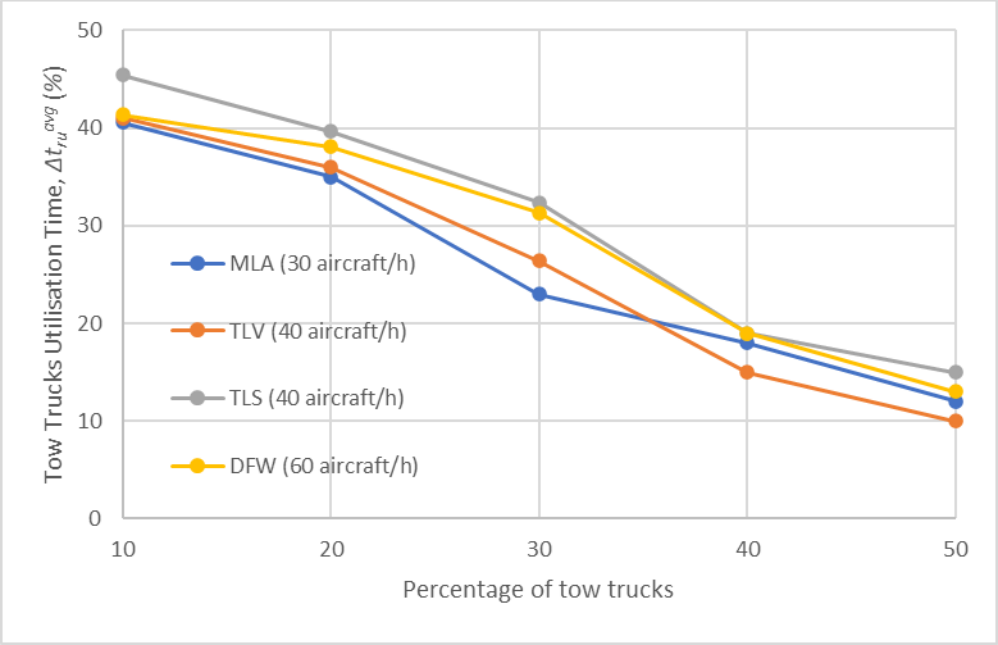


Fig. 48. Tow Truck Utilisation Time (%), Δt_{ru}^{avg} in *Test Scenario 2* using the *Deterministic Model*.

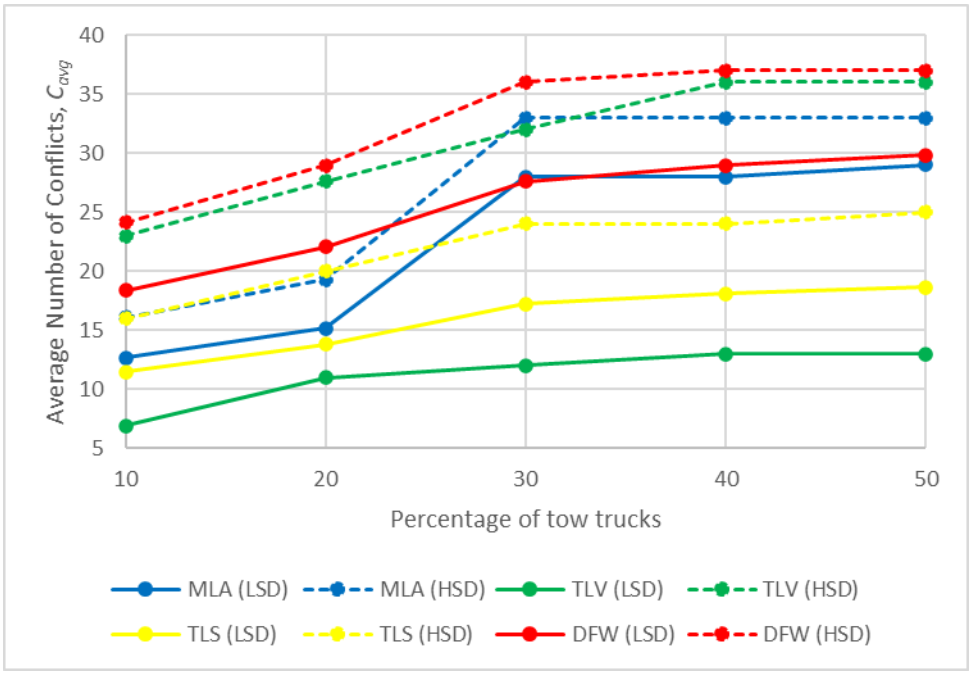


Fig. 49. Average Number of Conflicts, C_{avg} in *Test Scenario 2* using the *Probabilistic Model*.

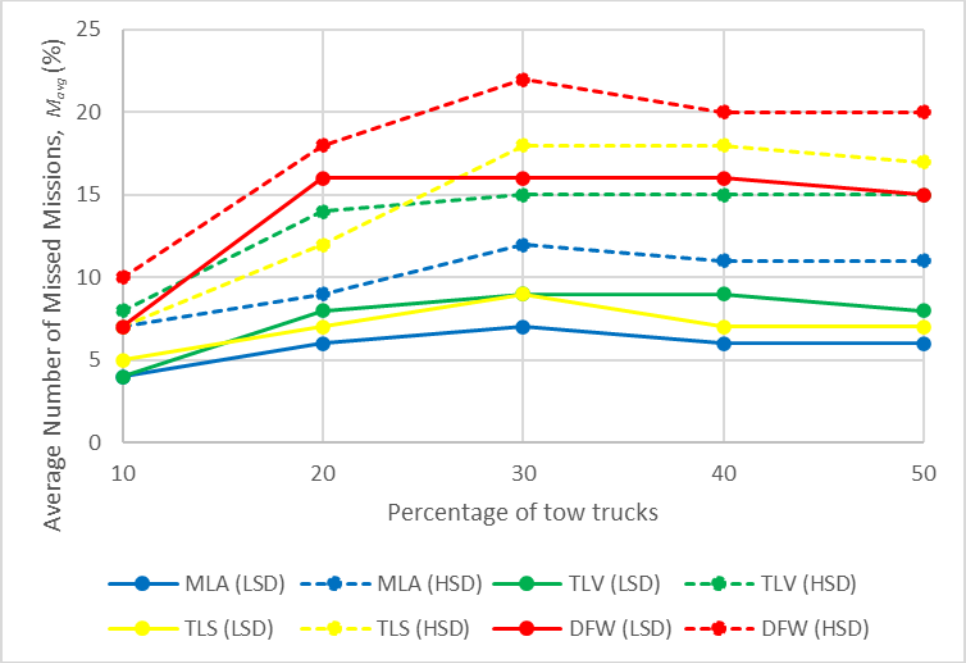


Fig. 50. Average Number of Missed Missions (%), M_{avg} in *Test Scenario 2* using the *Probabilistic Model*.

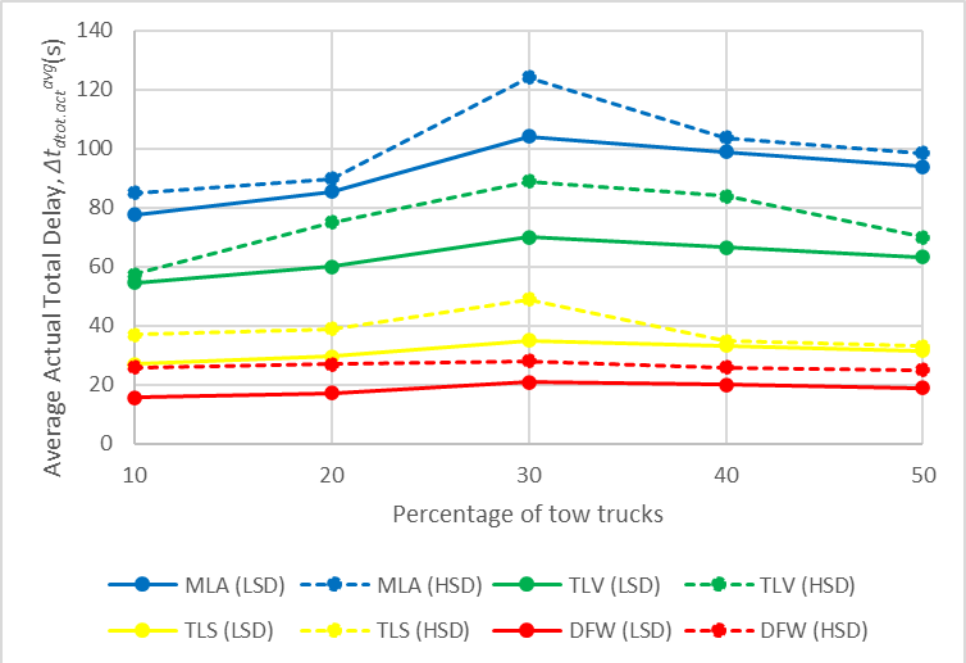


Fig. 51. Average Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ in *Test Scenario 2* using the *Probabilistic Model*.

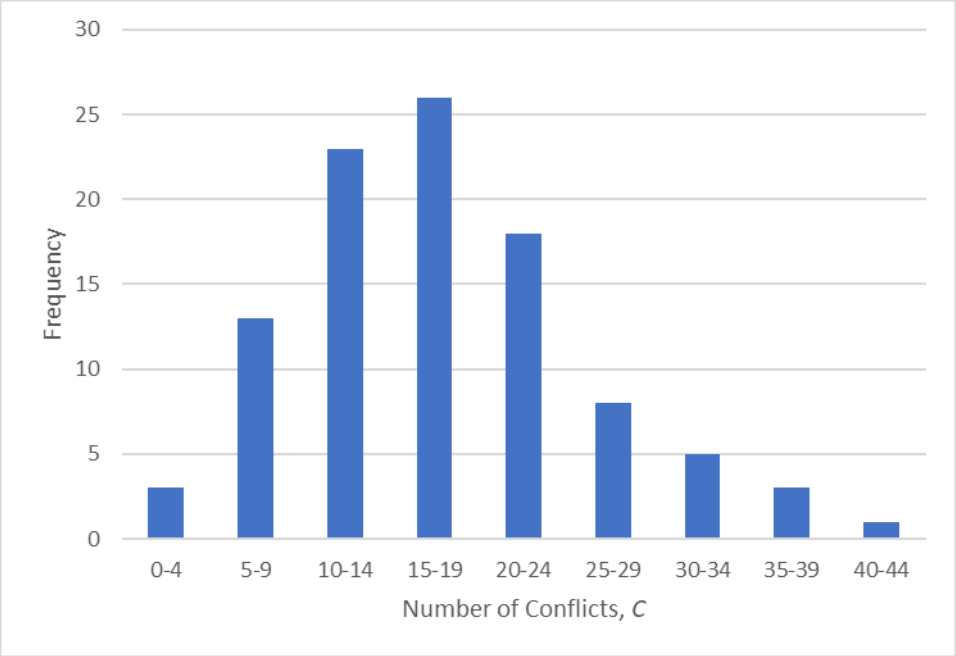


Fig. 52. Distribution of Number of Conflicts, *C* for 100 simulation runs in TLS and for a percentage of tow trucks equal to 30% in *Test Scenario 2* using the *Probabilistic Model* with LSD.

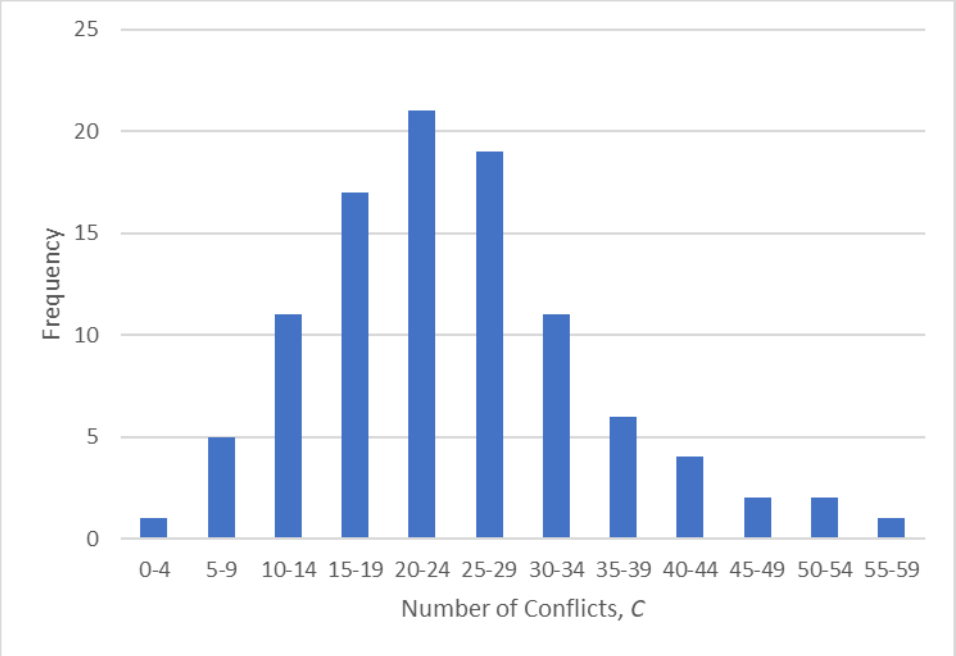


Fig. 53. Distribution of Number of Conflicts, *C* for 100 simulation runs in TLS and for a percentage of tow trucks equal to 30% in *Test Scenario 2* using the *Probabilistic Model* with HSD.

7.3 Results of Test Scenario 3

The results obtained for *Test Scenario 3* are shown in Fig. 54-Fig. 61, calculated using *Static Allocation* with *Fuel-Wise Approach*. Results for each level of traffic and for each ratio of tow trucks to aircraft are presented as average values for ease of reporting.

Fig. 54 and Fig. 55 show TA (%) and Δt_{tow} (%), respectively, for different percentages of tow trucks for each airport using the *Deterministic Model*. As observed for the same metrics in *Test Scenario 2*, both percentages initially rise with an increase in the percentage of tow trucks, but eventually level out. When the percentage of tow trucks exceeds approximately 30%, 90% (or more) of the traffic is handled by the tow trucks. Thus, only 10% (or less) of aircraft must taxi using their primary engines, and the percentage of towed aircraft does not increase significantly when the percentage of tow trucks exceeds 30%. The higher values for these two metrics in *Test Scenario 3*, compared to the outcomes of *Test Scenario 2*, could be attributed to the *Fuel-Wise Approach* since, when using this approach, the algorithm seeks to maximise the number of towed aircraft at the expense of taxi delays.

The trends of ΔF_s^{avg} (kg) and of Δt_{dtot}^{avg} (s), observed for various percentages of tow trucks at each airport using the *Deterministic Model* are shown in Fig. 56 and Fig. 57 respectively. These results are related to the towing time Δt_{tow} (%); in fact, for a percentage of tow trucks exceeding approximately 30%, the fuel savings do not significantly improve, while the delays do not increase any further. However, ΔF_s^{avg} (kg) in this case is slightly higher for each airport (for instance, 20 kg on average for a level of traffic of 30 aircraft per hour) than the fuel savings obtained in *Test Scenario 2* (shown in Fig. 47). On the other hand, Δt_{dtot}^{avg} (s) – which is represented by including the values obtained with the values for 0% tow trucks in *Test Scenario 1*, as shown in Fig. 39 – increases with percentage of tow trucks and levels off when the percentage of tow trucks exceeds 30%. This was expected as, while *Test Scenario 2* was carried out using the *Time-Wise Approach*, *Test Scenario 3* was carried out using the *Fuel-Wise Approach*, thus prioritising fuel savings over time delays.

Fig. 58 shows Δt_{ru}^{avg} (%) for different percentages of tow trucks for each airport using the *Deterministic Model* where, in line with expectations, Δt_{ru}^{avg} (%) steadily declines at all airports as the percentage of tow trucks rises. When compared to *Test Scenario 2* (shown in Fig. 48), Δt_{ru}^{avg} (%) shows slightly higher values; however, even in this case it never exceeds 50%, once again proving that battery performance is a crucial factor for tow truck utilisation and better management of the tow trucks using the *Dynamic Allocation* is needed to improve this metric.

Fig. 59-Fig. 61 show C_{avg}, M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s), respectively, for different levels of traffic for each airport under test using the *Probabilistic Model* with two different standard deviations (LSD and HSD). With an increase in the percentage of tow trucks, the trend of C_{avg} progressively climbs (with substantially larger values for HSD than for LSD) and stabilises at percentages greater than 30%, as observed also in *Test Scenario 2* (see Fig. 49), where similar

but slightly lower values were obtained. The values of M_{avg} (%) in this scenario are also comparable to those seen in *Test Scenario 2* (as shown in Fig. 50), with a very little increase, perhaps as a result of the fact that more aircraft were towed in this scenario. Finally, $\Delta t_{dtot.act}^{avg}$ (s) has higher values than those observed in *Test Scenario 2* (as shown in Fig. 51); this is explained by the fact that, when the *Fuel-Wise Approach* is used, Δt_{dtot}^{avg} (s) increases in the *Deterministic Model* (as seen in Fig. 57) in order to maximise fuel savings, and the effect of the delays is further amplified in the *Probabilistic Model*.

