Deicing scheduling

R.S. Verboon

Deicing scheduling

by

R.S. Verboon

to obtain the degree of Master of Science
in Computer Science track Information Architecture
at the Delft University of Technology,
to be defended publicly in the Colloquium room - EEMCS DI.01.180 on Thursday June 30, 2016 at 10:00 AM.

Student number:	4084896	
Project duration:	November 1, 2015 – June 30, 2016	
Thesis committee:	Prof. dr. C. Witteveen	TU Delft, supervisor
	Dr. J.A. Pouwelse	TU Delft
	P. van Leeuwen MSc.	NLR
	D. Goossens	Aviapartner

This thesis is confidential and cannot be made public until June 25, 2016.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

Imagine that you are flying to your favourite holiday, the winter sport holiday. Unfortunately it is already snowing at your airport of departure, Amsterdam Schiphol. Fortunately there is a deicing crew, which ensures a safe departure by *deicing* the aircraft. Deicing is an extra activity for the ground operations, and during this hectic time at the airport you would still like to depart on time to your destination. This thesis is written from the context of optimising the processes around the deicing of aircraft on Amsterdam Schiphol, more specifically the Aviapartner Operation.

This work provides a model for deicing operations which employ a deicing-at-the-gate operation. We have formulated this model in a Resource Constrained Project Scheduling Problem (RCPSP), and subsequently translated that into a Mixed Integer Linear Programming (MILP). We have shown that the computing bounds of an exact MILP implementation for our model is met at around 40 aircraft (activities). Therefore we introduced approximation solutions to cope with larger problem sizes. Finally we show a number of methods to cope with three disturbance scenarios we have defined.

The work was carried out for a number of parties, first of all there is the TU Delft, secondly I worked with the Nederlands Lucht- en Ruimtevaartcentrum (National Aerospace Laboratory) (NLR). For the more practical side of the operation we included Aviapartner, as they actually have a deicing operation. The challenge is to combine all the interests of the parties to make one thesis. TU Delft is mainly interested in the scientific portion of this work, while NLR and Aviapartner are more interested in the practical side, the software which is produced.

The committee is composed of four members. The committee represents the main parties involved in this project: Delft University of Technology, the Netherlands Aerospace Research Centre, and Aviapartner. Pim van Leeuwen is the daily supervisor for this project. He works at the Netherlands Aerospace Centre (NLR) in the field of air traffic management and airports. Cees Witteveen, is the supervisor from Delft University of Technology, and the chair of my thesis committee. Dave Goossens was our contact person at Aviapartner, and fulfilled the role of supervisor airside for the Schiphol operation. Finally, Johan Pouwels is the external member of this committee, he is an associate professor at Delft University of Technology.

I would like to thank my supervisors for guiding me through this process. Pim has been a great mentor during the course of this project, his input helped me improve the quality of my work. I would like to thank Cees for supervising this project, and taking a critical look at this work. Dave has been a great contact at Aviapartner, he helped us by showing us the operations at Aviapartner, which gave us good insights in the deicing operation. I would like to thank Johan for his contribution as a committee member. Finally, I would like to thank my friends and family for supporting me through my whole academic career.

R.S. Verboon
Delft, June 2016

Contents

1

Introduction

The air transport industry offers accessibility and mobility for goods and people. This makes it a vital component in our society and economy. Optimising air transport processes is a vital component to keep the air transport industry robust and cost effective. Before one can optimise a process in the air transport industry, one has to model the process. This thesis is focused on the deicing planning process, and optimising in the planning of the deicing activities.

This thesis is written from a computer science background, therefore the terms and processes of the air transport industry are introduced in this chapter, as they might be unknown to the reader. First, an overview of the different stakeholders is given, we will then we will give the main concept of communication between these stakeholders, the Collaborative Decision Making (CDM). Secondly, a description of ground processes is given, with a focus on the deicing process. Thirdly we will describe what influences this deicing process. We will give an overview of the current practises of Aviapartner on Amsterdam Schiphol. Finally, we will list what we think are the main practical problems.

1.1. Stakeholders for Deicing Operations

First of all, we focus on the major stakeholders involved in the air transport industry. We selected a number of stakeholders that represent the whole air transport industry. The selected stakeholders are as follows:

- **Airlines** such as KLM are the operators of flights. Airlines provide services for cargo and passenger transport. The airlines operate on various routes, for which they plan their aircraft. From the viewpoint of a passenger, airlines take responsibility for a smooth operation, they are the point of contact for the passenger. This can incur some challenges, as airlines are responsible for some processes while they have limited control over some processes, such as that of the ground handler.
- **Airports** are responsible for maintaining the facilities for aircraft to land, turn around, depart, and passenger boarding and de-boarding. Furthermore airports impose rules and regulations for other stakeholders regarding the processes in the aerodrome, and on the ground.
- **Air Navigation Service Provider (ANSP)**, are responsible for safely managing the air traffic. Different parts of the airspace are managed by different entities, for instance, at the airport traffic is managed by ground controllers.
- **Service Providers**, such as Aviapartner are responsible for various operations on the airport, this can include handling of luggage, providing staircases, and refuelling the aircraft. The most important aspect for these service providers is that all operations are executed safely, and on-time. Service providers are responsible for deicing operations on the ground.
- **Passengers**, and cargo, are the end-clients of the air transport industry. Passengers benefit by an on-time performing air transport industry. Often passengers can be critical when it comes to delay.

Figure 1.1: Five key CDM times graphically displayed. Two times for the inbound flight, and three for the outbound flight. Inbound is the process of the aircraft landing at the airport and taxiing to the gate. After the turn-around the aircraft becomes outbound, it leaves the airport. Image taken from [25]

For this work we focus on the operations on the airport. Here we see that the three major stakeholders come into play, and all hold different values. Airliners generally want a cost effective, and timely operation. This means they want high availability of runways and gates. The runways are managed by an ANSP, who generally want an operation which is robust and fair for all stakeholders concerned.

Aviapartner is one of the service providers on Amsterdam Schiphol Airport. Their services include baggage handling, check-in/boarding, and deicing.

1.2. CDM the Communication Standard for Actors

These stakeholders need to communicate with each other. It can be useful to share information about when processes around are ready, or are planned to be ready. To facilitate a formal process of communication about key times, the concept of Collaborative Decision Making (CDM) is employed. CDM is a concept/philosophy, whereby actors agree to share specific information[14]. The agreements include the communication of key times, and at which point these key times are communicated or updated.

The idea of CDM is to share information for an overall benefit. For instance, consider a delayed plane; this will impact the gate scheduling, as a delayed plane will depart later it might interfere with the next aircraft at the gate. Sharing this kind of information between the airline and e.g. the ground service provider will benefit the overall planning. Another example is that if the turnaround is delayed, the next plane will not be able to dock at the gate, so an alternative gate or stand might be needed. The earlier this delay is communicated, the better the planning will be able to adjust.

Since we write this thesis for the Aviapartner deicing operation on Amsterdam Schiphol, we consider the CDM implementation on Amsterdam Schiphol. The main actors are as follows[33]: the report for Aviapartner's operation on Amsterdam Schiphol.

1. Amsterdam Airport Schiphol (AAS) the owner of the airfield.
2. Luchtverkeersleiding Nederland (Dutch ATC) (LVNL) the operator of the airspace, and responsible for gate and runway allocation.
3. Airlines, such as KLM.
4. Meteo (weather) providers, such as the KNMI
5. Schiphol Airline Operators Committee (SAOC) a committee which also consists of ground operators such as Aviapartner.
6. Service providers (included in the SAOC) provide several services on aeroplanes.

The CDM shares information between parties. One of the most important pieces of information is the sharing of specific times. Traditionally there was limited information sharing between layers of the organisation, and a good implementation of CDM can overcome these hurdles. An implementation of CDM can lead to common situational awareness[14]. The result is that more layers of the organisation know key times, and can act upon that. For this report we will give an introduction to five key CDM times[25]. Figure 1.1 illustrates these five key times, the key times are as follows:

1. Estimated Landing Time (ELDT), after an aircraft enters Dutch airspace, LVNL updates the estimated landing time continuously.

Figure 1.2: Runways at Amsterdam Schiphol Airport. Notice that the 'Polderbaan' (18R-36L) is positioned at a significant distance from the main terminal building. Image taken from `https://nl.wikipedia.org/wiki/Luchthaven_Schiphol#/media/File:Schiphol-overview.png` under Creative Commons License.

2. Estimated In-Block Time (EIBT), when in final approach, the runway is known, together with the assigned gate, the taxi time can be calculated, this gives an approximation for the in block time.
3. Target Off-Block Time (TOBT), a ground handler has to put in the time it expects to be done with ground handling.
4. Target Start Up Time (TSAT), this time indicates the push-back and start-up of an aircraft. For TSAT there is a window of ten minutes around it to do the actual operation.
5. Target Take-Off Time (TTOT), the time that an aircraft is going to take-off.

These five times displayed are an illustration of what Schiphol calls its most important CDM times. There exist many more time points in the system, most notably the 'actual' times. These times denote the time at which a specific event happened.

An important notion here is that taxi times in Schiphol can differ greatly, if one lands on the 'Polderbaan' (18R-36L), a taxi to the Delta pier can take 15 minutes, see 1.2. This is also the reason for a separate mention of the EIBT and TTOT.

As Schiphol is a large airport, the TTOT is divided into slots. These slots are assigned to aeroplanes, if an aeroplane does not make his slot, the pilot has to request a new slot. When a new slot is requested there is a possibility of a delay. Therefore on of the things CDM aims to reduce, is the amount of unused take-off slots. By sharing information of future missed slots earlier, multiple parties in the chain of the air transport industry can anticipate on this delay. The next section will elaborate on the ground operations, these form an important process between the key CDM times.

1.3. Ground Operations

Between the landing and the take off, the ground operations take place. The ground operations are illustrated in figure 1.3. Generally when an aeroplane touches down, it first has to taxi to the gate. At the gate, the wheels are locked by blocks. After that, the passengers and crew have to disembark, the plane has to be fuelled, to be cleaned, catering has to load and unload, and baggage has to be loaded. After that the blocks are removed, and the plane can be pushed back, and depart. The process between the in-blocks and off-blocks is the turnaround.

If deicing is needed this will happen at two points in the turn-around process. In the case of an at-the-gate deicing, this will happen between the off-blocks and the push-back. In case of a remote

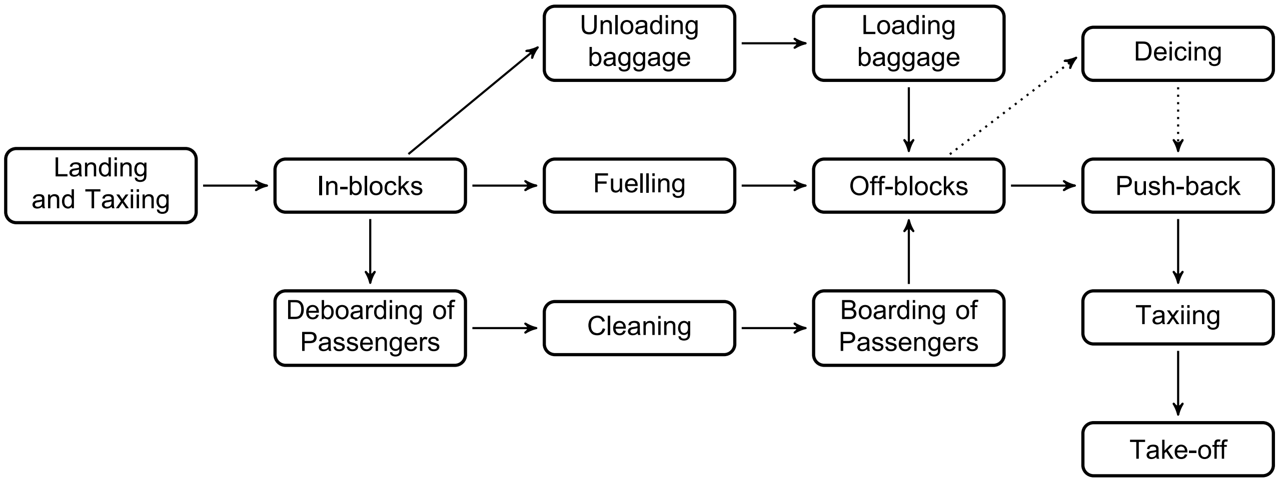


Figure 1.3: Ground operations needed for an aeroplane turnaround, note that the deicing operation is optional, and not present in most of the turn-arounds. This figure is a subset of the actions needed, note that operations as: jet-bridge coupling/decoupling, water services, technical services, and others are missing.

deicing, this will happen between taxiing. For this work we focus on a at-the-gate deicing operation, since Aviapartner only employees a at-the-gate deicing operation.

1.4. Deicing Operation

One of the parts of a non-standard ground operation is the deicing of aircraft. During winter circumstances keeping an aeroplane ice free is important to ensure safe flight. Ice built up on the critical surfaces of an aeroplane, such as the wings, can influence the aerodynamics of an aircraft[7].

The process of deicing is build up of two components: deicing and anti-icing[35]. Deicing is the removal of any ice buildup on the plane. Anti-icing is giving the plane a protective layer to prevent any ice from building up on the plane. For the rest of this work, we consider the terms deicing and anti-icing as one operation, and will generally refer to it as just deicing.

Considering the ground operations in section 1.3, the deicing can take place at two moments during the ground operations. It can take place at the gate, that means that before the push-back deicing operations are employed. The other option is that deicing takes place at a remote deicing operation stand, so after the push-back a plane will taxi to the deicing operation stands, and after deicing taxi further to the runway.

The deicing operations consist of spraying the so called deicing fluid on the aeroplane. The deicing fluids are composed of glycol and water. There are special trucks for deicing, the deicing trucks can contain a limited amount of deicing fluid. Before deicing operations can commence, the fluids have to be heated. The deicing fluids have an impact on the environment[24], and therefore the waste fluids have to be collected after operation for safe disposal.

The deicing fluids are composed of, most often, a glycol mixture. There are a number of producers of these fluids. The first part of a deicing treatment is the clearance of ice and/or snow, after that the deicing fluids are used to give the aircraft a protective layer. This layer has a limited time, depending on the conditions, it is safe to use, this time is called the holdover time. This time depends on the mixture of fluids used and the weather, severe conditions can shorten this holdover period, and thicker fluids can increase this holdover time. The time between the end of deicing, and the departure has to be shorter than the holdover time.

The captain is responsible for requesting deicing on his aircraft. When the pilot calls in a deicing, it can request the number of phases (one or two), and the areas to be deiced. The one phase deicing is de- and anti-icing together, and the two phase deicing is separate treatment of deicing and anti-acing. Different mixtures of deicing fluids can be more suitable for a certain situation.

The deicing operations occur only when certain conditions are present, and therefore are not part of the normal operation. This means that deicing usually leads to delays. Furthermore deicing operations will give an additional cost, depending on the contract this is by the ground service provider, or the airline itself. The risk of delays is a risk for the airlines. A good capacity planning can lead to a cost reduction by mitigating risks of a under- and over capacity, and might lead to a more robust planning. To make such a planning a, good understanding of the factors that influence it is needed, this includes the weather.

1.5. Weather Communication Standards

Weather influences the deicing operations, the fall of snow directly gives a need for deicing. Furthermore, cold and relatively humid conditions give reasons for deicing. In airports two important weather information streams are available, the information which describes the current situation: METAR, and an information stream with forecasts: the TAF.

1.5.1. METAR

METAR stands for Meteorological Aerodrome Report. In contrast to the TAF, a METAR message reflects the current situation. The METAR messages follow a standard format, an example start of a message is:

`METAR EHAM 201025Z 27013KT 9999 FEW033 09/04 Q1005 NOSIG=`[1]

The message starts with stating that it is a METAR message. The next for letters denote the ICAO code for the airport, `EHAM` is the ICAO code for Amsterdam Schiphol Airport. The next fragment (`201025Z`) denotes the date of the month `20`, and the time `1025Z` 10:25 zulu time, which is by computer scientist better known as UTC. This is followed by a part about the wind; `27013KT` means that there is wind from the direction 270 degrees, and has a speed of 13 knots. The `9999` block is about the visibility, the visibility is capped at`9999`, which means that the visibility for this message is more than 9999 meters. `FEW033` denotes that there are a *few* clouds at flight level 33, the flight level is the height from the ground in hundreds of feet. The temperature (`09/04`) is `09` degrees, and the dew point is `04` degrees centigrade. The air pressure (`Q1005`) is 1005 millibar. The last part denotes that in this message there are no significant changes (`NOSIG`) since the report before this one.

Winter Codes

After the visibility in the METAR message, there is a place for weather codes. These codes can, for example, display if there is precipitation, and what kind of precipitation. METARs, which contain snow (`SN`), freezing rain (`FZRA`), and freezing drizzle (`FZDR`), indicate that there is a high chance of conditions which require deicing.

1.5.2. TAF

The Terminal Aerodrome Forecast (TAF), serves as a forecast for the weather. The TAF follows a similar formatting as the METAR. The TAF message also includes from which point in time the message is valid and till when the message is valid. Consider the following example TAF[2]:

```
TAF EHAM 201029Z 2012/2118 27015KT 9999 SCT025
    PROB30 TEMPO 2015/2024 28018G28KT
    PROB30 TEMPO 2015/2019 7000 -SHRA SCT020TCU
    TEMPO 2019/2112 4000 SHRAGS SCT015 SCT020CB BKN030
    BECMG 2102/2105 24008KT
    BECMG 2109/2112 36015KT=
```

One can see that this message follows the same format as the METAR. For the TAF there is an extra element; the fourth word, `2012/2118` denotes the validity, this means it is valid from the 20th of the month at 12 o'clock till 21st of the month at 18 o'clock. Furthermore there is a 30% chance on temporary wind at direction 280 with gusts up till 28 knots. There is also a chance of 30% on light showers rain (`-SHRA`) with lower visibility (`7000`m), from 15 till 19 o'clock on the 20th of the month. The full definition of this forecast can be found online[3]

[1]Taken from `https://www.knmi.nl/nederland-nu/luchtvaart/vliegveldwaarnemingen` on the 20th of November 2015 at 11.39

[2]Taken from `https://www.knmi.nl/nederland-nu/luchtvaart/vliegveldverwachtingen` on the 20th of November at 15:00

[3]`http://aviationweather.gov/static/help/taf-decode.php`

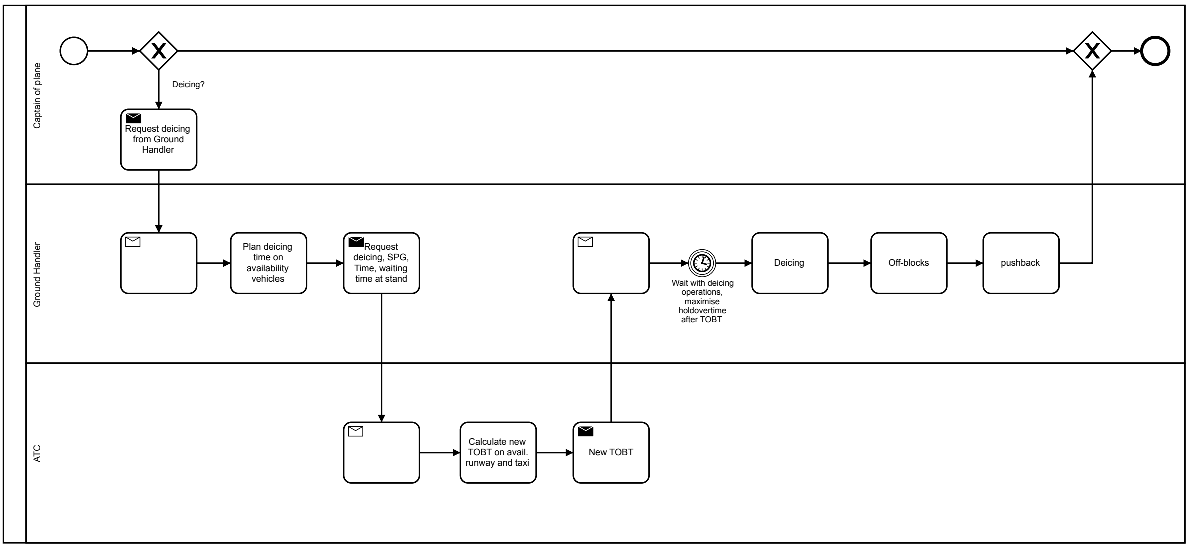


Figure 1.4: BPMN model of deicing at the gate for AAS, base on Official Schiphol CDM report[9]

1.6. Weather Influencing the Deicing Process

Given the defitions of the METAR and TAF messages, some weather conditions are more dangerous than others. The Association of European Airlines (AEA)[15] identifies that lower temperatures require a more intense deicing treatment. This is due to the properties of the deicing fluid that the holdover time is negatively correlated to the temperature. It also states that even as temperatures are above $0°C$, deicing can be necessary, especially in humid conditions, and if the wings are cold. This can create the so called 'cold soaked wing effect' that due to cold wings condensation occurs, and deicing is required.

The relative humidity can be calculated from the dew point[4]. The dew point is the temperature at which the current humidity will cause condensation. It is possible that at low temperatures ($< 0°C$) no deicing is needed, since there is a low absolute humidity.

At the NLR a Deicing weather type classification exists that based on the weather input, one could decide if deicing is necessary.

1.7. Deicing Request

We focus our work on that of a at-the-gate deicing operation on Amsterdam Schiphol[9]. The flow of a deicing request is visualised in figure 1.4. The start of a deicing call starts at the airline, this can be the pilot or an airline representative. The ground handler has the responsibility to update the TOBT, this is done by putting in a call for deicing in the CDM system. The call for deicing is preferably done 20 minutes for the TOBT at that moment[5]. ATC will approve or reject a deicing call, a reject means that the ground operator has to put in a request for a new time. In case of an approval, the new times will be in the CDM system.

For ATC it also depends on whether clean-up services are available. Due to the environmental impact of the fluids, the fluids have to be handled with care.

1.8. Deicing Capacity Forecasting

Deicing is not part of the 'normal' ground operations. As these activities take up time of personnel, extra personnel has to be available when deicing activities take place. The challenge is that deicing greatly depends on the situation, weather can change during the day, impacting the demand for deicing.

For assessing capacity needs, we define three levels of planning. Fist of all is the strategical planning, this is the long term planning. This strategic planning mainly concerns with long-time considerations, such as the acquisition of deicing vehicles. The pre-tactical planning, is the planning before

[4]http://andrew.rsmas.miami.edu/bmcnoldy/Humidity.html
[5]http://video.schiphol.nl/B2B/CDM/Procedures/12-2-CDM-De-Icing-Procedure.html

operations, this generally means the planning a day ahead of the day of operation. The tactical planning is the planning which is done on the day of operation.

1.8.1. Strategical Planning

Strategical planning is needed for each of the three resources as defined in the previous section. For vehicles one requires the acquisition and maintenance contracts for these vehicles. Fluids have to be bought from a supplier, a contract ensures the delivery of such fluids for an agreed price. Personnel has to be trained to have appropriate training for deicing activities. At the beginning of a deicing season, there has to be enough qualified personnel for all functions needed for deicing. This main components of strategical planning are outside the scope for the main portion of this work. However, to give the reader a complete picture of planning we note that prior to pre-tactical planning, strategical planning is necessary.

1.8.2. Pre-tactical Planning

Generally, a planning is made a day ahead. This planning is based on the available data. Overcapacity is a situation where too many resources are planned, but stay unused. Overcapacity will generally lead to a good customer satisfaction, as there is less chance of delay building up. However, overcapacity will lead to higher costs, as vehicles remain unused for a portion of their time. Undercapacity is a situation in which enough resources are available to ensure a good operation, that leads to a lower Quality of Service, which in result can lead to reputational damages and contractual damages. The challenge is to avoid under- and/or overcapacity by accurately making a capacity planning, this can be done by making a detailed pre-tactical planning.

For the pre-tactical planning, the following input data is used:

1. **Weather**, this is mainly based on the TAF.
2. **Schedule**, the scheduled departures from the companies having a contract for deicing. The schedule should include the estimated time of departure, the type of aircraft, and predicted (gate) location.
3. **Historic data**, Historic data can be used to optimise the planning, as some airlines might have higher chances of requesting deicing in certain conditions, and departure delays might be correlated with the origin of the aeroplane. The insights learned from historic data can extend beyond the two previous notions.
4. **Deicing vehicles**, the number of deicing vehicles available, and the characteristics of these vehicles, e.g. capacity, current fluid levels, etc. For example a vehicle might be broken down or scheduled for maintenance and therefore not be operational.
5. **Deicing times**, depending on the conditions and aircraft size, time for deicing varies. A table which provides a deicing time depending on aircraft size and weather condition is needed.
6. **Travel times**, a function or table is needed which can give travel times between various locations on the airport.

The output is a planning, which mainly consists of an assignment of vehicles to aircraft at a specific time. The planning takes into account the operational constraints, such as the travel times between gates. The constraints and goal are further explained in section 1.10.

1.8.3. Tactical Planning

The third level we devise is the tactical planning. In deicing conditions, the arrivals and departures of aeroplanes show significant delays. This means that the turnaround process, and the departure time of the plane increases in uncertainty. For example,a planning which is made in advance might need revision if departure times change.

The tactical planning deals with the planning for the coming hours. Often one has a few deicing requests then, these need to be planned and coordinated.

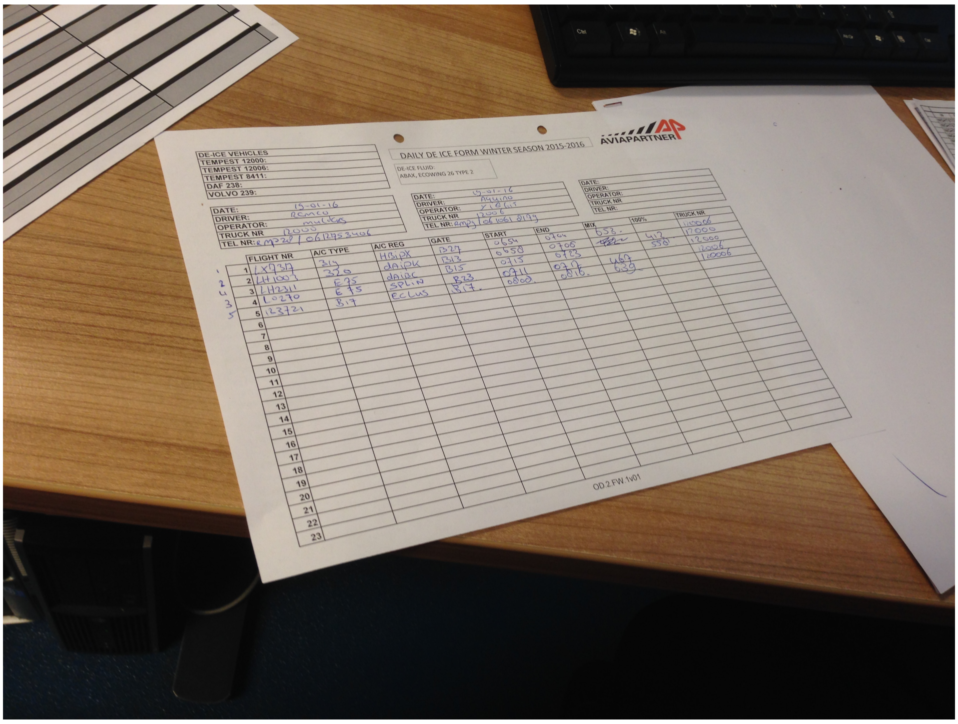


Figure 1.5: Deicing form for planning deicings at Flight Watch.

