Mobility & Built Environment
www.tno.nl
+31 88 866 00 00
info@tno.nl

TNO 2024 R11490V2 – 9 October 2025

Autonomy and self-organisation in airside baggage transport

BrightSky WP3.1

Author(s)	Ruben Fransen, Lola Sprenger, Coen van Leeuwen, Sylvie Poels
Copy number	2024-STL-REP-100354245V2
Number of pages	59 (excl. front and back cover)
Number of appendices	1
Sponsor	RvO
Number of copies	060.49014

Summary

This study developed central and decentral fleet control methods and a simulation model to evaluate baggage transport fleet performance for these different methods. The goal was to develop a simulation model to identify the impact of different ways of organisation on baggage transport on Schiphol Airport and to research possible gains of fleet collaboration by sharing tugs assets amongst fleets. Baggage transport processes have been mapped and modelled and subsequently instances with real world data have been simulated to compare self-organising baggage tugs with central controlled baggage tugs, which lead to the following conclusions.

Decentral control for system performance in limited computation time
Decentral control for baggage tugs shows better fleet performance in total on-time delivered bags and distance driven by baggage tugs compared to central methods in limited computation time. Fast computation time is crucial to ensure tug fleets can quickly reschedule given airport dynamics. The better fleet performance is a result of the decision making process in the decentral control method being more effective in trailer train formation leading to on average longer trailer trains and therefore making more effective use of available tug capacity. Hyperparameter optimisation in the shape of the set up ofthe single tug objective function is crucial in the decentral control method to ensure agent interactions lead to effective conflict resolutions and tug assignment decisions.

Characteristics of autonomous and human-operated tugs define scheduling constraints
Both the developed decentral and central control methods can schedule manned and unmanned baggage tugs. Unmanned, autonomous, baggage tugs have different vehicle properties, such as average driving speed, ability to handle situations and how they communicate with other airside stakeholders. These properties need to be well defined and known, as these properties impact the performance level and operational capability of the vehicles. Information on these properties and characteristics is therefore crucial as it constraints decision making fleet control methods.

Simulation and scheduling model for multi-fleet baggage transport
Above conclusions have been based on simulation results of 8 hours of realistic baggage transport demand for KLM Ground Services on Schiphol Airport. Discrete event simulation has been used in combination with scheduling heuristics for central control tower decision making and decentral decision making based on a Distributed Constrained Optimisation Problem (DCOP) formulation and solution methodology.

Sharing fleet assets can increase system performance
Having autonomous vehicles and autonomous decision making systems opens opportunities to scalable and automated asset sharing, for example through the sharing of baggage tugs amongst different airside fleets. Simulation of collaboration between the inbound and outbound baggage fleet of KLM Ground Services shows an increase in on-time performance of 55% compared to single fleet operations without asset sharing. This implies that the tug fleets can be used more effectively in collaboration with each other, which could be really useful in peak demand periods.

Contents

Abbreviations

ACO	Ant Colony Optimisation
DCOP	Distributed Constrained Optimisation Problem
DES	Discrete Event Simulation
FMS	Fleet Management System
LVNL	Luchtverkeersleiding Nederland
MDP	Markov Decision Process
MILP	Mixed Integer Linear Program
OR	Operations Research
OTP	On-time performance
PSO	Particle Swarm Optimisation
TLO	Teamleider Omdraaien – Team lead turnaround process.
TMS	Transport Management System
ULD	Unit Loading Device
VOP	Vliegtuig Opstel Plaats – Airplane Parking Stand
WMS	Warehouse Management System

1 Introduction

1.1 Research context

Amsterdam Schiphol Airport (shortened to Schiphol) ranks 4th in the list busiest airports in Europe, with over sixty million passengers in 2023. To maintain this position within Europe, the airport needs to keep innovating the airport ecosystem. This is the goal of the BrightSky project, a Dutch research and development project with 15 partners.

The luggage of all passengers that pass-through Schiphol needs to be transported to and from the airplanes. Schiphol is the worldwide transfer hub for airline KLM. A major share of the Schiphol passengers and their bags therefore require a transfer from an arriving plane to a departing plane in a limited amount of time. Transporting all the bags timely is a challenging task for the KLM ground handling service (called KLM Ground Services),, because they have a limited number of tug drivers and baggage handling crew. To ensure an optimal number of bags makes it to their flight in time creates the challenge of scheduling and operating the baggage handling assets as effectively as possible.

Figure 1: Train of two trailers with Unit Loading Devices (ULD) behind a baggage tug.

One of the challenges within the scheduling problem is a limited number of tugs and, mainly, a limited number of people to operate tugs and handle baggage. If schedules can be improved upon, then then system performance costs can be reduced. For example fewer resources, such as workforce, might be needed. Moreover, these resources are currently split between the incoming and outgoing flights for KLM Ground Services. This means that a tug operates only on incoming or only on outgoing flights. This fleet separation exists due to the different nature of the handling of the incoming and outgoing flights and different skills requirements for the tug drivers. Both fleets are controlled separately, but there are possibilities for sharing assets and drivers within these fleets that might benefit to the overall baggage transport performance. This concept of merging inbound and outbound fleets is not novel, it has been recognised by KLM Ground Services and seen as a potential direction for future changes. Controlling a fleet of vehicles and making the scheduling decisions of which asset performances which task can be organized in different manners from a central control tower to a distributed decision structure.

Another solution could be the automation of the baggage tugs. This requires further development of vehicle technology and creates requirements on the airside infrastructure for safe and smooth operations of autonomous vehicles.

Next to the physical technology, a system of autonomous vehicles driving around on the airside of Schiphol also raises questions on how to manage and control those vehicles. How to organize and control a fleet of autonomous vehicles? What level of autonomy and intelligence should those vehicles attain and how would this impact the baggage transport process? Should the vehicles be managed in a central way, or can they operate in a decentralised manner? This last question is not only relevant for the investigation of the automation of baggage tugs but is also interesting for reducing performance costs and lowering pressure on the work force by increasing the efficiency of used resources.

Currently, ground handlers assign tugs and tug drivers to transport tasks and therewith aim for effective asset use and transport performance. They collect all information on available assets and to be transported bags (tasks) into a single computer system and use scheduling methods to create scheduled assignments by matching tasks and tugs. We call this a 'central control method' because there is a single entity that oversees all transport assignments and assets and solves the scheduling problem as a hierarchal control entity. In general, the main advantage of such a central control method is having a full overview and full information. It therefore theoretically allows for optimal decision making.

An alternative to centralized control would be to decentralise the scheduling process, where there is not one single entity that decides which tugs execute which assignment. Instead, for example, the tugs could decide autonomously what a good assignment would be and execute this assignment. Such an approach could be inspired by natural, self-organising systems, such as beehives or ant colonies, that have many individual agents making decisions individually in the interest of the collective with a limited amount of communication. In case the baggage tugs are driving autonomously, could they also be organized in a decentral way, and would that benefit the overall performance of the airside baggage transport?

This study focusses on comparing these different control methods for the different airside baggage transport fleets (incoming and outgoing) and tries to answer the following main question: what type of control (central/decentral/hybrid) is most suitable for managing a system of multiple asset-sharing fleets of baggage tugs.The control methods have been applied to the situation of KLM Ground Services. A control method is considered suitable if it meets system requirements, including the requirements on the performance level of the operations that are being controlled. Performance goals for transport fleets in general can be summarised by maximising the amount of timely delivered baggage whilst minimising asset use and asset costs.

Multiple fleets can refer to the distinction that is made for resources allocated to the handling of either the incoming or outgoing flights. However, it can also refer to the difference in resource fleets per ground handler. Each handler has its own fleet of tugs. We focus on the first type of multiple fleets in this research; the fleets allocated to either incoming or outgoing flights. In other words, we focus on the multiple fleets within the operations a single ground handler.

A simulation model with different scheduling methods has been developed to research which type of control method is most suitable for the baggage transport processes under which conditions. This report describes the background and processes on the airside baggage transport, the fleet control methods that have been modelled, the developed simulation model and results of the different scheduling methods.

1.2 BrightSky

1.2.1 Overall project description

This report is the main deliverable of sub work package WP3.1 of the BrightSky project, that has been subsidised by the RVO mobility fund. This project describes itself in the following manner:

"The BrightSky project is an open partnership that aims to stimulate intensive and long-term collaboration with, and innovation at, many companies in the aviation sector. BrightSky is therefore a fertile breeding ground for R&D within and outside the aviation sector in general. BrightSky aims to allow Dutch aviation to play a leading role in Europe for a long time and to prepare itself to take advantage of new economic opportunities.

BrightSky emphasizes the following three integral aspects:
- ***Social innovation** – how do we obtain a healthy workforce where, with the right direction, we can develop talents in the direction demanded by technological developments and which meet the challenges presented by the market?*
- ***Digitization** – how can we use the opportunities that the innovative possibilities of digitization (storage, work content, scanning, data sharing and preventive maintenance) offer us?*
- ***Sustainability** – how do we approach the subject of sustainability (labour potential, ecosystem development, economic and environmental impact)? How do the work packages contribute to the sustainability goals for 2030?"*

This research relates to the overall BrightSky aspects as it focusses on further automation and digitization of the decision-making processes in fleet control and fleet organisation, with the goal to increase system efficiency and reduce energy use of the fleets, and reducing the work load of fleet planners at baggage handlers. Also enabling further development on full fleet automation and autonomous vehicles on the air side to reduce workload for air side workforce.

1.2.2 WP3 Smart Operations

BrightSky Work Package 3 researches an airport system that systematically makes the coordination and control of airport sub-systems more sustainable and digitized. Where digitization and automation are means to get to more resilient and efficient air side operations, making them more sustainable and reducing the workload for the air side workforce. The consortium with Vanderlande, T-Hive, Schiphol Airport, KLM Ground Services, Hogeschool van Amsterdam and TNO focusses on automation of baggage tugs on the airport of Schiphol. This relates to automation in the sense of autonomous driverless vehicles and autonomous control methods with digitized scheduling of fleet operations. Autonomous tugs have new properties which need to be considered in fleet management and they also provide opportunities for innovative fleet control methods. An example of this is automated decision making on tug level. Allowing tugs to make their own decisions instead of assigning tasks to tugs shifts the control from a central point to a decentral level. This leads to the focus of this sub work package 3.1: which fleet control method is most suitable?

2 Research plan

This chapter describes the plan of approach that was taken to analyse the impact of central and decentral control methods for the multi-fleet situation of the airside baggage transport. For the development of an effective simulation model a clear scope and modelling goal are required. The later paragraphs of this chapter go into the scope of this study, the scenarios to be analysed with the simulation model and the key performance indicators that are relevant for the airside stakeholders and the BrightSky consortium.

2.1 Research steps for multi-fleet baggage tugs

The overall goal was to develop a simulation model to evaluate different forms of organisation of baggage transport fleets.