Overall, the results obtained using the *Probabilistic Model* in this scenario reveal a low sensitivity to uncertainty, notably for LSD, proving the algorithm's robustness when employing *Static Allocation* with a *Fuel-Wise Approach*. These outcomes, together with the ones of *Test Scenario 2* (which employs *Static Allocation* with a *Fuel-Wise Approach*) indicate that *Static Allocation* is a viable way of dealing with uncertainty.

Fig. 62 and Fig. 63 show examples of the histograms obtained whilst computing the number of missed attachments, M. These plots were obtained for 100 simulation runs in TLV and for a percentage of tow trucks equal to 30% using the *Probabilistic Model* with LSD and HSD, respectively. Additionally, Table 29 in Annex 1 shows C_σ, M_σ and $\Delta t_{dtot.act}^{\sigma}$ obtained with the *Probabilistic Model* (both with LSD and HSD) for all the airports and for different percentages of tow trucks, in *Test Scenario 3*.

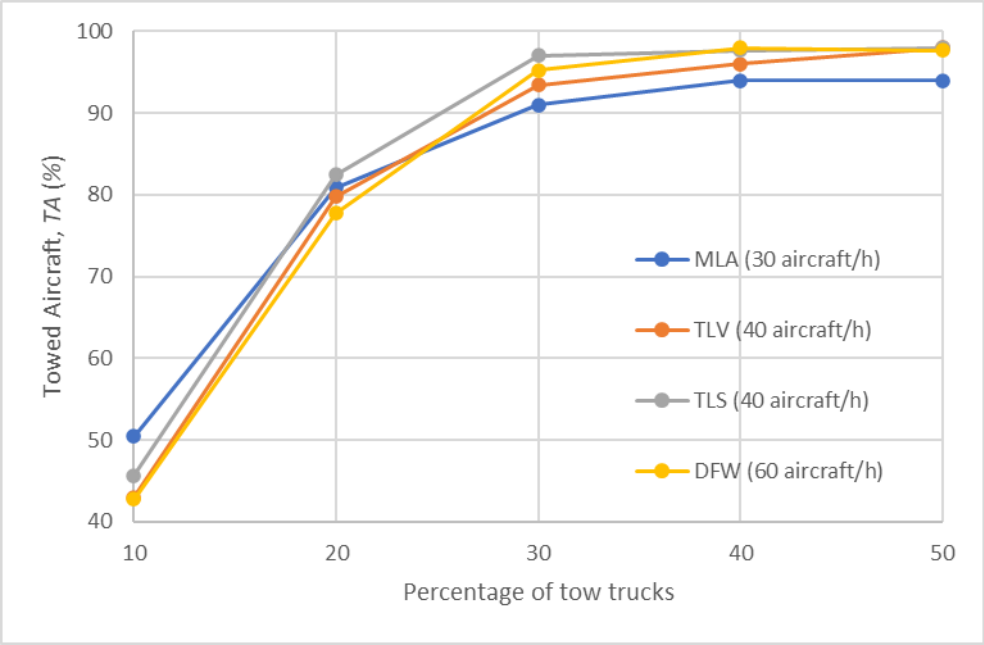


Fig. 54. Towed Aircraft (%), *TA* in *Test Scenario 3* using the *Deterministic Model*.

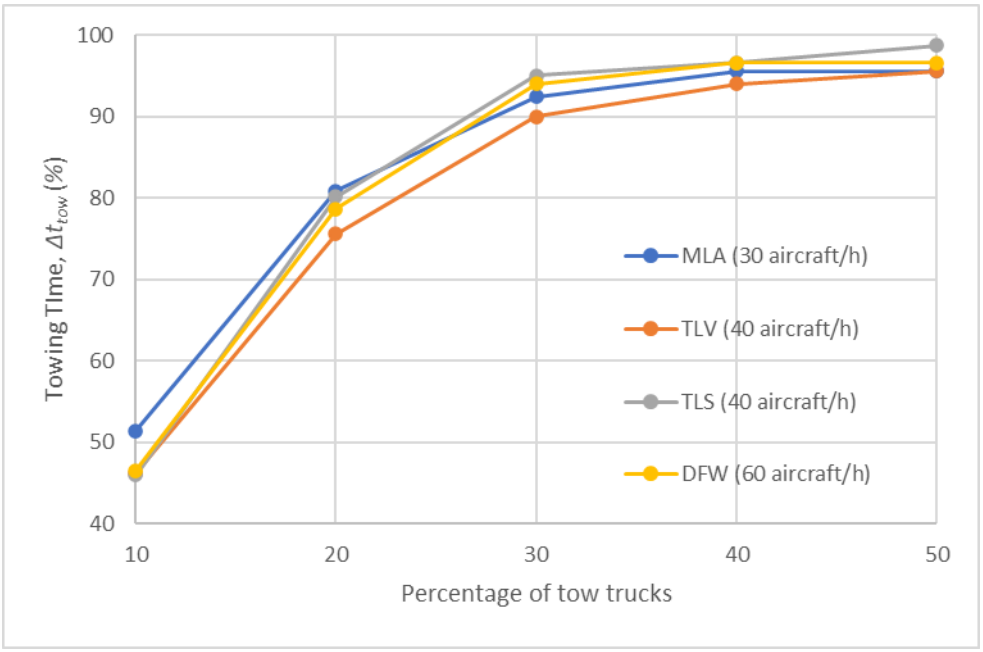


Fig. 55. Towing Time (%), Δt_{tow} in *Test Scenario 3* using the *Deterministic Model*.

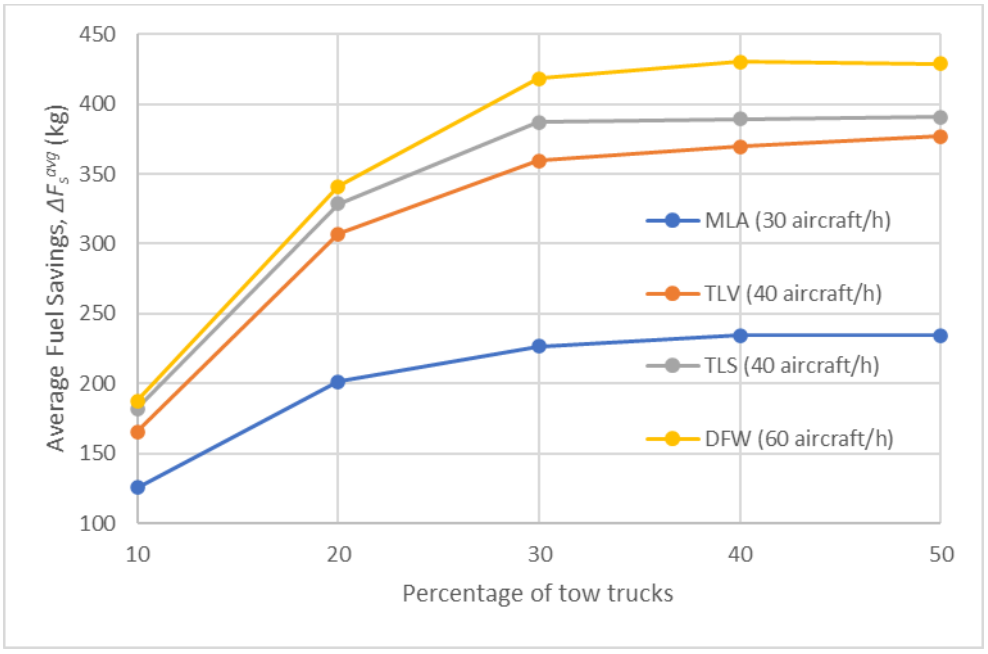


Fig. 56. Average Fuel Savings (kg), ΔF_s^{avg} in *Test Scenario 3* using the *Deterministic Model*.

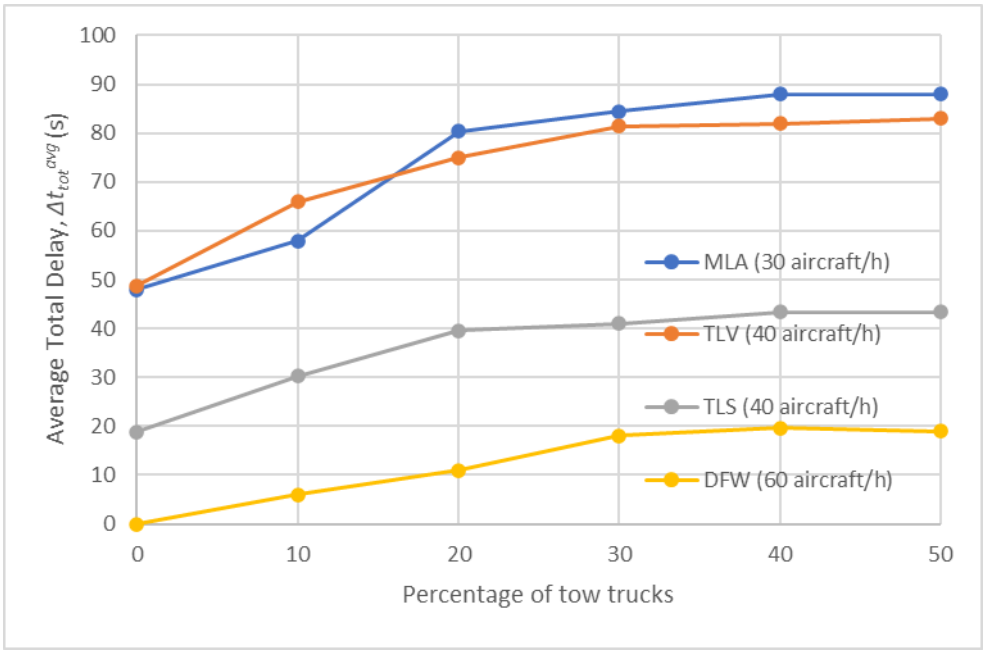


Fig. 57. Average Total Delay (s), Δt_{dtot}^{avg} in *Test Scenario 3* using the *Deterministic Model*.

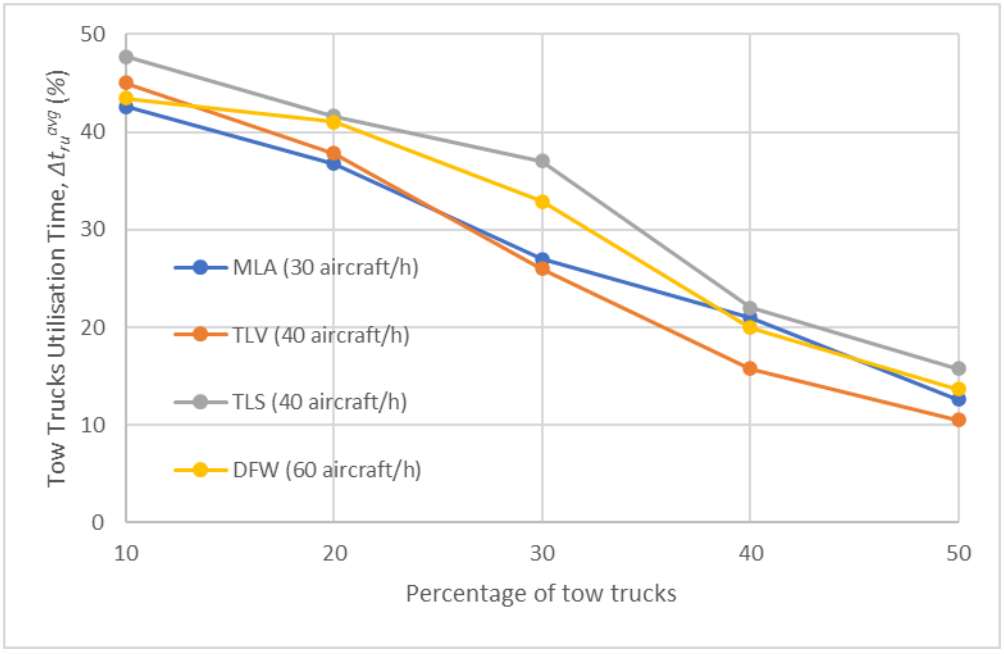


Fig. 58. Tow Truck Utilisation Time (%), Δt_{ru}^{avg} in *Test Scenario 3* using the *Deterministic Model*.

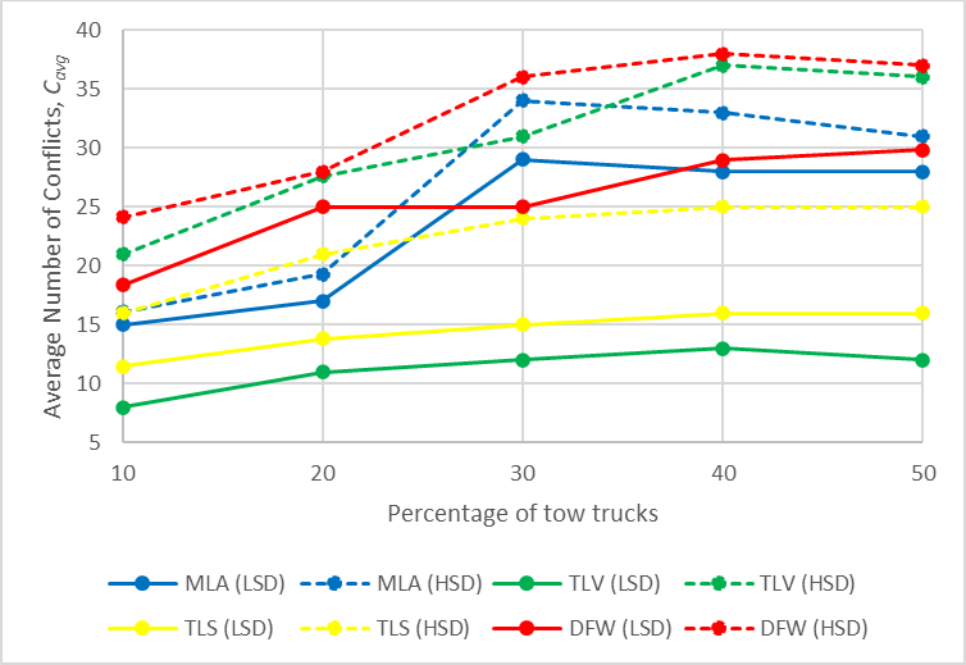


Fig. 59. Average Number of Conflicts, C_{avg} in *Test Scenario 3* using the *Probabilistic Model*.

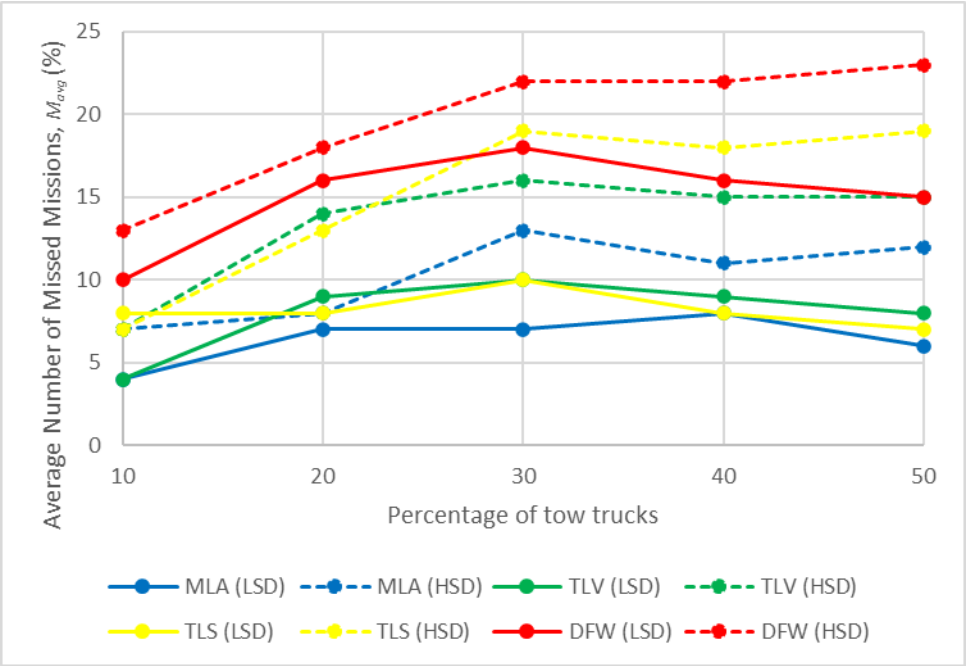


Fig. 60. Average Number of Missed Missions (%), M_{avg} in *Test Scenario 3* using the *Probabilistic Model*.