1.9. Methodology

This work is aimed at the deicing operations for a ground handler. More specifically the operation of Aviapartner at Amsterdam Schiphol. We want to limit the scope of our work the *planning* of deicing activities. We therefore do not consider for instance optimisations which might be made using different fluids or methods. We conducted a series of interviews and work visits to the Aviapartner operation from November 2015 until April 2016. These visits were aimed to establish a view of the current practises, objectives and concerns. Appendix A.2 and A.3 provide extra insight. The next sections will describe the current practise, the methods used for forecasting capacity, and we describe how we interpret the problem (challenge) Aviapartner faces.

We hypothesise that good planning can lead to improvement over the current method, the current planning methods can be characterised as *ad-hoc*.The operation of Aviapartner at Schiphol is characterised by a at-the-gate operation. This means that after the blocks are removed, deicing is applied. With five deicing trucks Aviapartner is a small, but significant, player on the Airport.

Our approach was to first conduct a number of interviews to assess the current deicing operations. Aviapartner has a deicing operation consisting of five deicing vehicles. The deicing operations mainly take place in the morning. This is due to *night stoppers*, aircraft that are parked outside during the night are likely to obtain ripe on the wings, and therefore require deicing in the morning.

The deicing operations are coördinated with operations planners (Flight Watch), they handle the communications with ATC and to the operations on the ground. The procedure is that after a request for deicing is received from the aircraft or from ground personnel, the planners write it down on a piece of paper, see figure 1.5. Then, on a first-come-first-served basis the deicing vehicles are assigned to aircraft. Before the start of each operation, Flight Watch will give an OK signal over telephone, after the operations the deicers call back with the detailed time the deicing took and the amount deicing fluid used in the deicing operation.

Forecasting the extra capacity needed for deicing is done using several input sources. While conducting interviews, we noted that the main sources are the flight schedule, METAR, TAF, and buienradar [6]. Each of these components has a limited span of forecasting capabilities. For example Buienrader can only forecast precipitation for a couple of hours in the future.

[6] `http://buienradar.nl`

Figure 1.6: Deicing vehicle from the Aviapartner operation. Note that the deicing vehicle has an open basket, e.g. there is no cover protecting the worker in the basket. This deicing vehicle requires two people to operate: a driver and a deicer. (image taken as a still from `https://youtu.be/5CP6grJi3AE`)

We presume that improving existing plannings can benefit the operational cost and quality of service. The first step in improving is understanding the current operation and operation needs of the deicing service.

1.10. Problem Formulation

To recapitulate, we noted that Aviapartner has a relative small deicing operation with five vehicles. Furthermore we see that for the deicing planning some steps are not planned, but ad-hoc operation is used. We presume that planning can lead to improvements. The first step is to provide a proper overview of the planning problem at hand.

In this section we provide a formal overview of the planning problem, this is done by splitting the problem into three categories. First of all, the resources are defined; resources are entities or products with limited availability. The resources are generally constrained by a number of constraints. Constraints define limits on the resources, they determine whether a solution is feasible. Finally, objective(s) define the 'goal' of the program, an objective defines the path at which solutions are considered improved.

1.10.1. Resources

Deicing operations depend on a number of resources, these resources are necessary to ensure smooth deicing operations. This work limits the resources to the 'physical' resources, and therefore omit more abstract resources, such as contract managers or existing relations. We define the main resources as follows.

Personnel
The deicing operations require trained personnel. For Aviapartner these are all round employees. A deicing vehicle requires two operators, one for driving the vehicle, and one for operating the hoses. The operator stands in a basket, which is supported by a beam on the truck.

Vehicles
The vehicles are specially designed for deicing operations, as can be seen in figure 1.6. Vehicles have two engines; one for driving, and another one for providing power for the fluid heater. The fluids are

stored in a boiler, and can be heated to obtain operating temperature. The vehicle has a beam with a basket for the operator. The beam can extend to reach the highest part of aeroplanes, the vertical stabiliser.

Fluids
The deicing fluids are stored in two big tanks at the airport. The fluids are then transported in the vehicles to be sprayed on the aeroplane. The temperature of the fluid has to be $60°C$ at the nozzle[15].

1.10.2. Constraints

The described resources are constrained by their physical abilities, and by rules and regulations. We note the following constraints for the Aviapartner deicing case:

1. At least one vehicle has to perform deicing operations on an aeroplane which wishes to receive deicing treatment. This is necessary for safety.
2. Two qualified operators have to be on each vehicle; one driver and one in the basket.
3. Vehicles can gather (cold) fluids from the depot. These fluids have to be heated to be operational. The heating takes approximately 30 minutes.
4. Each deicing operation takes a specific amount of time, this depends on the number of deicing trucks, the (weather) conditions and the aircraft type.
5. Vehicles can store a limited amount of fluid of each type.
6. There are a limited number of deicing vehicles (five).
7. Not all vehicles are able to deice every aeroplane, it depends on the extensibility of the beam and the height of the aeroplane; a vehicle must be able to reach the top of the plane.
8. Each aeroplane requires an amount of each fluids for it's operation. In case of a two-step operation, both types of fluid are used.
9. The deicing operations have to be finished in time such that the holdover time is not violated, and the aircraft can depart in time for its departure slot.
10. The deicing operations cannot commence before a clearance to start the operations is given, also known as the Target Start Up Time (TSAT).
11. Personnel has limited availability, and is bound by laws for working and rest times.
12. Since the planes are deiced at the gate, a travel time is incurred for vehicles and personnel travelling from one stand to another stand. This travel time also holds for driving to the fluid depot.

Furthermore, we can identify that not all processes are set in stone with regards to the certainty whether they are needed, what time they take, and/or what time the processes start. It is possible for vehicles to breakdown. For deicings, it depends on the conditions of the weather whether they are necessary.

1.10.3. Objectives

First of all, a schedule of personnel, vehicles, and deicings has to be valid, this means that each plane is deiced, while obeying the given constraints. Furthermore we might want to optimise this schedule to a number of objectives. We list an number of objectives in this section. These objectives can also be seen as Key Performance Indicators (KPI) for Aviapartner

Safety
As mentioned before, the most important aspect of deicing is providing safety. No or bad deicing will alter the aerodynamic behaviour of the plane, which could result in unsafe situations. Therefore this objective is modelled as a constrained. This implies that every valid planning automatically will satisfy this KPI

Profit
If a very basic definition of profit is taken; revenue minus costs. Optimising this KPI implies that we want to limit the costs, while maximising the revenue. The costs can be seen as the sum of stocking and purchasing fluids, personnel expenses, and vehicle operating expenses. The revenue generated depends on the type of contract Aviapartner has with its client.

Reputation - QoS
The operator of deicing is concerned about its reputation. There are multiple operators on Schiphol, that means that there is not only competition on price, but also on quality of the operation, the Quality of Service (QoS). The AEA estimates that delay will cost 90.8 EUR per minute to an airline for a connecting flight[34](e.g. a flight to a hub). We can define the quality as the ability to fulfil the clients wishes. This means that there is a timely and highly available service, that even a last minute request of deicing will cause a limited or no delay. We interpret reputation as fast request fulfilment.

Fairness
As multiple airlines are dependent on the service of Aviapartner, there can be situations were a competition for resources is present. For Aviapartner it is important to give all airlines with a similar service contract the same service. We assume that by applying a First Come First Served (FCFS) rule the fairness property is fulfilled. Other planning approaches might not use FCFS, but will not favour any airline or aircraft specifically. Therefore we consider fairness satisfied by the design of our algorithms.

Stability
Another goal of a schedule can be that changes in the planning will lead to minimal changes for the rest of the operation, there is a certain robustness and stability in the planning. This means that a small change in the operation will not lead to big changes in the planning. The air transport industry, and its clients, are concerned with an on-time delivery. Another consideration is that the air transport industry is prone to delays. The delays will lead to changing schedules. However, changes are prone to errors, and reducing changes might also be an objective. As less changes (more stability), will be easier to work on.

1.11. Operational Concerns

The deicing operations are coördinated with operations planners, they handle the communications ATC, airline Flight Operations Centre FOCs, and to the operations on the ground. During our interviews, we have observed a number of operational challenges:

- Vehicles can break down, usually they operate two of the five vehicles. There is a service contract with KLM ground services to service the vehicle. This enables them to quickly repair vehicles. However, the operations at the time of breakdown require another vehicle to enter the operations.

- The fluids for deicing have to be heated. Heating takes up to 30-45 minutes. The fluids in a tanker can be heated only three to four times, then they loose quality, and are no longer suitable for operations. When a vehicle has to apply many deicing treatments, it might need intraday refilling of deicing fluid, this also means that at each refilling a 30-45 minute warm-up time is needed before operations can recommence.

- Due to the size of Amsterdam Schiphol, driving from one gate to another gate can take a considerable amount of time. Furthermore, a variation in time is possible, as there can be traffic on the ring road.

- There can be faulty times communicated through the CDM system. Aviapartner might align its operation to a faulty time in the CDM system. This is thus an external party which influences the operation of Aviapartner.

- During wintry times, the airport can be covered in snow. The snow can also impact the movement of ground vehicles. Therefore travel times might change.

The deicing fleet of Aviapartner consists of five vehicles, two of an older type and three of a newer type. The fluids used are produced by Abax, and Aviapartner uses the type II fluid in a 'pure' ratio $(100/0)$, and in a $(75/25)$ mixture. Generally speaking the $75/25$ is used for the first part of a two-step procedure, or for the whole deicing treatment in case of a one-step procedure. The second part of a two-step procedure can be done with a (cold) pure glycol fluid. The choice for which procedure depends on the weather and the wishes of the captain, who has ultimate responsibility.

1.12. Practical Problems

From these objectives, constraints, and concerns, we can formulate a number of *practical problems* (PP). The practical problems are written from the perspective of the problem owner, for this work we consider the operator, Aviapartner, as the problem owner. The practical problems are closely related to the research questions, which will be introduced in the next chapter. The difference is that the practical problems are questions that originate from the main stakeholder; Aviapartner. The research questions are questions that are left to answer after we look at existing, scientific work.

We hypothesise that proper planning can improve the operation of Aviapartner. Proper planning can, partially, mitigate the effects of both under- and overcapacity. These effects have great impact on the airline transport industry. Therefore the main question or problem for Aviapartner can be summarised as follows:

PP: *How can Aviapartner optimise the deicing operations with regards to the key performance indicators?*

The main practical problem can be broken down in a number of sub-problems:

PP1 How can the pre-tactical planning phase of Aviapartner be improved, with regards to the objectives defined?

PP2 Given a planning for a day, how can one cope with changes in weather, delays, or sudden availability of resources?

1.13. Project Scope

This chapter introduced the general field, and tried to apply a scope towards practical problems. A side goal of this project is to build a planning tool, the dynamics of the tool are roughly outlined in figure1.7. One can see that a planning is based upon various input sources, and that the output will consists of a detailed schedule. This detailed schedule can be used to base resource estimations on. The main work is focused on the 'tool', or actually the algorithm in the tool that does the scheduling. We make estimations on the input based on real world traces.

The rest of the project is build up as follows. Chapter 2 look into existing work, from this we try to see if we can find enough elements in literature to build a planning algorithm for the deicing problem. A strict planning is the main hypothesis we have to optimise the resource prediction. In that chapter we list the shortcomings of existing work. The short comings are formulated as research questions, which can be seen as a refinement of our practical problems.

We built a complete model of the deicing operations in chapter 3. We do this with the use of advanced programming techniques, which will be explained in chapter 2. The advanced programming techniques have some limitations, as the model is rather complex.

To assess the complexity of our model and the corresponding implementation, we put our model through a series of tests in chapter 4. The tests have to first find out whether the applied techniques are suited for an operation with the characteristics of Aviapartner's operation.

To overcome some limitation we found in the experiments chapter, we present an alternative way to come up with solutions. We provide, alternatively to the exact approaches of the chapter before it, heuristics to solve our model in chapter 5.

We address the issue of the second practical problem, the operational uncertainties, in chapter 6. This is the last main chapter, as it combines both approaches from the exact solution, and from the heuristics.

Finally we conclude our work in chapter 7, in which we first provide answer to the research questions, and with that try to answer the practical problems.

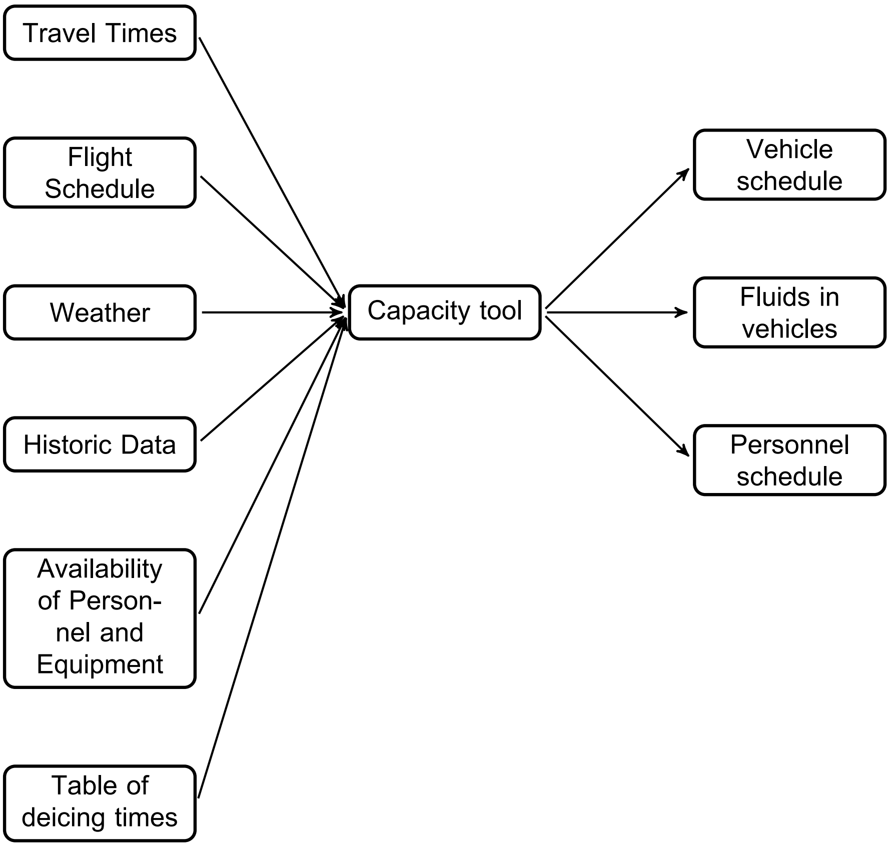


Figure 1.7: High level overview of a capacity forecasting tool

2
State of the Art

To find an answer for the practical problems presented in the previous chapter, we start off by looking at problems found in scientific literature. We do this by looking at specific literature about deicing planning. As deicing is part of the ground operations, we also include literature from ground operations in section 2.1.

As the problem we face is a special kind of planning problem, we zoom-out to more generic planning literature. We focus on one of the most generic formulations: the Resource Constrained Project Scheduling Problem (RCPSP) in section 2.3. The RCPSP exists in many formulations, some works introduce extra tools, or formulation, to model specific properties of their underlying problem.

Then, if we obtain enough information on how to model our problem, we have to solve it. We review the method of modelling the RCPSP in an advanced mathematical formulation in section 2.4.1.

The second practical problem mainly deals with the uncertainties we observed while conducting interviews at Aviapartner. We review existing work on uncertainties mainly on generic planning problems in section 2.5.

Throughout this chapter we will introduce research questions. These research questions arise as we observe that some methods have shortcomings in existing formulations, and that some methods require us to do specific research when suited for our model.

2.1. Deicing Planning

In the previous chapter we noted that deicing is part of the ground operations. The challenge we have is that we want to see how we can optimise the deicing operations for Aviapartner, in the pre-tactical and tactical phase. Our first exploration of literature focuses around literature that solves problems in the deicing and turn-around planning sphere.

The first work we consider is from Mao et al.[26]. They have built a model where they consider the scheduling and planning of deicing operations. They consider a system with a fixed location where the deicing operations take place (remote deicing station). Agents can reserve time slots for deicing. A slot is a set amount of time for deicing operations (e.g. 10:00am-10:10am). Deicing stations are constrained that they can only perform deicing operations on one aeroplane at one time point. If we consider a day of operation, there is a limited number of deicing stations, that means there is a limited capacity. Now, if a captain wants to receive a deicing treatment, it has to reserve (commit) a time slot. When a captain fails to make that slot, he decommits to that slot. The authors modelled that the closer one gets to the actual deicing operation, the better one can estimate the time the deicing operation has to start.

The key challenge for a captain in this system is *when* he has to commit to a deicing slot. In a traditional situation there is no penalty for decommitment, this results in a strategy whereby captains will reserve a slot very early, and when they cannot make it, just decommit. The authors used two strategies, one has a *decommitment* penalty, and the other does not have a decommitment penalty. Both strategies are based on First Come First Served (FCFS). A decomittment penalty changes the game for the captain, as he commits early in the turn-around process, he has an higher chance of not making the slot, and thus paying the decommitment penalty. As there is a value for the delay, and an optional value for decommitment, the captain may choose the strategy to reserve a slot if the expects the cost of reserving the earliest available slot in this round is lower than reserving a slot during the next reserving round.

The experiments showed that a 30 per cent reduction in total delay is possible with the implementation of a decommitment penalty over that of a FCFS without a decommitment penalty. When using a decommitment penalty, there is almost 75 per cent lower standard deviation in delay. Both results are for a scenario with 80 aircraft. So in short using a decommitment penalty will lead to improvements in both overall delay and a reduction in standard deviation.

Looking at the work of Mao we see that it is written from the perspective of a plane as an actor. It reviews a policy for an aircraft. This game theoretic behaviour is something which can be taken into account when planning deicing operations. However, for this work we mainly focus on the planning as a ground handler, and not from the perspective of an individual aircraft. Our setting still involves a number of actors, we still have to deal with captains and flight operations centres who request deicing. The scenario in this work shows us how other actors, aircraft captains, show game theoretic behaviour, and therefore might decommit to a deicing slot.

Furthermore, there is a fundamental difference, in that this work considers a remote deicing operation, whereas Aviapartner operates an at-the-gate deicing operation. This fundamental difference in approach makes the problem harder on the side of the ground-handler, as there is an introduction of travel times. These travel times also are present in a remote deicing operation, as aircraft have to taxi to the remote deicing stand, furthermore queueing of aircraft can occur in a remote deicing situation. However, from a viewpoint of a ground-handler, they have to drive the equipment around in a at-the-gate deicing operation.

2.2. Turn-around Planning

Deicing is one of the last operations in the turn around process. Therefore we widen our scope to literature concerning turn around planning. The main focus of turn around planning to find techniques that can forecast the capacity need for ground handlers, as is done by Oosterman et al.[29] and Van Leeuwen et al.[23]. They modelled the turn-around process on two levels. First the turn-around is modelled as turn-around activities which need to happen for each aircraft. They used a *temporal network* as a modelling technique. In this technique activities are modelled as nodes, and relations between the activities are added between nodes to denote when activities should finish or start depending on the other activity. From these temporal constraints one can calculate with relative ease when activities can take place. However, the ground-operations consists of many actors, all with their own constraints. The authors introduced a technique that uses these temporal constraints and minimises the amount of resources each actor needs. For example if there is flexibility for two overlapping time intervals for fuelling, only one fuelling truck is needed, whereas if there where overlapping time time intervals with no flexibility, two were needed.

The work of looking at the turn around process is useful to get a sense of the number of protocols involved in the turn around process. For our work, however, we want to specifically focus on deicing planning. We hypothesise that a detailed planning of deicing operations can lead towards a better understanding of the operations, and with this detailed planning a better resource capacity planning can be made. The authors of the previous work used decoupling often as a technique, with decoupling you choose specific values for certain sub processes, this means that one does not fully explore the search space. So in our situation we start forecasting the resource needs *after* we have made a detailed planning.

2.3. RCPSP, Universal Modelling Approach

As we have seen from the related work from both the deicing literature and the turn-around literature, that there is not enough information to make a detailed planning for the deicing operations that can lead to a better resource need prediction. The deicing literature did not review the situation we have, as we have a at-the-gate deicing operation. The turn-around planning focused on the decoupling, we want however to view the process as a whole. This has as an advantage that potentially optimal solutions are not missed, as might happen with decoupling. Furthermore temporal networks do not allow us to assign activities to deicing trucks, as an activity is assigned to another vehicle, the temporal relations change. Therefore we look at more generic planning approaches.

Considering our deicing problem, we formulated the constraints in 1.10. For our problem we have a number of essential resources, namely the deicing vehicles, personnel and fluids. Furthermore we have constraints such as the time at which point deicing operations can commence, and when they should optimally be finished.

Example 2.3.1. For our deicing case, consider this simple example of three aircraft which need to be deiced: $P = \{p_1, p_2, p_3\}$, two deicing vehicles $V = \{v_1, v_2\}$, each aircraft has a specific duration $t_{d_p} = 10$, i.e. deicing takes ten minutes. Furthermore, deicing vehicles can only work on one vehicle at a specific time. Each plane has a specific time after which operations can start $t_{i,r}$, and should ideally be finished before $t_{i,l}$. The challenge is to find for each plane p_i a suitable start time $t_{i,\text{start}}$, with a corresponding $t_{i,\text{end}} = t_{i,\text{start}} + 10$, while the constraints are adhered.

Now we try to model our problem onto that of the Resource Constrained Project Scheduling Problem (RCPSP). The RCPSP is a class of planning problems, which has been widely studied in literature[36]. The goal is to schedule a series of activities while maintaining resource constraints. An RCPSP consists of a number of processes (alternatively activities) $P = \{p_1, \ldots, p_n\}$. Each process has to be done at some point in time, and each activity has a specific duration d_i, activity i takes d_i time. Most RCPSP formulation do not allow preemption, which means that each activity has to be done without interruption. There are resources $R = \{r_1, \ldots, r_m\}$. Each activity can have resource requirements $c_{i,j}$ meaning that activity i requires $c_{i,j}$ resources from resource j. Furthermore there can be precedence constraints on activities, for instance it is necessary to complete activity one before we can start with the second activity (e.g. filling before driving).

An RCPSP also defines precedence relations, that means that for any activity p_i there can be a set of activities $B_{v_i} \subseteq P$ that defines activities which have to be finished before activity p_i can be started. An example would be that it is necessary to start a deicing truck before operations can commence.

A feasible schedule is an assignment of activities to a start times, which respects the resource constraints, and the precedence constraints. This means that for every p_i there is a corresponding $t_{i,\text{start}}$. Checking for feasibility consists of a number of (easy) tasks:

1. For every plane p_i there is a start time $t_{i,\text{end}}$, and that start time is after the release time $t_{i,r}$,
2. Verifying that each resource is only used by one activity at each specific time point.

Checking such an assignment for feasibility is an easy task. However, most of the times one wants specific quality assurances. We have set a 'deadline' ($t_{l,i}$), which is not explicitly formed as a constrained. So for example one might wants to ensure that deadlines are not excessively overwritten. There are different possibilities for optimisation criteria. Note that always the constraints have to be satisfied, before one can speak of a 'better' solution. We list a number of optimisation criterion frequently found:

1. *Makespan*, if one wants to optimise the makespan, one wants to make the duration between the start of the first activity, and the finish time of the last activity as short as possible. The start time of each activity (p_i) is written as $t_{i,\text{start}}$ and the finish time as $t_{i,\text{end}}$.

$$\text{minimise: } \left(\max_{p_i \in P} t_{i,\text{end}}\right) - \left(\min_{p \in P} t_{i,\text{start}}\right) \tag{2.1}$$

2. *Tardiness*, or *delay*, tardiness is the total amount of time (duration) between the finish time and the an optimal finish time. In the context of the air transport industry, tardiness is best explained as cumulative amount of delay. If we take the notation that for aircraft p the optimal time of end of deicing operations is $t_{p,d}$ we get that we want to optimise to the following criterion:

$$\text{minimise: } \sum_{p_i \in P} \max\left(0, t_{i,end} - t_{i,l}\right) \tag{2.2}$$

3. *Robustness*, a criterion used in RCPSP literature[3], it can be used in a multi-objective setting, where besides optimising for makespan one wants to optimise for a robust schedule. We define robustness for a planning as the extend to which the planning can withstand changes in a activity. A more robust planning will let, in case of a disturbed activity, other activities proceed without

mode	vehicle one used	vehicle two used	deicing time	fluid use vehicle one	fluid use vehicle two
1	yes	no	10 min.	450ℓ	0ℓ
2	no	yes	10 min.	0ℓ	450ℓ
3	yes	yes	7 min.	225ℓ	225ℓ

Table 2.1: Example of how multi-mode scheduling can work for our simple deicing problem example. There are three modes, each mode defines which resources, renewable and non-renewable, are used.

change. In the air transport industry it valued that activities are not influenced by changes of other activities. Abbasi et al.[1] translate this robustness concept to something they call *Floating Time (FT)*. In the case of our deicing problem example, consider an valid solution, from this solution we can derive for each vehicle a order of planes $\{p_i, p_j, ..., p_k\} = P_{v_a}$ = planes of v_a, make sure this list is sorted in increasing order of deicing start times. Now the floating time for vehicle v_a is defined as:

$$\text{FT}(v_a) = \sum_{\text{two consecutive } p_i, p_j \in P_{v_a}} t_{j,\text{start}} - t_{i,\text{end}} \tag{2.3}$$

Now to end up with one metric for the whole planning, one can take the weighted average of every vehicle planning:

$$\text{FT} = \sum_{v_a \in V} \frac{\text{FT}(P_{v_a})}{\max(1, |P_{v_a}| - 1)} \tag{2.4}$$

If the floating time is high, and if activities extend their expected duration, there is a higher chance the final makespan does not change, and no changes to schedule are required. For the air transport industry this robustness can be used as one flight might be delayed and other flights still require services to be completed on time.

2.3.1. Extensions of the RCPSP

As the RCPSP is such a widely used approach, there exist many extensions on the basic idea of the project schedule. For our work we require a couple of extensions. We list a couple of extensions we think are useful for modelling our deicing scheduling problem.

1. *Deadline and release times*, for every p_i there is a release time $t_{i,r}$ at which an activity can start, this corresponds to the time the deicing operations can start: the TSAT. Deicing activities are supposed to finish before a certain time, such that, after taxiing, the departure time can be made. We model this as: $t_{i,l}$, a deadline.