For the development of the simulation the following steps were foreseen:

1. Literature study
2. Airside baggage transport process mapping
3. Identify properties of autonomous and manual baggage tugs
4. Develop and implement scheduling methods for baggage tug fleets
5. Develop and implement a simulation tool to evaluate scheduling methods.

2.1.1 Literature study

Research in this part has been focused on optimisation methods for the problem. Online search tools Dimensions, Google Scholar and Scopus have been used to identify relevant literature.

Amongst others key words as:
- “baggage transport”,
- “airside”,
- “agent-based simulation”,
- “multi-agent”,
- “fleet management”,
- “scheduling”,
- “self-organising” and
- “decentral optimisation” have been used.

Next to first results from these search queries, also citations and references related to the relevant first results have been scanned for relevant knowledge and research. An overview of this can be found in chapter 3.

2.1.2 Airside baggage transport process mapping

Before developing simulation and scheduling models a detailed description of the airside baggage transport is needed. Interviews with and tours at project partners Schiphol and KLM Ground Servicess provided insight in current day processes and procedures. Additionally combining this with existing process descriptions led to the process description in chapter 4.

To structure the interviews and information gathering process the following steps have been taken to get a good understanding of, and describe the problem of the airplane turnaround process and the baggage handling transport:

1. Identify the stakeholders and their roles in the baggage transport
2. Make a flow chart model of the current information and control process for the airside baggage fleet.
3. Collect and analyse data on current day operations.

2.1.3 Autonomous baggage tugs

A relevant innovation for future airside baggage transport is the autonomous baggage tug. Due to labour shortages, there is an increased need for reduction in personnel needs.
A question for this research is what the differences of autonomous tugs and current manual operated tugs are and how this impacts the fleet control and fleet performance.

During this research the focus shifted to controlling a fleet of vehicles, which is independent from the fact that these vehicles are autonomous or human operated. Therefore this step only resulted in a qualitative description of the difference in properties and operations of autonomous vehicles, and how these differences could have impact on the fleet operations and the fleet control methodology.

The questions that have been answered are:

1. What are the differences in properties between human operated tugs and autonomous tugs?
2. What is the impact of these different properties on the airside baggage process?

2.1.4 Central and decentral multi-fleet scheduling

The main question of the research concerns the difference between central and decentral control on different fleet scenarios: single-fleet and multi-fleet. With single fleet we describe the situation where there are individual fleets having their fleet specific sets of tasks to execute. In case of the KLM Ground Services on Schiphol airport this is for example the inbound baggage transport fleet only executing the inbound baggage transport. With multi-fleet we describe the situation where multiple single fleets collaborate to share asset capacity to execute all tasks more effectively. For example, combining the outbound baggage fleet and the inbound baggage fleet. This is independent of how these fleets are organized, central or decentral.

We define central control as the situation where there is a single, central, entity which holds all information on assets and tasks and this entity also decides on the assignments of tasks to tugs. With decentral control, we define the opposite situation where the tugs are individual entities that decide which assignments they are going to execute and communicate or negotiate these decisions amongst each other.

One of the goals of the simulation model is to quantify the effect of collaboration in the multi-fleet scenario compared to the single fleet situation. Given the different options to organize the fleets, which also have impact on the fleet performance, there are already four different scenarios to simulate:

Table 1: Four different organisational scenarios with single and multi fleet, and central and decentral control.

	Single fleet	Multi Fleet
Central	Inbound – central control Outbound – central control	Inbound & outbound – central control
Decentral	Inbound – decentral control Outbound – decentral control	Inbound & outbound – decentral control

The baggage fleet operates on the airside of Schiphol. Schiphol as an infrastructure manager is responsible for the overall airport performance and there can on a tactical level steer the fleet operations by providing boundaries and regulations for the fleet control systems. Multiple ground handling services are operating at Schiphol. For this study, we focus on the ground handling services of KLM.

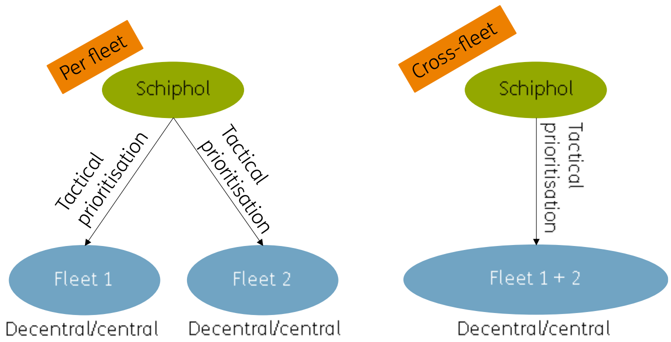


Figure 2: Concept of per fleet or cross-fleet tactical prioritisation by Schiphol. In the use case of this study, the fleets are owned by a single company, KLM Ground Services.

In the flowchart above, the fleet-blocks are the central control entities (fleet 1; fleet 2; fleet 1 + 2 combined). These entities control the vehicles at central level. At decentral level, the vehicles coordinate between each other how to organise themselves, within their own control entity (fleet 1; fleet 2; fleet 1 + 2). The control specifications are further explained below (see section "Steps"). All control methods need to take operational conditions and tactical prioritisation from Schiphol into account. An example of limiting operational conditions occur when there are weather conditions with limited sight, then it is forbidden for baggage tugs to cross airplane routes on the air side.

2.1.4.1 Single-fleet scheduling (2 scenarios)

The development of the scheduling model is an incremental process. The first goal for the scheduling model was to simulate a single fleet of baggage tugs. In more detail, a model that can simulate single fleet planning in a central and decentral control structure and compare the impact of the different scheduling approaches.

With single fleet we define the situation where there is one fleet of baggage tugs which has to transport a set of baggage shipments and need to decide which tug is going to transport which shipments. Where in the central control structure there is a single entity with all information and a decision making, scheduling method, to assign all batches to baggage tugs.

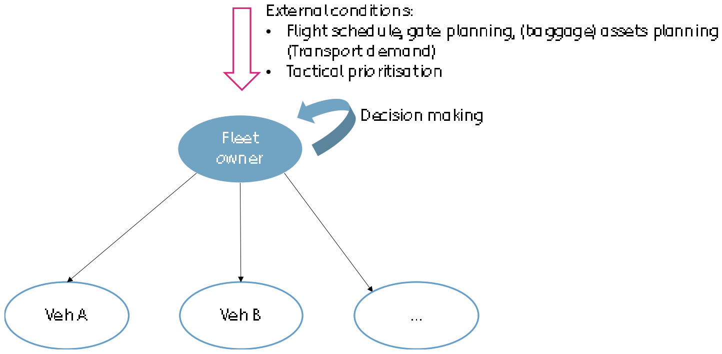


Figure 3: Concept of single fleet central control.

In decentral, or also called distributed, control scenario there is not a single central entity that makes the decisions. On the contrary, the decision making is distributed over multiple entities. In our situation we define the decentral scenario where the vehicles, baggage tugs, decide upon which tasks they are going to execute. It is important that decisions made by the different tugs are aligned, to ensure that effective decisions are made. This requires communication between the tugs to come to a mutually agreed decision making. The details of this decentral decision making and decision alignment can be implemented in various ways. These are further explained in the literature review in chapter 3 and in chapter 6 on the simulation model.

Note that the situation where the vehicles decide is one of the possible forms of decentral decision-making process. Another situation with multiple decision-making entities for subsets of the vehicles could also be considered a form of decentralisation. In other words, there would be a number of control towers, one for each subset of the fleet, which is not that different from the situation with one central control tower. It could also be seen as a situation with multiple fleets, which we will discuss in the next paragraph.

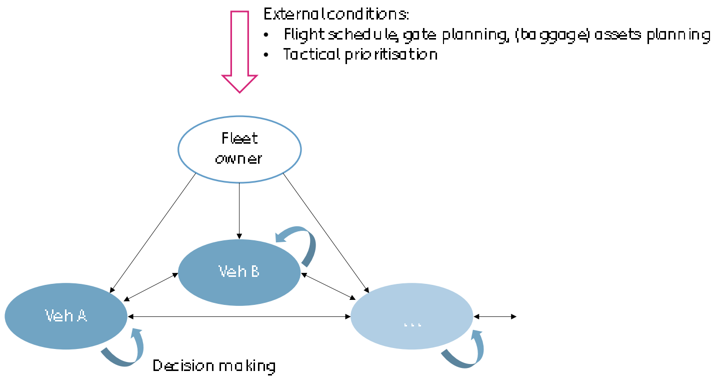


Figure 4: Concept of single fleet decentral control.

Developing the methodology for central and decentral scheduling and comparing these approaches has been done to provide insights in answering the following questions:

1. What are the differences between central and decentral planning of a (baggage) fleet?
2. What are the differences in organising and controlling a manned and unmanned fleet?
3. What are the differences (benefits and downsides) of the different control methods?

2.1.4.2 Multi-fleet scheduling (2 scenarios)

The next step for the scheduling models, after single fleet control approaches have been implemented, is the step to multi-fleet scheduling. In other words, to simulate scheduling of the two fleets in a cooperating scenario with a central and decentral control structure. This is illustrated by the following figures.

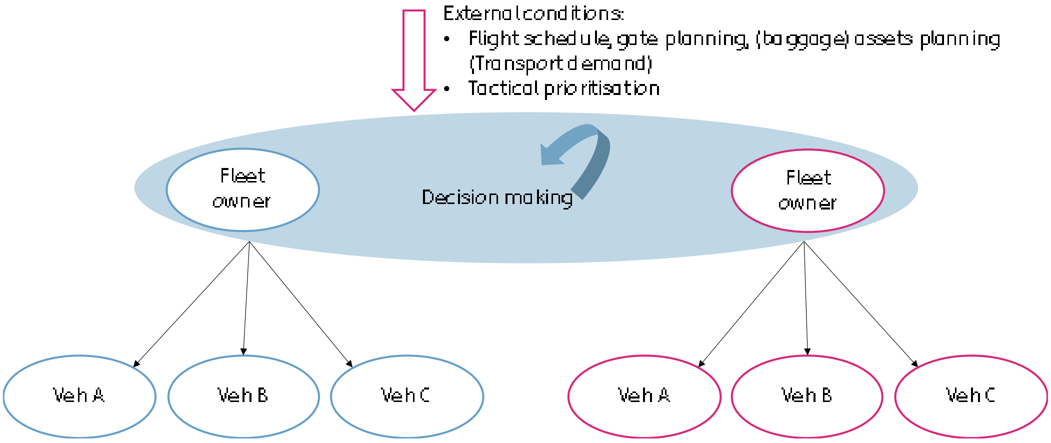


Figure 5: Concept of multi-fleet central control.