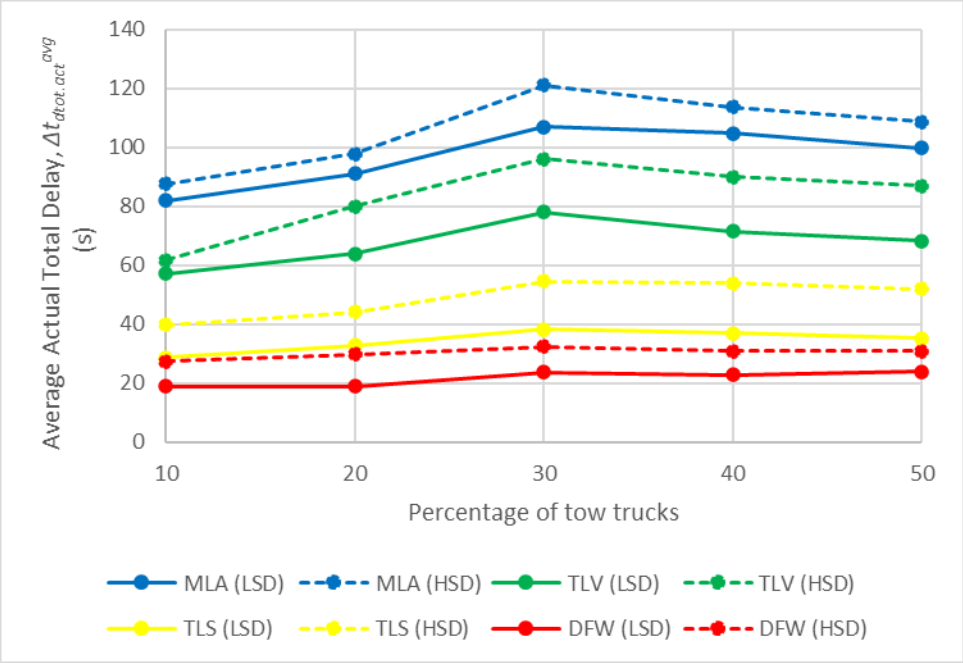


Fig. 61. Average Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ in *Test Scenario 3* using the *Probabilistic Model*.

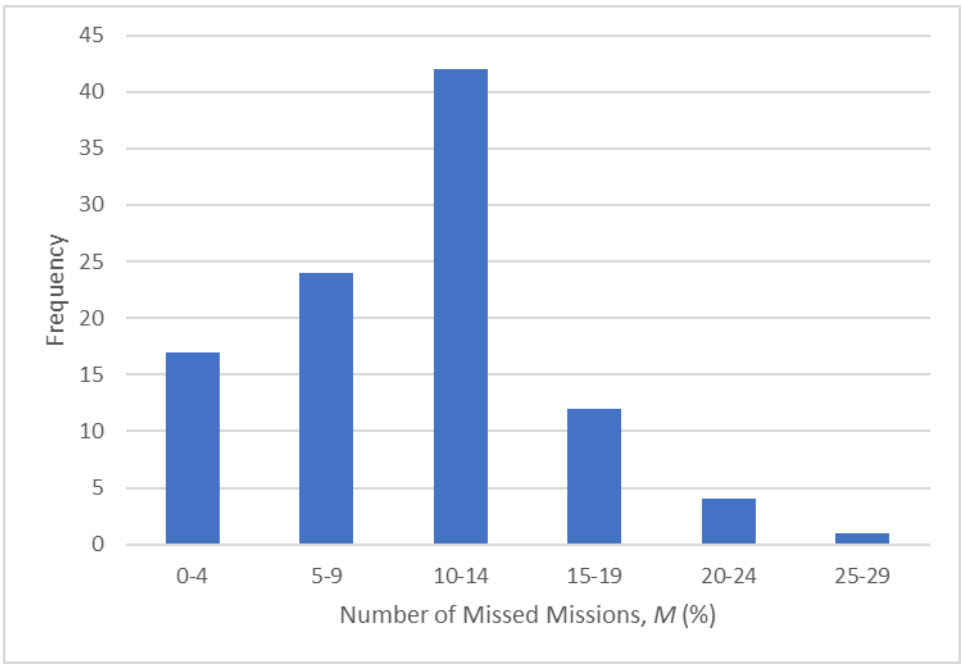


Fig. 62. Distribution of Number of Missed Missions (%), *M* for 100 simulation runs in TLV and for a percentage of tow trucks equal to 30% in *Test Scenario 3* using the *Probabilistic Model* with LSD.

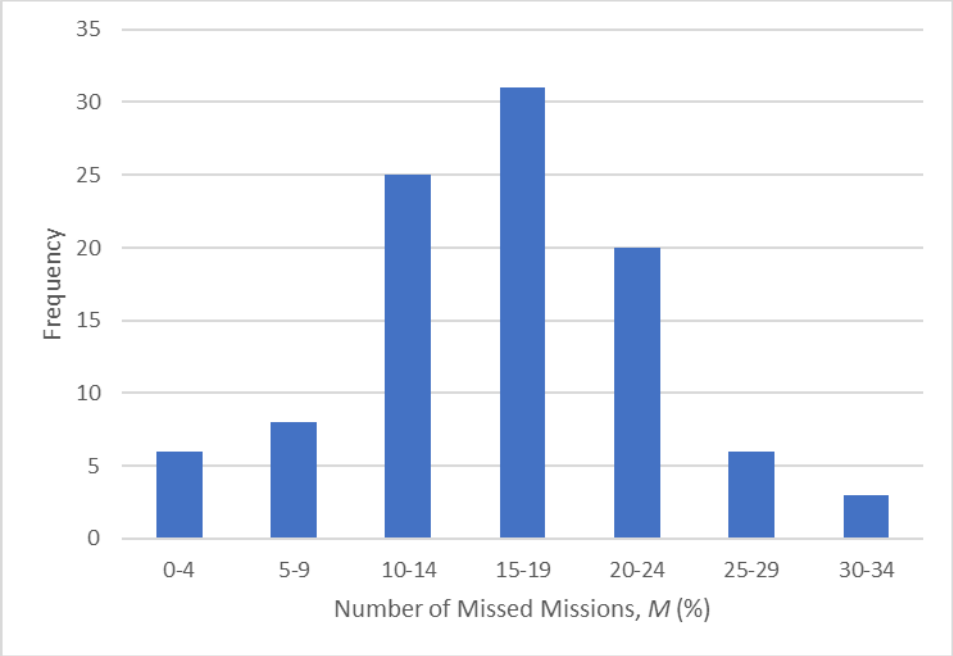


Fig. 63. Distribution of Number of Missed Missions (%), *M* for 100 simulation runs in TLV and for a percentage of tow trucks equal to 30% in *Test Scenario 3* using the *Probabilistic Model* with HSD.

7.4 Results of Test Scenario 4

The results obtained for *Test Scenario 4* are shown in Fig. 64-Fig. 70, calculated using *Dynamic Allocation* with *Time-Wise Approach*. Results for each level of traffic and for each ratio of tow trucks to aircraft are presented as average values for ease of reporting.

Fig. 64 and Fig. 65 show TA (%) and Δt_{tow} (%), respectively, for various percentages of tow trucks at each airport using the *Deterministic Model*. The values of the two metrics initially grow as the percentage of tow trucks increases, but finally stabilise for a percentage of tow trucks exceeding approximately 30%, as was observed in the previous two scenarios. However, in this instance, the values are slightly higher than the ones observed in *Test Scenario 3* (refer to in Fig. 54 and Fig. 55), and definitely higher than the ones observed in *Test Scenario 2* (as can be seen in Fig. 45 and Fig. 46), due to the higher efficiency of the algorithm when assigning tow trucks with the *Dynamic Allocation* approach.

As for the previous two metrics, the trend of ΔF_s^{avg} (kg) shown in Fig. 66 (observed for various percentages of tow trucks at each airport using the *Deterministic Model*) does not significantly change for a percentage of tow trucks over 30%. However, ΔF_s^{avg} (kg) in this instance is slightly higher than the values recorded for *Test Scenario 2* (shown in Fig. 47) and similar to the values obtained in *Test Scenario 3* (shown in Fig. 56), demonstrating once more the superior performance of *Dynamic Allocation* compared to *Static Allocation* when assigning tow trucks. With *Dynamic Allocation Approach*, the tow trucks are not required to return to a depot after each mission before being allocated to a new one; therefore, they are able to

complete a higher number of missions during the simulation, thus increasing the average fuel savings.

Fig. 67 shows Δt_{ru}^{avg} (%) for different percentages of tow trucks for each airport using the *Deterministic Model*. In keeping with the trend of the previous two scenarios (as seen in Fig. 48 and Fig. 58), Δt_{ru}^{avg} (%) steadily decreases at all airports as the percentage of tow trucks increases. Yet, Δt_{ru}^{avg} (%) has greater values, exceeding 50%, when compared to the prior two cases. However, Δt_{ru}^{avg} (%) never rises over 60%, demonstrating once more how important battery performance is to tow truck utilisation.

Fig. 68-Fig. 70 show C_{avg}, M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s), respectively, for different levels of traffic for each airport under test using the *Probabilistic Model* with two different standard deviations (LSD and HSD). The trend of C_{avg} rises steadily with an increase in the proportion of tow trucks (with noticeably larger values for HSD than for LSD), stabilising at percentages greater than 30%, as seen in *Test Scenario 2* (see Fig. 49) and *Test Scenario 3* (see Fig. 59). The values for *Test Scenario 4* are only marginally higher than those obtained in *Test Scenario 3* (likely due to the higher frequency of tow truck missions in the simulations), where similar but slightly lower values are observed. The plots of M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s) peak when the percentage of tow trucks is equal to 30%, then slowly decline, similarly to what was observed in *Test Scenario 2* (as shown in Fig. 50 and Fig. 51) and in *Test Scenario 3* (as shown in Fig. 60 and Fig. 61). However, the values in this scenario are significantly higher than those observed in the previous two scenarios; in fact, with *Dynamic Allocation*, the utilisation of the tow trucks is considerably high (as shown in Fig. 67) and, if a tow truck is immediately reassigned to a new mission after completing one, it may accumulate significant delays or fail to attach to the assigned aircraft. These results show that the algorithm is less resistant to uncertainties when using *Dynamic Allocation* by showing an overall higher sensitivity when compared to *Test Scenarios 2* and *3*.

Fig. 71 and Fig. 72 show examples of the histograms obtained whilst computing the actual total delays, $\Delta t_{dtot.act}$ (s). These plots were obtained for 100 simulation runs in MLA and for a percentage of tow trucks equal to 30% using the *Probabilistic Model* with LSD and HSD, respectively. Additionally, Table 30 in Annex 1 shows C_σ, M_σ and $\Delta t_{dtot.act}^{\sigma}$ obtained with the *Probabilistic Model* (both with LSD and HSD) for all the airports and for different percentages of tow trucks, in *Test Scenario 4*.

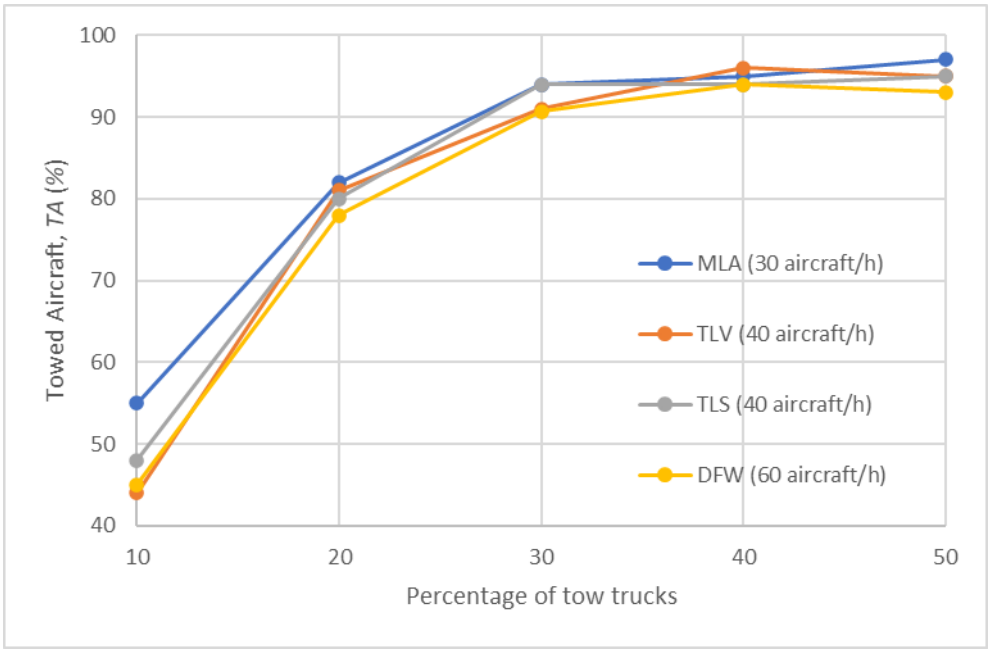


Fig. 64. Towed Aircraft (%), *TA* in *Test Scenario 4* using the *Deterministic Model*.

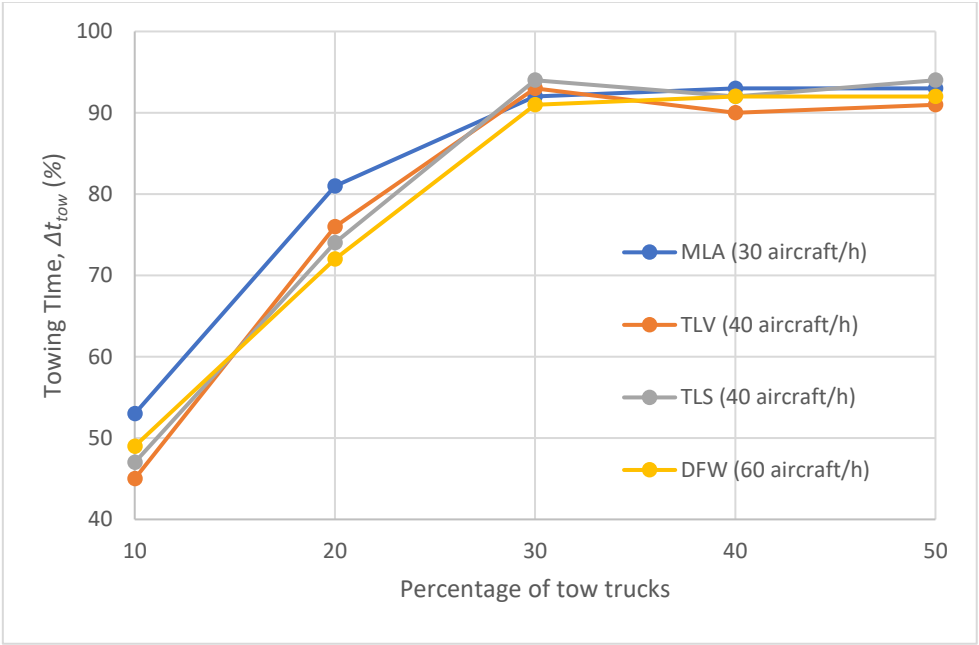


Fig. 65. Towing Time (%), Δt_{tow} in *Test Scenario 4* using the *Deterministic Model*.

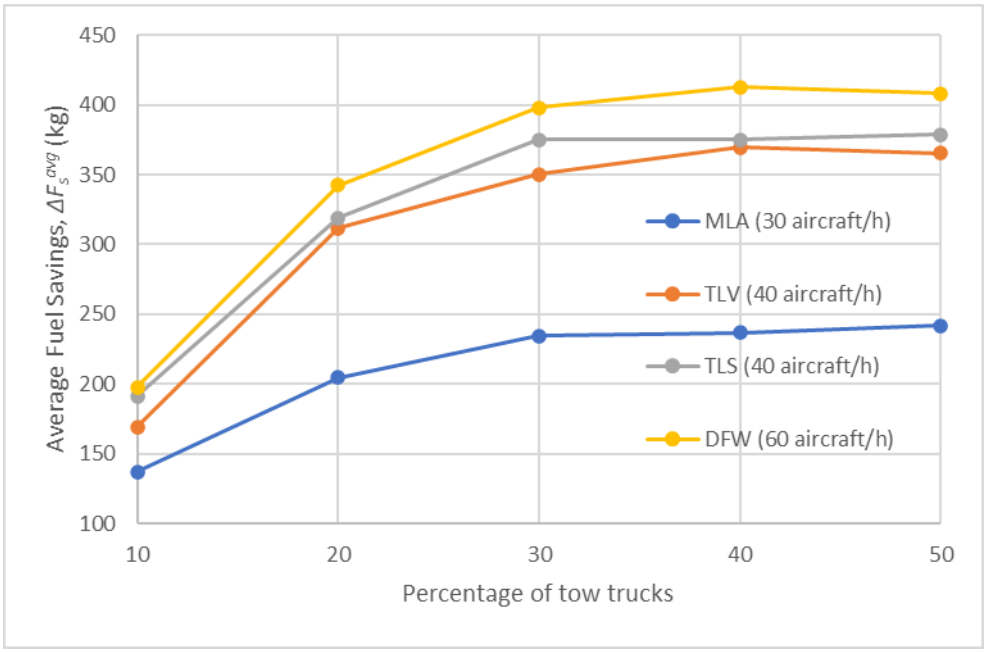


Fig. 66. Average Fuel Savings (kg), ΔF_s^{avg} in *Test Scenario 4* using the *Deterministic Model*.