2. *Transfer times*[22], sometimes transfer times are needed for resources. Some resources might be able to process a number of activities, however a transfer time might be needed between two consecutive activities. For example a deicing vehicle has to drive (transfer) from one activity (aeroplane) to another aeroplane, this time is the travel time or transfer time. In case of our example we can introduce a function $t : (p_i, p_j) \rightarrow t_d$ which takes two aircraft, and returns a travel time (t_d) between the two locations of the aircraft for a deicing vehicle. For example $t(p_1, p_2) = 6$ minutes the travel time between aeroplane one and aeroplane two is six minutes.

3. *Multi-mode scheduling*[8] is that activities can be processed in multiple modes, each mode has a specific resource requirement. If we consider our example, for plane one (p_1), there are two modes, one that it is deiced by vehicle one, and the other that it is deiced by vehicle two, this is displayed in table 2.1. We can model this into an RCPSP by changing the $c_{i,j}$ into a set of possible assignments, of which one has to be chosen: resource requirements for p_i are $\{v_a \lor v_b\}$.

4. *Non-renewable resources*[6] are resources that can be 'used', so they are not renewed. For our deicing problem, we can model fluids as non-renewable, since they once get depleted in a vehicle, and therefore a vehicle is no longer able to perform deicing operations. This resource usage is

also displayed in table 2.1. In our example we consider that there is for each vehicle 4500 litres of capacity for deicing fluids. Once this gets depleted there has to be a refilling, this means that the resource is somewhat renewable, as it can fill up at the fluid storage facility. Renewable resources, can in contrast to non-renewable resources, used indefinitely, while maintaining their capacity constraints.

These extensions offer almost enough degrees of freedom to model the deicing operations. We consider the RCPSP a universal enough modelling instrument to model our deicing operations. The RCPSP is able to grasp resources, constraints and objectives of our deicing problem.

However, if we consider the usage of fluids in vehicles we have a resource (fluid) that is not renewable, as it can be depleted, and not fully non-renewable, as refilling can restock the amount of fluid in the vehicle. To the best of our knowledge, we have not seen something like this fluid resource in literature, which is used in a setting where there is also transfer times and multi-mode scheduling. We therefore are left with the question how we can model this.

2.4. Solving RCPSP Planning Problems Exactly

Solving Resource Constrained Project Scheduling Problem (RCPSP)s is a NP-HARD problem[5]. This means that finding an optimal solution for large problems is virtually impossible. However, for many problems it is possible to provide algorithms that provide feasible instance, however these are not guaranteed optimal.

2.4.1. The Mixed Integer Linear Program

One technique for solving an RCPSP is to formulate it as a Mixed Integer Linear Programming (MILP). A MILP is a mathematical formulation of an optimisation problem. These linear programmes have been widely used to solve planning problems, this can be in the chemical industry[19], as well in other sectors. The MILP is one of the classical $\mathtt{NP-HARD}$ problems[16]. The advantage of translating our problem to this specific $\mathtt{NP-HARD}$ problem is that for the MILP multiple solvers exists (commercial and non-commercial), this implies that this field of MILP solvers is optimised and perfected.

The MILP is a mathematical formulation of an optimisation problem. As the reader might be unfamiliar with a MILP formulation, we will give a simple introduction to a MILP formulation. Consider a situation where we have the same two trucks as previously used in our example 2.3.1. We define an objective that we want to spread the number of deicing minutes as fairly as possible amongst the trucks.

A simple MILP would look something like this:

$$\min|y_1 - y_2| \tag{2.5}$$

$$\text{Subject to: } 450(1-p_1) + 450(1-p_2) + 450(1-p_3) = y_1 \tag{2.6}$$

$$450p_1 + 450p_2 + 450p_3 = y_2 \tag{2.7}$$

$$p_1, p_2, p_3 \in \{0,1\} y_1, y_2 \in \mathbb{R} \tag{2.8}$$

The first line of the formulation above, formula 2.5 is called the *objective*, we defined the objective as the absolute difference between the fluid in the first vehicle (y_1), and the fluid in the second vehicle (y_2). The lines after the subject to, equations 2.6 to 2.8 define the conditions for this MILP. The variable p_x defines by which vehicle the aircraft is deiced; $p_1 = 0$ states that the first plane is deiced by vehicle one, $p_1 = 1$ defines that the second vehicle will deice the first plane. Finally the last line (2.8) states that the p_x variables are Boolean, and the total fluids per vehicle are real numbers. This formulation is also an example why it is called a *mixed* integer linear program, since there are variables constrained by different sets of numbers.

Mula et al.[28] made a comparison between approaches found in the RCPSP and MILP research. The authors classify papers on the used techniques, an interesting approach of some papers is not only to use a MILP, but to use a Multi Objective Linear Programming (MOLP). A MOLP uses more objectives to optimise to. Marler et al.[27] give an overview of techniques for combining and handling multiple objectives. In the airline transport industry one might want to optimise to different criteria, where no clear trade-off can be made, for instance profit and customer satisfaction (on-time performance?). These trade-offs can be captured with the usage of multi-objectives, which why MOLPs might be used.

2.4.2. Time Formulation in MILP Formulations

Floudas et al.[11] review several approaches for building a Mixed Integer Linear Programming (MILP). They devise between discrete and continuous time formulations. First the authors describe for the discrete time formulation how one can plan an RCPSP. The authors apply this to a chemical process, for which there is an intermediate storage of chemicals. The authors make an variable for this intermediate storage for each time point, by then adding constraints they can ensure that this storage is not used past its capacity.

For the continuous time formulation, the authors use specific time points, to which they assign continuous time variables. This allows for a transition from a discrete formulated model to a continuous time formulated model.

If we have a time variable, lets say the start time of a deicing operation, we can formulate it in one of the two paradigms. We list for this example variable two example of how the start time of a deicing operation could be formulated:

1. *Discrete*, the variable t_s is translated to a defined number of sub-variables. Each sub-variable corresponds to a specific time. This results into a defined set of variables at which the activity can start. For instance only on exact minutes on one day of operation, so this one t_s variable will be translated into 1440 sub-variables $S = \{t_{s,1}, ..., t_{s,1440}\}$. To make sure there is only one start variable true, the following conditions are added: one for defining the domain of the variables: $\forall t'_s \in S : t'_s \in \{0, 1\}$, and one for that only one is true: $\sum_{t'_s \in S} t'_s = 1$.

2. *Continuous*, the variable t_s is translated to one single variable $t^*_s \in \mathbb{R}$. Often there are two functions needed, one which can translate time variables to one of that of the MILP time domain, and one which does the exact opposite.

From the above example it can be seen that the discrete formulation introduces more variables compared to the one in continuous formulation. Some authors stick with the discrete formulation, as some problems can only be expressed easier in this format.

2.4.3. MILP for the Deicing Problem

Taking these things into consideration, about how to formulate a MILP, and which time formulation to use. We want to research which formulation can be used for our problem. We have seen that the RCPSP knows many extensions that enable us to model specific properties of the deicing problem. The RCPSP is an instrument, modelling technique, that is used for many approaches. We think this instrument offers us a good starting point for building our model.

If one decides to use the RCPSP, there are more specifics to solving the model. We have seen that the most widely used approach is to model the defined RCPSP as a MILP, which is then solved by a MILP-solver. For our deicing problem we need to find the suitable RCPSP model, therefore we introduce the following research question:

RQ1 *How can one model the deicing problem into an RCPSP, while observing the defined resources, constraints, and objectives?* This question aims at providing a model for deicing operations, such that it can be solved with generic planning problem solvers. We have seen from literature that almost all elements can be used with existing formulation techniques. The question left open is how we can model fluids.

2.4.4. Scaling the Problem Size

As noted before, the RCPSP is NP-HARD[5]. Kone et al.[20] implemented RCPSPs with a discrete, exact time formulation and with the usage of approximation algorithms. Both implementations were submitted to a series of test-sets obtained from a standard library[17]. These test-sets are a collection of activities and constraints. We see that for the exact approaches only test-sets are used that are around 30 activities. For our work we want to test up to which point we can calculate a feasible schedule with usage of a MILP for our RCPSP. Therefore we introduce the following research question:

RQ2 *Can our exact formulation handle problems sizes which are representable for the Aviapartner practise?* The model given in the previous question is useful if it can be used on problem sizes the size of an actual operation. We want something that can run in under a couple of hours, since that is the amount of time a planner typically has to do the pre-tactical planning. The construction of the instances that our exact planner is solving, has to be based on actual operations at Aviapartner.

Furthermore, we could argue that we want to see if our model is useful beyond that of Aviapartner, as their similar operators. The main question here is what the limits are of our exact formulation, therefore we want to test how our model scales. We introduce the following research question:

RQ3 *How does the exact formulation scale?* We want to test up to which problem size our model can be used. Therefore this research question aims to provide an answer to which size we can stretch our exact formulation.

2.5. Dealing with Uncertainty

Planning and scheduling activities is only one side of the story. If we obtain a schedule for deicing activities, it is still the question how good this schedule remains in operations. Often one cannot capture the full state of constraints at the start of the project. For instance delay happens frequently in the airline industry, an it has a significant impact on the operation[37]. A delayed aircraft impacts the release time of activity this means that $t_r' >> t_r$, this also implies that activities that are scheduled to be deiced by the same truck might have to be rescheduled.

This leaves us with the question how we can deal with this *operational uncertainty*. Roughly we see two approaches, that not conflict each other, but both can deal with dealing with some operational uncertainties.

1. *Robust scheduling* Ben-tal et al.[4] describe a technique for scheduling activities in a robust manner, this means that the schedule is valid in a set of discrete operational conditions. For example it is possible that an aircraft needs deicing either at 10 AM, at 11 AM, or at noon. This scenario can, for example, occur if there is delay. Now a robust schedule will be valid in all of these conditions, practically this would imply that there is a 'reservation' on the vehicle from 10 AM to somewhere around 12:10 PM.

   The authors define two paradigms, one for *Robust* scheduling and optimisation, and one for *Stochastic* scheduling and optimisation. The main difference between these two paradigms is that the *Robust* paradigm assumes a set of known conditions, and not a distribution on variables.

   In the context of the deicing problem, this robust scheduling means that one makes an assignment of vehicles to aircraft, such that any delay can be absorbed by the schedule.

2. *Adaptive scheduling*, Considering any general planning technique, one can obtain a *baseline* schedule, this baseline schedule functions as an outline for the operation. If operational uncertainties present itself, adaption of the baseline schedule might be needed. We define an *online* policy as something that can adapt such a baseline schedule that it is again useful for the current state of operation. An adaptive schedule is more *resilient* to changes.

   Persson et al.[31] done this by combining *classical* and agent based optimisation techniques. The idea is that first an overall schedule (baseline) is formed using classical planning approaches. Then an agent can make or suggest changes when operational reality makes it necessary. This is written from a multi-agent setting. We do not model our problem as a multi-agent system.

The work of Hans et al.[12] surveyed planning techniques for project scheduling. The authors make the distinction between *proactive* and *reactive* scheduling. *Proactive* scheduling is making a schedule such that it can cope, up to a point, with uncertainties in the future. *Reactive* scheduling is that we adapt the schedule at the time of operation, and when these disturbances present themselves. Clearly, robust scheduling fall into the paradigm of *proactive* scheduling.

Mula et al.[28] provide an overview of papers of the last 10 years that investigate planning and scheduling papers. They make an important distinction between strategical, tactical, and operational planning, the main difference between these terms is the place in time before the operation when the planning happens. Strategical planning is planning months in advance, e.g. how many trucks do we need to buy? Tactical planning is planning a couple of days before operation, e.g. how do we assign trucks to jobs. And operational planning is planning with operational concerns on the day of operation, e.g. delays.

The relation here can be made that a robuster schedule might lead to a schedule which needs less adaption. For instance if there is more slack between activities, e.g. the time between two consecutive activities on a vehicle is of a significant amount, the schedule is more robust. If a plane has a small delay in a robust schedule, there is enough slack to take the delay.

Taking these considerations together, on one side we are left with the air transport industry, which is prone to delays and changes in schedule. On the other side, we have techniques for dealing with uncertainties, or operational uncertainties, the most important distinction we make is between proactive and reactive scheduling. Furthermore, we see that a combination of these is also possible, as there is overlap possible between the techniques. This leaves us with the research question:

RQ4 *How can we adapt plannings such that it can cope with operational uncertainties?* How can we model and subsequently deal with delays, resource failures, weather changes, etc.

The research question proposed here is two sided. First of all, we have to identify what these *operational uncertainties* are. In the introduction, we have identified some, with the use of interviews. However, these have to be translated to a model we use to build plannings. Secondly, we provide techniques based upon those found in literature and test these to the defined scenarios from the first side of the question.

2.6. Research Questions

Taking all the previous notions together, we can actually distill one overall research question:

RQ: How can one efficiently schedule deicing operations in a at the gate deicing operation?

To recapitulate these are the research questions asked in this work:

RQ1 *How can one model the deicing problem into an RCPSP, while obsserving the defined resources, constraints, and objectives?* This question aims at providing a model for deicing operations, such that it can be solved with generic planning problem solvers. We have seen from literature that almost all elements can be used with existing formulation techniques. The question left open is how we can model fluids.

RQ2 *Can our exact formulation handle problems sizes which are representable for the Aviapartner practise?* The model given in the previous question is useful if it can be used on problem sizes the size of an actual operation. We want something that can run in under a couple of hours, since that is the amount of time a planner typically has to do the pre-tactical planning. The construction of the instances that our exact planner is solving, has to be based on actual operations at Aviapartner.

RQ3 *How does the exact formulation scale?* We want to test up to which problem size our model can be used. Therefore this research question aims to provide an answer to which size we can stretch our exact formulation.

RQ4 *How can/do plannings cope with operational uncertainties?* How can we model and subsequently deal with delays, resource failures, weather changes, etc.

Finally we will conclude in chapter 7, in which we will recapitulate the answers to the research questions, and make the link with the practical problems from chapter 1.

3
Model for offline, pre-tactical planning

The previous chapter described the properties of An RCPSP, the RCPSP is a modelling approach that offers potential for our deicing planning problem. This chapter describes an RCPSP model for deicing operations, which is formulated as a Mixed Integer Linear Programming (MILP). Describing such an implementation of a MILP is a challenging process. Therefore we do this in a couple of phases. First we describe a simplified version of the deicing problem, with the use of this simplified version, we test whether to use continuous or discrete time formulation. Secondly, we describe an extended version of this model, which includes fluid usages. The goal is to build a exact planning instrument, this planning instrument is the middle part in figure 1.7. This is done by processing the input, and then providing a schedule for trucks and personnel. The planning phase is known as either *offline* or *pre-tactical* planning.

First we implement solutions for a relatively simple model in section 3.1, we test whether a continuous time, or a discrete time formulation is better. Then we extend this model in section 3.2 to incorporate the administration of fluid usages. A detailed version of the specific translation of real-world input to input of our model can be found in appendix C. This chapter will introduce a formal notation for the deicing case. Besides introducing symbols in the text, we have attached a *nomenclature*, which lists most symbols with a brief explanation.

3.1. Simple Model for Deicing Activities

This section will introduce a simplified version of the final model. Defining such a model requires us to describe exactly what is happening in the real-world. We have made great effort at describing the real-world situation in the previous chapters. However, we still require to make assumptions on some properties of the deicing problem. The assumptions we have made for the first model are as follows:

1. Every vehicle is able to deice any aeroplane.
2. Vehicles carry enough fluids, so don't need on-day refilling.
3. Start times of deicing are known, as the duration, and do not vary (deterministic). However, deicing duration times depend on aircraft type.
4. The weather conditions are known a priori.
5. The time needed for deicing can be determined from the aircraft type and weather condition.
6. Travel times between gates are known.
7. The location of aircraft, personnel, and vehicles is known.

First we provide the model in section 3.1.1. We provide two types of implementation, one with a discrete time formulation in section 3.1.3 and one with time in a continuous formulation in section 3.1.4. Finally, in this section we perform tests on the performance of each implementation, to assess which formulation should be used.

3.1.1. Model Formulation

The simple model, in contrast to the advanced model, does not consider fluid capacities for vehicles. This means that fluid is not modelled as a constrained or a resource. We define instances of this simplified model as $\mathrm{DEICING}(P, V, d)$: a set of planes $p_i \in P$, a set of vehicles $v_j \in V$, and a function which defines travel times. Each plane p_i has the following properties:

- $t_{i,r}$ time when a plane is ready for deicing,
- t_i time needed for deicing,
- $t_{i,d}$ a time when the plane is scheduled to be ready with deicing operations,
- g_i the gate at which the aeroplane is placed.

Objective

The objective is to find for each plane (p_i) a starting time of deicing $t_{i,s}$, such that this is after $t_{i,r}$. As we have seen in section 2.3, there are different possibilities for the objective. For now we limit us to minimising the *tardiness*, this loosely translates as the total amount of delay. Limiting delay is one of the key challenges in the air transport industry. In the context of this notation this can be defined as:

$$O = \sum_{p_i \in P} \max(0, t_{i,s} + t_i - t_{i,d}) \tag{3.1}$$

Constraints

Vehicles have to move from one location to another location, as aircraft are placed on different stands. Therefore we introduce a function $d : (G, G) \to \mathbb{R}^+$ the function takes two locations and gives the travel time for deicing equipment between these two locations, we assume here that the time taken to travel is the same for both directions.

The vehicles $v \in V$ are constrained by the fact that a vehicle can only deice one aeroplane at a time. Each plane has to be assigned to a vehicle with a start time of the deicing vehicle. This implies that a vehicle has a series of tuples, where each tuple is a combination of aeroplane and start time of deicing $(p_i, t_{i,s})$. In this sequence of tuples, it must hold that for any two consecutive tuples $(p_a, t_{a,s}), (p_b, t_{b,s})$, the start time of the second tuple is after that of the end time of the first: $t_{b,s} \geq t_{a,s} + t_a + d(g_a, g_b)$.

3.1.2. Proof of NP-hardness

If one proves the NP-hardness of a problem, one knows that this problem is not easily solvable. This means that the problem does not scale polynomially with the input size. The NP-hardness implies that the problem scales above polynomially, and for larger problem sizes might be practically unsolvable. In this section we proof the NP-hardness of the deicing problem. We consider a special variant of our DEICING problem, one without an objective function. This means that we model our objective function as a hard constraint (e.g. zero tardiness).

Consider the problem PARTITION, a $\mathrm{NP-HARD}$ problem[21]. The objective is to partition one set of numbers S into two sets of numbers S_1 and S_2, such that $\sum_{s \in S_1} s = \sum_{s \in S_2} s$.

The DEICING problem has the same inputs as the model in section 3.1.1: a set P, a set V and a function d. Furthermore for each aircraft (p_i) we have the location at a gate g_i, the time after which it can start $t_{i,r}$, the duration of the treatment t_i, and the time it should be ready $t_{i,d}$. We rewrite the objective into an extra constraint: the total tardiness should be below some set value. This means that the O from formula 3.1 should below some value: $O \leq O'$.

Given the problem of PARTITION, we give polynomial reduction: $\mathrm{PARTITION} \leq_p \mathrm{DEICING}$.

- P for each number $s \in S$ we insert a plane p_i in P, we set for that plane the location to a fixed location $g_i = 0$, time for deicing to that number $t_i = s$, the release time is set to $t_{i,r} = 0$, and the deadline $t_{i,d}$ to half the sum of the elements in S.
- $V = \{v_1, v_2\}$ We make two deicing vehicles, as we have two sets in the partition.
- $d : (G, G) \to 0$ We leave driving times out, this is a function which will return 0 for any two locations.

The proof consists of three parts, first we show that the reduction can be done in polynomial time, then we prove that a yes-instance of PARTITION will result in a yes-instance of DEICING(→), and finally we will show that a yes-instance of DEICING will result in a yes-instance of PARTITION(←).

1. The size of the input of PARTITION is $n = |input| = |S|$. The reduction consists of three steps, the first step takes $|S|$ time, the second and third step only take $O(1)$ time. Therefore the reduction takes $O(n)$ time, and is polynomial.

2. →: A yes instance of DEICING will have an assignment of aeroplanes to vehicles, such that there is no tardiness, e.g. each aircraft is finished before the deadline. Since we set the deadline for each aircraft at half the total deicing time, we know that each vehicle v_1 and v_2 will deice for half of the total time. If we take the deicing time of each aeroplane we can map that time from a vehicle to a corresponding partition. The two partitions are a yes instance of the PARTITION problem, since the sum of time is equal for both vehicles.

3. ←: Given a valid partition of a set of numbers S_1 and S_2; it holds that $\sum_{s\in S_1} s + \sum_{s\in S_2} s = \sum_{s\in S}$, and it holds that $\sum_{s\in S_1} s = \sum_{s\in S_2} s$. If we translate the assignment of these numbers to aeroplane according to the reduction function, we can assign vehicles to aircraft as in the two sets, e.g. a aeroplane is assigned to v_1 if its corresponding deicing time t_i is obtained from a number $s \in S_1$. Since now we know that the set of planes v_1 and v_2 deice directly corresponds to S_1 and S_2 we also know that the total deice time is the $\sum_{s\in S_1} s = \sum_{s\in S_2} s$. Therefore we know that we can make the deadline for each plane $\sum_{s\in S} s/2$.

Since we have proven the three things above, we have proven that PARTITION$\leq_p$ DEICING and since PARTITION∈NP − HARD we know that DEICING∈NP − HARD. This proof enables us to state that finding an exact solution for the DEICING problem is hard. This means that usually for large problem sizes, finding exact solutions is infeasible. This proof shows why it might happen that large problem sizes are unsolvable.

3.1.3. RCPSP Formulation in Discrete Time MILP

The following two sections are devoted to show the translation of an RCPSP into a MILP. The first step we have is building a Mixed Integer Linear Programming (MILP) from the RCPSP formulation in section 3.1.1. We will follow the notation for the MILP as briefly introduced in section 2.4.1. This version features a discrete time formulation. We choose to set our discrete interval at one minute; the time granularity is one minute. Each *minute* is translated to an individual Boolean variable $x_{a,m,t}$, where the a refers to a specific aeroplane p_i, the m refers to a vehicle v_j, and the t refers to a specific one minute time interval.

For the objective we use the minimal tardiness. We define tardiness (for the deicing problem) as the excess time past a deadline. For each aircraft we have the adjusted deadline d_a, this is the optimal finish time, minus the process time, e.g. if operations start after d_a, tardiness is incurred. Therefore the tardiness per aircraft a can be defined as: $\max(0, t \cdot x_{a,m,t} - d_a)$ (equation 3.1).

The first formulation we make of the simplified deicing model in a MILP is given in MILP 3.1

The first part of the MILP is the objective function 3.2, there are a number of objectives possible, for this one the minimal tardiness is chosen (cumulative delay).

If $x_{a,m,t}$ is true, that denotes the start time of activity a on vehicle m at time t. Analogous, if $x_{a,m,t}$ is false there is no activity start of deicing aircraft a by vehicle m at time t. Equation 3.3 states that each activity should be scheduled. There must be a truck who deices plane a on any t, ergo the sum is 1 in this equation. This equation also makes sure there is exactly one start time, and therefore only one truck who does the deicing.

Equation 3.4 ensure that each deicing only starts after the 'release time' $t_{a,r}$ of each aircraft a. The release time is needed in this process as deicing is only allowed as the jet bridge is removed from the aeroplane.

Equation 3.5 makes sure that each vehicle only processes one activity at a given time. This equation also makes sure that the travel times between activities are observed. This is done by taking two different aircraft, p_i and p_j, and making sure that the sum of those two on the same vehicle is not higher than one. The $d(g_i, g_j)$ denotes the travel time, and t_i the process (deicing) time. If p_i is scheduled to

$$\text{minimise} \sum_{a \in A} \sum_{m \in M} \sum_{t \in T} \max(0, t \cdot x_{a,m,t} - d_a) \tag{3.2}$$

$$\text{subject to} \sum_{m \in M} \sum_{t \in T} x_{a,m,t} = 1 \qquad (\forall a \in A) \tag{3.3}$$

$$\sum_{m \in M} \sum_{t \in T} t \cdot x_{a,m,t} \geq t_{a,r} \qquad (\forall a \in A) \tag{3.4}$$

$$x_{a,m,t} + \sum_{\tau = t}^{t + d(g_a, g_b) + t_a} x_{b,m,\tau} \leq 1 \qquad (\forall a, b \in A, a \neq b, \forall m \in M, \forall t \in T) \tag{3.5}$$

$$x_{a,m,t} \in \{0, 1\} \qquad \forall a \in A, \forall m \in M, \forall t \in T \tag{3.6}$$

MILP 3.1: Discrete time MILP formulation

be deiced by vehicle m at start time $t_{i,s}$, then the following $d(g_i, +g_j) + t_i$ minutes any p_j cannot start, therefore the sum of each $x_{j,m,\tau}$ should be 0.

The starting conditions for the vehicles are determined by the first scheduled aeroplane. Since this schedule will serve as an outline for the required personnel and deicing vehicles. For example when the first deicing operation is scheduled at 10 AM, a crew and vehicle have to start at 9 AM to heat the vehicle and drive to the deicing location.

3.1.4. RCPSP Formulation in Continuous Time MILP

The second implementation of the RCPSP into a MILP is done with a continuous time formulation, in contrast to one with a discrete time formulation of the previous section, continuous time uses a time scale without granularity, this means that there is no smallest step in time possible. The smallest time step in the discrete formulation was a minute.

We introduce a finite horizon mapping of real-world time points to time-points in the range $[0, 1]$. Since we consider day plannings, we can assume that operations for that day always start after midnight, and are finished a couple of hours after the next midnight. For example the time interval for the fourth of May starts at midnight and ends on 5 AM on the fifth of May. This means that noon on the fourth of May would have a corresponding value of $\frac{12}{29} \approx 0.41$, which is in the interval $[0, 1]$. For each aeroplane $a \in A$ we introduce a variable $t_{a,s} \in [0, 1]$.

First we define an objective for this MILP, we use the same tardiness function as defined for the discrete time formulation. We try to minimise $\min(t_{a,s} + t_a, t_{a,d})$ (equation 3.7) this states that any delay past the optimal finish time $t_{a,d}$ of an aircraft is considered bad. Note that this formulation of tardiness is different from the one used before, but is essentially the same.

For each aircraft-vehicle combination, we introduce $x_{m,a}$: a Boolean variable. If this Boolean variable is true, vehicle m does a deicing operation for aeroplane a. If two aeroplane are processed by the same vehicle, there must be enough time between the start-times for the actual treatment and travel-time. Consider that p_a and p_b are deiced by the same vehicle. Than if p_a is scheduled before p_b the start of deicing p_b is after the start of p_a plus the deicing time and driving time between them. This means that $t_{a,s} + t_a + d(g_a, g_b) - t_{b,s} \leq 0$. Analogous this can be formulated if p_b is processed before p_a: $t_{b,s} + t_b + d(g_b, g_a) - t_{a,s} \leq 0$. Since the activities can not be processed simultaneously by the same vehicle, it must hold that: $\min\left(t_{a,s} + t_a + d(g_a, g_b) - t_{b,s}, t_{b,s} + t_b + d(g_b, g_a) - t_{a,s}\right) \leq 0$. If there is no conflict one of these sides will be smaller than 0, if there is a conflict, both sides will be bigger than 0.