In the multi-fleet scenario with central control a similar control tower methodology as single fleet central control can be used. The central control method can take all combined tasks and combined assets of both fleets into account and use the same assignment methodology to get to a task assignment for all vehicles. Similar in the decentral scenario, all the vehicles that have to make decisions on their tasks would need access to information on all available tasks and communicate with other vehicles in the fleet to align the to be made decisions. Which contains the risk that a lot of vehicle-to-vehicle communication is necessary and is shown in Figure 6.

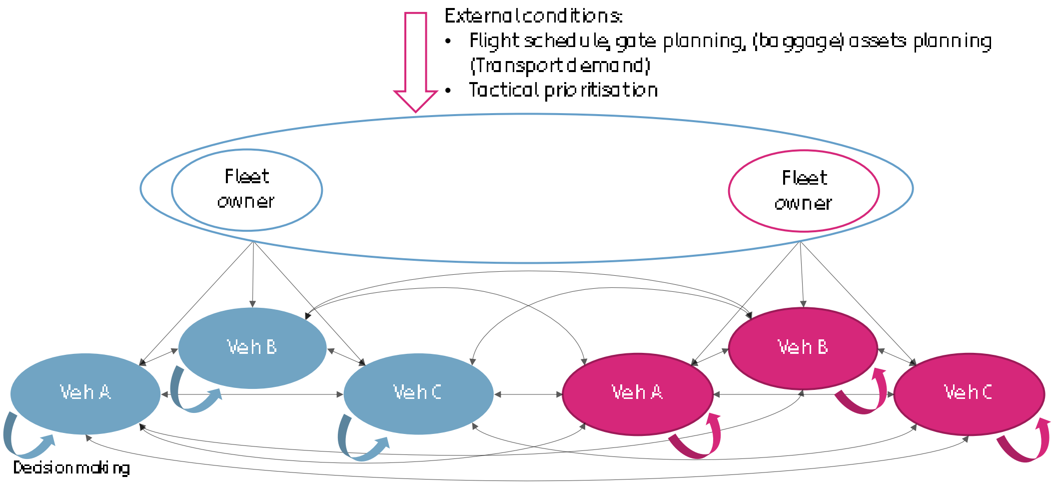


Figure 6: Concept of multi-fleet decentral control.

Further developing the scheduling methodologies to be applied on the multi-fleet scenarios provided scheduling models that give insight in answering the following questions:

1. What are the effects of sharing resources (vehicles and drivers) amongst fleets?
2. What are the differences (benefits and disadvantages) of the different control methods?

2.1.5 Discrete Event Simulation model

The goal of this last step is to evaluate schedules resulting from different control methods by simulation. Simulating execution of the schedules to identify impact on key performance indicators can be done by Discrete Event Simulation (DES). Where the environment to be simulated results from the airside baggage process mapping and the events to be simulated come from the scheduled events along with optional, additional, dynamics in the system. In that way the simulation can propagate the baggage transport process through time and monitor the simulated execution to get insight in performance indicator levels in different scenarios. More information about the used simulation tool is available in section 6.1.3.

Discrete event simulation as simulation methodology was chosen, because the to be modelled situations and processes have a clear event-based structure, in which the processes between events can be quite realistically modelled with known system dynamics. The events to be simulated are defined by the task assignment resulting from one of the scheduling methodologies. Up to now no other events, such stochastic events that represent real world disturbances have been included. Subsequently also no rescheduling to deal with disruptive events has yet been implemented.

2.2 Scope of study

At the start of the research this scope was defined. To clearly bound the research in this study and to define what the simulation should and should not entail. The following elements define the scope of the elements studied and modelled:

- All processes related to the transport of baggage from the airport baggage system to the airplanes and vice versa. This includes outbound and inbound baggage tug fleets Schiphol airside. This includes scheduling of picking-up empty trailers that are required for baggage transport, but it excludes additional repositioning of empty trailers to support the baggage transport process.
- To incorporate sufficient interaction between the fleets the simulation model focusses on the operation of a single baggage handler. The KLM ground handler (KLM Ground Services) operates separate fleets for inbound, outbound, and empty trailers, given current information from interviews with Schiphol and from literature (van Loenhout, 2019). The model simulates different fleets of a single baggage handler.
- The operational area for the baggage transport of this ground handler can be any of the aircraft parking stands (vliegtuigopstelplaats; VOP) or baggage depot, or on the roads between these locations on Schiphol Airport.
- Time frame of simulation: Focus on D-1 planning; schedule all vehicles for the baggage transport for the next day. The methods have been developed and tested for day ahead scheduling. In theory the control methods should also function with real-time disruptions, hence, should be able to do real-time rescheduling with updated inputs, but this has not been researched in this study.
- The developed control methods focus on scheduling vehicles (tugs) to baggage transports (trailers). See next paragraph on decision making for more details.

- Data on flight and gate schedule with baggage volumes and data on VOP-locations and assigned baggage laterals or baggage carrousels are considered model input.
- The developed simulation model focusses on the main processes of baggage handling during normal operations at Schiphol airport. No disturbances or stochastics are included up to now and process ends at delivering baggage (timely or not), or purposely not scheduling transport due to lack of assets. So-called miss-handled baggage is counted but no further handling processes are included.

2.2.1 Scope of decision making in fleet control algorithms

In this report we talk about fleet control, with this we imply the decision making of which tug is going to execute which transport task. In practice there is a bigger range of decisions to be made around operating a fleet of tugs, from how many tugs and drivers there are in the fleet to which tug has right of way on airside traffic junctions. Table 2 presents 5 layers of decision making and which decision maker has to make which decisions in the scenarios we define as central and decentral.

Vehicle decisions on which route to take and how to operate vehicles on traffic junctions are currently made by tug drivers and these can be effectively taken with local information and local interaction with other road users. Note also for the route choice there generally is only one logical route option between 2 locations on Schiphol airport and there is not much to decide upon.

Fleet decisions on how many vehicles in the fleet and specifying which assignments there are need to be executed is assumed to be predefined by the number of bags per flight, parking stand where a flight is handled and laterals assigned to this flight.

Following from these decision-making levels, in this study we compare the difference in decision making on which vehicle executes which assignment. These decisions can be made on vehicle level with decentral optimisation methods or with a central optimisation method in a so-called control tower.

Table 2: Overview and different levels of decision making in baggage transport.

Decisions	Decision maker Central	 Decentral	Input / Model / Out of scope
How many resources: vehicles + drivers Per activity (inbound, outbound, empties)	Fleet owner	Fleet owner	Input
From transport demand to demand task: how many carts / trains per flight/VOP. How many trips/tasks are there to assign vehicles to?	Fleet owner	Fleet owner	Input / basic rules on how to process different baggage (transfer, business class, ...)
Resource scheduling (which resources execute which demand task at what point in time)	**Fleet owner**	**Vehicle / driver**	**Model decision making**
Route choice	Vehicle / driver	Vehicle / driver	In general, each trip on Schiphol has only one logical route.
Traffic interaction	Vehicle / driver	Vehicle / driver	Out of scope

2.3 Scenarios of central and decentral control

The table below gives a summarized overview of the scenarios to simulate with the different control structures and vehicle fleets in the model. All scenarios focus on autonomous vehicles, to answer the question which impact different control method of autonomous vehicles have. The research question on the difference in control methods between human-operated and autonomous vehicles has not been simulated and only approached qualitatively, because for a control method the concept of a vehicle, an asset, does not significantly change if it has a or has no human driver. The qualitative results are presented in chapter 5. Comparing the different fleet control scenarios leads to 4 scenarios as presented in Table 3.

Table 3: Overview of scenario's to be modelled and simulated.

Scenario	**1a**	**1b**	**2a**	**2b**
Fleets:	Single fleet (outbound)	Single fleet (outbound)	Multi fleet (2) (outbound and inbound)	Multi fleet (2) (outbound and inbound)
Control:	Central	Decentral	Central	Decentral
Autonomous or human operated vehicles:	Autonomous	Autonomous	2x Autonomous	2x Autonomous

The scenarios will be experimented on various instances, datasets representing different periods of operations on the airside of Schiphol.

2.4 Performance Indicators

Performance indicators are defined to research the question what the impact of the different control methods on the fleet performance is. The main objective of the baggage handler as described is the on-time performance of baggage delivery, delivering as many bags in time to the departing flights or to the reclaim system for arriving passengers. In parallel the handler can maximize his profit by minimizing costs for asset use and required assets. Minimizing asset use can also reduce energy requirements and contribute to sustainability goals.

The following performance indicators have been implemented in the simulation model for evaluation and comparison of the fleet control systems:

1. the number of on-time delivered bags,
2. the total driving time of the tugs and
3. the number of used trailers.

The goals for the simulation model and control methods will be to maximize the number of on-time delivered bags (1.) and in parallel to minimize the asset use defined by total driving time (2.) and used number of empty trailers (3.).

This chapter described: the incremental scheduling model steps, the discrete event simulation model, scenarios for analysis and the performance indicators provided the base for this research. The remainder of the report will describe the outcomes of these executed steps.

3 Literature review

We have reviewed the state-of-the-art from two perspectives: (1) mathematical optimisation of scheduling problems and (2) applied research focused on airport process optimisation. In the following sections we provide a non-exhaustive overview of the available literature.

3.1 Optimisation of scheduling problems

3.1.1 General optimisation methods

Scheduling problems are a typical operations research (OR) application. A classic approach to solving such a problem is to formulate the problem as a (Mixed) Integer Linear Program. Examples of generic scheduling problems in operations research are job-shop scheduling problems [1] [2]; vehicle scheduling problems [3] [4]; vehicle rescheduling problems [5]; order scheduling problems [6]. A drawback of MILPs is that these are NP-hard problems [2]. This means that solution techniques that are guaranteed to find an optimal solution require a lot of computation effort. Running these algorithms may take a long time, especially for large problems with many decision variables. MILPs can be solved using solvers. These are commercially available, but also open source. The open-source solvers are easy to obtain, but often have lower computation efficiency.

Next to MILP-solvers there also exist metaheuristic methods to find solutions to scheduling problems. One category are meta genetic algorithms (see, for example, [4]). Metaheuristics provide a solution to the problem, but they do not provide guarantees about the quality of the solution. Some of these metaheuristic methods are based on decentral approaches, such as particle swarm optimisation [7] or ant colony optimisation [8], but these methods are generally used by a single central entity such as a control tower to find a solution to the scheduling problem. Sörensen 2015 [9] warns for the lack of comparison and scientific rigor with which these new evolutionary and metaphorical methods are tested.

Next to more classic OR-based approaches with solvers and metaheuristics, in recent years more and more examples of using machine learning to train decision making algorithms are being developed, by formulating the problem as a Markov Decisions Process (MDP) and training a reinforcement learning algorithm to make decisions. Research up to know shows that machine learning in general does not provide better schedules than above mentioned methods, as it is posed by challenges on how to effectively formulate the MDP and decision options, and these methods require intensive computational effort for training. For now, these methods are considered out of scope as it is not a goal of this research to further develop machine learning based decision making.