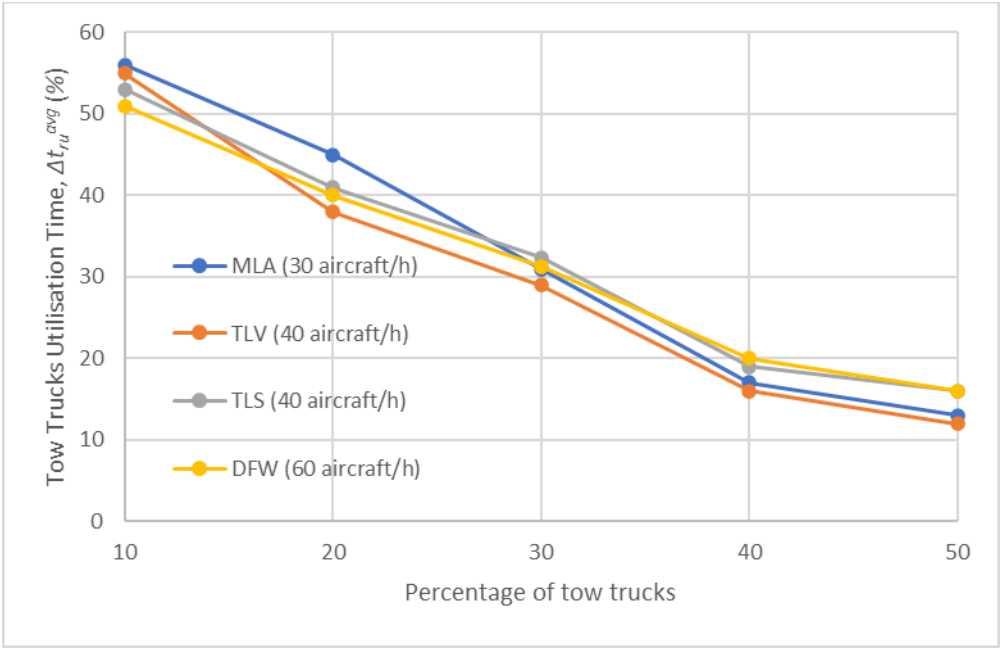


Fig. 67. Tow Truck Utilisation Time (%), Δt_{ru}^{avg} in *Test Scenario 4* using the *Deterministic Model*.

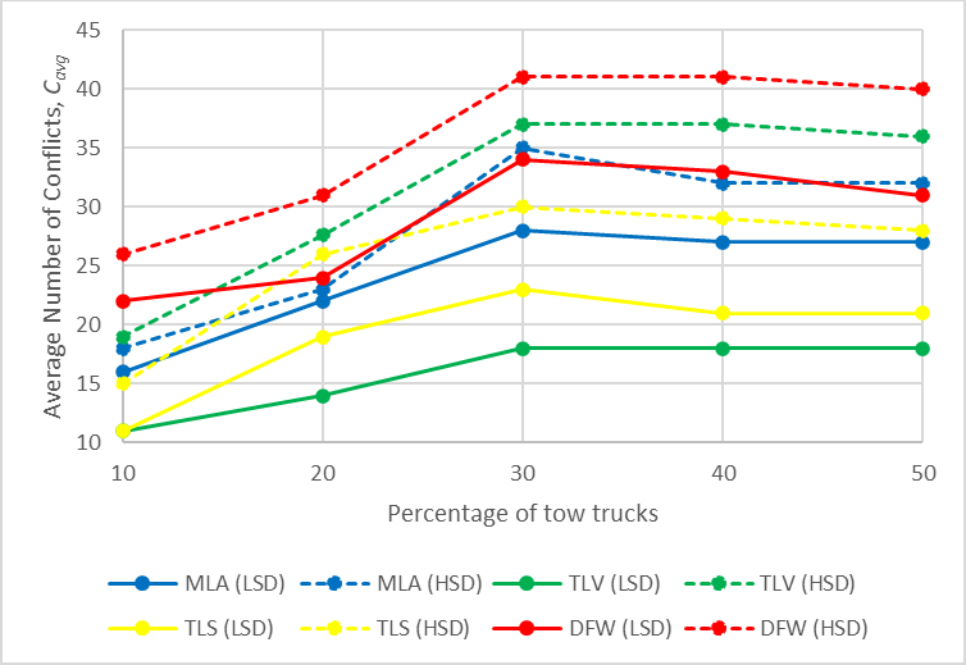


Fig. 68. Average Number of Conflicts, C_{avg} in *Test Scenario 4* using the *Probabilistic Model*.

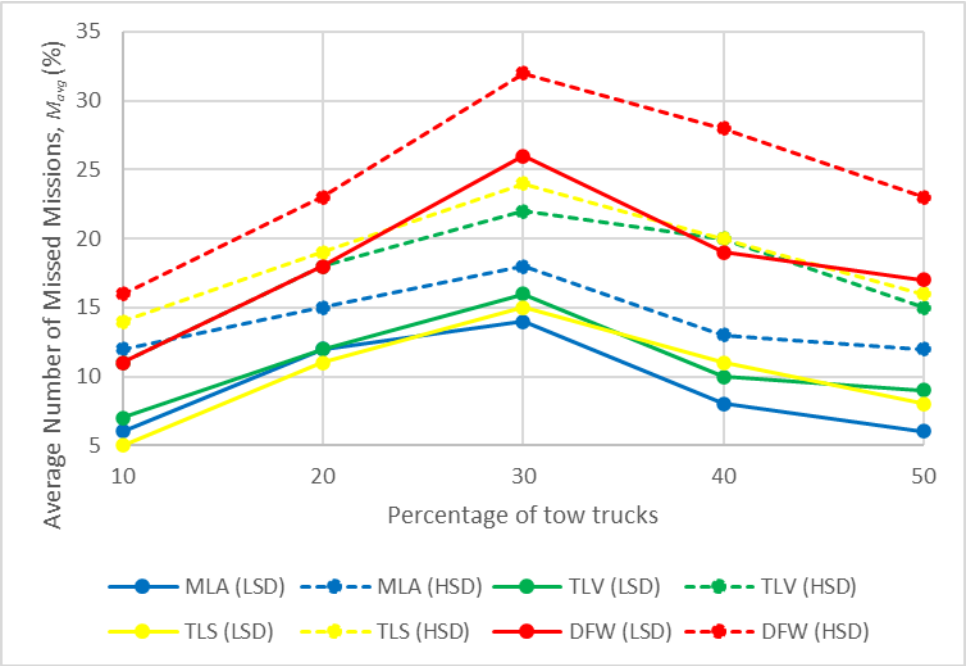


Fig. 69. Average Number of Missed Missions (%), M_{avg} in *Test Scenario 4* using the *Probabilistic Model*.

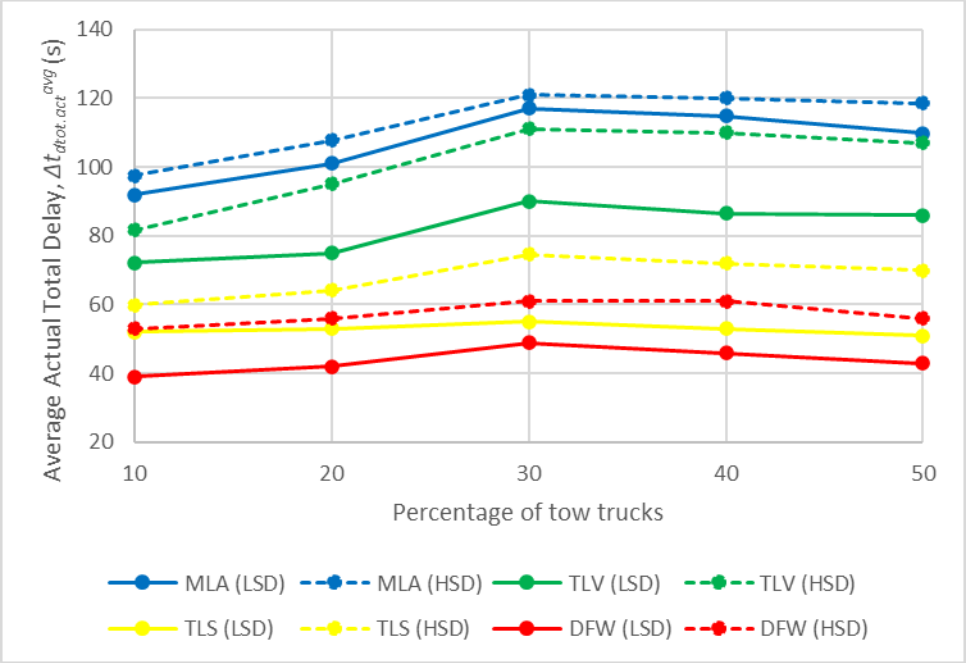


Fig. 70. Average Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ in *Test Scenario 4* using the *Probabilistic Model*.

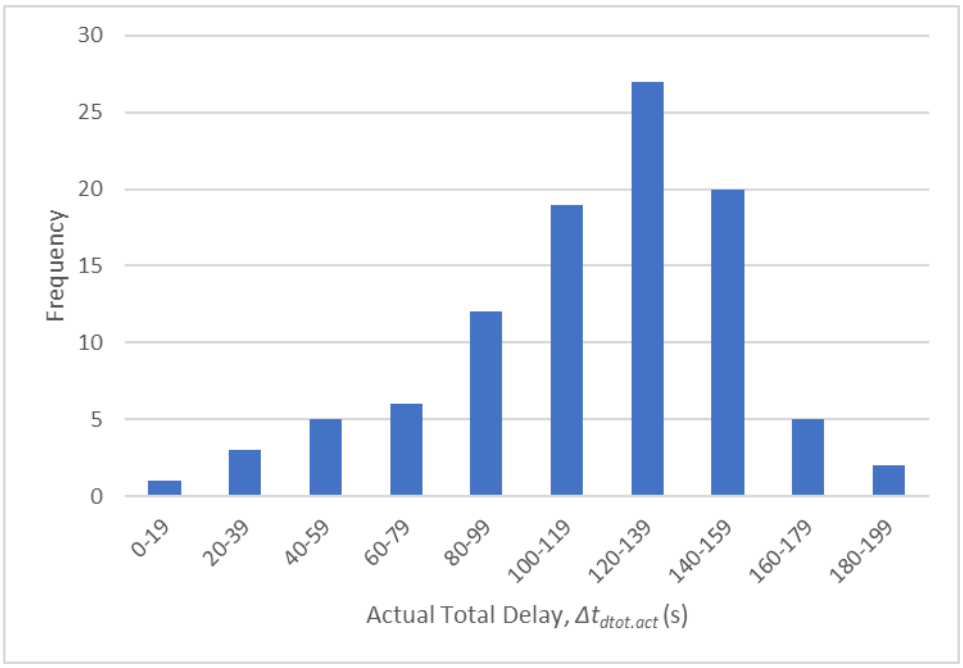


Fig. 71. Distribution of Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ for 100 simulation runs in MLA and for a percentage of tow trucks equal to 30% in *Test Scenario 4* using the *Probabilistic Model* with LSD.

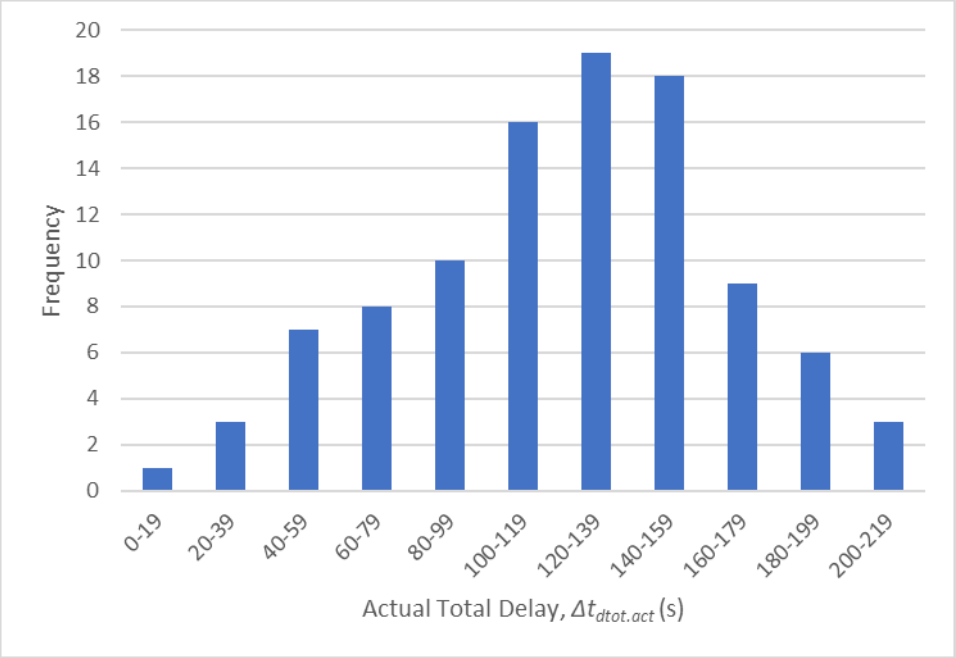


Fig. 72. Distribution of Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ for 100 simulation runs in MLA and for a percentage of tow trucks equal to 30% in *Test Scenario 4* using the *Probabilistic Model* with HSD.

7.5 Results of Test Scenario 5

The results obtained for *Test Scenario 5* are shown in Fig. 73-Fig. 80, calculated using *Dynamic Allocation* with *Fuel-Wise Approach*. Results for each level of traffic and for each ratio of tow trucks to aircraft are presented as average values for ease of reporting.

Fig. 73 and Fig. 74 show TA (%) and Δt_{tow} (%), respectively, for different percentages of tow trucks and for each airport using the *Deterministic Model*. Both percentages show an initial correlation with the number of tow trucks, but then gradually level out (as can be seen for the same metrics in *Test Scenarios 2-4*). In fact, when the proportion of tow trucks reaches around 30%, the tow trucks can handle 90% (or more) of the traffic, meaning that only 10% (or fewer) of aircraft must taxi using their primary engines. For all percentages of tow trucks, the values of these two metrics for *Test Scenario 5* are the highest among *Scenarios 2-5*. This improvement may be attributed to the combined use of the *Fuel-Wise Approach* (in which the algorithm seeks to maximise the number of towed aircraft at the cost of taxi delays) together with the *Dynamic Allocation* (in which the tow trucks are not required to return to a depot after each mission before being allocated to a new one, allowing each tow truck to complete a higher number of missions during the simulation).

The results of ΔF_s^{avg} (kg) and Δt_{dtot}^{avg} (s), which were observed for different percentages of tow trucks at each airport using the *Deterministic Model*, are shown in Fig. 75 and Fig. 76 respectively. These are related to the towing time, Δt_{tow} (%); in fact, for a percentage of tow trucks exceeding approximately 30%, the fuel savings do not significantly improve, whereas

the delays do not increase. However, ΔF_s^{avg} (kg) in this scenario is slightly higher for each airport than the fuel savings obtained in *Test Scenarios 2-4* (as observed in the case of for TA (%) and Δt_{tow} (%)), probably because of the combined use of the *Fuel-Wise Approach* and *Dynamic Allocation*. On the other hand, Δt_{dtot}^{avg} (s) – which is represented by including the values obtained with 0% tow trucks in *Test Scenario 1*, as displayed in Fig. 39 – increases with the percentage of tow trucks and its values are comparable to the ones of *Test Scenario 3* (shown in Fig. 57). This was expected as, while *Test Scenario 2* was carried out using the *Time-Wise Approach*, *Test Scenarios 3* and *5* were carried out using the *Fuel-Wise Approach*, thus favouring fuel savings over delays.

Fig. 77 shows Δt_{ru}^{avg} (%) for different percentages of tow trucks for each airport using the *Deterministic Model*. Similar to the preceding three scenarios, Δt_{ru}^{avg} (%) steadily declines as the percentage of tow trucks increases. However, when compared to *Test Scenarios 2-3*, Δt_{ru}^{avg} (%) shows higher values (above 50%) and slightly higher values when compared to *Test Scenario 4*. The figure, however, never exceeds 60%, underlining once more how crucial battery performance is for tow truck usage.

Fig. 78-Fig. 80 show C_{avg}, M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s), respectively, for different levels of traffic for each airport under test using the *Probabilistic Model* with two different standard deviations (LSD and HSD). As seen in *Test Scenarios 2*-4, the trend of C_{avg} steadily increases with a proportional increase in the percentage of tow trucks (with noticeably larger values for HSD than for LSD), stabilising at percentages greater than 30%. The values for *Test Scenario 5* are only slightly higher than those obtained in *Test Scenario 3* (likely due to the higher frequency of tow truck missions in the simulations), and similar to those observed in *Test Scenario 4*. The plots of M_{avg} (%) and $\Delta t_{dtot.act}^{avg}$ (s) peak when the percentage of tow trucks is equal to 30%, then slowly decline, similar to what was observed in the previous three scenarios. However, the values in this scenario are significantly higher than those observed in *Test Scenarios 2-3* and similar to those observed in *Test Scenario 4*; in fact, with *Dynamic Allocation*, the utilisation of the tow trucks is considerably high and, if a tow truck is immediately reassigned to a new mission, it may accumulate significant delays or fail to attach to the assigned aircraft. These outcomes demonstrate that the algorithm exhibits a larger overall sensitivity when adopting the *Dynamic Allocation* compared to *Test Scenarios 2-3*, indicating that the approach is less robust to uncertainties.

Fig. 81 and Fig. 82 show examples of the histograms obtained whilst computing the number of conflicts, C (s). These plots were obtained for 100 simulation runs in DFW and for percentage of tow trucks equal to 30% using the *Probabilistic Model* with LSD and HSD, respectively. Additionally, Table 31 in Annex 1 shows C_σ, M_σ and $\Delta t_{dtot.act}^{\sigma}$ obtained with the *Probabilistic Model* (both with LSD and HSD) for all the airports and for different percentages of tow trucks, in *Test Scenario 5*.