With the introduction of mapping time to a finite horizon interval, we assume that all time points introduced into the MILP are smaller than 1: for any $t_{a,s} \leq 1$. With this formulation of time, it implies that for any set of two planes (a, b), it holds that $t_{b,s} + p_b + d(b, a) - t_{a,s} \leq 1$. Then if we sum $x_{m,a} + x_{m,b} - 2$ it must hold that if these are assigned to the same vehicle m the sum is exactly 0, if not assigned to the same vehicle the sum is either -1 or -2. If we add this together, we get that $x_{m,a} + x_{m,b} - 2 + \min\left(t_{a,s} + t_a + d(a, b) - t_{b,s}, t_{b,s} + t_b + d(b, a) - t_{a,s}\right) \leq 0$, this can be seen in equation 3.10. The resulting MILP is listed in MILP 3.2.

$$\text{minimise} \quad \sum_{a \in A} \min(t_{a,s} + t_a, t_{a,d}) \tag{3.7}$$

$$\text{subject to} \quad t_{a,s} \geq r_a \tag{3.8}$$

$$\sum_{m \in M} x_{m,a} = 1 \qquad \forall a \in A \tag{3.9}$$

$$x_{m,a} + x_{m,b} - 2 + \min(t_{a,s} + t_a + d_{a,b} - t_{b,s}, t_{b,s} + p_b + d_{b,a} - t_{a,s}) \leq 0 \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a \neq b\} \tag{3.10}$$

MILP 3.2: Formulation of simple deicing with continuous time variables

The resulting MILP can be seen in MILP 3.2. Now if we try to put this in our solver, we have problem with the $\min$ operation. Therefore we have to get rid of the $\min$ operators. This is done by rewriting the MILP. This rewriting is done by introducing extra constraints, displayed in MILP 3.3

$$\text{minimise} \quad \sum_{a \in A} y_a \tag{3.11}$$

$$\text{subject to} \quad y_a - t_{a,s} - t_a \geq 0 \qquad \forall a \in A \tag{3.12}$$

$$y_a - t_a^* \geq 0 \qquad \forall a \in A \tag{3.13}$$

$$t_{a,s} \geq t_{a,r} \qquad \forall a \in A \tag{3.14}$$

$$\sum_{m \in M} x_{m,a} = 1 \qquad \forall a \in A \tag{3.15}$$

$$x_{m,a} + x_{m,b} + b_{a,b} + t_{a,s} - t_{b,s} \leq 3 - t_a - d_{a,b} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\}, \forall m \in M \tag{3.16}$$

$$x_{m,a} + x_{m,b} - b_{a,b} + t_{b,s} - t_{a,s} \leq 2 - p_b - d_{b,a} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\}, \forall m \in M \tag{3.17}$$

$$x_{m,a} \in \{0, 1\} \qquad \forall m \in M, \forall a \in A \tag{3.18}$$

$$b_{a,b} \in \{0, 1\} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\} \tag{3.19}$$

MILP 3.3: Rewriting 3.2 to remove min operations

In this simplified MILP the equations 3.16 and 3.17 replace equation 3.10. Furthermore, to remove the minimise operation in the objective, we introduce equations 3.12 and 3.13. The Boolean variable in 3.19 denotes a happened-before relationship, if $b_{a,b} = 1$ then this means that aircraft p_a is deiced before p_b, note that this is only relevant for activities scheduled on the same vehicle.

3.2. Advanced Model for Deicing Activities

For the more advanced model we want to incorporate fluid usages. Each vehicle has a limited amount of fluid it can carry. Filling on a day of operation is possible, however this requires the fluid to be heated, which takes a considerable amount of time: over 30 minutes. We choose to only implement this model with a continous time variable, our experiments showed this a faster approach (section 4.2.1).

The extension of the simple model is done in two steps, first we extend the *model* in section 3.2.1. After we have extended the formulation we provide an updated MILP in section 3.2.2, in which we present a solution for implementing the fluid in vehicles. The assessment of this implementation is left for chapter 4, where we conduct a number of experiments on this advanced MILP formulation.

3.2.1. RCPSP Formulation for Deicing with Fluid Constraints

The advanced model, with the integration of fluid usage, is an extension of the model formulated in section 3.1.1. Each aircraft p_a requires a specific amount of fluid f_a. This fluid has to come from a deicing vehicle m, each vehicle m has the amount of fluid it can contain C_m. It must hold that for each vehicle the amount of fluid in a vehicle is never below 0, or over the capacity of that vehicle C_m.

In RCPSP literature, there are a non-renewable resources and renewable resources. The fluid in the truck resource, is somewhere in between, it is non-renewable as a truck can get empty, and it is renewable as a truck can be refilled. Therefore we introduce a concept we call the *semi-renewable resource*. This is a resource that can be used as a non-renewable resource, and with the use of a *filling* action can be restored to some preset value.

3.2.2. RCPSP Translated into Continuous Time MILP

In this section we present an extension to the MILP presented in section 3.1.4. This enables us to make realistic forecast, were we can include deicing fluid usages. The main challenge for this formulation is the translation of the semi-renewable resource.

The solution we present for the fluid is to model the fluid resource as a series of non-renewable resources. Each element of this series is called a *stint*. After filling, a truck can deice a number of aeroplane in its stint. Now we introduce for each vehicle ten stints, this means that we limit the number of theoretical refillings of each vehicle to ten, we believe this is enough for a capacity estimation. Instead of assigning aeroplane to vehicles, we assign aeroplane to a vehicle stint. For each stint we calculate the begin and end times, then we make sure there is enough time between stints such that refilling and reheating is possible.

The formulation in MILP 3.4 is a more complex than that of the simple MILP (3.3). The first part of the equation (3.21, 3.22, 3.23) are equivalent to previous formulation. The variable of plane to deicing vehicle assignment has an extra parameter i for denoting the deicing stint ($x_{m,a,i}$). Equations 3.25 and 3.26 are almost the same as the previous formulation. Equation 3.29 ensures that each deicing stint does not exceed the capacity (C_m) of each truck. Equations 3.30 and 3.31 calculate the interval of a deicing operation stint, so $m^+_{m,i}$ is the end of stint i for vehicle m, and $m^-_{m,i}$ is the start of the stint. Equation 3.32 makes sure the end of a stint is is after the beginning of the same stint. Finally equation 3.33 ensures there is enough time between stints for filling the vehicle with deicing fluid. For our implementation we can suffice with only ten stints per vehicle, this is an overestimation, since refilling takes at least 30 minutes, so refilling ten teams would yield that a deicing trucks spends five hours of the day refilling.

$$\text{minimise} \quad \sum_{a \in A} y_a \tag{3.20}$$

$$\text{subject to} \quad y_a - t_{a,s} - t_a \geq 0 \qquad \forall a \in A \tag{3.21}$$

$$y_a - t_a^* \geq 0 \qquad \forall a \in A \tag{3.22}$$

$$t_{a,s} \geq r_a \qquad \forall a \in A \tag{3.23}$$

$$\sum_{m \in M} \sum_{i=1}^{10} x_{m,a,i} = 1 \qquad \forall a \in A \tag{3.24}$$

$$x_{m,a,i} + x_{m,b,i} + b_{a,b} + t_{a,s} - t_{b,s} \leq 3 - t_a - d_{a,b} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\}, \forall i \in I, \forall m \in M \tag{3.25}$$

$$x_{m,a,i} + x_{m,b,i} - b_{a,b} + t_{b,s} - t_{a,s} \leq 2 - p_b - d_{b,a} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\}, \forall m \in M \tag{3.26}$$

$$x_{m,a,i} \in \{0, 1\} \qquad \forall m \in M, \forall a \in A, \forall i \in I \tag{3.27}$$

$$b_{a,b} \in \{0, 1\} \qquad \forall (a,b) \in \{(a,b) | a, b \in A, a < b\} \tag{3.28}$$

$$\sum_{a \in A} f_a \cdot x_{m,a,i} \leq C_m \qquad \forall i \in I, \forall m \in M \tag{3.29}$$

$$x_{m,a,i} + t_a + t_{a,s} + d_{a,f} - 1 \leq m_{m,i}^{+} \qquad \forall a \in A, \forall i \in I, \forall m \in M \tag{3.30}$$

$$1 - x_{m,a,i} + t_{a,s} - d_{f,a} \geq m_{m,i}^{-} \qquad \forall a \in A, \forall i \in I, \forall m \in M \tag{3.31}$$

$$m_{m,i}^{-} \leq m_{m,i}^{+} \qquad \forall m \in M, \forall i \in I \tag{3.32}$$

$$m_{m,i}^{+} + t_f - m_{m,i+i}^{-} \leq 0 \qquad \forall m \in M, \forall i \in \{1, \ldots, |I| - 1\} \tag{3.33}$$

MILP 3.4: Advanced model in a MILP formulation.

4

Experiments on Offline Pre-Tactical Exact Implementation

The aim of this chapter is to first show whether the exact Mixed Integer Linear Programming (MILP) implementation of the previous chapter can handle problem sizes that correspond to the Aviapartner case. Furthermore we look how this solution scales. This chapter therefore aims to find answers to two research questions; **RQ**2: *Can our exact formulation handle problems sizes which are representable for the Aviapartner practise?* and **RQ**3: *How does the exact formulation scale?*.

To answer these questions we perform experiments. First we explain our experimental setup, in which we show how we have built representative problems for the Aviapartner operation. Furthermore we list assumptions we had to make to obtain representable data. Finally we describe the setup in terms of the software we used.

Secondly, we provide the results of our experiments. First we show which type of MILP implementation is the most efficient. Secondly, we experiment on problems representable for the Aviapartner case. Finally, we review the effects of scaling the problem in size.

4.1. Experimental Setup for Offline Planning

The question we try to answer is to find out whether our exact MILP formulation can handle plannings the size of that of Aviapartner. This setup therefore deals with a number of challenges we have, the first is that we have to define, what we think are representable planning problems for the Aviapartner case. Secondly, we will show what kind of setup we use for solving a MILP, this is a combination of software and hardware.

4.1.1. Experiment Data Construction from Operational Data

From Aviapartner we obtained schedules for the flights they have under contract. Only a subset of their airlines have a contract for deicing. For example, it is possible that an airline has it ground operations contracted at Aviapartner, however the deicing is contracted at KLM.

The data we obtained consists of a number of key times for each flight. The historic data consists of the following data fields[1]:

1. *Operator* The airline that operates the flight, for example KLM.
2. *Aircraft Type* The aircraft that is used for this flight, for example an Airbus A320.
3. *Registration* The registration of the airframe, for example PH-LAB.
4. *Flight* The flight code used by the airline to identify the flight, for example KL001.
5. *STD* The Scheduled Time of Departure (STD) is the time the aircraft is scheduled to depart.

[1] an example is given in appendix C.1.

6. *ETD* The Estimated Time of Departure (ETD) is the last estimate given of the departure time.

7. *Block* The block time is the time the blocks were removed.

8. *Routing* The destination airport is identified as the routing.

9. *Gate* The gate is the location were the aircraft is stationed.

For our experiments a number of fields are important. First of all, the STD defines the scheduled time. As deicing is closely to the actual departure time, we use the STD as an estimate for when the deicing operation takes place. Furthermore, we take the gate field as the location of the aircraft, we therefore assume that the gate assignment follows a similar pattern in the future. Furthermore the aircraft type is used to define the size of the aircraft, which implies an estimate for the time and fluid needed for deicing.

Our setup can be summarised as follows, the dataset is divided into days. On each day we filter whether a flight is contracted by Aviapartner. Subsequently we map each flight to an instance for the input of our program. Each flight will have a release time and objective finish time that corresponds to the STD. The deicing time and fluid use are based on the size of the aircraft and the weather[2].

4.1.2. Software & Hardware

The MILP is implemented in Java, using the Gurobi[30] optimisation library. We used Gurobi version 6.5.0 in combination with Java 1.8 (update 25). The choice for this library has been made, after seeing that Gurobi outperforms GLPK. The default stop conditions for the MILP solver are set to only stop if the exact solution is found; the gap between the bound and the actual solution is 0%.

The experiments are run on an Apple MacBook Pro (A1398) with a 2.2 GHz Intel Core i7 processor, which has four cores, and eight threads, furthermore the total system memory is 16 GB. Gurobi used all eight threads in our experiments.

4.1.3. Experiments

Given the model we have built in the previous chapter, and the setup, we conduct the following three experiments on our platform. The experiments are briefly introduced, the last two experiments are directly related to the research questions.

1. In section 3.1 we introduced two notations for the simple model, one with discrete time formulation, and one with continuous time formulation. The introduction of a variable for each time-point for each plane vehicle combination causes a large number of variables to be formed. This amount of variables increases the complexity greatly. Consider an example of a day where we try to plan 100 aircraft with 6 vehicles on an interval of 1200 minutes, this will introduce $100 \cdot 6 \cdot 1200 = 720000$ variables, in a brute-force algorithm, this could lead to a complexity of $O(2^{AMT})$, where A is the set of planes, M the number of vehicles, and T the set of time-units.

   So we set up a simple experiment whereby we increase the number of activities (aircraft) to be planned. We want to find out which implementation, discrete or continuous, scales better. The discrete time formulation has a considerable amount of Boolean variables, however the continuous formulation also introduces Boolean variables in the 'happened-before' relations, and introduces continuous time formulation. So the experiment is aimed at finding out which implementation is better.

2. The second experiment is concerned with testing the performance for problem sizes that represent a day of deicing operations at Aviapartner. We run the planning algorithm using one day of input as defined in section 4.1.1. Furthermore we assume that each aircraft needs deicing (worst-case). A problem size that represents a day at Aviapartner typically ranges between 30 and 50 aircraft.

3. The third experiment is aimed to finding the limit to this exact solution. We therefore put flights on the same day of operation, while keeping the assigned time (hours, minutes) the same. We increase the size, until we run into boundaries of the system.

[2] For a detailed overview of the settings appendix C.3 can be used

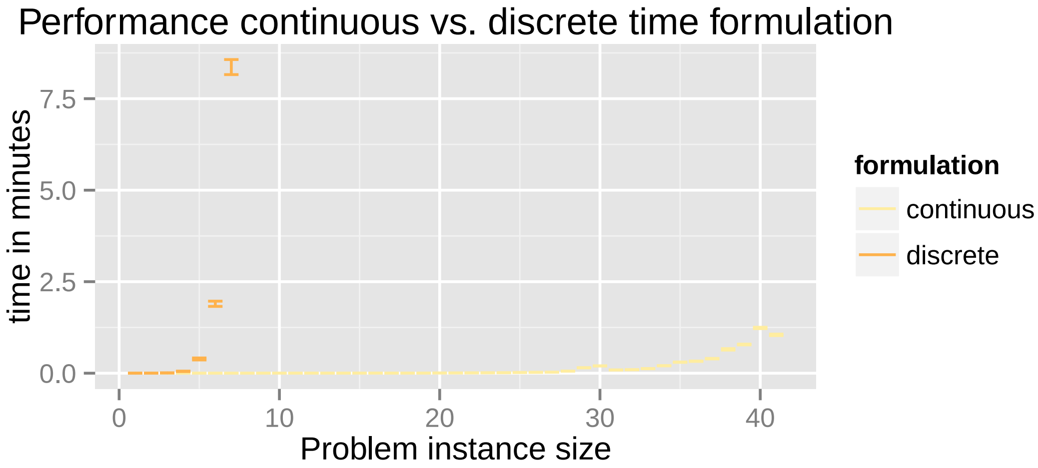


Figure 4.1: Experiment showing run-time for two exact implementations of the simple RCPSP model. Each problem size has been repeated seven times for each method.

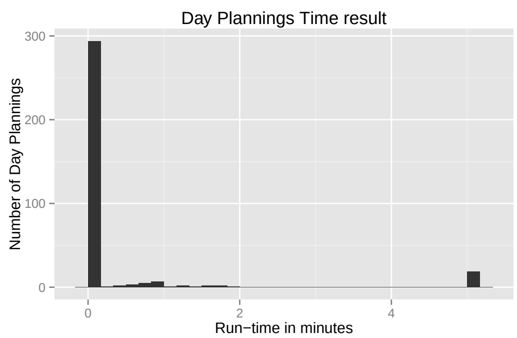


Figure 4.2: Day instances run on the exact MILP solver, note that the bar on the right side represents instances that exceeded the time limit of five minutes.

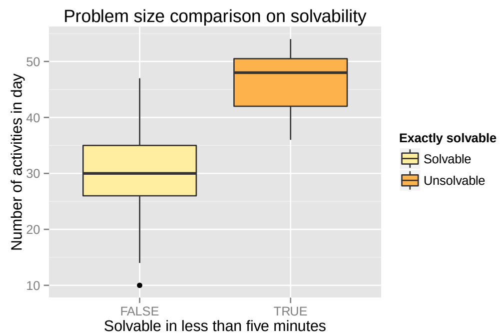


Figure 4.3: Size of the instances vs. the solvability under five minutes.

4.2. Experimental Results

The three experiments proposed in the previous section are run on our platform. We first list the results of the time formulation experiment. Secondly, we list results of problems that are based on day operations. After that we describe the results for when we increase the problem size.

4.2.1. Time formulation: Discrete or Continuous

The start of the experiments is to find out which approach is better in time, we compared the MILP formulation in discrete time (section 3.1.3) against an implementation which uses time in a continuous domain (section 3.1.4). The implementation of both algorithms has been done using the GLPK solver[3].

The results of this first experiment can be seen in figure 4.1. The graph uses error bars with a width of two standard deviation, each bar is the result of seven measurements. From the graph we can clearly derive that a MILP implemented in a continuous time domain outperforms one implemented in a discrete time domain. Therefore we presume that a continuous time scale is better for our advanced model.

4.2.2. Day Instances Solvability

For the day instances we used a pre-defined time limit of 300 seconds; the Gurobi solver had a max run-time of 300 seconds. This is done to prevent the computer from calculating too long. A pre-tactical

[3] https://www.gnu.org/software/glpk/

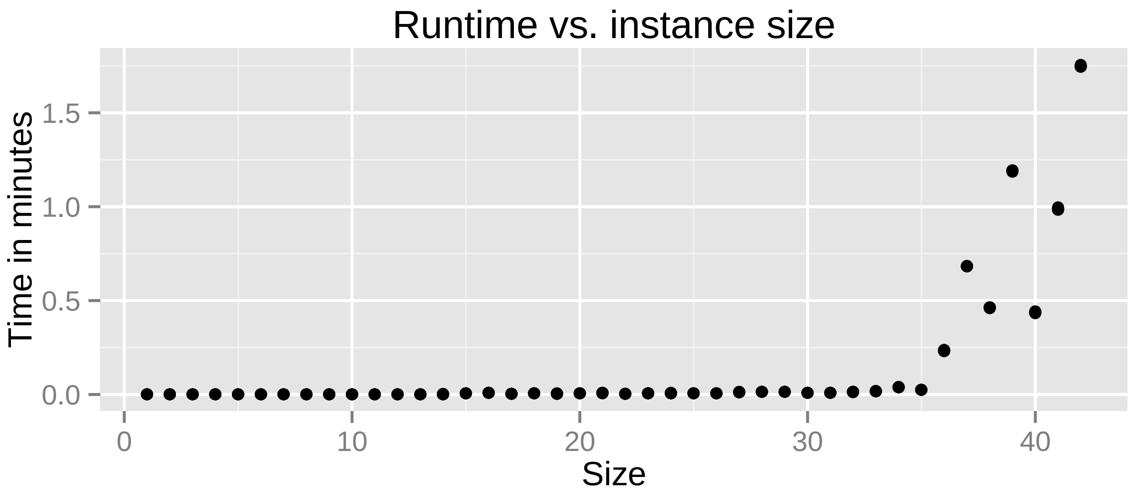


Figure 4.4: Problem size versus the run time of the MILP. Note that the next problem with size 43 took more than 5 hours to compute, therefore we left this out of the graph.

planning is typically made the day before the day of operation. The amount of calculation is limited, as a person has to review the planning, and take appropriate action upon it (e.g. planning personnel). Therefore we limit the time to something that can be done before the day of operation. We presume that 300 seconds is enough to make such a planning.

After we have ran the experiments, we plotted the results. The first plot we make with the data can be seen in figure 4.2. From this graph we can conclude that 20 of the 341, or less than 6%, of the day planning problems are not solvable by MILP implementation.

Further investigation led to the conclusion that most instances which were unsolvable, were rather large. Figure 4.3 gives an overview of problem sizes for solvable and unsolvable problems. The next experiment will further look into this bound on the problem instance size.

4.2.3. Size Limits on the Exact Offline Implementation

The results from the second experiment can be seen in figure 4.4. For each instance size we have repeated the experiment a total of seven times, the plot displays each experiment as an individual dot. Note that there is little spread for each problem size.

The graph shows that an problem with size 42 is solvable within two minutes[4]. The results suggest that there is an exponential relation between the problem size and the run-time. This corresponds to our earlier findings that the DEICING problem is a NP − HARD problem.

In this section one can see the limits of the exact approach: the MILP implementation can handle up to about 40 aircraft. The five minute requirement is a constraint that lets the end user still interpret a made planning. Some instances are not exactly solvable. Therefore we want to look at approaches that can handle larger instances.

Looking at alternative approaches for pre-tactical offline planning has two main reasons. First of all, we want to able to provide solutions for all problem sizes that Aviapartner faces. Secondly, Aviapartner is not the only operator facing the problem of planning deicing activities, therefore we want to see if there are solutions for problems larger than we have seen at Aviapartner.

Therefore it can be useful to look at non-exact approaches. If we discard the requirement that the solution should always be the most optimal solution to some objective, we can make solutions that try to be near optimal. The requirements for such an approach is that it takes the same inputs, and still produces a valid output. The valid output implies that there is a valid schedule produced by this

[4]The reader might notice that time reported for an instance of size 42 is less than two minutes Therefore he might wonder why we have not tested a larger instance, since there would be enough time during a thesis project to conduct these experiments. The problem that occurred was that on instance size of 43 there seemed to be a 'very' complex problem, as calculating that instance was cancelled after five hours of computing.

methods, which adheres the constraints. However, the objective might be worse than that of an exact solution.

One possible approach is to formulate a heuristic[13], such that planning can be done in a more efficient manner, these solutions are not guaranteed to be optimal. Heuristics are algorithms that work with a set of planning rules that generally speaking will provide a good objective value. Usually solutions are found much quicker than that of an exact solution. Related to our work, and our case, we introduce an extra, refined research question. We will answer this research question in the next chapter:

RQ3' *What type of heuristics can be used for the deicing problem?* Heuristics can be use to build non-exact planning algorithms. For our deicing case, we want to see if heuristics can be used. Heuristics are promising, as they do not suffer from the size limits as an exact algortihm does.

5

Heuristics for Offline, Pre-Tactical Planning

The previous chapter ended with the notion that our exact solution in a Mixed Integer Linear Programming (MILP) formulation can only handle a limited number of aircraft. As we might be interested in faster solution, we will use this chapter to introduce heuristics methods.

Heuristics are methods that use a set of planning rules that are aimed to provide a solution towards a certain direction. For example a *minimise cost* heuristic is reluctant to plan an extra deicing vehicle, as this would give a higher cost. Often heuristics can produce good results, for example in the classical computer science problem of Partitioning[21]. The trade-off for heuristics is that sometimes optimal solutions cannot be found. The main advantage of heuristics is that they are considered much faster, often heuristics scale polynomially.

First, this chapter will introduce a set of metrics on which we judge the performance of the heuristic algorithms. In the previous chapter we used run-time as a classifier on the quality off the algorithms. When using heuristics run-time is not useful, since heuristics are rather fast. Therefore we provide metrics based upon the Key Performance Indicator (KPI) identified in section 1.10.3

Secondly, we describe a number of heuristic methods, these methods will be based upon improving one specific KPI. Finally we test these methods, with a comparison on how the heuristics perform on specific KPI, and compare this to an exact solution. These KPI include the original tardiness objective, this means that we suggest an heuristic that has set the planning rules such that it should work toward the same objective as the exact solution. The previous chapter only used one KPI, tardiness, as the chapter aimed at looking if for a KPI an exact solution could be found. In this chapter the scope is widened to include the three main identified KPI.

5.1. Translating Objectives into Metrics

Section 1.10.3 provided a list of objectives for the planning problem. This section translates the objectives into constraints and optimisation functions. For this we introduce new ways of measuring the performance of planning solutions. The main objective will always be safety, therefore we model this a constraint and not as an objective. With safety as a constraint we ensure that every aircraft can receive deicing treatment. We provide three metrics: profit, Quality of Service (QoS), and cost in the following three sections.

Profit

Every deicing will incur a cost and make revenue. The costs are composed of the cost of labour, gasoline, deicing fluid use, and redemption of deicing trucks. Labour is one of the largest costs for deicing, and one we can optimise for.

The overhead time for starting up a vehicle, heating the fluids, and refilling it, is approximately an hour. This means that if we find a way to start up less vehicles, we can lower the costs.

We express costs in the function, of start-up, and running costs. We assume that labour follows from the time a vehicle is active, therefore we set the cost at running a vehicle at 100 units per hour.

Slack (minutes)	Grade
0-10	1
10-20	7
30+	10

Table 5.1: Scoring table for Robustness metric

The start up cost of a vehicle (e.g. using a vehicle) is set at 100 units, the cost of starting up, and post-operation activities.

QoS
Reputation of the ground handler depends on the Quality of Service (QoS). The reputation of a deicing operator is mostly determined by the on time performance of the operators. Therefore we use tardiness as a metric for QoS. Each activity has an optimal TSAT, each minute a deicing is late, will incur tardiness. The total amount of tardiness is used as a metric for QoS.

We make an assumption that any delay past ten minutes will incur a cost to the airline. If we take the most recent cost benefit analysis document from Eurocontrol[34], we can see that the overall cost of delay on the ground is an average of 90.8 EUR per minute (page 14). So any delay past the ten minute threshold will incur this cost. This cost is incurred to the airline, and not to Aviapartner. However, if the operations of Aviapartner leads to this cost for the airline, we can see this as a damage to the Quality of Service (QoS).

Robustness
The Robustness is given by the amount of changes needed for adopting to operational influences. As a metric we can take the amount of time between deicing activities. The more slack between activities, the more stable the system is. If for example an aircraft is delayed, and therefore requires deicing ten minutes past his planned time, then slack can compensate for this delay. This means that other aircraft that are planned to be deiced by this vehicle are still able to be deiced at their planned time. A more robust planning will lead to more stability, which is desirable for an operation that copes with a considerable amount of changes.

For this system we want to end up with a metric which gives an easy to understand value for the end-user. We therefore give a classification based system, which then is translated into a grade (1-10). From each grade derived using table 5.1 we then take the average, to end up with a final mark for the robustness of the planning.

5.2. Heuristic Planners

For each metric we provide an corresponding heuristic algorithm to plan deicing activities. First we provide one for minimal cost, than for maximal Quality of Service (QoS), and finally for maximal robustness.

Cost Heuristic Algorithm
The main factor attributing to the cost is the usage of deicing trucks, we therefore try to minimise the usage of such vehicles, such that there is a minimum of personnel required. The algorithm is displayed in algorithm 5.1.