3.1.2 Decentral optimisation problems

In this research we want to solve the scheduling problem in a decentral, distributed, manner with multiple decision agents.

When decentralising the scheduling problem there is a specific framework for mathematically formulating the problem, which is called Distributed Constraint Optimisation Problem (DCOP). Which formulates a decentral approach to the problem with multiple agents that optimise a collective decision problem.

> *"By definition of DCOP, the involved agents are part of a team and need to cooperate in order to perform well on the global task. Usually in DCOPs, cooperation between agents is achieved by passing messages from one agent to another."* [10]

DCOPs are also NP-hard [11]. DCOPs are derived from distributed constraint satisfaction problems (DCSPs), in which the goal is to find a set of assignments to variables that satisfies all constraints among agents [12]. Asymmetric DCOPs (ADCOP) are applied in cases where agents have heterogeneous constraint and cost functions.

There are multiple algorithms provided by the literature to solve DCOPs. We shortly discuss four different DCOP-solver approaches.
Van Leeuwen and Pawelczak [10] introduce CoCoA (Cooperative Constraint Approximation), which uses a non-iterative, semi-greedy approach with a one-step look ahead. It can solve ADCOPs and is, in general, better (solutions closer to the optimum) and faster than other (A)DCOP-solvers.

"Synchronous Branch and Bound", a derivative of the classic operations research algorithm, is introduced by Hirayama and Yokoo [12]. The usual branch and bound algorithm is adjusted for the distributed problem setting of DCOPs. Agents are not allowed to assign or change their variable values in parallel, and this algorithm can thus not be used to take advantage of parallel changes. The authors also present the "Iterative Distributed Breakout"-algorithm. An important drawback of this algorithm is that is not guaranteed to find a solution, even if a solution exists. It can also not determine if a solution exists.

Grinshpoun et al. [13] propose a privacy preserving "regional optimal"-algorithm. It is a set of local search algorithms, applied to solving DCOPs. Agents iteratively collect assignment information for a selected group of other agents and neighbouring agents outside of the group. The agents focus on finding an optimal set of assignments for the agents in their group. The runtime of this algorithm, however, is exponential with the size of the groups.

The question for this research now is which method to use to solve the decentral baggage transport problem. Part of the DCOP methods have been compared by Fioretti et al. [14]. From these comparisons we find that CoCoA performs better than other ADCOP solving methods on computational requirements and is known to be more privacy preserving. Moreover, it is known from earlier work [6] that CoCoA provides effective schedules for transport planning problems and is therefore deemed fit to also solve the decentral baggage transport problem.

3.2 Optimisation of airport processes

There is also literature available on optimising airport processes besides research on general scheduling problems. This can be split into two categories: allocation of static resources and optimising (part of) the operations chain for incoming or outgoing flights. From these two categories, it is already apparent that there is a lack of research on the allocation of dynamic resources, e.g., vehicles. For completeness, however, we touch upon the two abovementioned categories. For example, [15] focus on allocating baggage chutes and baggage belts, with a case study at Copenhagen Airport.

The allocation of baggage carousels is also considered in [16]. [17] focus on the optimisation of assigning baggage of departing flights to piers in the baggage-handling facilities. Other literature researching baggage handling can be found in [18] (discrete event simulation of inbound baggage handling), [19] (optimising outbound baggage handling through the assignment of facilities and decision on handling start time) and [20] (optimising inbound baggage handling through allocating infeed stations and baggage carousels). [21] is a master thesis with baggage handling at KLM as focus point. It improves an earlier model that has been created for KLM for the assignment of flights to baggage halls.

Besides the allocation of static resources, we also discuss some literature that focuses on optimising (part of) the operations chain. [22] has researched the multi-objective scheduling of ground handling processes as a whole. A short state of the art is given, in which only baggage scheduling for transfer flights is discussed. An extensive overview of the modelling techniques is given. These include constraint programming, the vehicle routing problem with time windows and multi-objective optimisation. [23] have also modelled the transfer baggage scheduling problem using a static mixed integer model, which is also used in a dynamic environment. This is applied to Frankfurt Airport and solved using a commercial MILP-solver. [24] uses a variant of the vehicle routing problem to model short transfer baggage. Their model is an integer programming model which is solved using a parameterized greedy algorithm. [25] investigates the estimation of the demand for ground resources and inclusion of flight delay in ground resource planning for KLM. Machine learning methods are applied for both subjects.

The literature research has shown that there is a gap in the literature for the scheduling of baggage tugs and trailers, in other words, for the dynamic resources. This is also emphasized in [19] and [4]. [26] investigate workforce (i.e., personnel) planning. They first determine the staff requirements and follow this with an optimisation over the handler shifts and task assignments. Both steps are formulated with an ILP and solved with commercial solver CPLEX. [4] is one of the very few papers which focus on the baggage vehicle scheduling problem. They apply a genetic algorithm to the problem. We will extend the literature by applying both central and decentral methods to this specific problem.

To benchmark the to be developed methods we choose to use a greedy heuristic method based upon First-In First Out principle, as it is an effective and fast method that is easy to implement, just as [6]. The central method initially was based on an MILP as used in above literature, but due to computation time restriction an alternative central method was needed. An extended version of the greedy heuristic was developed in this study, which shows to provide decent schedules in limited computation time. For the decentral approach the DCOP-solver CoCoA [10] as it is an effective method to solving DCOP-problems.

4 Airside baggage transport process

In this chapter the simulated processes are described in detail. Before we go into the details of the baggage transport process, we first provide an overall description of the airplane turnaround process, of which the baggage (un)loading are essential subprocesses. Followingly, we continue with the baggage transport process and how current fleet operations are managed and transport tasks are assigned to baggage tugs.

4.1 Airplane turnaround process

The airplane turnaround process is considered to be everything that happens on an airplane parking stand (*Dutch*: vliegtuigopstelplaats, abbreviated as VOP) between arrival of an incoming flight and departure of the outgoing flight. For sake of simplicity, we assume that all airplanes arrive and depart from the same VOP in a single turnaround process. This is true for the majority of airplanes visiting Schiphol Airport. In this section we describe the actors taking part in the turnaround process and that are relevant for the baggage transport process. Other actors and details on the turnaround subprocesses and a flowchart of the airport turnaround processes are provided in Appendix A.1.2.

4.1.1 Actors

The three most relevant actors in the baggage handling process on the airside are:

1. Airport
2. Airline
3. Ground handler

The general roles of these actors are:

1. Airport manages the airport infrastructure. The main goal is to ensure smooth and safe operations of the airport. They minimize flight delays and want all flights to adhere to flight and VOP schedules. The airport controls Baggage Handlings Systems and VOP schedule.
2. Airline operates the aircraft and optimises on cost minimization in their flight network.
3. Ground handler is contracted by an airline to operate all ground operations for the airplane turnaround process. Concerning the baggage process, they are responsible for loading/unloading bags and cargo to/from airplanes and transporting bags to and from the VOP.

These actors have their own objectives in the baggage turnaround processes, which need to be well defined and understood to develop the scheduling models.

4.1.2 Turnaround objectives

The turnaround process is a cooperative process of the different actors. Each actor has their own stake in the turnaround process, these can be complimentary with other actors, but they can also be conflicting. Therefore, we consider this a cooperative and not a collaborative process. Cooperation occurs when different parties work together in certain processes to achieve a good performance in line with each individual party's interest, there is no continuous common goal, because of possible conflicting interests. Collaboration occurs when different parties work together with a shared, common goal. The different objectives of the different actors, making this a cooperation, are explained below.

The goal of the airport is that the target turnaround time can be achieved (goal of airport) and airport operations run as scheduled and delays are minimized. The airport is evaluated on timeliness of operations.

The airline wants to maximize its turnover, which is done by minimizing the airplane operating costs and maximizing operating turnover. For the turnaround process this means there is a balance between minimizing costs of using airport services and minimizing sunk costs for passengers and/or bags that miss their flights. For example, waiting a few more minutes could imply some more bags and passengers can make the flight, which saves the airline costs for replacing flights for miss handled bags or missed transfers. This decision can conflict with the airport goal to achieve on-schedule departure.

The main objective of the baggage handler is to ensure the airplane turnaround process is executed as smooth as possible. In other words, the baggage handler wants to provide a good quality service to the airline, which pays the handler for their services. They have to do this with a limited amount of available assets, which in peak moments can require decision making on which flight or airline to give priority over another.

4.2 Airside baggage process

For the development of simulation and decision-making models for the airside baggage transport a clear and detailed description of the process steps and decision making for these processes is needed. We first give a global description of the baggage flow and then specify all process rules and different baggage flows there are on Schiphol airport.

4.2.1 Departure and reclaim baggage flows

The baggage flow on Schiphol airport is separated into the 2 major flows, departure baggage flow and reclaim baggage flow:

- Departure baggage travels via Schiphol baggage system and enters the system via check-in or transfer processes.
- Reclaim baggage is unloaded on reclaim baggage assets which are only connected to the baggage reclaim carrousels.

In more detail, the bags at Schiphol Airport can be categorized into 4 different baggage flows:

1. Airplane to reclaim
2. Airplane to baggage system (transfer)
3. System to Airplane (check-in + transfer)
4. Airplane to airplane (tail-to-tail) (transfer)

Figure 7 shows how the bags flow across the airport and which flows traverse the road network and require tugs and trailers.

The ground handler decides which of the transfer processes, via the baggage system or tail-to-tail, is used for which bags. Minimal connecting times, minimal time for a passenger and bag to transfer from flight to flight, are defined by the airline. For intercontinental flights this is 50 minutes and for transfer connections within Europe this is 40 minutes. These times are too short for transferring bags via the Schiphol baggage system. In general, 70 minutes is the required minimal handling time for transfer bags via the baggage system.

Therefore, this short connection baggage (SHOCON) can be treated differently and transferred from arrival airplane directly to a departing airplane (provided security regulations allow it), this is called tail-to-tail baggage. This is an expensive process as assets, tugs and drivers, need to put in effort for a relatively small number of bags.

In case baggage does not make it in time to the transfer flight, it is labelled as NOC (no connection) baggage and buffered for a next flight to the passenger's destination. This is not always allowed due to security regulations.