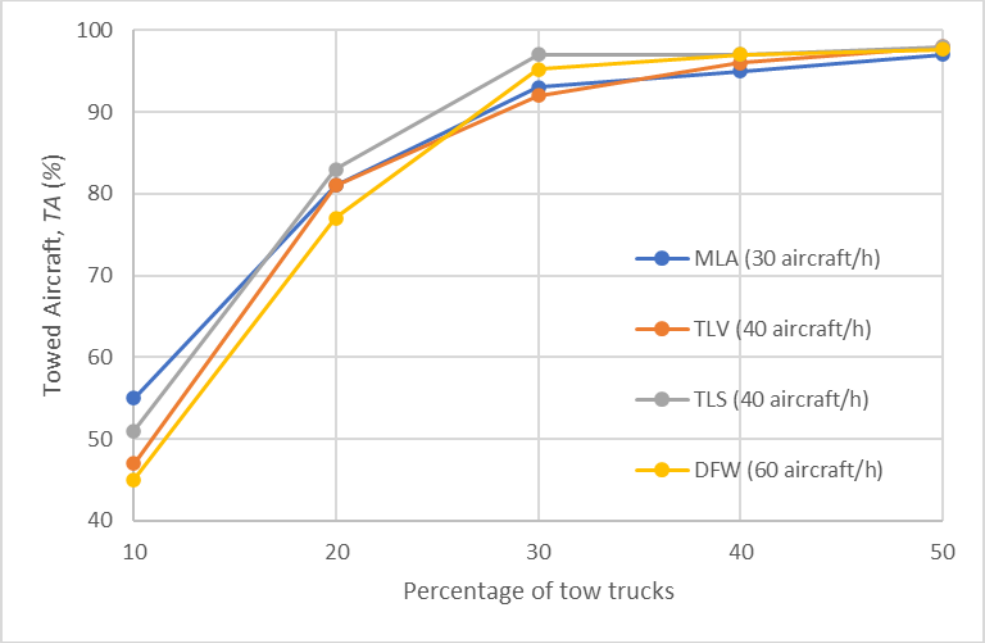


Fig. 73. Towed Aircraft (%), *TA* in *Test Scenario 5* using the *Deterministic Model*.

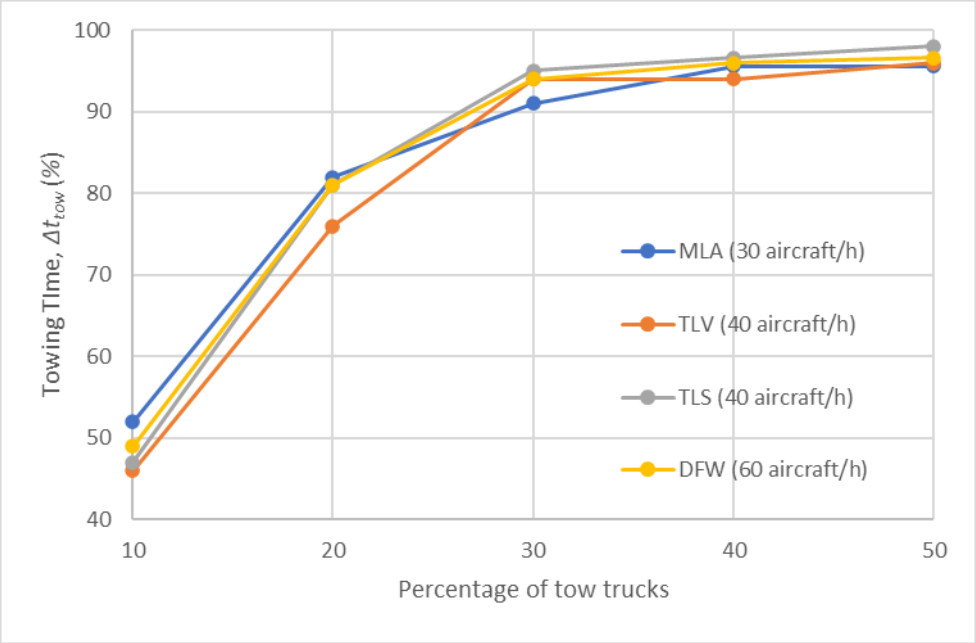


Fig. 74. Towing Time (%), Δt_{tow} in *Test Scenario 5* using the *Deterministic Model*.

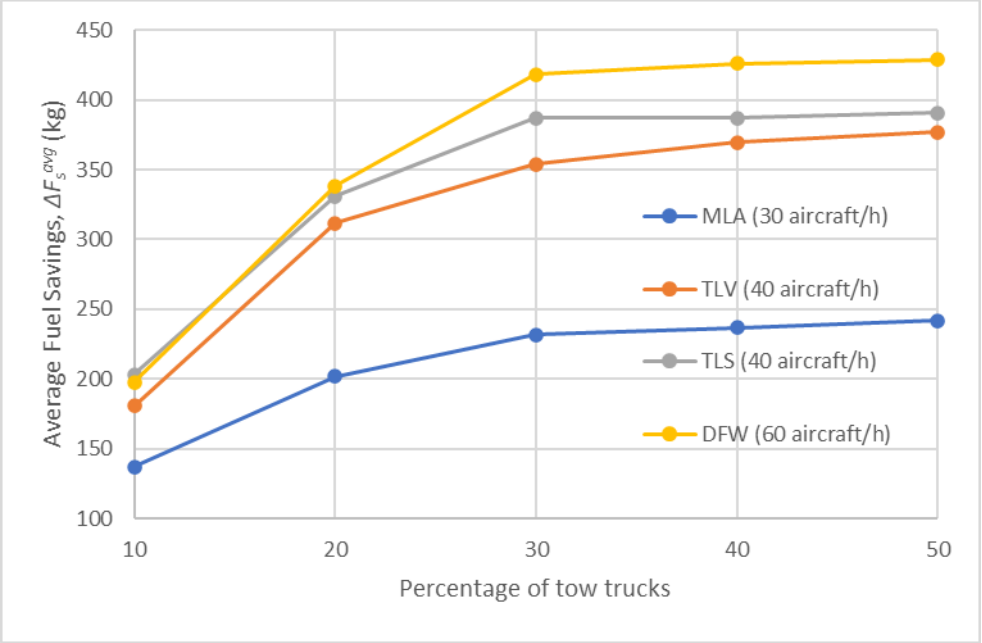


Fig. 75. Average Fuel Savings (kg), ΔF_s^{avg} in *Test Scenario 5* using the *Deterministic Model*.

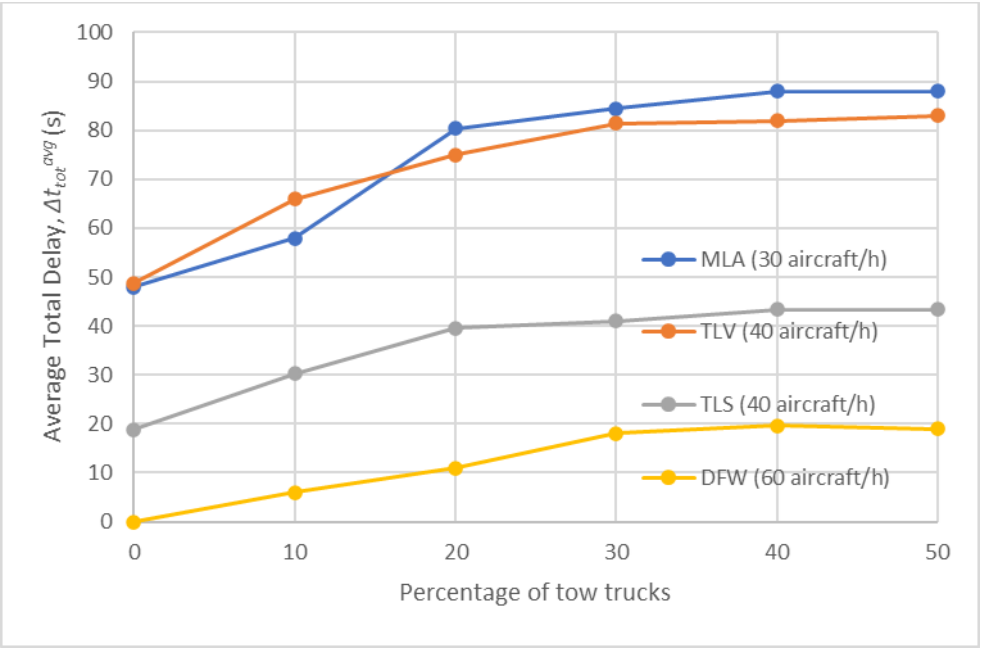


Fig. 76. Average Total Delay (s), Δt_{dtot}^{avg} in *Test Scenario 5* using the *Deterministic Model*.

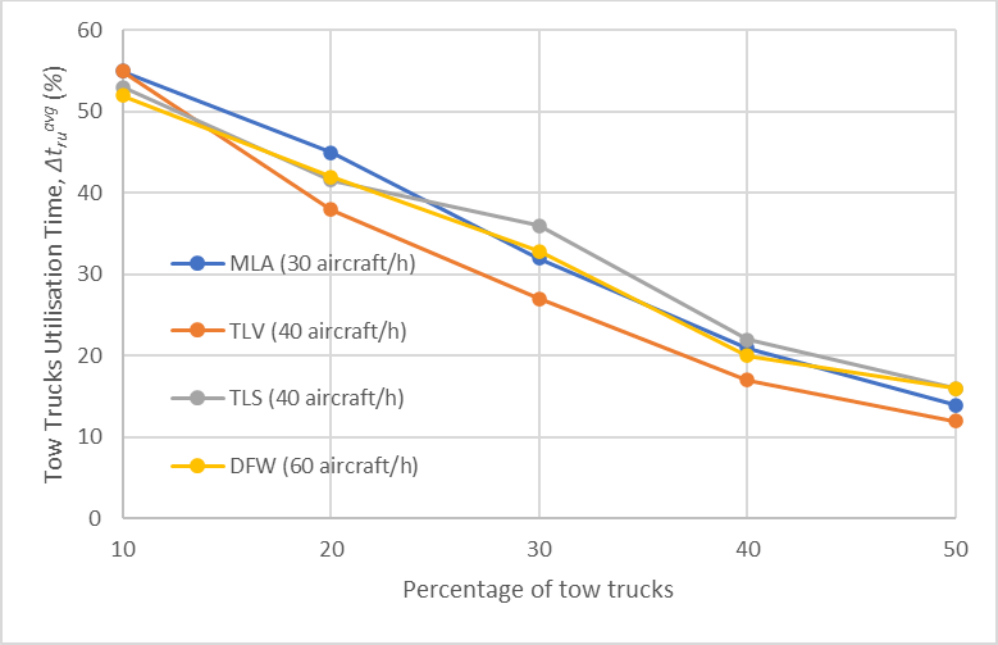


Fig. 77. Tow Truck Utilisation Time (%), Δt_{ru}^{avg} in *Test Scenario 5* using the *Deterministic Model*.

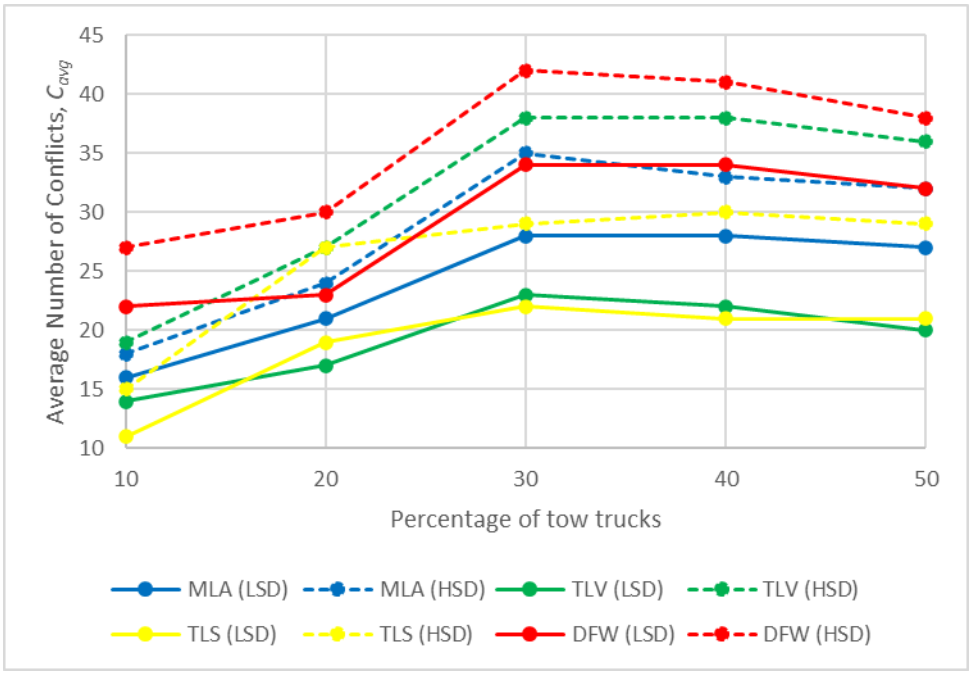


Fig. 78. Average Number of Conflicts, C_{avg} in *Test Scenario 5* using the *Probabilistic Model*.

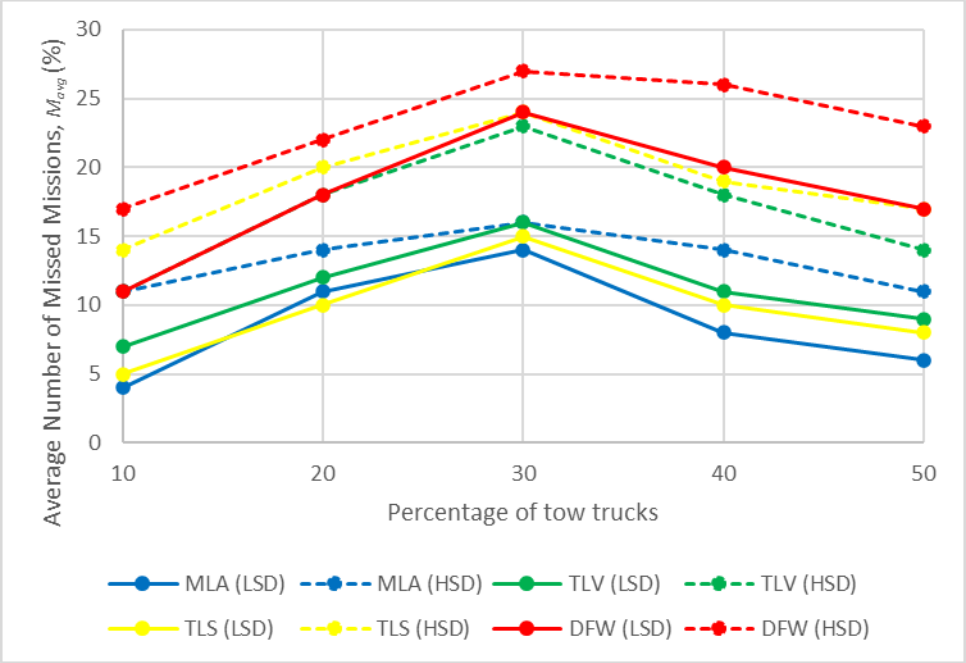


Fig. 79. Average Number of Missed Missions (%), M_{avg} in *Test Scenario 5* using the *Probabilistic Model*.

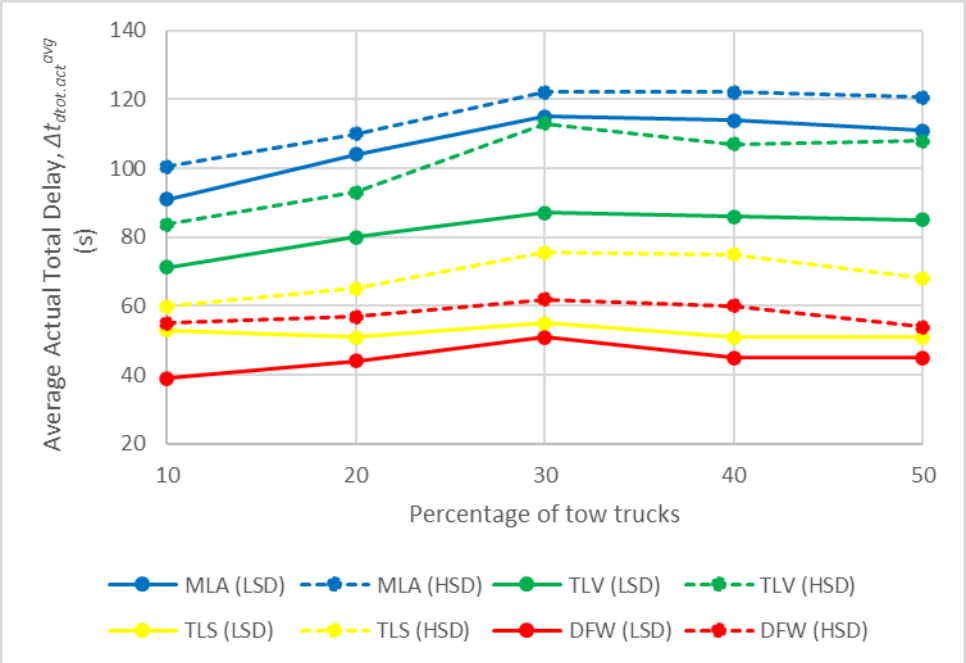


Fig. 80. Average Actual Total Delay (s), $\Delta t_{dtot.act}^{avg}$ in *Test Scenario 5* using the *Probabilistic Model*.