The algorithm keeps two sets, one set for unused vehicles V, and one for vehicles in operations V'. This algorithm first looks if there is a vehicle available, which is already used, which can deice an aircraft with a maximum delay of 15 minutes to the aircraft. If not, a new vehicle is used, if that is not available there is a first fit at the available trucks. There is a threshold on the amount of fluid the vehicle should at least contain. If the amount fluid falls below the threshold after an deicing activity, the vehicle has to refill.

QoS Heuristic Algorithm
The QoS heuristic algorithm is essentially the same as the cost heuristic algorithm. We adapt the algorithm in the first if statement, such that there is no delay possible. This means that one would use an unused vehicle earlier. This also could prevent a ripple effect, as delays might cause delays.

$V \leftarrow$ vehicles
$V' \leftarrow \emptyset$
$P \leftarrow$ planes sorted by scheduled departure time
$\ell \leftarrow$ threshold for deicing fluid (e.g. 1000ℓ)
for all $p \in P$ **do**
    v^*
    if $\exists v \in V'$ at scheduled time of p minus 15 **then**
        $v^* \leftarrow v$
    else if $|V| > 0$ **then**
        $v \leftarrow$ one element of V
        $V \leftarrow V \setminus \{v\}$, $V' \leftarrow V' + \{v\}$
        $v^* \leftarrow v$
    else
        $v \leftarrow$ first available $v \in V'$
    end if
    assign p to v^*
    if fluid level of $v^* \leq \ell$ **then**
        Assign refilling to v^*
    end if
end for

Algorithm 5.1: Minimal cost heuristic algorithm, that provides a solution to the deicing problem. Minimising cost is done by not requiring extra vehicles.

Robustness Heuristic Algorithm

For the robustness heuristic algorithm we want to design an algorithm that will produce a schedule that can withstand changes the best. As a heuristic, we want to maximise slack, slack allows us to adapt to a changing schedule the easy. An overview of the implementation of the maximise robustness algorithm can be seen in algorithm 5.2.

The algorithm is quite similar to that of the cost heuristic. Only instead of minimising the time between activities, we now want to maximise the time between them. Therefore we first assign aircraft to vehicles that do not contain any activities yet. After this phase, we assign vehicles to the one with the longest inactivity.

5.3. Experimental Results for Heuristic Pre-Tactical Planners

In this section we investigate how an exact MILP solution compares to heuristic solutions. First the exact MILP planner builds a solution for a day of Aviapartner data. The heuristics make a planning on the same data. We compare the solutions on the heuristics as described in the previous sections. For each of these four plannings we calculate the performance on three metrics.

The results can be seen in figure 5.1. The results are displayed in a box-plot. The 1 on the vertical axis means that a certain metric solution performs as good as the exact solution. A score lower than 1 means that the metric outperformed the exacts solution for that specific metric.

It should hold that no metric should outperform the exact solution on the *tardiness* metric, the exact solution uses the *tardiness* as an optimisation criterion. From the figure we can see that no heuristic outperforms the exact solution on tardiness (e.g. the first, fourth and seventh box-plot).

The Max Quality of Service will provide for some problems a better cost, which means the total *active* time of the deicing vehicles is lower. We see indeed that the robustness criterion is better than that of an exact planning for most input problems. So the max-QoS policy will in general give an better cost and more robustness.

The max robust algorithm shows the lowest numbers for the slack, so we can state that it is indeed the most likely to produce a robust schedule, judging from generated slack. However, we see that there is higher cost for generating this robustness. The cost is for all problem instances higher than the exact solution. Also we see that the cost is the highest of all heuristics. Therefore we conclude that if one uses a max robustness algorithm, there is a trade-off between robustness and cost.

The min cost algorithm will most likely give the lowest cost of all policies. The cost is the lowest of all heuristic solutions, and it is for most instances much lower than the exact solution. Also for this

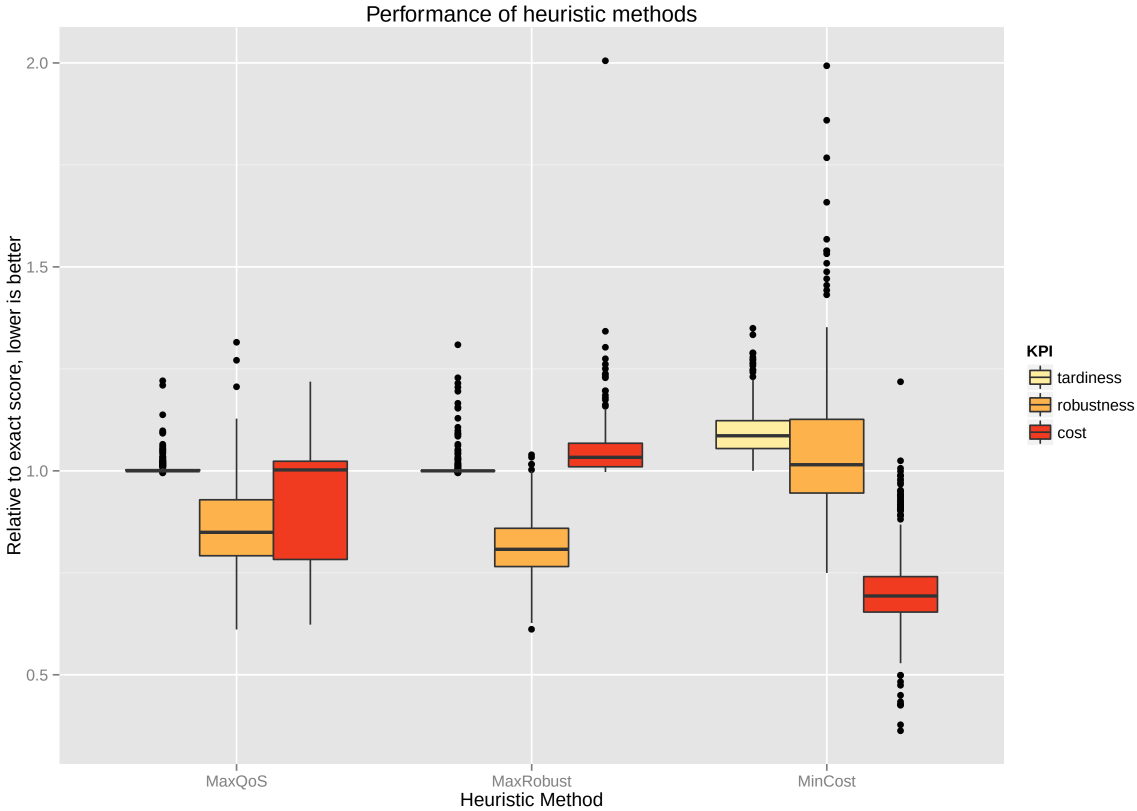


Figure 5.1: Heuristic performance, grouped by algorithm, for each algorithm we show three box-plots how it relatively performed to the exact solution. For all metrics, a value lower than 1 means that it outperformed the exact solution on that metric. Generally speaking, a score lower than one is more desirable, since that means the algorithm performs better on that metric to the exact solution on that metric.

```
V ← vehicles
V′ ← ∅
P ← planes sorted by scheduled departure time
ℓ ← threshold for deicing fluid (e.g. 1000ℓ)
for all p ∈ P do
    v*
    if V ≠ ∅ then
        v ← one element of V
        V ← V \ {v}, V′ ← V′ + {v}
        v* ← v
    else
        Sort V by non-decreasing completion time of activities
        v* ← first element in V, minimise slack
    end if
    assign p to v*
    if fluid level of v* ≤ ℓ then
        Assign refilling to v*
    end if
end for
```

Algorithm 5.2: Algorithm for planning deicing activities while maximising the robustness. Robustness is interpreted as the time between consecutive activities.

algorithm there is a trade-off, since the tardiness is worse than the exact solution, and the tardiness is the highest of all heuristics. This is possible since sometimes cost can be reduced by not employing a vehicle, so preventing start-up costs and extra refilling time. By design, the min cost heuristic can introduce a delay for not starting up a new vehicle, this will limit the cost, but a delay inherently will worsen the tardiness score.

We see that there are trade-offs between the various metrics. The max robustness and min cost outperform heuristics and the exact solution on the metric they are designed for. The max quality of service will deliver a schedule that is likely to have the same tardiness as the exact solution. This is also true for the max robust heuristic, it is likely to set the same tardiness scores as the exact solution.

Apart from the academic work, we have built a web interface to interact with the planning algorithms. Figure 5.2 shows the planning screen. It shows that a trade-off can be made for various heuristic algorithms on KPI scores.

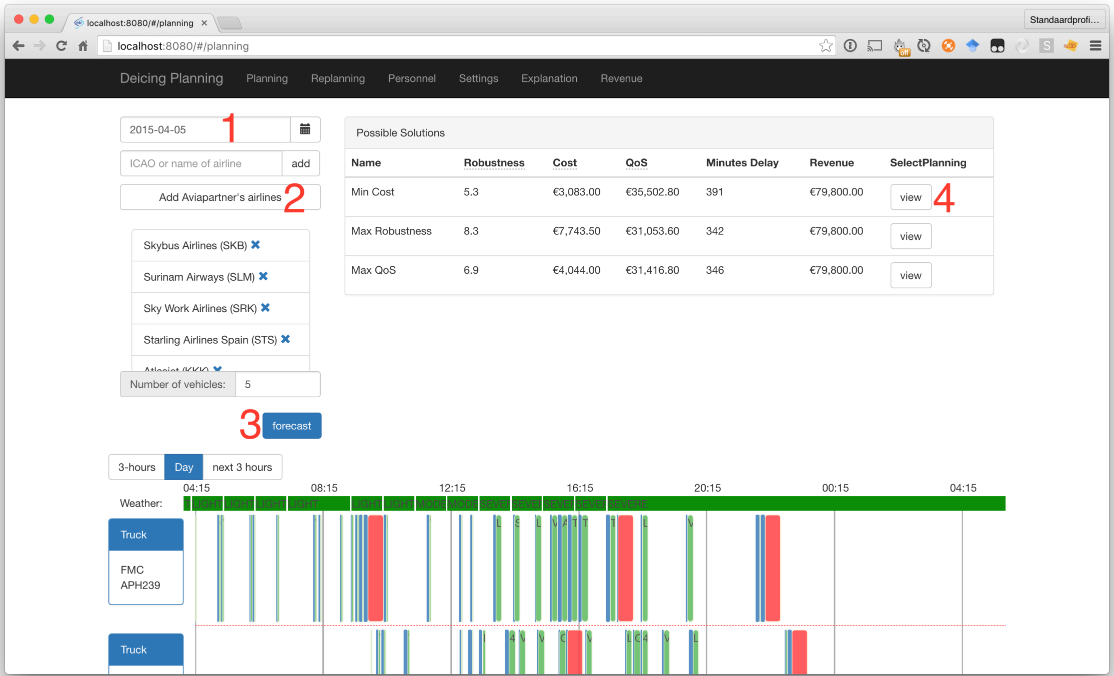


Figure 5.2: The graphical user interface, to interact with plannings. The red numbers indicate the steps to take to interact with the program. 1) one selects the date to plan deicing activities on, 2) one selects the airlines to include in the planning, 3) the 'forecast' button will give results of various planning algorithms on the right side of the screen, and 4) one can select a planning and with 'view' one can view a more detailed view of the planning at the bottom of the screen.

6

Online, tactical planning model and experiments

For now, we have limited our work to pre-tactical, offline, planning approaches. We have build a model, and we have showed exact and non-exact implementations for this model. This chapter will look at tactical (online) planning, this is the planning which is done in a dynamic environment, there can be disturbances from the operation that give the need to alter the planning.

For our work we consider a number of distinct disturbances. A disturbance is an event which gives a need to adapt an existing planning. The disturbances we consider are based on the operations of Aviapartner. These disturbances, shown in table 6.2, are categorised by name, a description, and we describe the effect on the planning.

There is a difference between the 'vehicle' disturbance and the other disturbances. If we consider the relation of vehicle-aircraft assignment in a baseline schedule, these need to be broken in case of a vehicle breakdown. If we consider a situation where a aeroplane a is scheduled to be deiced at 6 PM, and the vehicle breaks down at noon, then the aeroplane has to be deiced by another deicing vehicle.

This chapter will look into work that relates to dealing with unexpected events, when a baseline schedule is present in section 6.1. Then we provide five methods for dealing with these uncertainties in section 6.2. The setup of our experiments to test these approaches for dealing with disturbances is given in section 6.3. The results of these experiments are in section 6.4.

6.1. State of the Art

The challenge we face is coping with that of a changing schedule. This section reviews a number of approaches found in literature, which relate tot the replanning of deicing activities. The main work we found was in the field of multi-agent setting.

6.1.1. Re-planning an Multi-Agent setting

A Multi-Agent System (MAS) is a system composed of several agents which jointly commence in an operation. MAS[10] is a different paradigm than a centralised approach. A centralised approach considers a single actor, which makes decisions for all processes upon the information it has available. In a MAS setting, information can be distributed amongst actors which make. Traditionally deicing operations are centrally coordinated. For example, we end up with a MAS if we let the deicing trucks decide what deicing operations they operate.

A MAS offers a number of advantages, if the amount of information is more limited to an individual actor than a centre planning instance, a individual actor has less to reason about. Furthermore, message complexity can be reduced as not all information has to be shared. For planning and re-planning purposes it is possible to let agents for themselves decide the order of operation. One of the drawbacks of a MAS setting is that you can miss optimal, or better, results as one would get in a centralised approach. A MAS can create an information asymmetry.

Komenda et al.[18] review a setting where there are agents that have a plan, and then agents have to repair this plan. The plan has to be altered because unexpected events occur that cause the need for change. In this setting agents make actions to traverse among states. An agent has a begin state and a final state which he wants to reach. By doing actions an agent can change states, and end up in the final state (goal). If a disturbance occurs, the planned path is not possible anymore, either a transition between states, or a state is missing. To deal with these disturbances, the authors devised a number of approaches. The approaches vary in the degree of repairing actions. The first algorithm the authors introduce is a Back-on-Track algorithm, this algorithm uses a MAS-planner to repair the states missed, and continue the rest of the plan. The second algorithm, called *lazy*, just tries to get the actor in such a state that it can execute the remainder of the plan (*suffix*). To do this a *prefix* is calculated, such that after the *prefix* the *suffix* can be executed. This *lazy* repair might entail that not all tasks are executed.

This idea of keeping part of the original plan, or making a new plan is the central idea we focus on in this chapter. The previous chapters discussed how one can make a offline baseline schedule. This chapter will introduce a number of methods that can deal with disturbances. They will vary in the degree they use the baseline schedule.

6.2. Dealing with Disturbances

When a base schedule is ready, a planning of deicing vehicles and aircraft, operational factors might require to adapt the schedule. For instance if a plane is delayed, the deicing operation has to be delayed as well. To deal with these operational concerns, we present three classes of strategies for changing the baseline schedule to a realistic schedule. Generally speaking there is a limited amount of time to calculate a renewed feasible schedule, since changes happen close to the actual time of operation. The degree they vary in is how much they use the baseline schedule. The three classes are as follows:

- C1 **Adaption - Save plan** With repair actions, you try to get back to the original plan, for instance in case of a delay, you re-plan that aircraft, and stick to the original plan for the other aircraft.
- C2 **Heuristics to save plan** One reschedules aircraft, such that most of the plan remains intact, this can be in case of a delay that several aircraft are rescheduled.
- C3 **Heuristics without use of original plan** One discards the original plan, and re-plans every action needed to end up again with a feasible plan.

With these classes in mind we introduce a number of approaches we devised for handling changes in the plans. For each method we identify the class, and which disturbances it can handle. We list a detailed description of each method. Table 6.1 lists five methods for dealing with delay. Each of these methods is used in a delay and weather severity increase scenario. The vehicle breakdown can only be handled by two approaches, therefore only these two are used in that scenario.

6.3. Experimental Setup

There are a number of disturbances we want to test the approaches of the previous section. The disturbances are based on the real world scenarios. We use the disturbance formulated in table 6.2. The next three subsections deal with each type of disturbance.

6.3.1. Delay Scenario

We have obtained a dataset where for every flight a number of key times are listed. We take the Scheduled Time of Departure (STD) as indicator for a baseline schedule. We believe that an updated, and therefore 'final' Estimated Time of Departure (ETD) is representative for an actual time of deicing.

If we want to be precise, we should use the off block time as a start point for deicing operations. We were not able to obtain significant amount of scheduled, estimated, and actual off block times. The difference in time to the time of departure is assumed to be constant. Therefore we can assume that the time of departure can be used to make pre-tactical plannings, and take the ETD as representative for the delayed time.

6.3.2. Vehicle Breakdown Scenario

The vehicle breakdown scenario is based upon real world events at our ground handler. During interviews we learnt that it frequently occurs that one of their vehicles has a breakdown on the day of

Name	Class	Can deal with Vehicle Breakdown?	Description
Simple (naive)	C1	no	From the baseline schedule we keep the assignment from aeroplane to vehicle intact, and we keep the order of the planes intact. This means that if there is a delay for an aeroplane, this can cause delays for other aeroplanes
Local Optimal	C2	no	From the baseline schedule, the assignment of vehicles to aircraft is kept, only the order of these operations per vehicle is changed. This is done using a first-come-first-served approach. Refilling is planned using a threshold.
Local change (Look Ahead)	C2	no	This is almost the same as the previous method, so from the baseline schedule, the assignment of vehicles to aircraft is kept, only the order of these operations per vehicle is changed. Now we use a look ahead to plan deicing, and refilling.
Globally (FCFS)	C3	yes	The baseline schedule is no longer needed in this method. Each deicing is rescheduled on its own; for each job a suitable vehicle is found. There is no look-ahead. Refillings are planned using a threshold, e.g. if the amount of fluid is below 600l in the vehicle, a vehicle is sent for refilling. This method uses a First Come First Served (FCFS) approach.
Look Ahead (Globally)	C3	yes	One discards the baseline schedule, and replans activities with a heuristic. There is a limited look-ahead, a number of future jobs (e.g. ten) are taken into account when planning deicings and refillings.

Table 6.1: Methods for dealing with changes in the schedule. The classes refer to how the baseline schedule is used.

Name	Description	Effect on planning
Delay	The target take-off time is delayed, this can be due to outside events causing the plane to depart at a later time.	The deicing operations have to be done just before take-off, therefore the operations have to be done later than planned.
Vehicle breakdown	A deicing vehicle breaks down. This can be due to malfunctioning equipment.	The deicing capacity is reduced, other vehicles have to take the load of the missing vehicle.
Weather severity increase	The weather gets worse, the conditions change such that the deicing operations take longer, and might require additional fluids.	The vehicles are able to deice less planes per hour, and require more frequent refilling.

Table 6.2: Table with disturbances

operation. Such a vehicle breakdown usually means that the vehicle will be out of operation for the rest of the day. We therefore introduce a vehicle breakdown scenario, in which the vehicle breakdown occurs at noon on a day of operation.

6.3.3. Weather Severity Increase Scenario

Another possible scenario is that the weather is increasing on the day of operation. This can have as an effect that the operations take longer, and more fluid is needed for the deicing.

We model this as an increase by assuming a duration increase of 25%, and a fluid usage increase of also 25%. We do not have enough detailed information to make better assumptions of the effect of weather severity increases. We think the effects on fluid usages and duration are left for other disciplines.

So in short We take day traces of Aviapartner's operation. For each day we make a baseline schedule based on all flights in the schedule, based on the Scheduled Time of Departure (STD).

For our experiments we make an exact approach with Gurobi. We limited the execution time to 60 seconds, as the MIP-gap mostly is very small at this moment. After this we run heuristic algorithms for the various scenarios.

6.4. Results of Replanning with Disturbances

This section reviews the results of the experiments proposed in the previous section. We adhere the order that was provided in the previous section. So first we show the results for the delay scenario. Secondly for the vehicle breakdown scenario. Thirdly, we show the results for the weather severity increase scenario.

The delay experiments are displayed in figure 6.1. We used box plots to display the distribution of hours delay. On the vertical axis the total hours of delay is plotted, from an operational view, fewer delay is desirable. Judging from the graphs we can see that the simple methods yields sub-par results, especially the third quartile is way above the other third quartiles. The Look Ahead method outperforms any other method. In the look ahead we used a look ahead of ten instances.

Figure 6.2 shows the results of the scenario with vehicle breakdowns, there is a small improvement from going to a globally optimal to a look-ahead strategy. The vehicle breakdown scenario can only be solved by algorithms that have a global scope. For this scenario we have to break some vehicle-aircraft assignments, as a vehicle is broken down.

The last scenario is to do rescheduling with a weather severity increase. We plotted the success of each method on the scale of total hours delay. Note that there is always delay, as we measure delay from the original departure time. We have made box-plots in figure 6.3.

For this scenario we see that a look-ahead outperforms the other strategies. Especially the local change, and the simple policy perform worse. The look-ahead policy can, by design, take more of the future into account. This is therefore more likely to outperform other methods.

In all scenarios we see that a look-ahead outperforms the other strategies. By design, The global

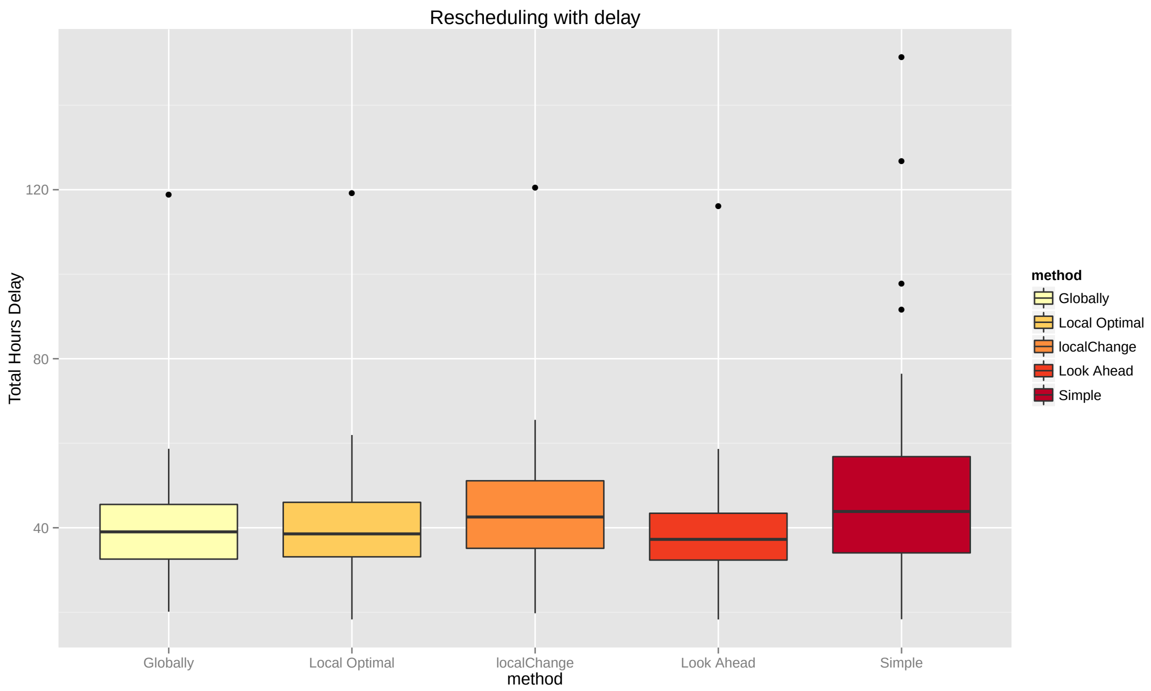


Figure 6.1: Performance for delayed planes, with several methods tested

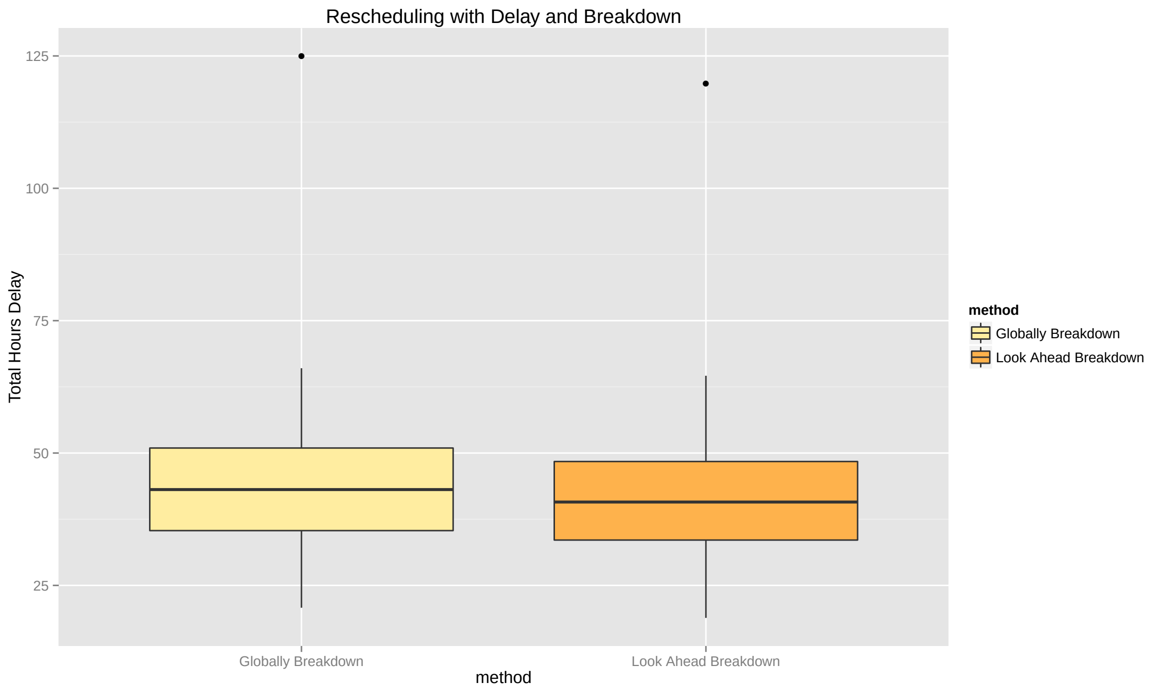


Figure 6.2: Performance of tactical methods where a vehicle breaks down at noon.