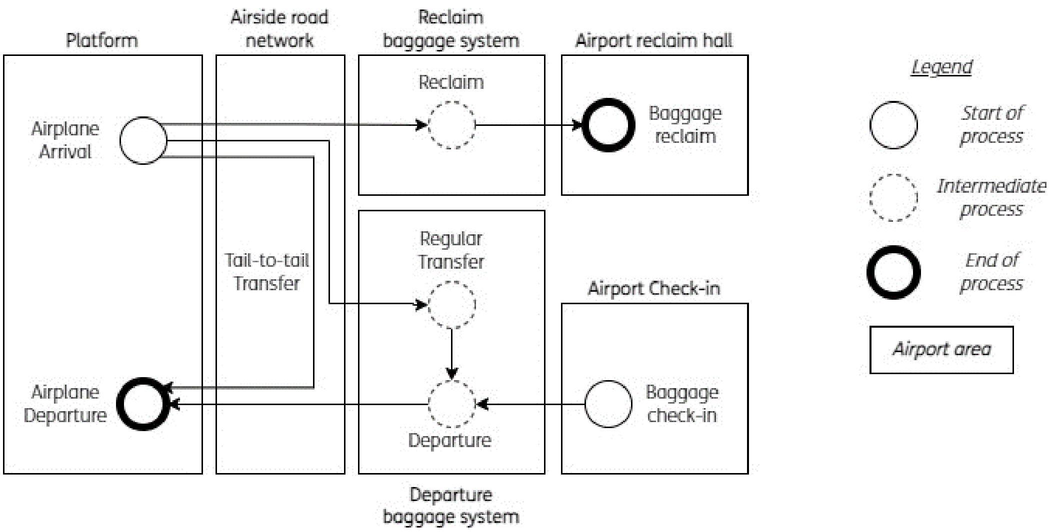


Figure 7: Generic overview of baggage flows on Schiphol Airport.

4.2.2 Baggage transport process rules

Figure 8: Lateral with bags on Schiphol Airport.

Within the baggage transport processes there are different steps with different systems and actors handling the bags.

For the scheduling decisions and for the simulation model the following relevant process rules have been identified from interviews with Schiphol and KLM Ground Services and from the Schiphol-CDM-Operations-Manual:

1. Baggage can be driven to VOP if destined plane has landed and is inbound to the VOP.
2. In general one lateral (see Figure 8) is assigned to a single flight for certain period of time and a carrousel (see Figure 9) can be assigned to multiple flights in parallel.
3. A driver with tug is send to pick-up inbound baggage trailers at the VOP several minutes after a plane has arrived on blocks.
4. Drivers do not wait at VOP to return the empty trailers. In general, the driver decouples loaded trailers and leaves aircraft loading for the VOP-crew under supervision of the ground handling team lead (TLO: TeamLeider Omdraaien).
5. Driver is expected to drive shortest / direct route between VOP and baggage asset. In general, there is only one logical route between each VOP and baggage asset; a driver is generally not going to circle around a terminal pier and drive extra distance. This can exceptionally occur in case of road works or other road disruptions.
6. A tug is allowed to drive a train of up to maximum 6 trailers on the airside. For some baggage basements lower maximum of 3 or 4 trailers of specified.
7. In general, the transport of outbound baggage is segregated into 3 outbound trains, to spread the workload for the ground-crew effectively over time. To prevent all bags being delivered just before flight departure, leaving almost no loading time for the ground crew. Depending on the number of bags the number of trains can be changed.

Figure 9: Carrousel for bags on Schiphol airport.

4.2.3 Length of trailer trains

The number of trailers that are needed to get all bags from a lateral to an airplane or vice versa is determined by the number of bags or number of ULDs that need to be transported. Each flight has a loading manifest which defines number of bags or ULDs are aboard (arrival) or need to be loaded (departure). The average number of bags per trailer lies between 35 and 40. Each ULD requires one trailer. In the simulation model we define a 'batch' as the set of bags that fill one trailer for loose bags or each ULD as a 'batch'. This means a batch is equal to a single trailer load.

4.2.4 Empty trailers and empty ULDs

For picking-up batches a tug needs empty trailers. Empty trailers are generally stored on general parking stand, but also on VOPs and baggage areas (reclaim, claim). Next to empty trailers also empty ULDs are needed for loading bags into a wide-body aircraft. For now, in this study, it is assumed that empty ULD-trailers contain an empty ULD that is reusable, similar to an empty trailer. In practice, the empty ULD-scheduling is more complex, which is ignored for now. This implies that storage and availability of ULD is not included in the developed model. Making the model not directly applicable in practice, as location and availability of empty ULDs impacts fleet performance.

4.2.5 Baggage categories and prioritization

In parallel with different categories of passengers, there are also different categories of bags.

In general, there are 3 segregations flight, in order of prioritization:

1. Business class
2. Transfer
3. Economy class

In some cases, a flight has multiple destinations e.g. there can be a flight from Amsterdam to Bonaire and Curacao, in that case there are 6 segregations, 3 for each of the destinations. Sometimes even more segregations possible. Each segregation can have a different prioritization for the ground handler.

4.2.6 Example of exceptions and disturbances

As in every process, exceptions and disturbances can occur that distort the process out of regular handling, the most common exceptions and consequences are:

1. A delayed flight
    a. Checked-in/Claimed Baggage can be buffered in baggage system or in handlers' baggage trailers.
2. Last minute-change of VOP
    a. This implies that assigned resources have to change their destined locations, which can imply higher or lower travel time from the location of previous task.
3. Bags that miss their flight
    a. Bags that have missed their flight go to the Miss-Handled-Baggage process.

Note that in practice, next to these examples, a lot of different disruptions might occur. To develop a practical applicable fleet management system requires exception management. Which can either be implemented in the system or otherwise would require external intervention by a human. Up to now we did not yet include such exceptions and exception handling in the developed simulation model, but this might be interesting for further research.

4.3 Tactical scheduling for baggage handling

The developed decision making and simulation models focus on decisions in the transport process between VOPs and Laterals of the baggage systems. Which VOPs or lateral is assigned to a flight and how many tugs, trailers and drivers are available are considered input. However, it is good to note that there are other actors and processes that define and decide on these inputs. The so-called tactical decision-making process can, for example, take into account staff shortage of a ground handler and subsequently optimise assignment of laterals to reduce driving distance to VOPs.

This level of decision making is out of scope for developed models, but we briefly describe these processes to give broader understanding of the baggage handling process and factors that determine and influence the operational scheduling process in the simulation model.

4.3.1 Assignment of VOP and laterals to flights

For the resource allocation in airside baggage transport services, it is relevant to know:

1. which flight is going to land on which VOP and
2. via which baggage assets the suitcases are handled.

There are some schedules that are made on tactical level that are input for the to be developed airside baggage transport simulation. Tactical decisions, that have to deal with the number of resources available to execute baggage transport, mainly concern the number of crew members available. Moreover, strategic decisions such as the size of the vehicle fleet and infrastructural developments influence the process.

Most relevant tactical decisions that influence the boundary conditions and constraints for the operational baggage transport problem are:

1. The VOP that is assigned to an aircraft.
2. The assigned baggage assets (locations of lateral, carrousel, buffer areas…)
3. Available baggage handler crew
    a. Drivers for baggage tugs
    b. Asset crew for loading and unloading trailers at assets.
    c. VOP crew for loading and unloading trailers at VOPs and/or laterals
4. Repositioning of empty trailers

The scheduling of flight, VOP and assets is an ongoing process, ranging from a seasonal schedule to a tactical schedule which is made from 7 days in advance to an operational schedule for real time operations. This operational schedule of current day is called D-0. For schedules that lie in the future, this schedule is called D-1 (one day in advance) up to D-7 (one week in advance).

The tactical schedules D-7 up to D-1 is managed by Schiphol and updated in accordance with involved stakeholders, also called the sector partners.

4.3.2 Baggage handler crew scheduling

The ground handler is responsible for assigning vehicles, drivers and (un)loading crew to the baggage handling process. There are different ground handlers active on Schiphol. The drivers of the tugs and the tugs are assets of the ground handlers and they need to ensure they have enough assets available to execute the baggage transport process. This requires maintaining a pool of skilled people and assigning duty rosters to ensure enough crew members are available in the operational process. The available number of tug drivers and tugs during the day is considered input for the simulation model.

4.4 Airside and baggage handling data

Next to description of the processes to model, also empirical data is required for simulating airside processes as realistic as possible. For the development of the model and model scenarios data has been provided by Schiphol Airport and KLM Ground Services. In this section we describe the data that has been used.

The following three datasets have been provided:

- Geographical information system (GIS-data)
  - Location of VOPs and laterals
  - Road network connecting VOPs and laterals.
- Airport Flight data
  - Arrival and departure of planes at which VOPs
  - Passenger count per flight
  - Laterals of entry and departure flights
- KLM Baggage handling data
  - Bags per passenger
  - Share of transfers and reclaim for incoming flights

These data have been used to create input data for the simulation model. Cleaning, combining, and restructuring of the data was required to get to cohesive datasets as the provided data had different structures in naming conventions and taxonomy. One of the steps was to create a uniform list of locations and location names. Subsequently the GIS-data has been used to create a distance matrix with all travel distances between these locations at the airport. See Figure 10 for all VOPs and the road infrastructure.

Figure 10: Overview of VOPs and road infrastructure at Schiphol Airport.

4.4.1 Baggage transport problem data

The flight and baggage data has been used to generate so-called transport problem descriptions for the simulation model, defining the number of trailer loads for the model to schedule and simulate.

The flight data from Schiphol provided the following information per flight:

- VOP
- Laterals for reclaim and departure baggage
- The arrival or departure time
- Number of passengers

This data has been analysed for completeness and correctness, and to give an insight in the transport problem for the baggage handling fleet. For example, Figure 11 shows the distribution of the workload, in other words when the peak moments for the baggage handling crewing are during the day. This data however lacked information on number of bags per passenger and the transfer baggage. Therefore, this data has been combined with data from KLM.

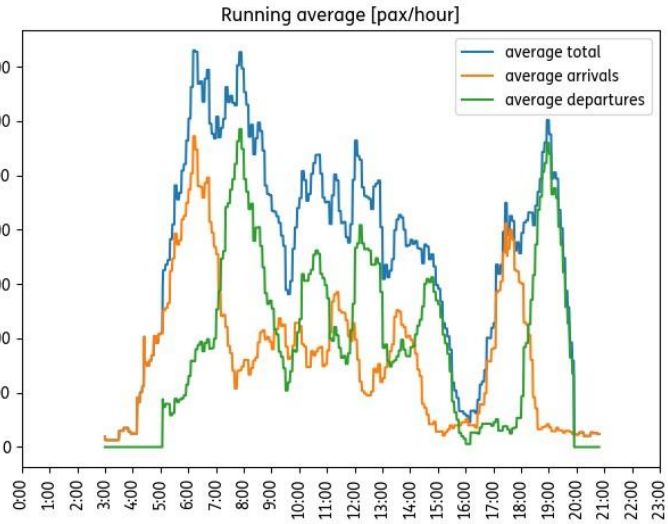


Figure 11: Distribution of passengers during a day at Schiphol Airport, average over 6 different days in 2021 and 2022.

KLM Ground Services provided a dataset with the number of bags per flight and shares of transfers per flights. As this data contains commercial sensitive information, we anonymised this data by abstracting only a distribution of the shares of transfers destinations per flight and bags per passenger. These distributions have been used in combination with the Schiphol data to generate realistic transport problem descriptions in format as provided in Appendix A.2.