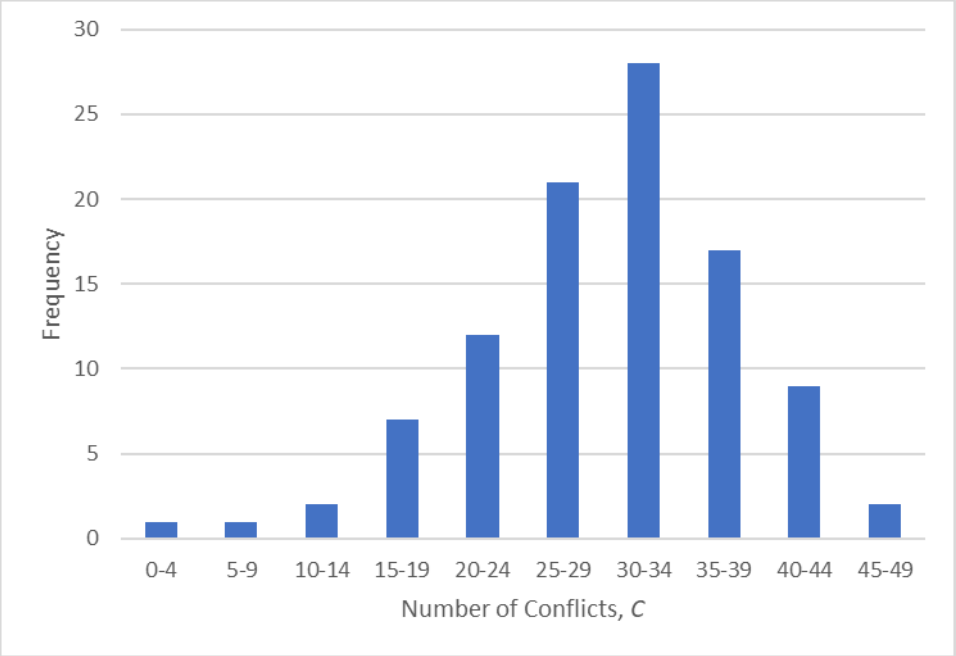


Fig. 81. Distribution of Number of Conflicts, *C* for 100 simulation runs in DFW and for a percentage of tow trucks equal to 30% in *Test Scenario 5* using the *Probabilistic Model* with LSD.

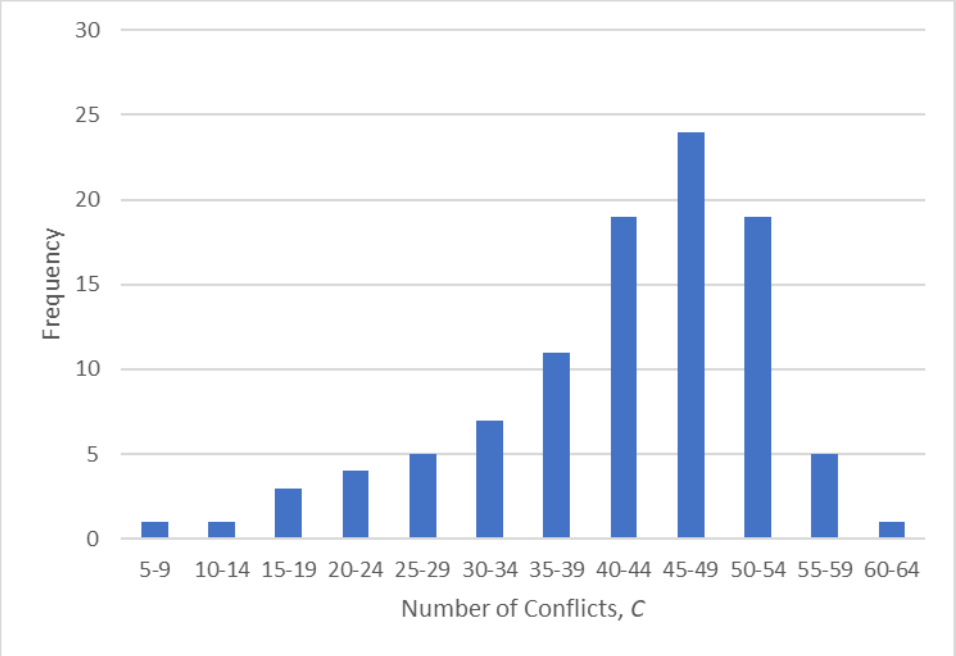


Fig. 82. Distribution of Number of Conflicts, *C* for 100 simulation runs in DFW and for a percentage of tow trucks equal to 30% in *Test Scenario 5* using the *Probabilistic Model* with HSD.

7.6 Results of Test Scenario 6

The results obtained for *Test Scenario 6* are shown in Table 26, calculated using *Dynamic Allocation* with *Time-Wise Approach* and *Deterministic Model*. The purpose of this test scenario was to evaluate the relationship between tow truck performance and battery performance. As can be observed from Table 26, for lower discharge rates, TA (%), Δt_{tow} (%), Δt_{ru}^{avg} (%) and ΔF_s^{avg} (kg) have higher values; in particular, TA (%) and Δt_{tow} (%) exceed 80%, suggesting a consistent improvement of the tow truck performances when compared to the base scenario (i.e. nominal values of r_{bdh} and r_{bdl}). On the other hand, an increase in discharge rates results in a sharp decline in the value of the metrics. This could be because the tow trucks are frequently not assigned to the aircraft due to their low battery level.

Large variations in the metrics for relatively small percentage changes in discharge rates demonstrate the importance of battery performance for tow truck performance and for determining how many tow trucks should be deployed to meet demand corresponding to various traffic levels.

Table 26. Relationship between tow truck performance and battery performance in *Test Scenario 6*.

Discharge rates percentage variation (%/min)	r_{bdh} (%/min)	r_{bdl} (%/min)	TA (%)	Δt_{tow} (%)	Δt_{ru}^{avg} (%)	ΔF_s^{avg} (kg)
-0.75	1.25	0.25	85	81	45	337
-0.50	1.50	0.50	83	79	44	329
-0.25	1.75	0.75	82	76	42	325
0	2.00	1.00	78	73	40	315
+0.25	2.25	1.25	75	71	39	305
+0.50	2.50	1.50	72	67	37	288
+0.75	2.75	1.75	65	56	31	258
+1.00	3.00	2.00	55	47	27	221

7.7 Results of Test Scenario 7

The results of *Test Scenario 7* are shown in Table 27. These were obtained using *Dynamic Allocation* with *Time-Wise Approach* and *Deterministic Model*. As can be observed from Table 27, the performance metrics are almost identical to those obtained in *Test Scenario 6* with a 0%/min discharge rate percentage variation. These results demonstrate that the algorithm remains effective over a complete day of aircraft ground operations.

Table 27. Results for *Test Scenario 7*.

TA (%)	Δt_{tow} (%)	Δt_{ru}^{avg} (%)	ΔF_s^{avg} (Kg)
77	72	40	317

7.8 Discussion

Seven scenarios were used to evaluate the algorithm – one of which examined the performance of the *Flight Scheduler* (*Test Scenario 1*), while the other five examined the performance of the *Tug Dispatcher* under various conditions (*Test Scenarios 2-6*). One of the key results of *Test Scenario 1* is represented by the average delay of the aircraft. Generally, the algorithm prefers increasing the waiting time at the stand (for departing flights) or next to a runway (for arriving flights) as this results in fewer delays than selecting a different (longer) route to the destination, so in fact, the delays accumulated before starting to taxi are typically higher than the delays accumulated while taxiing. This is also beneficial for fuel consumption, because departing aircraft can start their engines later, while arriving aircraft use less fuel by waiting instead of taxiing.

It is interesting to examine how the *Tug Dispatcher* performs in *Test Scenarios 2* and *4*, which uses the *Time-Wise Approach*, and *Test Scenarios 3* and *5*, which use the *Fuel-Wise Approach*. With the first approach, the algorithm attempts to minimise delays, whereas, with the second approach it aims to maximise fuel savings; the variation is modest both in terms of delays and fuel savings but, when added up across all flights, these variations might make a significant difference. The small average difference between the two approaches may be explained by the fact that, in essence, minimising delays or maximising fuel savings will both result in an overall fuel saving. As a result, optimising one of these two objectives will inherently have a positive impact on the other.

Another comparison that can be made is between the *Tug Dispatcher* performance in *Test Scenarios 2* and *3*, with *Static Allocation*, versus that of *Test Scenarios 4* and *5*, with *Dynamic Allocation*. The algorithm is able to assign tow trucks to a higher number of flights because of *Dynamic Allocation*, which allows a tow truck that has just completed a mission to be re-assigned straightaway to another mission instead of being required to first return to a depot; as a result, in *Test Scenarios 4* and *5* the algorithm performs better for a wide range of metrics, such as the percentage of towed aircraft, the tow truck utilisation time, and the fuel savings. However, when tested with the *Probabilistic Model*, the system performed poorly in *Test Scenarios 4* and *5*, particularly in terms of delays and missed attachments. This can be attributed to the fact that with the *Dynamic Allocation* the utilisation of tow trucks is further maximised in these scenarios; therefore, adding systemic uncertainty has a knock-on impact that interferes with the intended tow truck missions. This shows that the *Dynamic Allocation*

approach does not always have a favourable impact on the algorithm, especially when there are large margins of uncertainty.

In *Test Scenarios 2-5*, it was observed that the tow truck utilisation never surpasses 60% of the total simulation time. This may be partially due to the unavailability of conflict-free routes for the tow trucks in situations of high volumes of traffic, but another factor is the battery performance, since tow trucks need to occasionally recharge their batteries periodically. Furthermore, in *Test Scenario 6* it was observed that relatively small variations in the tow truck battery discharge rates have a significant impact on the tow truck performance. These results demonstrate the importance of battery performance for tow truck-based electric taxi operations. In addition to deploying rapid charging tow trucks, other ways to increase the efficiency of the tow trucks include: expanding the number of depots, placing them in strategic locations around the airport, and increasing the number of charging points (i.e. parking slots) in each depot. Finally, *Test Scenario 7* validates that a 6-hour simulation effectively represents an entire day of air traffic scenarios.

The majority of the results obtained in the test scenarios vary depending on the airport being tested. For volumes of traffic proportional to an airport's dimensions, larger airports with an extensive network of taxiways perform better than smaller airports with fewer alternatives to taxi, both in absolute terms and when compared to ideal (shortest) paths. This could be due to the fact that a limited number of taxiways is more likely to lead to bottlenecks and reduce the number of tow truck-based operations. This result suggests that the proposed algorithm might perform better for larger airports, whereas alternative algorithms or human intervention might be needed to improve performance for smaller airports.

8 Conclusions

After presenting and discussing the test results, this chapter draws conclusions by summarising the work, identifying the scientific contributions of this work, and indicating potential topics for further research.

8.1 Summary of Work

This work introduced an algorithm to automate and optimise taxi operations using autonomous tow trucks. The algorithm generates conflict-free routes for aircraft and tow trucks at a strategic level; reduces taxi-related delays and fuel consumption; and makes maximum use of tow trucks for taxi operations. Moreover, it can be tuned to enhance certain desirable performance aspects.

First, a literature review was carried out on related work in order to identify a research gap and more clearly describe the goal of the PhD research. Then, an airport environment for taxi operations with tow trucks was set up; an algorithm was designed and implemented in line with the research objectives; and a simulator was developed to evaluate the performance of the algorithm in the context of airport taxi operations with tow trucks. A number of performance metrics were defined and a large number of simulations were carried out for different algorithm settings and test scenarios. In particular, two different models were implemented to simulate the vehicle motion: a deterministic model and a probabilistic model. The latter attempts to model real-world uncertainties associated with taxi operations. Finally, the results were presented and discussed.

The results demonstrate that the algorithm is successful in: limiting delays with respect to the flight schedule (even for high volumes of traffic); exploiting the use of tow trucks; and maximising fuel savings. Additionally, the results indicate that the algorithm can be tuned to further improve some performance aspects over others.

8.2 Scientific Contributions

The scientific contributions arising from this work are the following:

- A strategic, centralised algorithm for tow-truck-based taxiing (for departing and arriving aircraft). This algorithm is capable of generating conflict-free routes both for aircraft and tow trucks, while accounting for taxi delays, fuel consumption and tow truck utilisation. Moreover, it can accommodate multiple active runways, different levels of traffic and a varying number of tow trucks. Also, it takes into account the battery charge of the tow trucks, the location of the depots, and the number of

available charging stations in each depot, thus allowing for a more realistic estimation of the number of tow trucks that may be required to complete the taxi operations;

- A fully autonomous solution that does not require intervention from ATCOs to issue clearances or to solve conflicts (as it guarantees conflict-free routes by design); therefore it would enable the introduction of tow trucks for taxi operations without significantly incrementing the ATCO workload;
- A system that can be tuned to prioritise taxi delays over fuel consumption (*Time-Wise Approach*) or to prioritise fuel consumption over taxi delays (*Fuel-Wise Approach*) as required. In addition, the algorithm can be adjusted so that tow trucks can be assigned to aircraft by either using *Static Allocation*, in which case they must be parked in a depot, or using *Dynamic Allocation*, in which case they can be assigned to an aircraft from anywhere in the airport;
- The scalability, applicability of the algorithm's solutions to different taxiing environments, and its and robustness to unforeseen events, guaranteed by a large number of tests in a range of scenarios which include four airports of varying sizes and geometries; different active runways at each airport; varying numbers of aircraft per hour; different ratios of tow trucks to aircraft per hour (with the number of charging points per depot scaled in accordance with the number of tow trucks); and varying degrees of uncertainty (such as different vehicle speeds and unexpected delays) during the simulations;
- The assurance that, in the absence of uncertainty, the solutions provided are always conflict-free, even in conditions of high volumes of traffic. The test results show that nearly 70% of flights required short delays of up to 3 minutes to ensure that adequate traffic separation is maintained at all times. Furthermore, a fleet of tow trucks with a size equal to 30% of the number of aircraft per hour can tow over 90% of the aircraft. This knowledge can be used to determine how many tow trucks would be needed to cater for a certain level of traffic and for a particular type of airport, as well as the amount of fuel that could potentially be saved, in order to find a suitable trade-off between the expenses associated with a fleet of tow trucks, and the benefits of tow truck-based taxiing operations.

8.3 Practical Considerations for the Real-World Implementation of the System

An assumption of this study is that departing aircraft would not require engine warm-up, a premise that diverges from practical realities. In actuality, tow trucks would need to disconnect from the aircraft at a certain distance from the runway, enabling the necessary engine warm-ups and checks. This adjustment necessitates modifications to both the settings of the algorithm, as well as to the airport infrastructure. Specifically, it would require the

establishment of dedicated bays along the taxiways. These bays would serve two primary functions: facilitating the detachment of tow trucks and accommodating the aircraft during their engine warm-up and pre-departure check procedures. Incorporating such features aligning the algorithm's operations with the real-world needs of aircraft preparation before take-off.

Moreover, the operational procedures for using tow trucks in aircraft taxiing must be clearly defined and standardized across the industry. These procedures would include guidelines for maximum taxi speeds to ensure safety, detailed attachment and detachment protocols that synchronize with the aircraft's readiness for take-off, and specific emergency procedures in case of mechanical failure of either the tow truck or the aircraft. Training for both airline and airport personnel in these new procedures will be critical to ensure smooth operations and quick responses to any irregular situations.

A key logistical consideration in implementing tow truck-based taxiing is the strategic acquisition of a tow truck fleet tailored to the specific needs of each airport. The size and composition of this fleet are influenced by several critical factors. These include the size and layout of the airport, the volume of aircraft movements per hour, and the typical size and weight of the aircraft serviced. Additionally, economic factors play a significant role. These encompass the initial cost of purchasing and maintaining the tow trucks and their depots, the potential savings from reduced fuel consumption, and the financial benefits and tax incentives associated with adhering to environmentally-friendly policies. These considerations vary significantly depending on the specific airport and country in which they are implemented, and must be carefully evaluated by all stakeholders involved, including airport operators, airline companies, and third-party service providers.

Additionally, modifications to airport infrastructure are essential to support the implementation of the system, including constructing depots equipped with electric charging points to accommodate the fleet of tow trucks. Furthermore, adequate road infrastructure must be developed to connect the depots with the rest of the airport efficiently. The optimal number and strategic placement of the depots will vary based on each airport's specific operational demands and spatial layout, ensuring that tow trucks are readily available when needed and can move efficiently throughout the airport.

Finally, the system should integrate ATC interventions, thereby enabling the provision of clearances and the facilitation of route adjustments as directed by ATC. This necessitates the development of an HMI that allows ATC personnel to seamlessly interact with the algorithm. Such an integration will ensure that the automated system not only complements but also enhances the decision-making capabilities of ATCOs, ensuring a more dynamic, responsive, and safe airfield operational environment. Incorporating ATC interventions will bridge the gap between the current and proposed systems, showcasing a comprehensive approach that leverages automation while respecting the indispensable human oversight in air traffic management.