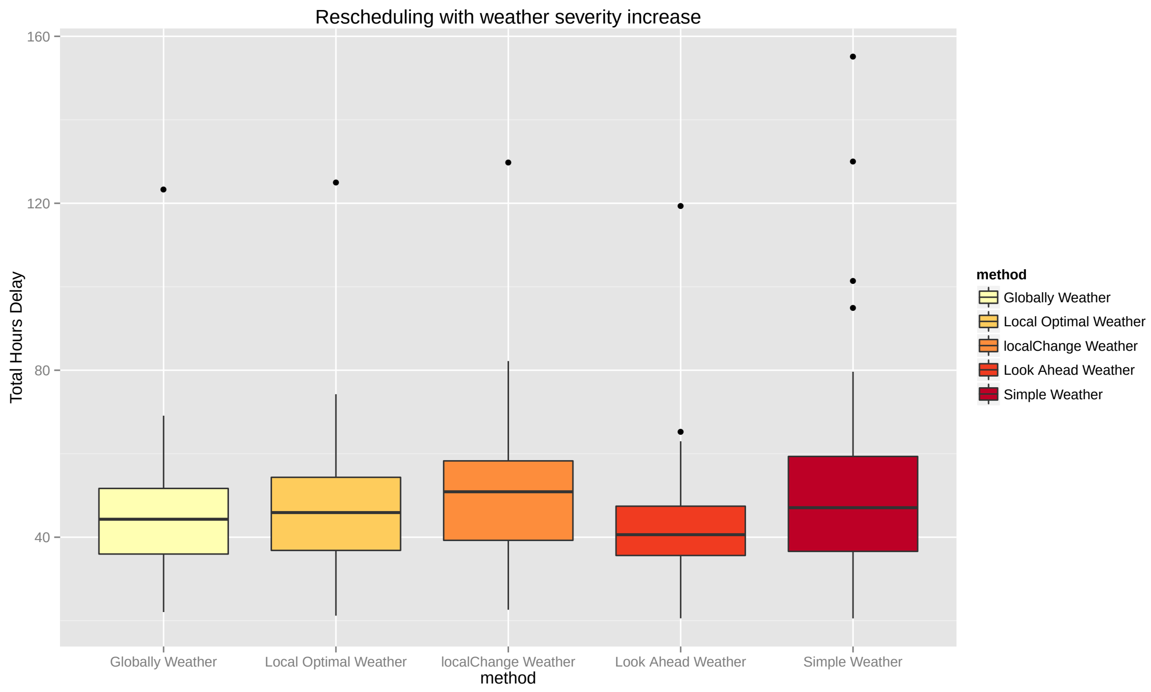


Figure 6.3: Performance of rescheduling methods when the weather increases in severity.

optimal strategy is effectively a look ahead strategy where the look ahead is one. Therefore it stands to reason that the look-ahead will outperform the global algorithm.

7 Conclusion

The work started by introducing a problem of how to optimise the deicing activities at Aviapartner. We hypothesised that making a detailed planning can lead to a more accurate capacity estimation. From investigating the operations at Aviapartner, we have listed a number of practical problems. For the work of this thesis we derived, on the basis of the practical problems and existing work, a number of research questions.

In the conclusion we answer these questions in reverse order; we first give answer to the research questions in section 7.1. From this we try to give answers to the practical problems in section 7.2. We discuss our results and provide indications for future work. Finally we provide recommendations for the Aviapartner operation.

We first looked into related work from the field of deicing and turn-around planning. The work in deicing planning looks at deicing scheduling from another perspective, that from the captain. The turn-around planning looked at the whole turn-around process, and used non-exact approaches for estimating the resource capacity need. We therefore extended our scope to the Resource Constrained Project Scheduling Problem (RCPSP). We state that the RCPSP is universal enough to be used in our deicing problem. We used extensions of these RCPSP to model our problem, such as multi-mode scheduling, non-renewable resources, and transfer times.

The main method for solving an RCPSP in an exact manner is to use a Mixed Integer Linear Programming (MILP) implementation. For this mathematical formulation there exist many solvers. For our work we mainly used GLPK and Gurobi. Due to the complexity of a MILP, there is usually a limit on how many activities we can schedule exactly with a MILP from an RCPSP model.

7.1. Research Questions

In this part of the conclusion we recapitulate answers to the research questions. We will do this in the order they were presented in chapter 2.

RQ1 *What is a suitable model for deicing operations, and how can one solve such a model?*

We have built an RCPSP model for our deicing operation. We showed that the deicing problem is $\mathsf{NP-HARD}$. That coincides the statement from academic literature that RCPSP models are hard to solve.

One of the main challenges we noted while reviewing academic literature, is that to our believe there is not something that can model the deicing fluid in the deicing vehicle. This fluid is on one side a non-renewable resource, as a vehicle can get empty, and on the other side a renewable resource, as it can be refilled.

We modelled the fluid resource with the usage of what we call stints. We introduced the concept of a semi-renewable resource, and modelled it as consecutive non-renewable resources. Each non-renewable resource has its own specific stints. With the help of constraints we make sure the stints do not overlap, and make sure there is enough time between stints to model the refilling and heating time. In a stint the fluid acts as a non-renewable resource, and between stints the vehicle is refilled.

Driving between gates is modelled as transitions for resources. Vehicles have to transition, drive, from gate to gate to apply deicing treatment. The driving times were implement with an approximation function.

For the objective we used the minimal tardiness. Tardiness is incurred if an activity finishes past its deadline. Minimising tardiness is equivalent to minimising the delay.

We state that a suitable model for deicing operations is an RCPSP. We extended the RCPSP to include the constraints and objectives of the deicing problem. We showed the corresponding MILP formulation, such that it can be solved by a MILP solver.

RQ2 *Can our exact formulation handle problems sizes which are representable for the Aviapartner practise?*

We used a real-world trace dataset from Aviapartner to find answers to research questions. Problem instances were derived using the Scheduled Time of Departure (STD) as a time when deicing operations should commence. The problem sizes for a day of deicing operations ranged from 30 to 50. The solver was set to a calculation limit of five minutes. Our implementation could solve 94% of these problem sizes exactly. The other instances were solved, only with a remaining *gap* of $< 1\%$, and therefore not completely solved.

RQ3 *How does this exact formulation scale?*

We used the same data as in the previous research question, only we scaled the size of the problem size. The expectation is that the time needed for solving instances scales exponentially with the problem size, since it is NP-hard to solve.

The experiment we performed confirmed our expectation that the time needed to solve increasing problem sizes scales exponentially. In our experiment problem sizes up to 42 were solvable, this confirms the previous finding that not all problem sizes of Aviapartner are solvable exactly.

We therefore extended our search to approaches that do not solve the deicing RCPSP in an exact manner. We therefore looked at *heuristics*, and introduced the following research question.

RQ3'*What type of heuristics can be used for the deicing problem?*

We introduced three different planning heuristics, which were all based upon key performance indicators (KPI) derived from the practises at Aviapartner. Since the run-time of these algorithms is negligible, we assessed the performance of the algorithms on the three performance criteria. We showed that the three policies performed best for the key performance indicator they were designed for. Heuristics scale better than exact solutions, therefore heuristics can make plannings for larger day instances.

RQ4*How can/do plannings cope with operational uncertainty?*

With the use of interviews we devised three scenarios, for what we believe are representable of disturbances found at Aviapartner. The three scenarios we identified are: delay of deicing ready time, vehicle breakdown, and weather severity increase. The delay scenario is modelled after historical traces from Aviapartner. For the weather severity increase and the vehicle breakdown, we provided our own interpretation on the effects on the conditions for these scenarios.

We introduced five methods to deal with these disturbances. These methods varied in the amount of data they used from the baseline schedule. In a simple approach both the order of deicing activities and the assignment from vehicle to aircraft is kept, while respecting delay. In more loose scenario, the order is not kept, so there is change possible on a vehicle. In a global scenario, the baseline schedule is discarded, and a new planning is made. For the latter two, we introduced a *look-ahead* scheduling methods, which makes a heuristic planning on a partial view of the future.

We see that the best deal for dealing with disturbances is to use a look-ahead planning, this means that one re-plans the coming 10 activities using a heuristic. The naive approach, to change as little as possible produces appealing results.

7.2. Practical Problems

The research questions answered in the previous section leave us with the question how we can relate this to our practical problems. The practical problems arose from interviewing with Aviapartner.

Aviapartner's operation is characterised by its relative small size of only five deicing vehicles. The current practise is to make little usage of very formal planning when it comes to pre-tactical and tactical planning. In this work we showed how one can plan operations for the pre-tactical and tactical phase. The two practical problems are: *How can the pre-tactical planning phase of Aviapartner be improved, with regards to the objectives defined?*, and *Given a planning for a day, how can one cope with changes in weather, delays, or sudden availability of resources?*.

For the pre-tactical planning we have showed how one can make a formal planning width the use of scientific planning approaches. At this point it is not possible to give a conclusive answer to the first practical problem, since validation has not been performed. This validation would entail a side-to-side comparison of the old planning methodologies and our pre-tactical planner. We do, however, provide an in indication that formal planning is possible for the operation of Aviapartner. We have shown multiple heuristics that can optimise towards a specific KPI for Aviapartner.

For the tactical operation, we showed that a look-ahead can provide improvement over other approaches we tested. However, the current operation at Aviapartner is not formalised in a model, this means that a theoretical comparison is not possible. Therefore we have not shown that an actual improvement at Aviapartner is possible.

7.3. Recommendations

The practical problems were aimed at providing advice to Aviapartner how the operations for deicing planning could be optimised. During our work we found a number of practical things one could implement to optimise the deicing operation. These are as follows:

1. For our work, it would have been helpful if there was more data on the details of the deicing operations. Especially on the driving times between gates, since that would give a better estimation of the operation for Aviapartner. Furthermore, requests for deicing are not registered at which time the request is made. To give a realistic estimate on how far one can look-ahead this data is required.

2. We found that the look-ahead offers an advantage over not using a look-ahead strategy. As we pointed out, we could not provide evidence that this works in practise. However, from the interviews we learnt that better prediction of fluid levels could have led to less unexpected required refillings, which lead to delayed aircraft. We recommend the Aviapartner practise to incorporate a technique to look-ahead in the queue, and adapt the planning to requested operations.

3. We think that a more graphical overview of the deicing activities can help the planners to get a better overview of the situation. This recommendation is partially out of the scope of this work. However, we have built a preliminary graphical interface to interact with a plannings from our planning algorithms.

7.4. Discussion

In the discussion we argue the validity of the results, and discuss what our improvements to the status quo is. First we argue whether our theoretical results can actually make an impact on the real world situation.

We based our experiments on historical data we obtained from the Aviapartner practise. There was a limited availability of key time points. This limits the validity of the results found. We presume that the STD is a representable time point for the time of deicing operation. However, in the whole CDM implementation on Schiphol, this is much more fine-grained. The time when an aircraft can be deiced depends on the slot it is assigned to. Further assessment would be necessary using a shadow mode trial. Furthermore, we assumed the weather conditions to be known for the future. In reality the weather conditions are better modelled with probabilities, as it is hard to predict weather conditions. Especially precipitation is hard to predict.

The look-ahead strategy we used in the tactical planning, might in theory make significant improvements. However, this might be too simplistic to state, as some information is known in the theoretical model, but not in practise. For example, we presume in our work a theoretical departure time, and an updated one in the tactical planning. In real-world there might be an update of this departure time as we have made our look-ahead, this means that our theoretical look-ahead was based on no longer valid information.

Our experiments showed that the preferred method for replanning is to discard the original planning. This does not mean that the offline planning does not serve a purpose, it can still be used to make a capacity estimation on the resources needed for the operation. However, the plan holds not so much value if most of it is discarded. The air transport industry is one which is prone to changes in schedules, we have seen an ad-hoc operation at Aviapartner, which might not be the worst thing to do in such a dynamic environment.

In the research field of RCPSP we noted that we could not find the appropriate tools to model a semi-renewable resource, a resource that is not strictly non-renewable or renewable. Therefore we introduced a semi-renewable resource. We showed how one can implement this semi-renewable resource, with the usage of stints, into a MILP formulation in continuous time. The drawback of our MILP implementation of this semi-renewable resource is that one should have a theoretical upper-bound on the number of stints the semi-renewable resource is going to make.

This work has not provided a validation of the results for a real-world operation. We have hinted that certain approaches look promising for the real-world operation. However, to establish answers for the practical problem, proper validation is needed. This is left for future work.

7.5. Future Work

This section presents work that can be done in the future. The future work is aimed at two aspects, the work that can be done on the research side, and the work that can be done on the practical side.

While scheduling the deicing activities, we presumed the times of departure, driving time, and deicing time to be known. In real-world these times often vary, the times depend on numerous factors, such as personnel speed, jet-streams, etc. We suggest that future research can look into these distributions of times. This probability on conditions should also be considered for the weather. Especially precipitation, which influences the length of deicing operations, varies from time to time, and is hard to predict. For future research we suggest that one could provide a planning algorithm that works with probabilities on certain conditions.

Furthermore, we introduced a concept of 'semi-renewable' resources. This is something we have not seen in this form in literature. This concept can be researched further, there might be more efficient methods to implement this into a MILP, or to implement this with the use of heuristics.

For real-world future work we suggest to make a stronger connection between real-world planning and theoretical planning. We showed how a theoretical planning can be made, and how it can be adapted in a real world scenario. We also know that in real-world human factors play a role. Therefore it is advisable to use a shadow-mode trial validation in case one wants to implement one of our strategies. In a shadow-mode trial, the new algorithms run on the same information as the original planner has, only the orders of the 'old' planner are still followed. However, we see potential that one of our theoretical approaches can be used to assists planners in the current operation.

So to sum up, we have made contributions towards the theoretical modelling world with our semi-renewable resource. Furthermore we showed a formal model for the deicing problem, showed how to model this an RCPSP, and translated that into a MILP. We showed the effect of real world disturbances on the planning, and showed what heuristics one can use to deal with this.

A

Software design document

This document aims to provide a design overview of the software that is going to be designed for Aviapartner's deicing operation.

A.1. Actor analysis

This section lists the relevant actors involved in the deicing process. For each actor we provide its role and interest.

Actor	Interest
Aviapartner Fligth Watch	Monitors flight operations, and does overall communication on day of operation via: CDM, CISS, radio, Telex (SITA), e-mail and operators on the ground. Also is responsible for assigning deicing vehicles to aircraft, and updating CDM, CISS accordingly.
Aviapartner's Coordinators	Are responsible for assigning crew to specific roles and tasks on the day of operation.
Airliner's Flight Operation Centres	Monitor operations, and assisting captain.
Aviapartner's Planbureau	Make staff planning, generally day(s) ahead of day of operation
KLM equipment services	Responsible for maintaining and repairing equipment of Aviapartner
Meteo Provider (e.g. KNMI)	Provide weather information for the airport
Captain	Ultimate responsibility for the passengers and the aircraft. Also has a role in on-time performance. A captain typically requests the deicing.
Aviapartner Deicing Supervisor	Checking aircraft for quality of deicing operations and advising on the treatment; has a role in the safety of the operation.

To further analyse these actors, a power/interest diagram[2] has been made. Some actors are aggregated, such as all the devisions of Aviapartner. The diagram in figure A.1 describes the amount of interest and power over the deicing process.

A.1.1. Scenario Deicing

The scenario reflects a typical deicing operation at Aviapartner. This scenario is based on a day of operations and the days prior to the days of operation.

1. First the Planbureau will plan capacity for deicing operations, this is communicated with the deicing supervisor, they base their opinion on whether deicing will happen on the day of operation

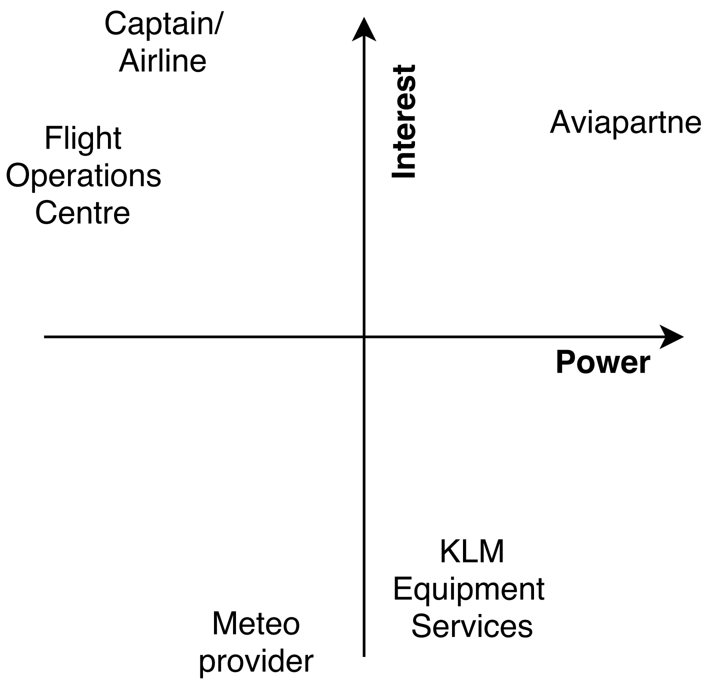


Figure A.1: Interest/Power diagram for stakeholder involved in the deicing operations of Aviapartner

on weather information from KNMI. The weather will influence the whether deicing in necessary, and the severity of the conditions influences the time needed for deicing operation.

2. The deicing supervisor makes sure that deicing vehicles are operational, he makes sure that there is enough capacity to deice the expected aircraft. This means that there are enough vehicles and personnel operational at specific time points.
3. On the day of operation, a captain will call in a deicing, this is for instance done over the radio, this can also be done via communication with the ground crew.
4. Flight watch will write down the request for deicing.
5. Flight watch will update the CDM with a request for deicing, after granted:
6. Flight watch will assign a deicing vehicle to the aircraft.
7. Flight watch will communicate the details to the crew of the deicing vehicle.
8. Before deicing can start, the crew asks permission from Flight watch, after granted they start
9. After the operation the deicing crew will call Flight watch with the times of deicing and the amount of fluids used.
10. Flight watch will enter details of the deicing in the CISS system.

A.2. Customer Processes for pre-tactical planning

From the interviews conducted at Aviapartner, we tried to capture their rules for deicing pre-tactical planning. We noticed that there is no pre-defined assignment of vehicles to aircraft, this is mainly due to the uncertainty of exact departure times, and uncertainty in the need for deicing.

For estimating the extra capacity in personnel for deicing aeroplanes, the answer was to use common sense. This common sense is mainly reflected in a few best-practises for determining capacity needs. To our belief, the best practises can be summarised as follows, as pictured in figure A.2

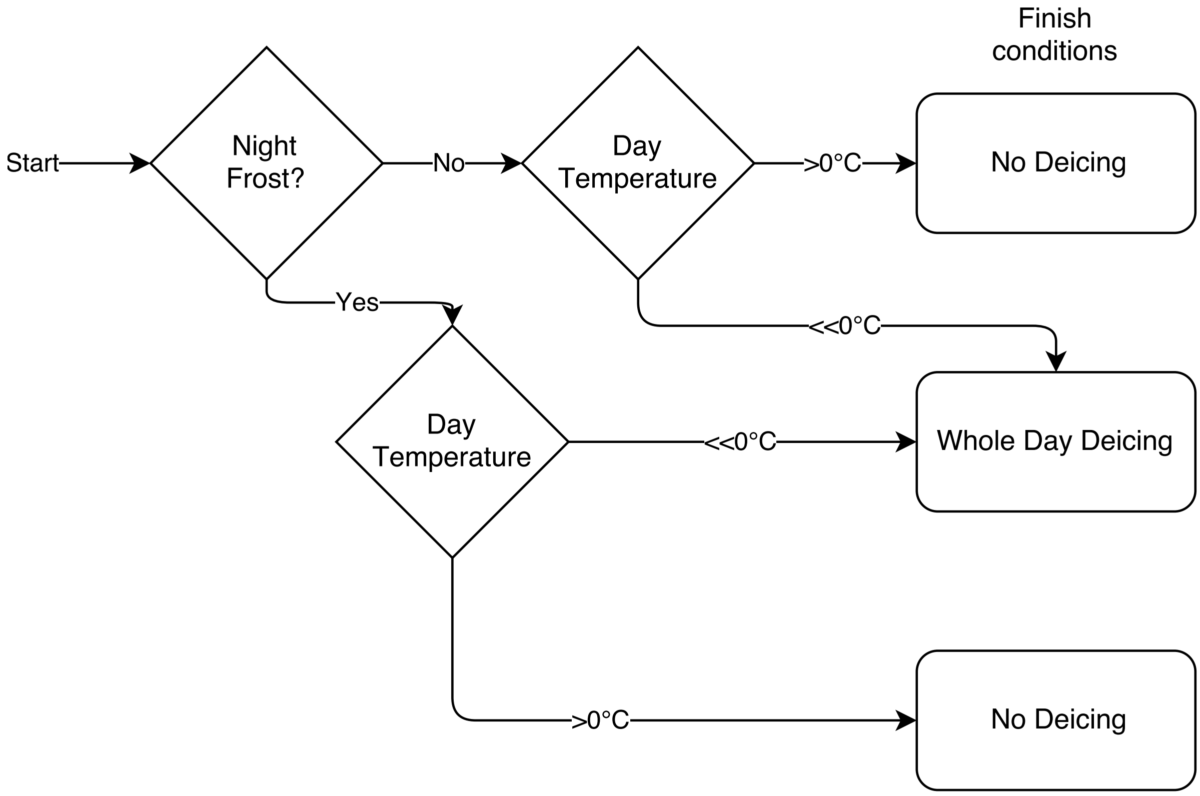


Figure A.2: Deicing process diagram for pre-tactical operations

1. If night frost (or temperatures close to 0°C) is expected, it is likely that all night stoppers require deicing,
2. If temperatures are around or just above freezing point, turnaround aircraft are not likely to request deicing. However, there are conditions were this is possible since flying at altitude cools down the fuel in the wings and this can cause condensation on the ground.
3. If precipitation is expected around or below freezing point, deicings can be expected for all aircraft. Precipitation will cause other effects, such as cancelled flights, and longer deicing operations, since a two-step procedure is needed. The two-step procedure is employed to first remove ice from the aircraft, and then give it an extra protective layer, such that the time the aircraft is exposed to fresh precipitation is minimised.

From this point the planner knows what flights might require deicing, from this point one can plan the capacity needed. As a general rule at Aviapartner, if there are only night-stoppers, they generally plan two vehicles, and warm-up a third. If the third is actually needed, people from the normal operation can be included in the process.

For the night-stoppers it is possible that three aircraft are scheduled to depart at the same time.

A.3. Customer Processes for tactical planning

The general process for captains or airliner representatives to call in deicing is described in section A.1.1. Requests for deicing are processed in a first-come-first-served (FCFS) manner. Captains will receive a sequence number, indicating their position in the deicing queue.

Generally, the aircraft don't actually depart at the same time, as some might be ready earlier, or some delay is introduced as there are no deicing vehicle available.

A.4. Our added value for Aviapartner

Fist of all we identify the objectives for Aviapartner as an organisation. The objectives include: Safety, Profit and Quality of Service (QoS).

The airliners are under a considerable amount of competition resulting in thin margins, this also causes the ground handler fees to be under pressure. This also holds for Aviapartner, there are a few options to ensure profit, this is either lowering the costs, or improving the revenue. At Aviapartner the profit remains an important key performance indicator.

During the observations at Aviapartner we noted that several parts of the process might eligible for improvement, or further investigation. We identified parts of the process where there might be room for improvement, these are as follows:

1. Improve pre-tactical planning forecasting, now this still involves manual checking of information sources, this can be automated.
2. Online, or tactical, planner. This can be an semi-automated process for handling deicing requests, and planning (assigning) these to vehicles.

A.5. Data needed for in- and output

For the two improvements above we each need data as input, for each we specify the output.

A.5.1. Pre-tactical planning

The **input** can be characterised as follows:

1. Weather information from KNMI,
2. Historic information (estimates, distributions) on:
   (a) Travel time for deicing vehicles between gates,
   (b) Deicing times in various weather conditions,
   (c) Fluid use for various aircraft types.
3. Schedule of contracted deicing airlines

The **output** for the pre-tactical planning consists of a estimation for personnel and deicing vehicles. It trivially follows that if you plan to have x deicing vehicles operational on certain time intervals, how much personnel is needed.

A.6. Logic of underlying decisions

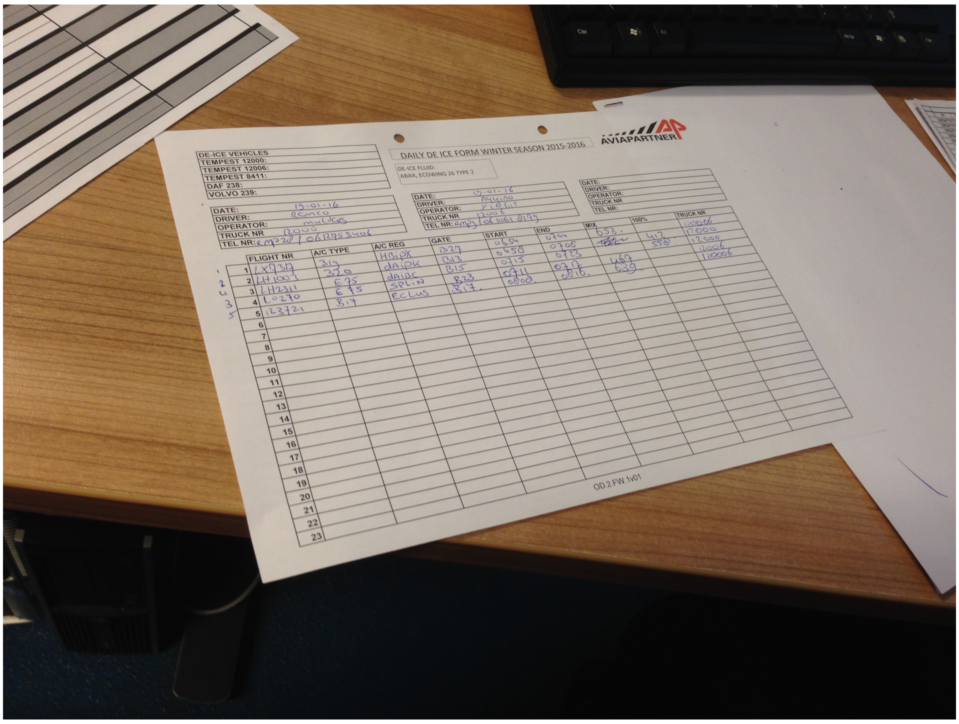
We split the logic into two phases. In the first phase we look at the pre-tactical planning the idea is to implement a number of strategies from the theoretical computer science. An overview of these strategies can be found in section 3.2.

The online planning involves the process on the day of operation. First of all, on the day of operations there are changes possible compared to the baseline schedule. Mostly, these changes are small corrections, like a couple of minutes. Secondly, it also possible that there are large changes, which can have a significant impact on the operation, for instance a delay of an incoming flight of 2 hours.

The task of online strategies is to deal with these changes and events, such that it is still possible to have an operation of sufficient quality, while keeping control of operational expenses. The insights gained for tactical planning with these deep approaches will benefit the development of online strategies.

A.7. GUI suggestions

During the interviews we noticed that there is some reluctance towards IT solutions. Therefore we suggest to build tools which provide assistance, and do not completely remove the need for human intervention. This means that a human operator has to approve the decisions of a system, or that the IT systems merely provide a suggestion, with information why the system made the suggestion.



AVIAPARTNER

DAILY DE ICE FORM WINTER SEASON 2015-2016

DE-ICE VEHICLES
TEMPEST 12000:
TEMPEST 12006:
TEMPEST 8411:
DAF 238:
VOLVO 239:

DATE:
DRIVER:
OPERATOR:
TRUCK NR
TEL NR:

	FLIGHT NR	A/C TYPE	A/C REG	GATE	START	END	MIX	100%	TRUCK NR

Figure A.3: Deicing form for planning deicings at Flight Watch.