5 Autonomy in baggage handling

This study focusses on how to manage an autonomous fleet of baggage tugs. To answer this question, we first define what further autonomy of systems could implicate in the baggage handling process.

In this chapter we elaborate on the autonomy on two different levels:

1. Automation of baggage tugs
2. Automation of fleet scheduling

Firstly, what does autonomy mean for baggage tugs and what does this implicate for managing a fleet of tugs? Secondly, what is needed to make the scheduling of the tugs a more autonomous, automated, process? The developments of making the vehicles, tugs, autonomous are researched in WP3.3 and WP3.4 of BrightSky.

Some operations that are closely related can also be automated, but these are out of scope for this study. These are for example the (un)loading of trailers or ULDs, (un)loading the airplanes or connecting trailers to form trailer trains behind a tug. Note that out of these examples the (un)loading trailers are currently already being executed autonomously on the Schiphol airside. The number of robot arms executing these processes is limited so only a small share of all bags are handled this way and a major part still requires manual (un)loading.

5.1 Autonomous vehicles

Sub work packages WP3.3 and WP3.4 look into automation of the baggage transport tugs. This change also has impact on how to organize and how to communicate with the vehicles from an organisational and scheduling perspective. We shortly describe what will change and how this will influence the scheduling process.

For an automated scheduling algorithm, the impact of changing a tug from human operated to autonomous is small. For a scheduling algorithm it will remain a similar asset with similar capacities, of driving trailers with or without baggage around the airport. Properties of the asset might change, such as the response time to communication, the airside routes the vehicle is allowed to access or the driving speed. These properties are very relevant to consider when making a schedule and assigning tasks to vehicles. Therefore, the different properties of human operated tugs and automated tugs need to be determined and evaluated critically.

Next to task assignment and execution it is also relevant to know the level of autonomy of a tug. Is it equipped to plainly follow ‘orders’ from a scheduling system or would it also include more intelligence.

For example, to solely change its assigned route in case of disruptions, or even provide feedback or advice to a scheduling entity to update or improve the scheduling process.

When looking at a fully decentralized fleet of autonomous vehicles, the vehicles would need sufficient information technology to incorporate and facilitate the algorithms for decentralized scheduling methodology, including the communication between autonomous vehicles to allow for negotiation and ensure tasks are effectively assigned across the fleet.

5.2 Autonomous asset scheduling

Autonomy in asset scheduling means that without intervention of humans, clear schedules and tasks are generated by IT systems. This requires algorithms that can access and process available data on the scheduling problem and provide schedules that contain tasks for the baggage tugs. As described before these scheduling systems can be either centralized or distributed systems. Methods to solve these scheduling problems are described in the literature review in section 3.1.

Advantages of automating the scheduling process are reduced workload for the human workforce and also in general, if implemented correctly, computer systems can create more effective schedules than human planners. Subprocesses at Schiphol Airport, at ground handlers and at airlines, already use automated scheduling and optimisation methodologies. Due to interaction of different stakeholders and subprocesses there might be possibilities for improvement for alignment of subprocess schedules, to optimise the whole turnaround process. This is one of the future goals of one of the ground handlers at Schiphol. For this ground handler to internally achieve this would already require a challenging combination of different subprocess optimisers. When looking at multiple stakeholders the alignment of such subprocess optimisers would be even more challenging due to conflicting interests.

In general, challenges of automated scheduling systems are:

- availability of quality real-time data of the to be scheduled processes and assets, and
- the tailoring of scheduling algorithms to work well for the specific planning problem.

These tailor-made solutions are hard to steer or adjust in case situational changes occur which are not included in the original design of the scheduling system. In such cases human intervention could be needed to keep schedules and operations up and running. In general humans provide a lot more flexibility in problem solving then specified tailor made scheduling algorithms. Therefore, to guarantee continuity of operations in case of unforeseen situations, requires having a back-up plan of approach with possible human intervention.

In this study a centralized and decentralized (distributed) scheduling approach has been developed for incoming and outgoing baggage transport vehicles. These are described in the following chapter.

6 Simulations

To obtain information about the performance of the various scheduling methods, a simulation environment is built to extract key performance indicators (KPIs). The input of the simulations is the data that was provided by KLM Ground Services and Schiphol, as described in chapter 4. In this chapter, the simulation methods that are developed in this WP3.1 of BrightSky project are described. All following functionalities have been implemented using Python 3 and can be run on a laptop computer (i7 1.90 GHz, 16 GB RAM).

6.1 Architecture

First, the overall simulation architecture is defined as shown in Figure 12. As shown in this figure, the different software components are implemented as stand-alone applications, which communicate with one another by sharing files. In Figure 12 these components are shown as white rectangles, and the files that are their input or output are shown as blue rounded rectangles. The functionalities of the three components are defined in the following sections.

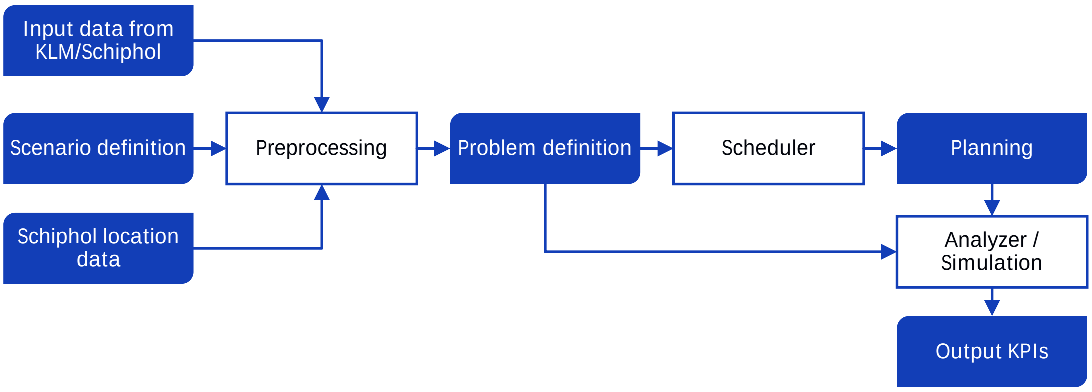


Figure 12: A schematic depiction of the overall structure of the simulation model

6.1.1 Preprocessing

The first component is the preprocessing application, which makes sure the data is provided in a way that is compatible with the other components.

The preprocessing application takes input data, which consists of three parts:

1. The input data from KLM Ground Services and Schiphol. This contains information about flights, departure and arrival times and locations, and amount of luggage to be handled.
2. Scenario definitions. This is the additional configuration that describes the conditions for the scheduler, i.e., the amount of tugs and trolleys to be included in the scenario, but also conditions on how many trolleys a tug can operate.
3. The location data. The input consists of shape files that includes relevant locations on Schiphol airport. This includes baggage drop-off points, baggage belts, VOPs, but also the topology of the roads.

The input data is processed by a set of pre-processing functions, which generates a standardized format problem definition file. By using a standard data format for the problem description, which are provided to stand-alone test applications, we can test the components in isolation. The full data format is not included in this document, but an example of a problem definition is provided in Appendix A.2. This standard problem definition includes a section on the fleets of tugs, trailers, flights, jobs and locations. All information needed to describe the problem should be included in this file.

The problem definition can be used by simulation software, or by any application that can analyse the problem and provide some output parameters. For instance, one could analyse the problem to get information about the number of flights, pieces of luggage or available resources. Another example would be a simple component to determine if certain conditions apply or whether all defined timings are consistent. The concepts in the data format have been designed to be self-explanatory, yet self-contained for a potential simulation or scheduling component.

6.1.2 Scheduler

The scheduler component takes a problem definition and applies any of the scheduling algorithms as described in chapter 7. The output of this process consists of a new file with the same format, but now with an additional section for the planning. This planning contains a set of assignments, which determines which tug must take which action at which moment.

The possible actions that correspond to assignments are:

1. couple trailers,
2. load batches,
3. drive from one location to another,
4. unload batches, or
5. decouple trailers.

This planning file can be provided to the simulation component to assess the validity and the quality of the schedules. By analysing the schedule in a separate component, we ensure a separation of concerns, and have a higher chance of finding any errors. The simulation component makes sure the schedule is valid, in the sense that the timings are possible according to the routes provided in the location descriptions. Also, it makes sure resources are used one by one, i.e. a trailer is always only used by a single tug at a time, and a batch is loaded only once. Finally, it will determine which batches are handled in time, and will generate a report consisting of a set of KPIs that will determine how good a planning is. Potentially, there could also be a feedback loop between the simulation and the scheduler, by sharing files, or even a real-time interface, for instance if in the simulation some unforeseen circumstances would be encountered. However, at the moment this is not implemented.

6.1.3 Simulation

The simulation component as described in the previous section is implemented using a discrete event simulation. The tool that was used to create this simulation is called DynAA [27], which is a simulation library developed by TNO.

DynAA has been used in multiple domains; initially it was created for modelling sensor networks, with a heavy focus on the communication side of the system. It has been extensively used as the core driver for the NetSquid[1] simulator, which models quantum communication networks.

Discrete event simulations operate by modelling any state changes in the world as discrete events, which occur at a specific time. In the period between two events, the world is assumed to be static, therefore the simulator only needs to keep track of changes at the events and can jump from one event to the next one. Any object in the simulated world can schedule an event at a time in the future and can register as a listener for future events. An event has an entity as a source is always of a specific type. An event type might be for example: BATCH_LOADED or START_DRIVING. A schematic representation of the objects in DynAA is shown in Figure 13.

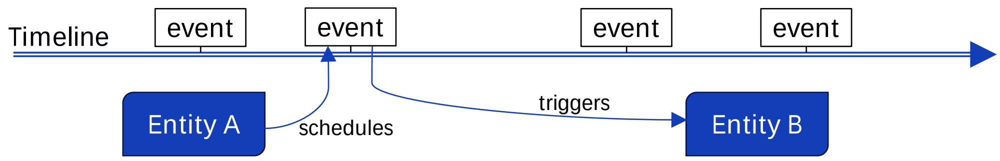


Figure 13: A schematic overview of the objects and relations in a Discrete Event Simulation.

An example of how this describes relations in the BrightSky simulation is that of a trailer loading a batch of luggage. Both the trailer and the batch are modelled as entities, so both can schedule and listen for events. At some point a function is called of the trailer object, with an argument the batch that needs to be loaded. This function checks some conditions and updates the state of the trailer and *claims* the batch. Claiming the batch means that the batch will schedule an event called *START_RESOURCE_CLAIM* at the time of being claimed, and at the same time will schedule an event after a specific time of type *FINISH_RESOURCE_CLAIM*. Upon the latter event, the trailer again triggers because the loading is now finished, and thus updates its state again, and immediately schedules an event *BATCH_LOADED*.