8.4 Potential Areas of Future Work

The proposed algorithm offers strategic solutions in which route and tug assignment are pre-computed ahead of time. In practice, however, this approach might not be sufficient, as taxi operations are subject to a certain degree of uncertainty, which can disrupt the decision made by the strategic algorithm. This was confirmed by the simulations which were carried out using a probabilistic model, which resulted in potential conflicts, missed attachments between aircraft and tow trucks, and additional delays. Hence, future work should focus on adopting – in addition to strategic solutions – a number of tactical solutions that can operate in real-time and respond to unexpected occurrences, such as, for instance, the crossing of the taxiways or the runway by other ground vehicles like coaches, fuel trucks, and other service equipment.

In this work, a constant velocity vehicle model was used, and the same velocity was applied to all vehicles (i.e. aircraft and tow trucks). This model could be improved in further work by modelling acceleration and deceleration, setting a higher velocity along straightaways and a lower velocity during turns, and taking the vehicle's turn radius into account. Moreover, the velocity may differ between tow trucks and aircraft, between different types of aircraft and between different types of tow trucks.

In the proposed system, only one type of tow truck was implemented, which was assumed capable of towing all types of aircraft. For future enhancements, the system should be designed to incorporate different types of tow trucks, each tailored to handle specific categories of aircraft. This diversification would allow for more specialised and efficient operations, catering to the unique requirements and characteristics of different aircraft models.

Furthermore, in this work, the tow truck battery discharge and recharge rates were updated on the basis of three predetermined parameters and were assumed to be constant. In the future, a more realistic battery charging and discharging model could be implemented, and the algorithm could be tested for different values of battery parameters in order to better quantify the impact of battery performance on algorithm outcomes, and the number of tow trucks needed at various airports to cope with different volumes of traffic.

Also, a total of four tow truck depots were implemented for each airport considered in this work and a fixed ratio of charging points to tow trucks was used in each simulation. In future work, the system could be tested with different values of these parameters, in order to assess their impact on the tow truck-based taxiing operations.

Furthermore, other uncertainties could be explored during testing in future work, for instance in the attachment and detachment time; in the exit point from the runway (for landing aircraft); in the entry point to a runway (for departing aircraft); in the battery discharge and recharge rates; and in the acceleration and deceleration of the vehicles (if implemented with a new motion model).

Moreover, the future work should include the development of an HMI specifically designed to ensure that the ATCOs maintain comprehensive control throughout the operation. This interface should allow the ATCOs to seamlessly integrate the tow truck-based taxiing system into their existing workflows, providing them with real-time data, alerts and controls. The HMI should be intuitive and responsive, enabling quick adjustments to taxi routes and schedules based on airport conditions and ATCO decisions. This development will play a crucial role in maintaining safety and efficiency, ensuring that human oversight complements automated processes.

Finally, it would be paramount to validate the proposed solution in a real-world environment by conducting extensive trials with end users such as pilots and ATCOs to gather feedback and identify any practical challenges or areas for improvement. These trials should simulate various operational scenarios to test the robustness of the system under different conditions, including emergency situations. Gathering direct input from the end users will not only enhance the functionality of the system but also ensure that it meets the rigorous demands of actual airport operations. This step would be crucial for gaining broad acceptance and ensuring the effective deployment of the system in real airport environments.

References

[1] "www.eurocontrol.int," Eurocontrol, 10 March 2023. [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-data-snapshot-40-taxi-times#:~:text=Typically%2C%20flying%20from%20one%20of,time%20of%20over%2020%20minutes. [Accessed 12 May 2023].

[2] I. C. A. Organization, "Consolidated statement of continuing ICAO policies and practices related to environmental protection – Climate change," 2019.

[3] M. Zhang, Q. Huang, S. Liu and H. Li, "Assessment Method of Fuel Consumption and Emissions of Aircraft during Taxiing on Airport Surface under Given Meteorological Conditions," College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2019.

[4] O. T. Pleter and C. E. Constantinescu, "A review of flight trajectory optimisations," *The Journal of Navigation,* 23 May 2022.

[5] C. Celis, V. Sethi, R. Singh and P. Pilidis, "On Optimisation of Environmentally Friendly Aircraft Engine Cycles," *Journal of Air Transport Management,* 2015.

[6] "skybrary.aero," SkyBrain, 2023. [Online]. Available: https://skybrary.aero/articles/surface-movement-radar. [Accessed 4 April 2024].

[7] "Eurocontrol," 2023. [Online]. Available: https://www.eurocontrol.int/service/advanced-surface-movement-guidance-and-control-system. [Accessed 4 April 2024].

[8] "skybrary.aero," SkyBrain, 2023. [Online]. Available: https://skybrary.aero/articles/electronic-flight-bag-efb. [Accessed 4 April 2024].

[9] T. Nikoleris, G. Gupta and M. Kistler, "Detailed estimation of fuel consumption and emissions during aircraft taxi operations at Dallas/Fort Worth International Airport," *Transportation Research Part D, Elsevier,* 2011.

[10] E. Fleuti and S. Maraini, "Taxi-Emissions at Zurich Airport," 2017.

[11] N. Dzikus, J. Fuchte, A. Lau and V. Gollnick, "Potential for Fuel Reduction through Electric Taxiing," in *11th AIAA Aviat. Technol. Integr. Oper. Conf.*, 2011.

[12] Y. Jiang, Z. Liu, Z. Hu and H. Xhang, “A priority-based conflict resolution strategy for airport surface traffic considering suboptimal alternative paths,” Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, USA, 2020.

[13] “op.europa.eu,” European Commission, 2011. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/296a9bd7-fef9-4ae8-82c4-a21ff48be673. [Accessed 15 March 2023].

[14] “ec.europa.eu,” European Commission, 2020. [Online]. Available: https://ec.europa.eu/clima/policies/transport/aviation_en. [Accessed 15 March 2023].

[15] “www.iata.org,” International Air Transport Association, December 2022. [Online]. Available: https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet---fuel/. [Accessed 4 April 2023].

[16] “www.fitchratings.com,” Fitch Ratings, 25 February 2022. [Online]. Available: https://www.fitchratings.com/research/corporate-finance/oil-price-shock-is-a-material-risk-for-us-european-airlines-25-02-2022. [Accessed 4 April 2023].

[17] “www.statista.com,” Statista, 2023. [Online]. Available: https://www.statista.com/statistics/591285/aviation-industry-fuel-cost/. [Accessed 4 April 2023].

[18] “https://www.sesarju.eu/,” Single European Sky ATM Research Joint Undertaking, 2019. [Online]. Available: https://www.sesarju.eu/perfectflight. [Accessed 15 March 2023].

[19] “www.aviationhunt.com,” Aviation Hunt, 31 July 2022. [Online]. Available: https://www.aviationhunt.com/single-engine-taxi/#:~:text=Single%20engine%20taxiing%20(SET)%2C,)%20and%20hydrocarbon%20(HC).. [Accessed 4 April 2023].

[20] G. S. Koudis, S. J. Hu, A. Majumdar, W. Y. Ochieng and M. E. J. Stettler, “The impact of single engine taxiing on aircraft fuel consumption and pollutant emissions,” *The Aeronautical Journal,* 2018.

[21] “www.icao.int,” International Civil Aviation Organization, 21 March 2023. [Online]. Available: https://www.icao.int/environmental-protection/pages/SAF.aspx. [Accessed 4 April 2023].

[22] "www.energy.gov," Department of Energy (USA), 2023. [Online]. Available: https://www.energy.gov/eere/bioenergy/sustainable-aviation-fuels. [Accessed 4 April 2023].

[23] J. Bosch, S. de Jong, R. Hoefnagels and R. Slade, "Aviation biofuels: strategically important, technically achievable, tough to deliver," *Grantham Institute Briefing paper No 23,* November 2017.

[24] M. G. W. Groot and P. C. Roling, "The Potential Impact of Electric Aircraft Taxiing. A Probabilistic Analysis and Fleet Assignment Optimization," in *AIAA Aviation*, 2022.

[25] WheelTug, [Online]. Available: https://www.wheeltug.com/. [Accessed 15 March 2023].

[26] "tec.ieee.org," IEEE TRansportation Electrification Community, 2014. [Online]. Available: https://tec.ieee.org/newsletter/march-april-2014/electric-green-taxiing-system-egts-for-aircraft. [Accessed 15 March 2023].

[27] "aviationweek.com," Aviation Week, 2013. [Online]. Available: https://aviationweek.com/air-transport/honeywellsafran-joint-venture-tests-electric-taxiing. [Accessed 15 March 2023].

[28] "magazine.groundhandling.com," Ground Handling International, 22 February 2023. [Online]. Available: https://magazine.groundhandling.com/news/vueling-becomes-wheeltugs-first-european-lcc-customer/. [Accessed 16 March 2023].

[29] M. Soltani, "Electrification of Airport Operations: Electric Powered Tow-Truck Utilization in Taxiing Operations," Department of Mechanical and Industrial Engineering, Montreal, Quebec, Canada, 2019.

[30] A. L. Salihu, "Impact of On-Ground Taxiing with Electric Powered Tow-Trucks on Congestion, Cost, and Carbon Emissions at Montréal-Trudeau International Airport," Concordia University, Montreal, Quebec, Canada, 2020.

[31] "TaxiBot," 2020. [Online]. Available: https://www.taxibot-international.com/. [Accessed 15 March 2023].

[32] "https://www.airport-technology.com/," Airport Technology, 19 February 2015. [Online]. Available: https://www.airport-technology.com/news/newsfrankfurt-airport-deploys-fuel-saving-taxibot-for-taxiing-aircraft-4517031/. [Accessed 15 March 2023].

[33] R. Camilleri and A. Batra, “Assessing the environmental impact of aircraft taxiing technologies,” in *ICAS*, Shanghai, 2020.

[34] “www.timesofisrael.com,” The Times of Israel, 20 February 2015. [Online]. Available: https://www.timesofisrael.com/israels-fuel-saving-taxibot-deployed-in-german-airport/. [Accessed 16 March 2023].

[35] “maltats.com,” MATS, [Online]. Available: https://maltats.com/about/. [Accessed 14 March 2023].

[36] S. Zaninotto, J. Gauci, G. Farrugia and J. Debattista, “Design of a Human-in-the-Loop Aircraft Taxi Optimisation System Using Autonomous Tow Trucks,” in *AIAA*, Dallas, Texas, USA, 2019.

[37] S. Zaninotto, J. Gauci and B. Zammit, “A Testbed for Performance Analysis of Algorithms for Engineless Taxiing with Autonomous Tow Trucks,” in *DASC*, San Antonio, Texas, USA, 2021.

[38] “2021.dasconline.org,” DASC Conference 2021, 2021. [Online]. Available: https://2021.dasconline.org/best-paper-awards/. [Accessed 14 March 2023].

[39] S. Zaninotto, J. Gauci and B. Zammit, “An Engineless Taxi Operations System Using Battery-Operated Autonomous Tow Trucks,” in *ICAS*, Stockholm, Sweden, 2022.

[40] “www.sesarju.eu,” Sesar Joint Undertaking, 1 June 2013. [Online]. Available: https://www.sesarju.eu/newsroom/brochures-publications/mota-modern-taxing. [Accessed 19 April 2023].

[41] F. Lancelot and M. Causse, “Human-in-the-Loop Multi-agent Approach for Airport Taxiing Operations,” 2015.

[42] Z. Chua, M. Cousy, M. Causse and F. Lancelot, “Initial Assessment of the Impact of Modern Taxiing Techniques on Airport Ground Control,” in *The International Conference*, 2016.

[43] Z. Chua, F. Andre and M. Cousy, “Development of an ATC Tower Simulator to Simulate Ground Operations,” in *AIAA Modeling and Simulation Technologies Conference*, Dallas, Texas, USA, 2015.

[44] “http://ihmaero.recherche.enac.fr/index.php?article3/mota-project-summary,” 15 October 2022. [Online].

[45] R. Morris, M. L. Chang, R. Archer, E. V. I. Cross, S. Thompson, J. L. Franke, R. C. Garrett, W. Malik, K. McGuire and G. Hermann, “Self-Driving Aircraft Towing Vehicles: A Preliminary Report,” Austin, Texas, USA, 2015.

[46] R. Morris, M. L. Chang, R. Archer, E. V. Cross II, S. Thompson, J. Franke, R. C. Garrett, C. Pasareanu, W. Malik and G. Hemann, “SafeTug Semi-Autonomous Aircraft Towing Vehicles,” NASA, 2016.

[47] G. Sirigu, M. Battipede, J.-P. Clarke and P. Gili, “A fleet management algorithm for automatic taxi operations,” in *International Conference on Research in Air Transportation*, Philadelphia, Pennsylvania, USA, 2016.

[48] G. Sirigu, M. Cassaro, M. Battipede and P. Gili, “A Route Selection Problem Applied to Auto-Piloted Aircraft Tugs,” in *WSEAS Transactions on Electronics, International Conference on Applied and Theoretical Mechanics*, Venice, Italy, 2017.

[49] G. Sirigu, M. Cassaro, M. Battipede and P. Gili, “Autonomous taxi operations: algorithms for the solution of the routing problem,” in *AIAA Information Systems-AIAA Infotech @ Aerospace*, Kissimmee, Florida, USA, 2018.

[50] P. H. Abreu and E. Oliveira, “Comparing a Centralized and Decentralized Multi-Agent Approaches to Air Traffic Control,” 2014.

[51] Y. J. Chiang, J. T. Klotowski, C. Lee and J. S. B. Mitchell, “Geometric algorithms for conflict detection/resolution in air traffic management,” in *36th IEEE Conference on Decision and Control*, San Diego, California, USA.

[52] X. Wang, H. Peng, J. Liu, X. Dong, X. Zhao and C. Lu, “Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck,” *Defence Technology,* February 2022.

[53] H. Udluft, “Decentralization in Air Transportation,” Delft, Netherland, 2017.

[54] A. Vemula, K. Muelling and J. Oh, “Path Planning in Dynamic Environments with Adaptive Dimensionality,” Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

[55] E. Gawrilow, E. Köhler, R. H. Möhring and B. Stenzel, “Dynamic Routing of Automated Guided Vehicles in Real-time,” Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2007.

[56] E. Gawrilow, M. Klimm, R. H. Möhring and B. Stenzel, “Conflict-free vehicle routing. Load balancing and deadlock prevention,” Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2012.

[57] M. Zhang, Q. Huang and H. Li, “Multi-Objective Optimization of Aircraft Taxiing on the Airport Surface with Consideration to Taxiing Conflicts and the Airport Environment,” College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2019.

[58] L. P. Rosa, D. M. Ferreira, L. L. B. V. Cruciol, L. Weigang and D. X. Jun, “Genetic algorithms for management of taxi scheduling,” TransLab, University of Brasilia, Brasilia, Brazil.

[59] J. Smeltink, P. de Waal and R. van der Mei, “An Optimisation Model for Airport Taxi Scheduling,” in *in Proceedings of the INFORMS Annual Meeting*, Denver, Colorado, USA, 2004.

[60] P. C. Roling and H. G. Visser, “Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming,” *International Journal of Aerospace Engineering,* 2008.

[61] J. Montoya, W. Zachary, S. Rathinam and W. Malik, “A Mixed Integer Linear Program for Solving a Multiple Route Taxi Scheduling Problem,” NASA Ames Research Center, Moffett Field, California, USA, 2010.

[62] G. Clare and A. Richards, “Optimization of Taxiway Routing and Runway Scheduling,” *IEEE Transactions on Intelligent Transportation Systems,* 2011.

[63] L. Adacher, M. Flamini and E. Romano, “Rerouting Algorithms Solving the Air Traffic Congestion,” Università degli studi Roma Tre, Rome, Italy.

[64] M. Philips and M. Likhachev, “SIPP: Safe Interval Path Planning for Dynamic Environments,” Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

[65] K. J. C. Fransen, J. A. W. M. van Eekelen, A. Pogromsky, M. A. A. Boon and I. J. B. F. Adan, “A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems,” Eindhoven University of Technology, Eindhoven, Netherlands.

[66] J. van den Berg, D. Ferguson and J. Kuffner, “Anytime Path Planning and Replanning in Dynamic Environments,” Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht, Netherlands.

[67] N. Zhao, N. Li, Y. Sun and L. Zhang, “Research on Aircraft Surface Taxi Path Planning and Conflict Detection and Resolution,” *Journal of Advanced Transportation,* 2021.

[68] L. Li and G. Jiawei, “Research on aircraft taxiing path optimization based on digraph model and Dijkstra algorithm,” in *IOP Conference Series Materials Science and Engineering*, 2021.

[69] M. F. Ibrahim, A. Z. A. Bakar and A. Hussain, “Genetic Algorithm-based Robot Path Planning,” Department of Electrical, Electronics and Systems Engineering, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia.

[70] Eurocontrol, “www.eurocontol.int,” [Online]. Available: https://www.eurocontrol.int/service/estimated-block-time-update-service. [Accessed 19 January 2023].

[71] ICAO, “www.icao.int,” International Civil Aviation Organisation, 3 April 2015. [Online]. Available: https://www.icao.int/APAC/Meetings/2015%20ATFM_SG5/WP21%20ATFM%20Terminology%20and%20Communications.pdf. [Accessed 12 February 2023].

[72] Studyflight.com, “www.studyflight.com,” [Online]. Available: https://studyflying.com/eta-estimate-time-of-arrival/#:~:text=What's%20ETA%3F,will%20arrive%20over%20the%20aerodrome.. [Accessed 19 January 2023].

[73] “www.skybrary.aero,” SKYbrary, [Online]. Available: https://skybrary.aero/articles/calculated-take-time-ctot. [Accessed 12 February 2023].