A.7.1. Existing UX

The existing GUI for tactical planning consists of a form with deicings, as in figure A.3, this is at Flight watch. The planboard of the coordinators, and the section which is dedicated for deicing can be seen in figure A.4.

A.8. Implementation process

The wish of the NLR is to maintain the quality of the software such that is understandable and extensible. This requires that the software documentation and implementation is of sufficient quality. To maintain this quality a number of measures, on controls are implemented.

A.8.1. Quality control

Maintaining the quality of software is important to ensure the continuity, extensibility, and reliability of the software. We take the following controls for ensuring the quality of the work.

1. Using of a version control system: Git,
2. Usage of Unit tests,
3. Usages of Pull requests, for each feature a pull request,
4. Documentation on the open endpoints of the software,
5. Package control with Maven,
6. Usage of widely available packages for common tasks.

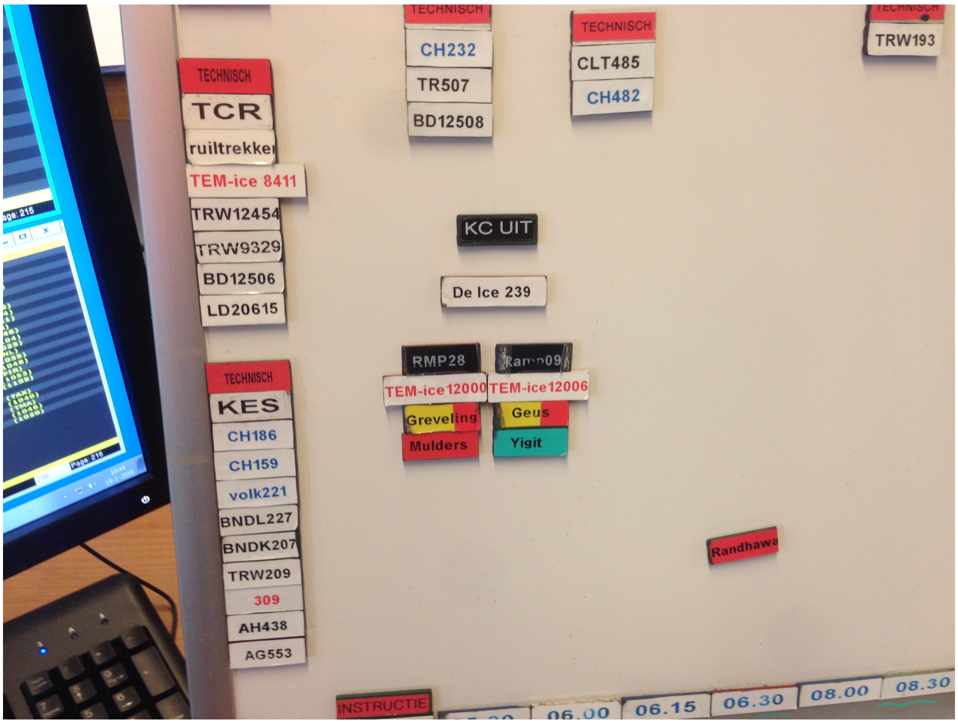


Figure A.4: Planboard used by the coordinators. Note that people are assigned to the deicing vehicles.

B

Software Solution

For the thesis a software solution has been implemented. The tool enables users to build a planning for deicing activities. This implementation makes use of several technologies. The total implementation can be seen as a *technology stack*.

The process incorporated decisions of which techniques and packages are used. This chapter aims to explain the process and the reasons for choosing specific parts of the technology stack.

B.1. Global Design

The first design decision we took was to make an application which uses a web based user interface. Traditional applications have an interface which is usually coded in the same language as back-end of the application, such as Java Swing or JavaFX. A global overview of the technology stack can be seen in figure B.1

A web-based user interface uses end-points of the application, this is usually done with a defined API. The most common API is to use a REST API. The back-end specifies the points and methods at which the front-end can call the application. The incorporation of this design has a number of advantages:

1. Strict end-points allow for alternative user interfaces to be created for this application with relative easy. One can build a mobile application using the same end-points of the back-end.

2. Defining end-points allows for better testing, as often the end-points are well defined and well specified. This will increase the quality of the application.

3. Web based front-ends can run on almost any computer, since all major operating systems can run a browser.

This design principle has become a standard, in my opinion, for developing applications. It is a pattern observed much more frequently these days. The following sections will first explain the design of the back-end, and then we will focus on desing principles of the front-end.

B.2. Back-end Technologies

The back-end incorporates a number of functions. First of all it handles the primary planning process. Furthermore it serves data, such as airline data and weather data to the user.

B.2.1. Primary Programming Language

The first thing to choose is a primary programming language. This choice has been made relitevelty fast, as no other candidates seemed a good option. My familiarity with Java, and that of my supervisor are the primary reasons to choose Java as a primary programming language.

Choosing Java, allowed me to choose other technologies on top of that. The following sections will give an overview of selected technologies, such as that of the webserver.

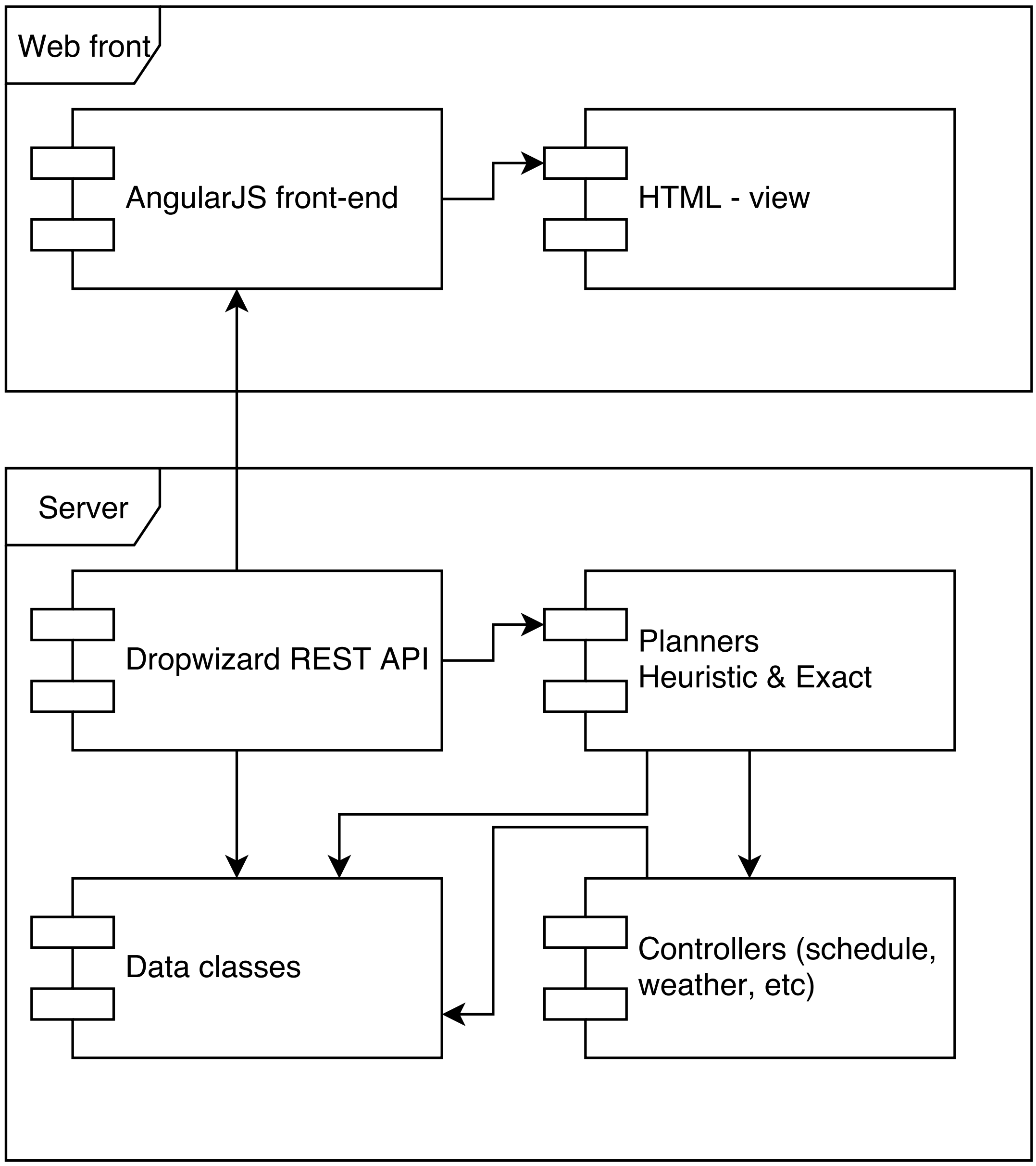


Figure B.1: Software technology stack overview for the deicing planning tool.

B.2.2. Web Server Technology: Dropwizard

The requirements for designing a back-end with a server such that it can serve a web front-end require a web server technology package. The first requirement for this is that it can easily work with with the primary programming language, Java, this leaves some popular choices out of the picture, such as Node.JS and Apache http server.

The choice for the server-side web-framework has been made for Dropwizard[1]. This framework is a collection of well-developed techniques. Further it boasts an easily usable quick set up, such that one can immediately start developing. Dropwizard was advised by a number of friends, as it is easy, and works with Java.

B.2.3. MILP solvers

MILP solvers are used for solving the planning problems. The main solver used is Gurobi. This solver is a commercial solver. Installation details for this solver can be found on their website[2]. The choice for Gurobi has been made, as it outperforms GLPK by a mile. Furthermore, Gurobi offers an relative easy to use Java API, and is available to students for no cost.

After installing Gurobi, one needs to un-comment the dependency of Gurboi in the `pom.xml` file. After that in the class `planner.implementation.MilpContGurobi` one needs to uncomment the code. Furthermore in the `web.resources.PlanningResource` one needs to uncomment the line that specifies the MILP in the method `getAllPlannings`.

B.2.4. Other Packages

This section aims to explain choices for other packages. These packages mostly handle very specific tasks.

1. *Maven*[3] is a tool mostly used for dependency management. The tool allows to easily specify on what the project depends on. Furthermore, Maven allows the software to be packaged such that the dependencies are included. Maven also allows for specific life-cycles to be executed easily, such as testing, installing and packaging.

   This tool is primarily chosen as it was very familiar to the author. It is seen as a best practise for managing dependencies in a Java project.

2. *Spring dependency injection*[4], Spring is used for dependency injection[32], this means that the singleton pattern is not necessary in this piece of code. That means that the software is of better quality.

3. *JodaTime*[5] is a package for managing time. Due to the nature of the work, in the air transport industry, time zones and good management of that are very relevant.

B.3. Web Technologies

The web front-end is composed of several techniques. Generally speaking a division into three techniques is made: 1) HTML - what information is displayed, 2) CSS - how it looks, and 3) JavaScript (JS) - how it interacts. For the Layout we have chosen Bootstrap, and for the JS side mainly Angular.JS does the job.

The web code is place in the project under the folder `/afstuderen/deicing/src/main/resources/assets/`. There is a folder for the style-sheets, called 'CSS', a folder for the Javascript files in 'js', and a folder for HTML partials in 'views/partials'.

B.3.1. Lay-Out

The User Experience (UX) greatly depends on how the application looks. One of the most popular tools for rapid development with HTML and CSS is to make use of the Bootstrap[6] framework. This

[1] http://www.dropwizard.io
[2] http://www.gurobi.com/
[3] https://maven.apache.org/
[4] https://spring.io/
[5] http://www.joda.org/joda-time/
[6] http://getbootstrap.com

framework is designed by Twitter. It offers a set of well-designed web elements, and offers a good structure for positioning elements on a web page, such that it looks good any screen display size.

This framework has been chosen, because I had already familiarised myself with this framework. On top of that, it offers a great ’wow’ factor, as applications using Bootstrap will generally have a good look-and-feel. Furthermore Bootstrap offers easy to use templates, and numerous components to make use of.

B.3.2. Web Application Framework: Angular

Angular is a framework implemented in Javascript, which offers a set of tools one can use to develop a web-page. Angular combines functions to make calls to the API of the back-end, and a framework to update the front-end dynamically.

Angular has been chosen because it is widely used, moreover I had familiarity with it. Angular removes cumbersome steps of putting information from JS into HTML view.

B.4. Class Diagrams

This section explains in more detail the data-classes of the program. It extends on the diagram in figure B.1. We extended this view with four extra class diagrams.

For reading the class diagrams it is good to take the following thinks into consideration. The classes with class names in italics represent abstract methods, this means that the classes are extended and the extensions will actually be used. Secondly, the dotted lines represent a ’use’ relation, this means that a method takes an argument or returns something that is of that specific pointed class. The lines without interruption represent an ’extends’ relation.

A class has on top the class name. Under the class name, there are a number variables listed, generally the variables are *private* (-), however we have listed most variables as *public* (+) while in fact they are private. This due to the fact that incorporating all *getters* and *setters* makes the diagrams less readable.

Controller classes are used to control specific data classes. These classes are instructed by the Spring context.xml file to read a specific data file, and with that file construct data classes of the data package.

B.4.1. Airline Classes

Figure B.2 give an overview of the classes related to airlines. The class `AirlineResource` implements a number of methods that can be used by the web front end. The resource returns a list of airlines on a specific search query. This is used in the front-end by airline selection tool.

B.4.2. Activity Classes

Each planning consists of an assignment of activities to vehicles. The diagram in figure B.3 gives an overview of all classes that represent activities. On top we find an abstract class `Activity`. Every activity has a start and end time; an interval.

We have chosen to make *composite* activities, a composite is a set of one actual activity and it might contain driving activities. The reason for this is to give an overview that driving and the activity in itself are closely related. The deicing or refilling are linked to driving to the activity, and sometimes driving from it.

Static activities are activities that take place on one specific place, hence the incorporation of a data field ’location’. The two static activities are deicing and refilling. Deicing has some data fields that represent the expected amount of fluid that is going to be used, and the amount of fluid that is left in the vehicle it has been deiced by. There is a pointer to the flight of the deicing activity.

B.4.3. Planning Classes

The activities of the previous section have to be listed and assigned to vehicles, therefore we present in this subsection the various classes used for containing activities. Figure B.4.

The planning resource is a class that handles the REST end of the application. It gives plannings based on the requests given in a `PlanningRequest` format. The return value is a collection of `VehiclePlanning`s in the class `PlanningCollection`. Over this collection one can calculate various

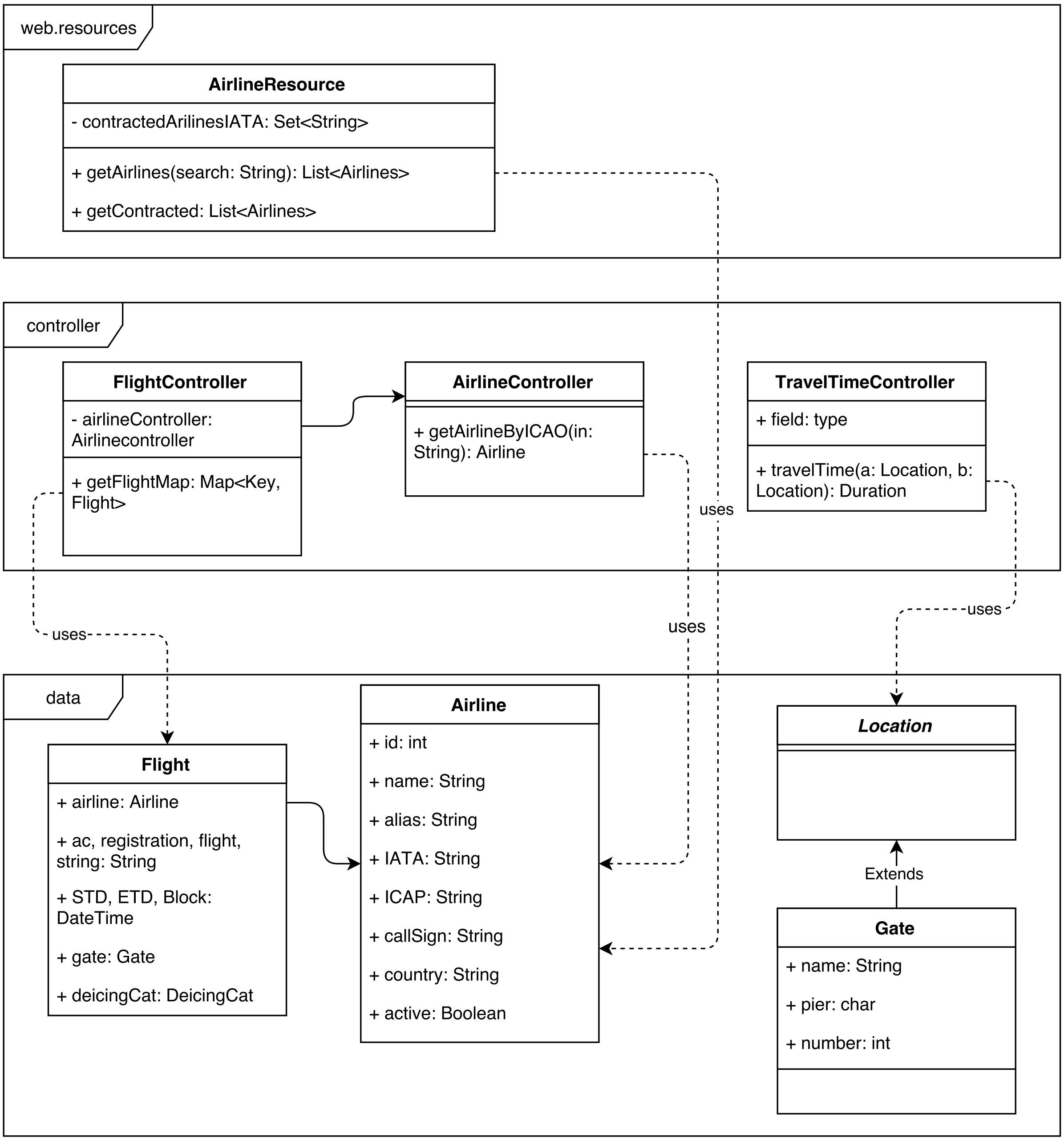


Figure B.2: Classes used in the context of airlines. On the top is the controller, which makes a REST endpoint for HTTP requests to connect with. The AirlineResource connects to the controller and uses airlines as result type.

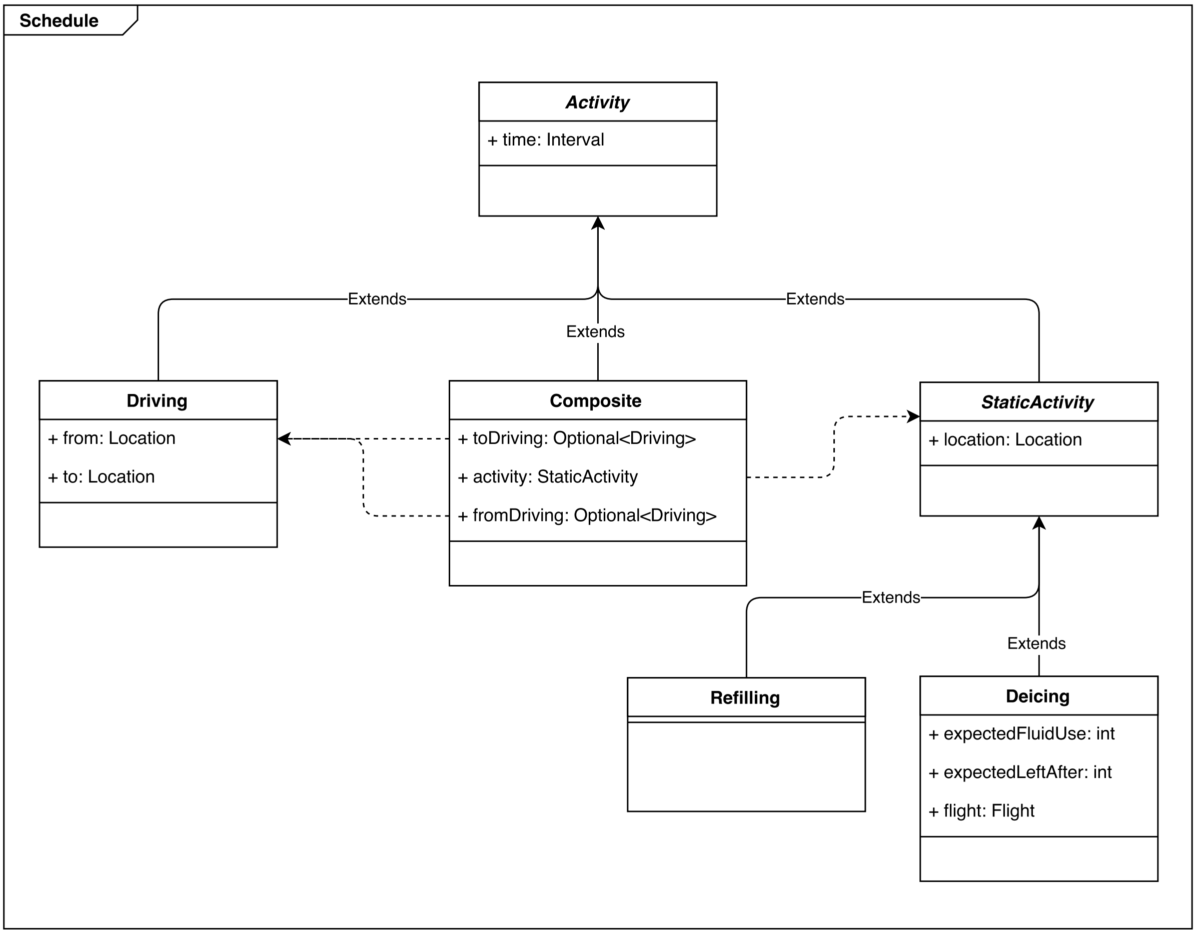


Figure B.3: Classes used to represent activities. The top class is an *abstract* class *Activity*, from which the actual activity classes are extended. A composite is a collection of activities, there is always a *StaticActivity* (Refilling or Deicing), and there can be optional driving activities.

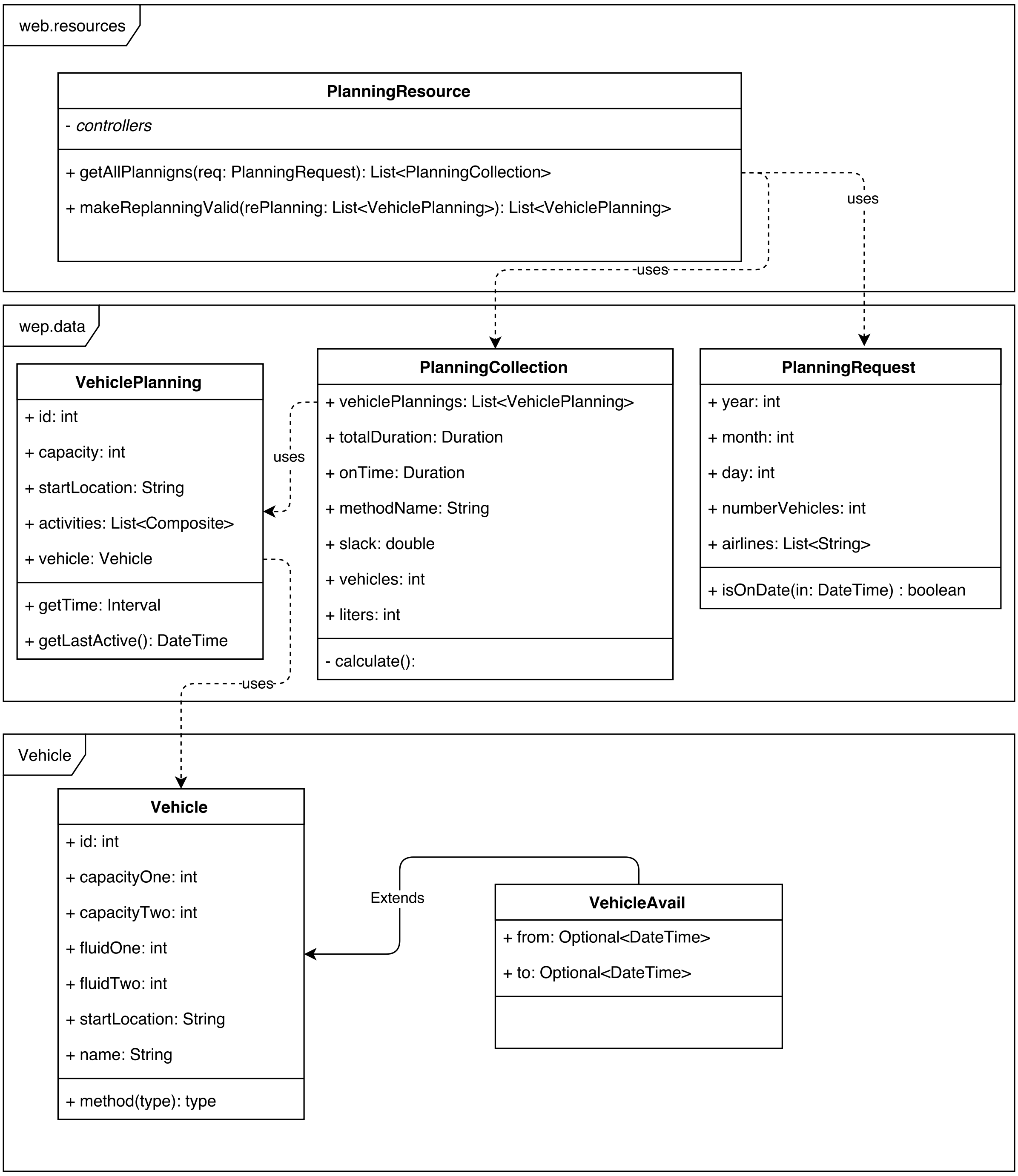


Figure B.4: Classes used for the planning. On the top is the class used for the web API. The web API specifies endpoints in the REST architecture. The framework will convert an appropriate JSON request to a PlanningRequest class. The solvers (image B.5 will plan such a request, and return a valid planning. The PlanningResource will then send a PlanningCollection back that corresponds with the request.