By describing relations between objects as events, and corresponding event handlers, allows for a very flexible way of modelling systems. Entity additions or removals can occur seamlessly, without the need of predefining all permutations of cause and effects.

6.2 Relation to external simulators

In this research a specific simulation environment and scheduling model have been developed to answer the research questions. The model has been developed with the awareness that BrightSky consortium partners and possible other external parties have knowledge and tools that have overlapping functionalities within existing tooling. Options for connecting existing models and simulation environments have been discussed, but differences in requirements, technological and legal possibilities lead to the conclusion to develop a tailer-made model to answer the posed research questions with possible future connections of modules in mind. This paragraph gives a brief overview of the relation of the developed simulation model to existing software at Vanderlande, T-Hive and KLM.

[1] https://netsquid.org/

6.2.1 Terminology

Before we go into details on existing tools, we first briefly describe the terminology. The software that generally provides insight in the number of shipments to be transported is called Warehouse Management System or Warehouse Execution System. As the name says the tool oversees the number of products in stock and the requests for new stock delivery or pick-up of stock. In our simulation model we call such a request a Job that defines a number of Batches (trailer loads) that need to be transported from or to an airplane.

A next step in the process flow is that these batches need to be assigned to empty trailers and a tug to transport them. A system that assigns tasks to vehicles is generally called a Fleet Management System (FMS) or Transport Management System (TMS). How these different functionalities are integrated in the developed simulation model and in current day applications at consortium partners is described below.

6.2.2 Tooling at Vanderlande, T-Hive and KLM

The simulation model developed for this research covers decision making processes that the KLM Ground Services executes in present day operations. To optimise transport of baggage to their flights KLM has developed an optimisation tool, called BagPro, that allocates tugs and tug drivers to transport baggage on the airside of Schiphol. In short, BagPro requires input information on to be transported baggage, available tugs and tug drivers and provides a schedule for the tugs and tug drivers as output. The internal methodologies have not been shared, but it is a central optimisation method, that covers functionalities of WMS and FMS systems. It also covers functionality of WMS systems as the tool can decide which shipments to transport and which shipments not to transport to optimise operational costs. Given the commercial sensitivity it is not possible to use or directly connect BagPro to the developed models in this research.

In parallel to BagPro, Vanderlande and T-Hive develop tooling to orchestrate fleets of (autonomous) baggage tugs or pods. They call this the FLEET system, which contains separate tooling for WMS and FMS functionalities. In theory these functionalities could be interchanged or connected to developed decision-making models in this research. However due to differences in level of detail and uncertainty in specifics, requirements and possibilities this has not been done up to now. The main difference between BagPro and the FLEET system is that BagPro includes business specifics of the airline and optimises with the objective of minimizing operational costs for the airline. For example it can prioritize over which bags to ensure make their plane based on costs that would be incurred to the airline, costs that can differ per flight or destination. Where the FLEET system (currently) is not aware of all airline operational costs that are impacted by the systems decisions.

An abstract overview of the different systems compared to the central and decentral control methodologies has been sketched in Figure 12.

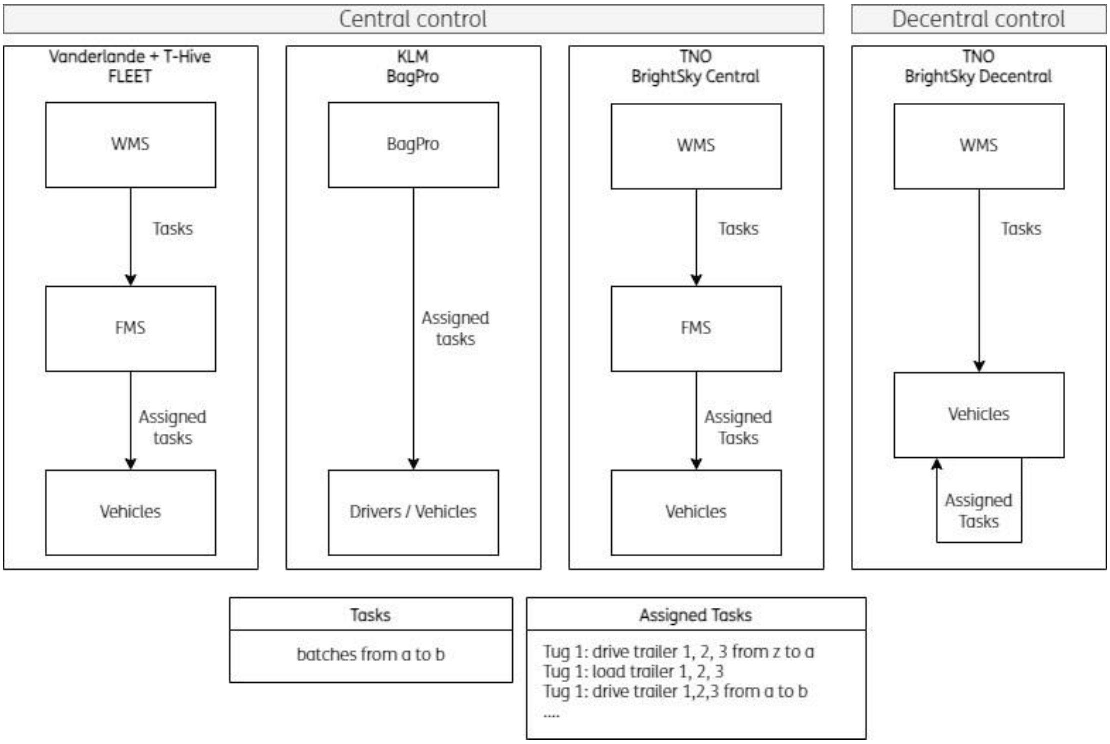


Figure 14: Abstract overview to compare basic functionality and setup of developed control methods and tooling at consortium partners.

7 Multi-fleet scheduling methods

The core part of the simulation model is the scheduling of tasks for the baggage fleets, or in other words the baggage fleet control methods. This chapter describes the problem formulation and how the problem is solved using central and decentral techniques.

7.1 General problem formulation

The main objective of the problem is to deliver as many batches on time as possible. The decision variables are the assignments of tugs and trailers to jobs. Each tug and each trailer can only be assigned to one job at any point in time. For every part of every journey of every job, there is a fixed time that needs to be spent. This depends on the number of bags that is being transported for the loading and unloading of bags.

The process of picking up a job and delivering the baggage is as defined as follows:

1. Tug picks trailers up:
   a. Drive to trailer 1
   b. Couple trailer 1
   c. Drive to trailer 2
   d. Couple trailer 2
   e. ...
2. Drive to job start
3. Load baggage on trailers
4. Drive to job destination
5. Uncouple trailers
6. Unload baggage

The execution time of each of these jobs is added to the schedule for the tug-trailer-combination that executes the job. The tug and corresponding trailers are only available after the baggage is unloaded. We assume that there is one optimal route per origin/destination-combination. Combined with the driving speed of tugs, we get the travel times for the origin/destination-combinations. These travel times are stored in a matrix. For origin/destination-points inside baggage basement, only the travel time from outside up until the closest entry point of the specific basement is included. Travel time inside the basement is excluded. The number of trailers that a tug can drive with is limited and has a differentiation for the baggage basements.

7.2 Model approaches

The goal in this research is to schedule tugs and trailers according to a given objective. The tugs and trailers will be scheduled jointly but are separate entities. The main priority is maximising the number of on-time baggage deliveries. For outgoing flights, this means that the baggage is delivered to the VOP before the final load deadline.

For incoming flights, it is a bit different. In this case, there is an internal deadline by which baggage should have reached the internal baggage systems of Schiphol Airport, counting from touch down. In this case, the goal is to deliver the baggage to the internal baggage systems before that deadline. This optimisation is done through scheduling tug-trailer combinations. This problem is only non-trivial in case the total number of tugs and trailers are restrictive. However, if the tugs and trailers are not restrictive, another goal could be added: minimise the total number of vehicles needed in the process.

The problem can be formulated as a Mixed Integer Linear Program (MILP) and categorized as a version of problems such as the job shop scheduling problem, vehicle scheduling problem and the order scheduling problem. These formulations are for the central approach. MILPs are NP-hard. The decentral approach to solving the scheduling problem in this research is formulated through a Distributed Constraint Optimisation Problem (DCOP). Optimisation methods for both MILPs as for DCOPs are discussed in the following sections.

7.2.1 Central approach

We consider two types of central optimisation: heuristic methods and solvers that are guaranteed to find the optimal solution, given enough computation time and the existence of an optimum. The heuristic methods are not guaranteed to find an optimal solution but are in general much more scalable and require less computation time. Both options will be discussed in the following sections.

7.2.1.1 Heuristic methods

We have implemented two greedy, heuristic methods. They are both originating from the same principle: earliest deadline first. We denote this with the “first in, first out”-solver (FiFo-solver). The algorithm consists of sorting the list of assignments to be planned based on their deadline, in which the assignment with the earliest deadline is the first on the list and the assignment with the latest deadline the last. This assignment is then coupled to the first available tug. This is followed by the search for available trailers to load the batches on. This is also done based on the first and closest available trailers.

The second method that we apply as heuristic is a “smarter” version of the FiFo-solver. We have named this the FiFoPrime-solver (FiFo’-solver). The improvements are detailed below.

The decision to use First-In First-Out methodology is driven by the problem specification with time critical deadlines and the need for a computationally fast benchmark method. By sorting the available tasks on their deadline and subsequently assigning them in that order to available tugs generally leads to quite effective schedules [6]. Improving this greedy method by adding more reasoning to the assignment was done to have an alternative central approach next to the MILP approach, which showed to high computation times.

Search for trailers based on available time and travel time to location
In most cases, multiple trailers must be gathered for an assignment. The FiFo-solver tries to assign trailers to batches based on the time that it costs to get to the available trailer-locations. This may, however, lead to trailers being gathered from multiple locations for a certain assignment. This can result in inefficiencies for total distance driven. The FiFoPrime-solver has replaced this behaviour by trying to gather the total number of necessary trailers at one location. In this improved algorithm, locations with at least as many available trailers as needed are listed. This is followed by sorting the list based on travel time to each of these locations of trailer-groups. The trailer-group that is closest by is then assigned to the job.

Only schedule batches that will arrive before deadline + slack
When a certain batch will be delivered late, resulting in a late flight, the FiFo-solver just keeps on trying to schedule subsequent batches of that flight in time. In time here indicates the original deadline and slack. This slack is the wiggle room that is available for delivering after the deadline. If the baggage cannot be delivered before the deadline + slack time, the baggage is added to the category ‘mishandled baggage’. Adding this baggage to separate ‘mishandled baggage’-category instead of still trying to deliver batches as soon as possible (even when these are past the deadline + slack) allows for a break in the lateness. Continuing to schedule batches, even when there are already late batches for a specific flight can lead to a propagation of lateness in the schedule. This is changed in the FiFoPrime-solver, in which a timeliness check is done before actually scheduling the batches. This timeliness check involves the comparison of earliest possible delivery time and deadline (including slack). Batches that can be delivered on time will be scheduled, while the remaining batches are marked as too late and ignored. This last step, ignoring late batches, follows from the principle that late batches will miss their flight and will be reassigned in the system. This is not the flow that we are scheduling and hence, we ignore batches that we can no longer schedule to arrive on time.