[74] “www.aviation.stackexchange.com,” 9 February 2017. [Online]. Available: https://aviation.stackexchange.com/questions/35357/how-is-calculated-takeoff-time-ctot-calculated. [Accessed 12 February 2023].

[75] “www.eurocontrol.int,” January 2020. [Online]. Available: https://www.eurocontrol.int/publication/introducing-eurocontrol-network-manager-operations-centre-nmoc. [Accessed 30 May 2023].

[76] “www.skybrary.aero,” September 2021. [Online]. Available: https://skybrary.aero/articles/network-manager-operations-centre-nmoc. [Accessed 30 May 2023].

[77] "/www.eurocontrol.int," Eurocontrol, 23 December 2016. [Online]. Available: https://www.eurocontrol.int/article/what-is-a-slot#:~:text=Normally%2C%20the%20aircraft%20should%20take,to%20reapply%20for%20a%20slot.. [Accessed 12 February 2023].

[78] H. Khadilkar and H. Balakrishnan, "Estimation of Aircraft Taxi-out Fuel Burn using Flight Data Recorder Archives," Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2011.

[79] L. Yang, Y. Suwan, K. Han, J. Haddad and H. M., "Fundamental diagrams of airport surface traffic: Models and applications," *Transportation Research Part B: Methodological,* 2017.

[80] X. Hu and W.-H. Chen, "Genetic algorithm based on receding horizon control for arrival sequencing and scheduling," *Engineering Applications of Artificial Intelligence,* 2005.

[81] L. Glomb, F. Liers and F. Rösel, "A rolling-horizon approach for multi-period optimization," *European Journal of Operational Research,* 2022.

[82] A. Bischi, L. Taccari, E. Martelli, E. Amaldi, G. Manzolini, P. Silva, S. Campanari and E. Macchi, "A rolling-horizon optimization algorithm for the long term operational scheduling of cogeneration systems," *Energy,* 2019.

[83] M. Khachay, Y. Kochetov and P. Pardalos, "Mathematical Optimization Theory and Operations Research," in *18th International Conference, MOTOR 2019*, Ekaterinburg, Russia, 2019.

[84] B. A. Hendrickson and M. H. Wright, "Mathematical Research Challenges in Optimization of Complex Systems," Department of Energy, USA, 2006.

[85] I. Modos, "www.towardsdatascience.com," Towards Data Science, 21 February 2023. [Online]. Available: https://towardsdatascience.com/mixed-integer-linear-programming-formal-definition-and-solution-space-6b3286d54892. [Accessed 20 April 2023].

[86] J.-L. Marcelin, "Optimization of the boundary conditions by genetic algorithms," in *HAL Open Science*, 2012.

[87] A. Richards, "Trajectory optimization using mixed-integer linear programming," University of Bristol, Bristol, UK, 2005.

[88] N. Ploskas and N. V. Sahinidis, “Review and comparison of algorithms and software for mixed-integer derivative-free optimization,” *Journal of Global Optimization,* 2021.

[89] J. Carr, “An Introduction to Genetic Algorithms,” in *Withman College*, 2014.

[90] A. Palet, “Performance Evaluation of Path Planning Techniques for Unmanned Aerial Vehicles,” in *Dept. Computer Science & Engineering, Blekinge Institute of Technology*, Karlskrona, Sweden, 2016.

[91] J. Peters, D. Stephan, I. Amon, H. Gawendowicz, J. Lischeid, L. Salabarria, J. Umland, F. Werner, M. S. Krejca, R. Rothenberger, T. Kötzing and T. Friedrich, “Mixed Integer Programming versus Evolutionary Computation for Optimizing a Hard Real-World Staff Assignment Problem,” in *Twenty-Ninth International Conference on Automated Planning and Scheduling*, Berkeley, California, USA, 2019.

[92] A. Javaid, “Understanding Dijkstra Algorithm,” *SSRN,* January 2013.

[93] M. Zhou and N. Gao, “Research on Optimal Path based on Dijkstra Algorithms,” in *3rd International Conference on Mechatronics Engineering and Information Technology*, Dalian, China, 2019.

[94] V. Martell and A. Sandbeg, “Performance Evaluation of A* Algorithms,” Faculty of Computing, Blekinge Institute of Technology, Karlskrona, Sweden, 2016.

[95] R. Ramadiani, D. Bukhori, A. Azainil and N. Dengen, “Floyd-warshall algorithm to determine the shortest path based on android,” in *IOP Conference Series Earth and Environmental Science*, Banda Aceh, Indonesia, 2018.

[96] R. Jordan, M. A. Ishutkina and T. G. Reynolds, “A Statistical Learning Approach to the Modeling of Aircraft Taxi TIme,” in *Digital Avionics Systems Conference*, Salt Lake City, Utah, USA, 2010.

[97] “MathWorks,” Matlab, [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed 17 April 2023].

[98] “www.sqlite.org,” SQLite, [Online]. Available: https://sqlite.org/index.html. [Accessed 17 April 2023].

[99] “www.autodesk.com,” Autodesk, [Online]. Available: https://www.autodesk.com/. [Accessed 17 April 2023].

[100] S. van Oosterom, M. Mitici and J. Hoekstra, “Dispatching a fleet of electric towing vehicles for aircraft taxiing with conflict avoidance and efficient battery charging,” *Transportation Research Part C,* 2023.

[101] M. Zoutendijk and S. van Oosterom, “Electric Taxiing with Disruption Management: Assignment of Electric Towing Vehicles to Aircraft,” 2023.

[102] M. Zoutendijk, M. Mitici and J. M. Hoekstra, “An investigation of operational management solutions and challenges for electric taxiing of aircraft,” *Research in Transportation Business & Management,* 2023.

[103] A. L. Salihu, S. M. Lloyd and A. Akgunduz, “Electrification of airport taxiway operations: A simulation framework for analyzing congestion and cost,” *Transportation Research Part D,* 2023.

[104] “www.sciencedirect.com,” ScienceDirect, 2010. [Online]. Available: https://www.sciencedirect.com/topics/economics-econometrics-and-finance/monte-carlo-simulation. [Accessed 27 April 2023].

[105] S. Raychaudhuri, “Introduction to Monte Carlo Simulation,” in *Winter Simulation Conference*, Miami, Florida, USA, 2008.

[106] S. Gordon, “The Normal Distribution,” Mathematics Learning Centre, University of Sydney, Sydney, New South Wales, Australia, 2006.

[107] M. Taboga, “www.statlect.com,” StatLect, 2021. [Online]. Available: https://www.statlect.com/glossary/probability-density-function. [Accessed 27 April 2023].

[108] “www.maltairport.com,” Malta International Airport, 2023. [Online]. Available: https://www.maltairport.com/corporate/traffic-development/statistics/. [Accessed 27 April 2023].

[109] “X-Planes Reviews,” [Online]. Available: https://xplanereviews.com/index.php?/forums/topic/925-scenery-review-lmml-malta-international-by-justsim/. [Accessed 3 March 2023].

[110] “Annual Report 2020 Ben Gurion International Airport,” Ben Gurion International Airport, Tel Aviv, Israel, 2020.

[111] “X-Plane Reviews,” [Online]. Available: https://xplanereviews.com/index.php?/forums/topic/5542-scenery-review-llbg-airport-ben-gurion-xp-by-aerosoft/#comment-10446. [Accessed 3 March 2023].

[112] “www.toulouseairport.net,” Toulouse-Blagnac Airport, 2023. [Online]. Available: https://toulouseairport.net/statistics/. [Accessed 27 April 2023].

[113] “Javier Blog,” [Online]. Available: https://theblogbyjavier.com/tag/lfbo/. [Accessed 3 March 2023].

[114] “ACI World,” Airports Council International, 2022. [Online]. Available: https://aci.aero/2022/04/11/the-top-10-busiest-airports-in-the-world-revealed/. [Accessed 27 April 2023].

[115] “www.statista.com,” Statista, 2023. [Online]. Available: https://www.statista.com/statistics/1268446/number-aircraft-operations-at-dallas-international-airport/. [Accessed 27 April 2023].

[116] “Federal Aviation Administration,” [Online]. Available: https://www.faa.gov/air_traffic/publications/atpubs/aip_html/part3_ad_2.0_texas.html. [Accessed 3 March 2023].

Annex 1

Table 28-Table 31 show C_σ, M_σ and $\Delta t^{\sigma}_{dtot.act}$ obtained with the *Probabilistic Model* (both with LSD and HSD) for all the airports and for different percentages of tow trucks, in *Test Scenarios 2-5*. Besides the fact that the values of C_σ, M_σ and $\Delta t^{\sigma}_{dtot.act}$ are higher when calculated with HSD compared to those calculated using LSD, the results of *Test Scenarios 4* and *5*, which use *Dynamic Allocation*, are generally higher than those of Test Scenarios 2 and *3*, which use *Static Allocation*. This can be explained by the maximisation of tow truck utilisation, which has a knock-on effect that interferes with the intended tow truck missions when systemic uncertainty is added.

Table 28. C_σ, M_σ (%) and $\Delta t_{dtot.act}{}^{\sigma}$ (s) in *Test Scenario 2* using the *Probabilistic Model*.

Test Scenario 2							
Percentage of Tow Trucks (%)	**Airport**	**LSD**			**HSD**		
		C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)	C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)
10	MLT	7.2	1.9	18.9	12.6	5.0	30.9
	TLV	6.5	4.3	13.1	19.2	5.8	24.3
	TLS	7.0	3.9	8.0	9.7	5.8	13.5
	DFW	8.1	5.3	7.4	13.3	11.9	12.0
20	MLT	7.6	2.3	16.9	13.0	4.3	26.8
	TLV	6.3	4.3	13.5	19.5	6.7	25.2
	TLS	7.0	3.8	10.2	9.9	6.1	15.9
	DFW	8.2	6.7	6.6	12.2	11.7	12.4
30	MLT	8.9	1.8	17.3	14.4	5.6	32.9
	TLV	7.8	3.7	14.1	20.6	5.2	23.8
	TLS	7.4	3.5	9.9	11.3	6.1	15.7
	DFW	8.4	6.2	6.3	14.2	11.2	12.8
40	MLT	9.0	1.9	16.7	12.5	4.6	32.4
	TLV	7.6	4.1	17.1	18.2	6.8	25.5
	TLS	8.2	3.5	9.8	11.1	5.6	17.1
	DFW	10.1	6.8	6.9	13.9	12.4	13.8
50	MLT	7.3	1.8	19.8	14.8	5.1	24.7
	TLV	6.0	4.3	13.8	20.6	6.6	24.0
	TLS	8.4	4.3	7.9	11.3	6.8	13.5
	DFW	8.4	5.5	6.2	15.8	12.4	12.5

Table 29. C_σ, M_σ (%) and $\Delta t_{dtot.act}{}^{\sigma}$ (s) in *Test Scenario 3* using the *Probabilistic Model*.

Test Scenario 3							
Percentage of Tow Trucks (%)	**Airport**	**LSD**			**HSD**		
		C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)	C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)
10	MLT	7.2	1.7	17.7	12.1	5.6	27.1
	TLV	6.2	4.0	15.7	19.6	6.0	23.7
	TLS	7.3	4.1	10.3	9.4	6.2	14.1
	DFW	7.9	6.6	7.3	12.3	10.6	12.8
20	MLT	6.9	2.2	20.1	12.5	4.9	25.5
	TLV	6.3	4.5	16.9	19.8	6.5	23.6
	TLS	7.2	3.8	8.8	10.3	6.7	15.7
	DFW	8.2	6.3	6.6	13.0	10.9	11.7
30	MLT	7.7	1.8	18.5	14.6	4.6	30.8
	TLV	7.9	4.2	16.0	24.1	6.3	29.3
	TLS	9.0	4.1	9.6	12.3	5.4	17.1
	DFW	8.1	6.7	6.6	15.6	12.3	14.5
40	MLT	6.9	2.1	15.5	14.8	5.7	25.7
	TLV	6.0	4.5	14.5	22.0	6.3	29.3
	TLS	8.7	4.0	10.0	12.2	5.8	17.1
	DFW	8.2	5.7	7.8	15.9	10.1	12.2
50	MLT	8.6	2.0	15.5	14.2	4.6	26.0
	TLV	6.2	3.4	14.8	20.9	6.7	26.8
	TLS	8.5	4.4	8.3	10.2	5.1	14.5
	DFW	9.2	6.9	6.4	14.6	9.8	11.7

Table 30. C_σ, M_σ (%) and $\Delta t_{dtot.act}{}^{\sigma}$ (s) in *Test Scenario 4* using the *Probabilistic Model*.

Test Scenario 4							
Percentage of Tow Trucks (%)	**Airport**	**LSD**			**HSD**		
		C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)	C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)
10	MLT	8.3	1.9	17.6	14.0	5.4	28.3
	TLV	7.7	4.0	16.2	22.9	6.5	27.3
	TLS	8.1	4.2	9.3	12.0	6.2	15.0
	DFW	9.7	6.0	7.5	14.3	11.6	13.4
20	MLT	9.1	2.1	18.7	15.8	5.4	32.1
	TLV	7.9	4.1	15.9	21.8	6.8	28.3
	TLS	8.9	4.2	9.3	12.6	6.8	15.4
	DFW	10.0	6.8	7.9	14.7	12.1	13.4
30	MLT	8.5	2.0	19.4	17.3	6.5	30.1
	TLV	7.6	5.1	18.4	25.8	6.4	33.3
	TLS	9.3	4.9	9.7	12.1	6.6	16.2
	DFW	9.2	6.8	7.7	15.4	12.5	16.7
40	MLT	9.7	2.4	21.7	15.4	6.1	30.3
	TLV	8.8	4.4	16.3	22.7	7.0	31.8
	TLS	9.6	5.1	11.7	11.1	7.2	15.3
	DFW	10.8	6.9	8.0	16.7	12.3	14.8
50	MLT	9.7	2.4	19.2	14.1	6.1	30.5
	TLV	7.8	4.4	17.3	21.3	7.1	30.0
	TLS	9.7	4.8	11.0	11.3	7.7	16.7
	DFW	11.1	6.5	8.8	16.8	11.9	15.1

Table 31. C_σ, M_σ (%) and $\Delta t_{dtot.act}{}^{\sigma}$ (s) in *Test Scenario 5* using the *Probabilistic Model*.

Test Scenario 5							
Percentage of Tow Trucks (%)	**Airport**	**LSD**			**HSD**		
		C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)	C_σ	M_σ (%)	$\Delta td_{tot.act}{}^{\sigma}$ (s)
10	MLT	8.2	2.1	18.9	13.4	5.1	28.0
	TLV	6.8	3.8	15.2	21.9	6.4	25.4
	TLS	8.5	4.3	9.3	11.6	6.2	14.5
	DFW	9.2	6.3	7.7	15.2	10.7	13.1
20	MLT	9.1	2.0	19.8	15.6	5.6	29.1
	TLV	7.2	4.0	15.6	22.6	6.5	26.6
	TLS	9.0	4.2	9.7	11.6	6.6	16.6
	DFW	9.2	6.3	7.9	15.7	11.0	14.0
30	MLT	9.7	2.2	19.3	16.1	5.4	35.4
	TLV	7.8	4.4	15.2	21.6	7.2	31.0
	TLS	9.1	4.1	9.5	12.5	7.3	15.4
	DFW	10.8	7.3	7.0	15.4	12.1	13.3
40	MLT	8.4	2.6	18.1	17.6	5.8	37.7
	TLV	8.3	4.8	19.0	22.1	7.1	28.2
	TLS	9.2	5.1	10.0	13.3	7.8	17.1
	DFW	9.5	6.5	8.8	17.7	11.1	16.3
50	MLT	8.8	2.3	18.5	15.1	6.4	31.3
	TLV	7.7	4.9	18.7	21.3	6.7	31.2
	TLS	8.6	4.8	9.8	12.2	6.2	15.9
	DFW	11.5	7.7	7.9	17.6	11.1	15.3

L-Università ta' Malta

FACULTY/INSTITUTE/CENTRE/SCHOOL Institute of Aerospace Technologies

DECLARATION OF AUTHENTICITY FOR DOCTORAL STUDENTS

(a) Authenticity of Thesis/Dissertation

I hereby declare that I am the legitimate author of this Thesis/Dissertation and that it is my original work.

No portion of this work has been submitted in support of an application for another degree or qualification of this or any other university or institution of higher education.

I hold the University of Malta harmless against any third party claims with regard to copyright violation, breach of confidentiality, defamation and any other third party right infringement.

(b) Research Code of Practice and Ethics Review Procedure

I declare that I have abided by the University's Research Ethics Review Procedures. Research Ethics & Data Protection form code (N/A).

☒ As a Ph.D. student, as per Regulation 66 of the Doctor of Philosophy Regulations, I accept that my thesis be made publicly available on the University of Malta Institutional Repository.

□ As a Doctor of Sacred Theology student, as per Regulation 17 (3) of the Doctor of Sacred Theology Regulations, I accept that my thesis be made publicly available on the University of Malta Institutional Repository.

□ As a Doctor of Music student, as per Regulation 26 (2) of the Doctor of Music Regulations, I accept that my dissertation be made publicly available on the University of Malta Institutional Repository.

□ As a Professional Doctorate student, as per Regulation 55 of the Professional Doctorate Regulations, I accept that my dissertation be made publicly available on the University of Malta Institutional Repository.