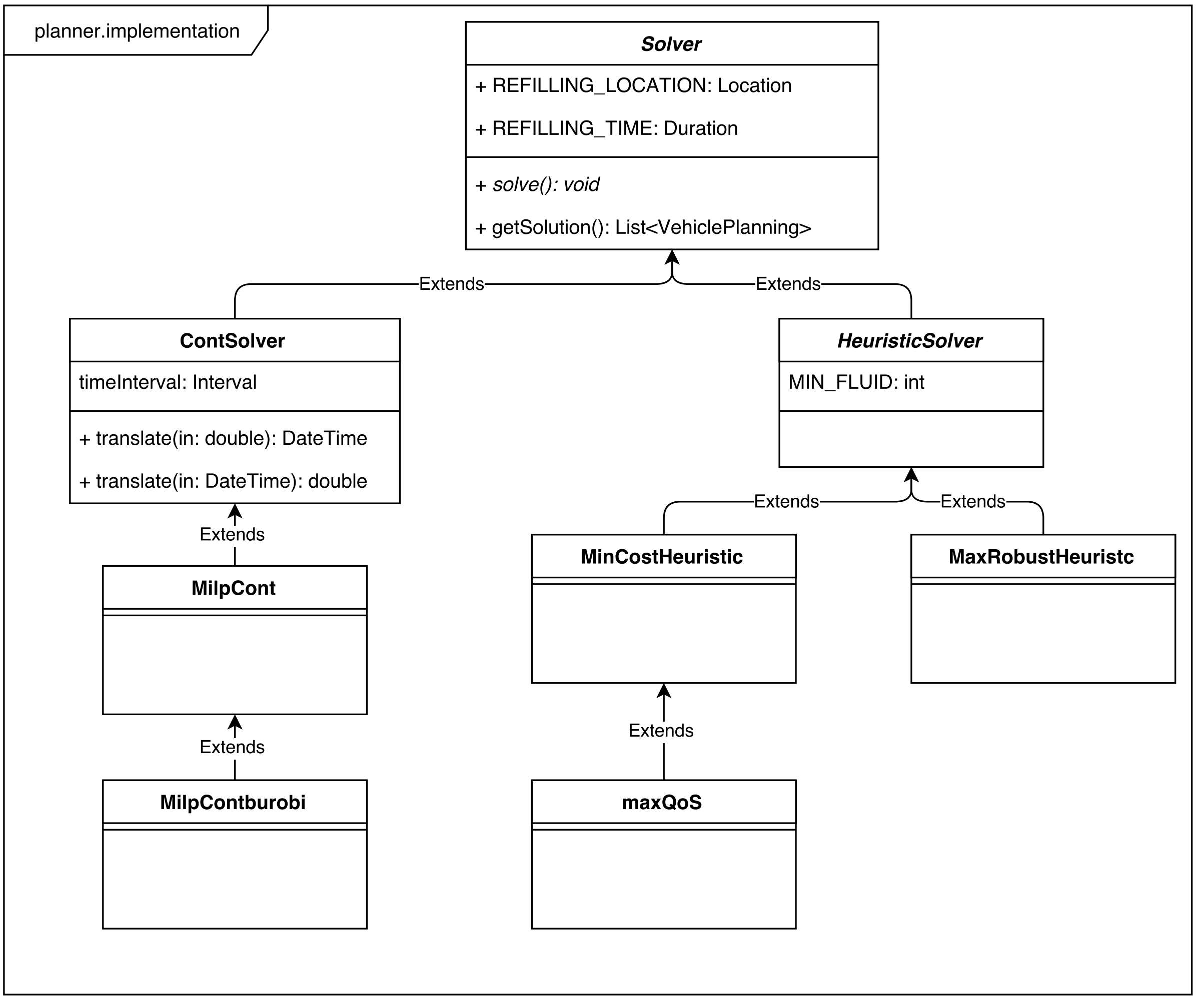


Figure B.5: Classes as Solvers. Every solver extends the *abstract* class *Solver*. There are two main extensions, one for heuristics, and one for exact solvers. The ContSolver is an abbriviation of Continuous time solver, where two implementation are used, one Milpcont in GLPK, and one MilpContGurobi which uses Gurobi. The three heuristic methods extend from the *HeuristicSolver* class.

KPIs to assess the quality of the planning. A vehicle planning is basically a list of composite activities and a vehicle.

The class `VehicleAvail` extends the class `Vehicle`, it adds optional times from when or until when the vehicle is available.

B.4.4. Solver Classes

The previous seciton explains how the result is represented and the input for the planning. The actual classes that calculate these plannings are displayed in figure B.5. The abstract class solver is extended by exact approaches and heuristic approaches.

B.5. Installation & Running

First of all one has to make sure that a number of software packages are installed. The following have to be installed:

1. A *Java SDK 8* or newer.
2. *Maven*, the tool to install dependencies.
3. *Git*, to install and download the software.

Running the application is relatively easy. We consider a situation where the project is hosted in a Git repository. The first step would be to obtain the full source code like this:

1. First clone the repo: `git clone git@github.com:rubenverboon/afstuderen.git` Or an alternative repo.
2. Now we need to install two jars in the folder `deicing/` These are the `GLPKSolverPack.jar` and the `SCPSolver.jar` which can be downloaded from here.
3. Then we need to place the two data files in `/afstuderen/deicing/src/main/resources/flightStatistics/` the files `aviaData.csv`.
4. If one wishes to have the exact solutions on top of the heuristics, one should follow the steps as defined in section B.2.3
5. Now we should move our unix shell to the folder `/afstuderen/deicing/`
6. Install dependencies with `mvn clean package`
7. And then run with one of these commands: `java -jar target/deicing-1.0-SNAPSHOT.jar` or `mvn spring-boot:run`
8. Use a browser and navigate to `ttp://localhost:8080`

B.5.1. Running Software from an IDE

This guide will let you run the software from your IDE. This gives one the opportunity to make easy changes to the project. We will give this guide from an Eclipse point of view. The author of this work prefers IntelliJ, however Eclipse is used here. Make sure the following steps have been completed:

1. An Eclipse installation, for older Eclipse versions a Maven plug-in has to be installed.
2. Steps 1 through 6 of the previous guide.

Before we list how to import and run this application in Eclipse, it might be wise to make sure that the application is not longer running, as one might have done by doing steps 7 and 8 from the previous installation guide.

1. Start Eclipse, with your default path.
2. On the top corner select the menu 'File'

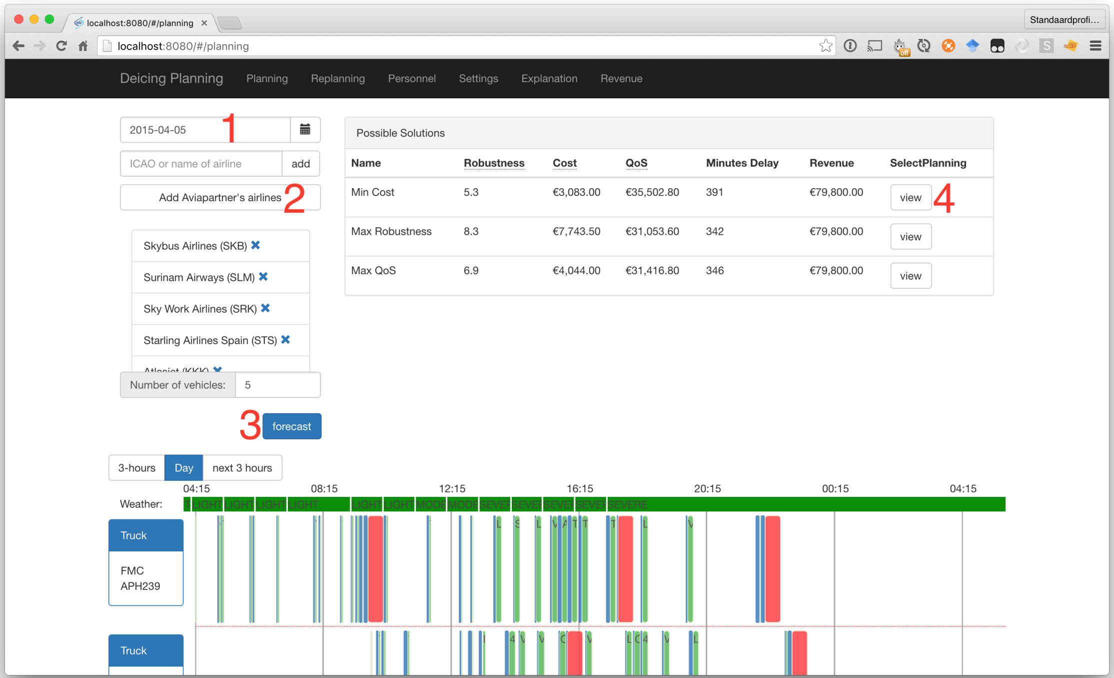


Figure B.6: The program viewed from a browser. Note the order of instructions (1,2,3,4).

3. Then select 'Import'
4. From the import menu look for the folder titled 'Maven' and select the option 'Existing Maven Projects', and click 'Next >'
5. For the root directory browse to the location we have checked the code out in step 1 of the previous installation guide (e.g. `git clone`). Make sure we select the folder `deicing` as the `pom.xml` file is located right in that folder.
6. If one has selected the correct folder, one should see that the 'Projects:' box will list a 'nl.nlr.aoap.deicing:...' pom select this project and click on 'Finish'
7. to make sure all Maven dependencies are satisfied, right click on the root of the project folder in the 'Package Explorer' (right side of screen).
8. From the Maven tab, select 'Update Project', and press 'OK'
9. locate the class `Main.java` ('src/main/java' > 'default package'). Right click on the class and select the option 'Run as' and then 'Java Application'
10. The program will run.

B.6. GUI Results and Usage

The resulted program is implemented with a web-GUI. The choice for this has been explained in the previous sections. The main program can be reached, after starting the server as in the previous section, by navigating the browser to `http://localhost:8080`. For now, it only works with Chrome. The steps for building a planing are illustrated in figure B.6, the steps are as follows:

1. The first option is to select a date. For now, the tool only works with (pre-loaded) historic data. The calender button on the right side of the input field will load a 'date-picker'.

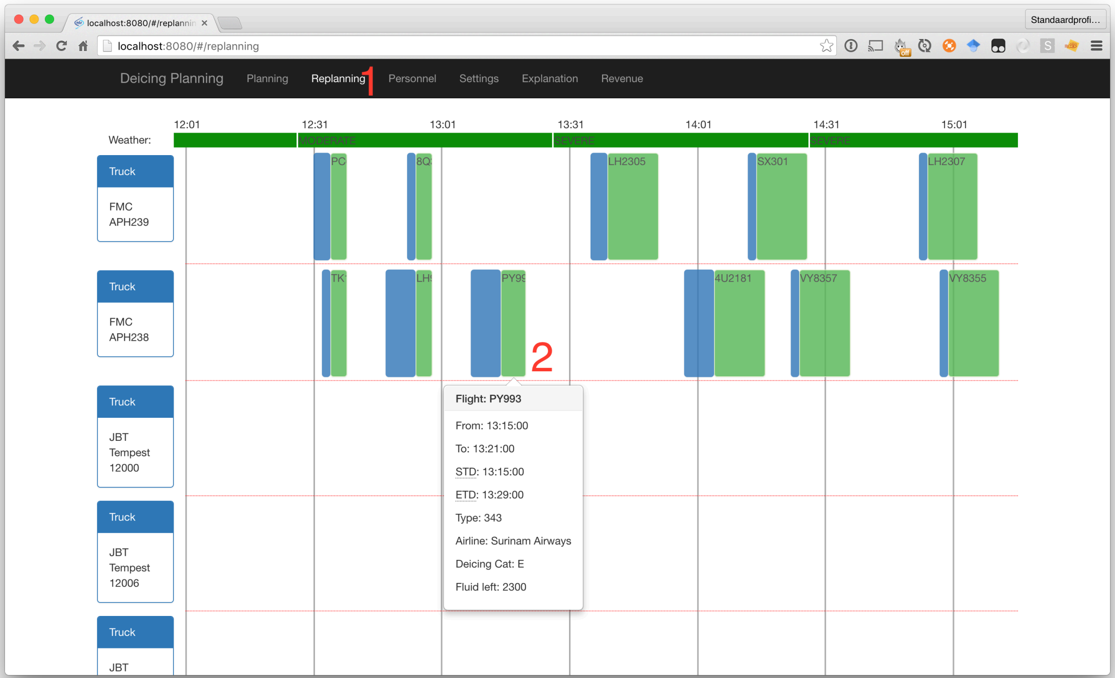


Figure B.7: The replanning screen

2. The airline selection field has a button that will add all Aviapartner's contracted airlines with one click. Furthermore individual addition of airlines is possible by using the text-field. Deletion is possible by clicking the cross.

3. The 'forecast' button will make a planning for the day. In the table on the right-top side of the screen this will be visible.

4. From this table a planning can be selected to be both viewed, this is done by clicking the 'view' button. The selected planning is also the planning which is used for the replanning section of this work. The planning will be displayed on the bottom of the screen (one might need to scroll down). The vehicles are on the vertical axis. Time, and below that weather, are displayed on the horizontal axis. All planned activities are displayed in the roster.

The replanning phase is illustrated in figure B.7. The replanning is done by dragging an deicing activity towards another vehicle. The whole planning is then sent to the server, to be recalculated. This is done in the `MakeValidAgain` class in Java. This 'valid' planning is then sent back to the web front-end to be displayed to the user. The server checks for filling inconsistencies, and for if there are no overlapping activities. Activities coloured in blue are driving activities, activities in red are filling (deicing fluid) activities, and activities in green are deicing activities. The steps of replanning are as follows:

1. On the top menu selection, select the 'replanning' tab.

2. Next one can replan activities. This done on the selected planning (the 'view' button). Only deicing activities (green) can be replanned. By pressing the mouse and dragging the activity around, the activity can be assigned to another vehicle.

C

Input and output for experiments

This appendix describes how we obtain the input for the deicing planner, and the output it generates. The input is based on assuming certain conditions, which are translated into specific times. The output consists of a schedule. Finally we describe what settings we used while running the experiments.

C.1. Input

The input for the model consists of a couple of components. We consider the weather, the flight schedule, and availability of other resources.

Weather

We assume there is a reliable information source for weather information. This source is able to forecast a 72hr period. An assumption is that a forecast can be created in the form of a time and a Deicing Weather (DIW) type. In section 1.6 we said there is an NLR classification for the DIW. This classification is on a scale of zero to three, where zero means no deicing conditions present, and three means that there is a severe weather condition which requires intense deicing. For our first model we take updates of the DIW by every hour. A sample of such an input would look like this:

```
05-04-2015 00:30;1
05-04-2015 01:30;1
05-04-2015 02:30;2
05-04-2015 03:30;1
05-04-2015 04:30;1
```

This means that on the 5th of April 2015, the DIW from 2:30 till 3:30 is type 2. This model is a simplistic interpretation of a weather forecast. Real-world weather forecasts are much more uncertain. Often weather forecasts present *chances* on certain weather conditions, as we have seen in an exmaple Terminal Aerodrome Forecast (TAF) in section 1.5.2

Flight Schedule Times

For flight times we take an historic trace of CDM data. For now we assume the deicing has to be done just before TSAT. For deicing operations the engines have to be turned off, the jet-bridge removed, and the push-back truck is usually connected. After the deicing operations are done, the aircraft is pushed back and engines can be started, this is thus at the Target Start Up Time (TSAT). Aviapartner has a number of contracted airlines, that means these airlines are able to request deicing services from Aviapartner. An example of this data looks like this:

Operator	A.C	Registration	Flight	STD
SWR	AR1	HBAAB	LX001	2009-01-01T07:00:00.000+01:00
BMR	319	GDDEF	BD001	2009-01-01T07:20:00.000+01:00
IBE	321	ECYYY	IB0003	2009-01-01T08:05:00.000+01:00
SWR	AR1	HBLLL	LX004	2009-01-01T08:50:00.000+01:00
SWR	32S	HBXYZ	LX055	2009-01-01T09:50:00.000+01:00

ETD	Block	Routing	Gate
2009-01-01T07:20:00.000+01:00	2009-01-01T07:05:00.000+01:00	ZRH	D16
2009-01-01T07:36:00.000+01:00	2009-01-01T07:17:00.000+01:00	LHR	D18
2009-01-01T08:30:00.000+01:00	2009-01-01T08:07:00.000+01:00	MAD	B10
2009-01-01T09:08:00.000+01:00	2009-01-01T08:57:00.000+01:00	BSL	D10
2009-01-01T10:54:00.000+01:00	2009-01-01T10:37:00.000+01:00	ZRH	D20

Travel Times
For vehicles we make an assumption for how long it takes to travel from one location to another location. Schiphol Airport has various piers, the first model for travel time is as follows:

$$t(a,b) = \left|\text{rank}(a) - \text{rank}(b)\right| + 2 \tag{C.1}$$

The function rank will return the alphabetical number of a given pier letter, so rank(E8) will return 5. This means that the travel time only depends on the letter of the pier. For instance if you travel from pier H to pier D it will take $|8 - 4| = 4$ minutes.

Deicing times
An estimation is taken for deicing times using times from previous SESAR experiments. We classify aeroplanes in sizes A to F, weather types in classified in four steps: 0-3, and type of precipitation.

Vehicles
The vehicles can hold a number of fluids of each type. We assume that every aircraft in its class uses the same amount of fluid.

C.2. Output

The output consists of a schedule for vehicles. This schedule communicates the activities likely to happen for each vehicles. The algorithm tries to plan as little vehicles as possible, this will minimise the operational cost.

C.2.1. Baseline Schedule

This output serves as a baseline schedule for the operation. It shows what one can expect for operations in assumed weather conditions. Often things change on the day of operations and changes are needed to the baseline schedule.

C.2.2. Vehicle planning

Personnel is tied to a vehicle, this means that the personnel planning automatically follows from a vehicle planning. One thing we can observe from this planning is the amount of slack, or robustness in the planning. If a planning is very tight, in a perfect world it is very efficient. However, if there is a small delay, this will cause the whole planning to be disrupted. Therefore small gaps, or slack, can prevent the cascading of delays. Slack makes the planning more robust for delays.

C.3. Settings for experiments

Below one can find the setting we used while performing the epxeriments. There is a combination between aircraft type and deicing time, and between aircraft type and deicing fluid usage. The sizes of the aircraft correspond to that found in the AEA documentation[15]. Typically the time and fluid usage depent on the weather conditions, however we model the weather as static, and only in the weather severity increase situation let the weather conditions change (or the effect of this change).

C.3.1. Deicing times

Aircraft Type	Time (minutes)
A	6
B	11
C	12
D	12
E	13
F	14

C.3.2. Fluid usage

Aircraft Type	Litres fluid used
A	300
B	400
C	500
D	600
E	700
F	800

Glossary

AAS Amsterdam Airport Schiphol. 2

AEA Association of European Airlines. 6

ANSP Air Navigation Service Provider. 1, 2

ATC Air Traffic Control. 2, 6, 8, 11, 75

CDM Collaborative Decision Making. 1–3, 6, 11, 51, 71

DIW Deicing Weather. 71

EIBT Estimated In-Block Time. 3

ELDT Estimated Landing Time. 2

ETD Estimated Time of Departure. 32, 44

FCFS First Come First Served. 11, 15, 16, 45

FOC Flight Operation Centre. 11

holdover time that deicing layer is protective against new ice formation in certain weather condition. 4

ICAO International Civil Aviation Organization. 5

KLM Koninklijke Luchtvaart Maatschappij (Royal Dutch Airlines). 1, 2, 11, 31

KNMI Koninklijk Nederlands Metereologisch Instituut (Royal Netherlands Meteorological Institute). 2

KPI Key Performance Indicator. 10, 11, 37, 41, 50, 51

LVNL Luchtverkeersleiding Nederland (Dutch ATC). 2

MAS Multi-Agent System. 43, 44

METAR Meteorological Aerodrome Report. 5, 6, 8

MILP Mixed Integer Linear Programming. iii, 19, 20, 23, 25–29, 31–34, 37, 39, 49, 50, 52

MOLP Multi Objective Linear Programming. 19

NLR Nederlands Lucht- en Ruimtevaartcentrum (National Aerospace Laboratory). iii, 6

QoS Quality of Service. 11, 37–39

RCPSP Resource Constrained Project Scheduling Problem. iii, 15, 17–20, 22, 23, 25, 26, 28, 49, 50, 52, 77

SAOC Schiphol Airline Operators Committee. 2

SESAR Single European Sky ATM Research. 72

STD Scheduled Time of Departure. 31, 32, 44, 46, 50, 51

TAF Terminal Aerodrome Forecast. 5–8, 71

TOBT Target Off-Block Time. 3, 6

TSAT Target Start Up Time. 3, 10, 18, 38, 71, 77

TTOT Target Take-Off Time. 3

UTC Coordinated Universal Time. 5

Nomenclature

C_m Capacity in litres for deicing fluid in vehicle m

$d : (G, G) \rightarrow T$ Driving time function d, take two gates as arguments, and returns a time needed for driving T.

f_i Litres of deicing fluid needed for aircraft p_i.

g_i The location of aircraft p_i, hence the g for gate. From the total set $g_i \in G$

p_i Aircraft with number i from the total set $p_i \in P$

$t_{i,d}$ The time when aircraft p_i is scheduled to be done with deicing operations (deadline).

$t_{i,r}$ Time when aircraft p_i is ready for deicing. This corresponds to the TSAT. This can be after $t_{i,d}$

$t_{i,s}$ Time when aircraft p_i starts deicing operations, this time is generated by a planning algorithm.

t_i Time, duration, needed for deicing of aircraft p_i

v_j Vehicle with number j from the set $v_j \in V$

MILP formulation

a Aircraft a, from the total set $a \in A$, corresponds to a specific p_i. Symbol is based on RCPSP *activity*.

$b_{a,b}$ Happened before relationship. Denotes that aircraft a is deiced before aircraft b.

d_a Adjusted deadline

I The set of *stints* I denotes the set of stints used for deicing vehicles. Between stints there is a refilling.

m Vehicle m, from the total vehicle set $m \in M$. The symbol is based on the RCPSP *mode*.

$m^{+}_{m,i}$ End of sting number i, where $i \in I$, for vehicle m

$m^{-}_{m,i}$ Start of sting number i, where $i \in I$, for vehicle m

$x_{a,m,i}$ Boolean variable, if true denotes that aircraft a will start deicing by vehicle m in stint i.

$x_{a,m,t}$ Boolean variable, if true denotes that aircraft a will start deicing by vehicle m at time t.

$x_{m,a}$ Vehicle m deices aeroplane a if true

Bibliography

[1] Babak Abbasi, Shahram Shadrokh, and Jamal Arkat. Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied mathematics and computation*, 180 (1):146–152, 2006.

[2] Fran Ackermann and Colin Eden. Strategic management of stakeholders: Theory and practice. *Long Range Planning*, 44(3):179–196, 2011.

[3] Mohammad A Al-Fawzan and Mohamed Haouari. A bi-objective model for robust resource-constrained project scheduling. *International Journal of production economics*, 96(2):175–187, 2005.

[4] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[5] Jacek Blazewicz, Jan Karel Lenstra, and AHG Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.

[6] KLEIN Bouleimen and HOUSNI Lecocq. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2):268–281, 2003.

[7] Michael B Bragg, Andy P Broeren, and Leia A Blumenthal. Iced-airfoil aerodynamics. *Progress in Aerospace Sciences*, 41(5):323–362, 2005.

[8] José Coelho and Mario Vanhoucke. Multi-mode resource-constrained project scheduling using rcpsp and sat solvers. *European Journal of Operational Research*, 213(1):73–82, 2011.

[9] Duivenoorde et al. Derogee, Kelder. *Airport Collaborative Decision Making*. Amsterdam Airport Schiphol, 2015.

[10] Jacques Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.

[11] Christodoulos A Floudas and Xiaoxia Lin. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139(1):131–162, 2005.

[12] Erwin W Hans, Willy Herroelen, Roel Leus, and Gerhard Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega*, 35(5):563–577, 2007.

[13] Patrik Haslum and Hector Geffner. Heuristic planning with time and resources. In *Sixth European Conference on Planning*, 2014.

[14] HH Hesselink, JL Raoul, and D Bowen. A cdm standard to guide implementation in europe. *Optimal Use of Scarce Airport Capacity, The Hague, April 2007*, 2008.

[15] AEA De icing/Anti-icing Working Group. Recommendations for de-icing / anti-icing aeroplanes on the ground. Technical report, AEA, July 2015.

[16] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[17] Rainer Kolisch and Soenke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European journal of operational research*, 174(1): 23–37, 2006.

[18] Antonín Komenda, Peter Novák, and Michal Pěchouček. Decentralized multi-agent plan repair in dynamic environments. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1239–1240. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[19] E Kondili, CC Pantelides, and RWH Sargent. A general algorithm for short-term scheduling of batch operations—i. milp formulation. *Computers & Chemical Engineering*, 17(2):211–227, 1993.

[20] Oumar Koné, Christian Artigues, Pierre Lopez, and Marcel Mongeau. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1): 3–13, 2011.

[21] Richard E Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106 (2):181–203, 1998.

[22] Doreen Krüger and Armin Scholl. Managing and modelling general resource transfers in (multi-) project scheduling. *OR spectrum*, 32(2):369–394, 2010.

[23] Pim van Leeuwen and Cees Witteveen. Temporal decoupling and determining resource needs of autonomous agents in the airport turnaround process. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 185–192. IEEE Computer Society, 2009.

[24] Linda Luther. Environmental impacts of airport operations: Maintenance, and expansion. *LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE*, 2007.

[25] CJ Madern. *Collaborative Decision Making at Schiphol Airport: Challenges of Decision Making in a Multi-actor Environment*. PhD thesis, TU Delft, Delft University of Technology, 2014.

[26] Xiaoyu Mao, AW Ter Mors, Nico Roos, and Cees Witteveen. Agent-based scheduling for aircraft deicing. In *BNAIC 2006: 18th BeNeLux Conference on Artificial Intelligence, Namur, Belgium, 5-6 October 2006*, 2006.

[27] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[28] Josefa Mula, David Peidro, Manuel Díaz-Madroñero, and Eduardo Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204(3):377–390, 2010.

[29] JEG Oosterman, Remco Ravenhorst, C Witteveen, and P Van Leeuwen. Determining resource needs of autonomous agents in decoupled plans. In *BNAIC 2008: The 20th Belgian-Netherlands Conference on Artificial Intelligence, Enschede, The Netherlands, 30-31 October 2008*, 2008.

[30] Gurobi Optimization et al. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, 2012.

[31] Jan A Persson, Paul Davidsson, Stefan J Johansson, and Fredrik Wernstedt. Combining agent-based approaches and classical optimization techniques. In *Third European Workshop on Multi-Agent Systems*, 2005.

[32] Dhanji R Prasanna. *Dependency injection*. Manning Publications Co., 2009.

[33] Schiphol. Cdm at schiphol. `http://www.schiphol.nl/B2B/CDM/InformationAboutCDM/CDMAtSchiphol.htm`, 2016. Accessed: 2016-05-13.

[34] D Teunissen. Standard inputs for eurocontrol cost benefit analyses. Technical report, EUROCONTROL, 11 2015.

[35] Scott K Thomas, Robert P Cassoni, and Charles D MacArthur. Aircraft anti-icing and de-icing techniques and modeling. *Journal of Aircraft*, 33(5):841–854, 1996.

[36] Vicente Valls, Francisco Ballestín, and Sacramento Quintanilla. Justification and rcpsp: A technique that pays. *European Journal of Operational Research*, 165(2):375–386, 2005.

[37] Cheng-Lung Wu and Robert E Caves. Aircraft operational costs and turnaround efficiency at airports. *Journal of Air Transport Management*, 6(4):201–208, 2000.