Only couple/decouple if necessary
The FiFo-solver couples exactly as many trailers as there are batches to be scheduled and decouples the trailers as soon as the corresponding batches are at their destination. For the next assignment, however, the tug will need to gather trailers again. If there are trailers that have been available for a longer period, the tug will gather these, instead of the trailers that have been uncoupled from it during the previous assignment. In the FiFoPrime-solver, trailers are only uncoupled if there are more coupled to the tug than necessary for the next assignment. This removes unnecessary steps.

7.2.1.2 Solvers

It is possible to formulate the given problem as a Mixed Integer Linear Program (MILP). There exists software (both commercial and open source) that can solve such an MILP. These solvers guarantee to find an optimal solution if they are given enough computation time. The computation time, however, scales exponentially with the size of the problem. For already a small subsection (5 tugs, 100 trailers, 20 batches) of the problem, the solver (Coin-OR branch and cut, available through PuLP in Python, [28]) that we chose took multiple hours to find a solution. This was not entirely unexpected, as the problem consists of many assignments for both tug to job and tug-trailer-combination to batch. Because of the computation time, we have not used the central MILP-solver for our final analyses. A not explored alternative would be to use commercial solvers, which might be better suited to solving large problems. These were however not available for this project.

7.2.2 Decentral approach

The decentralized approach is based on a Distributed Constraint Optimisation Problem (DCOP) formulation of the baggage fleet scheduling problem. In DCOPs, agents have to find assignments to variables, which in our problem correspond to combinations of trailers and batches. This means that every tug is represented by an agent, who is responsible for claiming a set of batches to deliver and finding the right set of trailers to do so. The agents will have to communicate with one another to share some information about assignments. This is done to find an assignment which leads to an overall assignment, that is beneficial for the whole group.

7.2.2.1 Cost function definition

A cost function needs to be defined, that determines how “good” or “bad” an assignment is. Ideally this cost function eventually represents the lateness of a batch of luggage, and at the same time holds some information about the priority of the batch, as well as the possibility to deliver the batch at all. Therefore, the following cost function for an assignment x is used in our scheduler:

$$C_x = \lambda_x - n_x + \frac{s_x}{n_x+1} \tag{1}$$

Where λ_x is a lateness penalty, indicating that an assignment in which any batch is too late will have a high cost. Then, n_x indicates the number of trailers that is used, so that longer trains are preferred over short trains, and finally s_x is the time required to deliver all batches in the assignment. This indicates that the agents will somewhat greedily choose assignments that they can quickly complete, instead of finding the assignments that are nearly going to be too late. However, because of the lateness penalty, the agents will still try to find assignments minimizing any late deliveries. This is quite a fine balance between assigning batches near their deadline versus choosing batches as quickly as possible.

In addition to the “preference” cost function, there is also the “overlapping assignment” cost function which states that two agents must never have assignments in which the same batch is assigned to two agents.

$$C_{xy} = \begin{cases} \phi, & \exists_b (b \in x) \cup (b \in y) \\ o, & otherwise \end{cases} \tag{2}$$

Here we indicate that the cost affected by assignments x and y is equal to some penalty ϕ if there is some batch in x and in y. We introduce a virtual assignment θ that corresponds to not doing anything to allow the agents to avoid this penalty. The cost of this “do nothing” assignment is defined as C_θ. Then, as long as the following holds:

$$\lambda < C_\theta < \phi \tag{3}$$

the agent will prefer delivering a batch too late over doing nothing, but the agent will also prefer delivering a batch late over attempting to deliver a batch for a second time.

7.2.2.2 Cooperative Constraint Approximation

To solve the DCOP-formulation of the baggage fleet scheduling problem we use the Cooperative Constraint Approximation (CoCoA) algorithm. Van Leeuwen and Pawełczak 2018 [10] introduce CoCoA (Cooperative Constraint Approximation), which uses a non-iterative, semi-greedy approach with a one-step look ahead. This algorithm can cope with asymmetric constraints and is faster than other (A)DCOP solvers.

CoCoA has three key ideas:

1. “A one-step look ahead to consider the effect of an assignment on the cost of neighbours. This is especially effective when a neighbour is constrained in its choices;
2. A unique-first approach, such that an agent will only assign a value if it is a unique local optimum for its variable. If it cannot find a unique solution, the decision will be delayed until more information is available;
3. A state machine to spread and keep track of the algorithm’s activity, prevent dead-locks or endless loops.” [10]

Effectively, this means that any agent will inquire with its neighbours what the impact of his decision will be on *them.* Then, he will assign the value that minimizes the cost for himself, and his neighbours. A neighbour in this context, means any agent that he shares a constraint with, and since in our problem, we cannot assign any duplicate batches, every other agent is a neighbour.

7.2.2.3 Method adaptation for domain specific application

Similarly to the approach proposed in [6], a pruning step is used to reduce the domain size of the variables, and at the same time reduce the number of constraints that an agent needs to consider. Both reductions have a large impact on the search space, which means that the algorithm will converge faster. At the same time, the reduction of the number of constraints will make the problem graph less dense, which we know from the literature will improve the found solution of CoCoA [29]. This pruning step is implemented by making “trains” of batches and trailers, selecting only the first batches that can be delivered at the time of running the algorithm, and selecting the top n combinations as candidates. Then, when the DCOP algorithm runs, only these top n trains are considered when solving the problem.

Once a solution has been found for the DCOP problem, every agent has assigned a value to its variable, meaning every tug has selected a set of trailers and batches to deliver. The corresponding batches are removed from the queue, and the process repeats until there are no more batches to be delivered.

7.2.2.4 Hyperparameter optimisation

The above-described cost function and methodology adaptations are a result of iterative improvements of the multi agent decision system. In between the implementation iterations, results of the CoCoA-algorithm have been studied to identify possible improvements in single agent decision making and the agent interaction protocol.

The adjustments were in either of the following categories:

1. Elements to include in cost function (see equation 1);
2. Value of weight parameters (see equation 3).

Equation 1 shows the final cost function, taking into account: (1) lateness of batch deliveries λ_x, (2) length of trailer train n_x and (3) total time required s_x. Elements that have been tested in development iterations, but did not make the final cost function are for example time remaining until batch delivery deadline, difference of batch deadlines within a train and penalising additional trailer (de)coupling.

The calibration of weight parameters has been done by evaluating scheduling results: if the resulting scheduled train lengths were rather short with mostly one or two trailers, the weight value for train length has been increased, or, if many batches were assigned with late delivery, then the penalty for lateness has been increased and the function for estimating has been improved.

7.2.2.5 Implementation

A Python implementation of this algorithm was developed and in parallel a decentral BrightSky scheduler was made that creates a DCOP-formulation of the general problem formulation to feed into the algorithm. Scenarios with both central heuristics and decentral CoCoA method have been simulated and the results are presented in next chapter.

8 Results

Experiments have been run with the developed simulation model and the different scheduling methods using the baggage transport data as described in the section Airside And baggage handling data. We first compare the results of the different scheduling approaches (central and decentral). Following this, the different fleet scenarios of single- and multi-fleet are compared.

Unfortunately, no data on real world execution of baggage tug fleets was available. This would have been preferable as benchmark values. Alternatively, the basic and greedy assignment method First-In First-Out (FiFo) (see 7.2.1.1) is used as a reference method in the results.

8.1 Scheduling problem size and context

The schedules have been calculated for two problem sets: a small and a large problem set. In the small problem set we have used 20 tugs which have to deliver bags for 100 jobs. In the large problem set we have used 60 tugs which have to deliver bags for 500 jobs. A full day of baggage transport consists of approximately 4000 jobs and at most 60 tugs for the KLM ground handler. This implies that the large problem set represents approximately 1/8-th of a day, similar to scheduling 4 hours ahead. The current day operations system of KLM, BagPro, schedules 15 minutes in advance. Scheduling longer periods in advance is not deemed effective due to the high level of uncertainty and dynamics in the airport processes and scalability issues in the scheduling problem. We have experienced a similar scalability problem with the MILP approach, therefore heuristic methods have been developed to schedule longer periods of time. Dynamics, such as disturbances or changes of airplane timing, are not (yet) included in the simulation model and therefore not of impact in this study. This study schedules a longer period ahead to create more extensive results to provide more input to the comparison of the scheduling methods.

8.2 Key performance indicators

Six performance indicators have been used to evaluate the performance of the scheduling methodologies on the baggage transport problem. The main objective of the baggage transport fleets is to ensure that as many bags as possible reach their flight in time. A more efficient scheduling method will result in the same fleet being able to deliver more bags in time. Hence, the number of on-time delivered the bags is the most important indicator. The total driving time of the tugs is reported to indicate the resource efficiency of the scheduling methods. To this purpose, also the number of used trailers is reported. This will especially be interesting for comparing the FiFo- and FiFoPrime-schedulers, as the FiFo-solver does not reuse trailers, while the FiFoPrime-solver allows the same trailers to remain attached if they can be reused for the next job. Lastly, the average length of the trailer trains are reported. This is an indication of the amount of batches that can be combined into one ride and tells something about the efficiency of the rides.

Note that there are three more indicators than described in the plan of approach in paragraph 2.4. The increase in number of indicators is due to the fact that the indicator in total driving time of tugs is separated into driving with loaded and with empty, which added 2 new indicators. The third additional indicator that has been added is the average train length. This indicator gives insight in the impact of the different scheduling methods on the train formations and therewith making more effective use of available tug capacity.

The main performance indicators that are presented in this chapter are therefore:

1. number of on-time delivered bags,
2. total driving time of the tugs,
3. total driving time with empty trailers,
4. total driving time with loaded trailers,
5. number of used trailers,
6. average length of trailer train.

The decentral method, the DCOP-solver, has a random component. To be able to report on the results of the DCOP-schedules we have run each experiment 100 times for the DCOP-solver. This is shown in the report by showing the average value for the DCOP-schedules in the coming sections. The spread of the results of the 100 DCOP-schedules is shown in the figures, in which the standard deviation is presented through a whisker plot with one standard deviation below and above the mean as end points.

8.3 Single fleet comparison of central and decentral solution methods

The results of comparing the central FiFoPrime- and decentral CoCoA-methods in the single fleet scenario are presented in this section. Before we go into the fleet performance details we first compare the methods on computation time requirements.

8.3.1 Computation time requirements

FiFoPrime currently has a much lower computation time compared to the decentral method. For example with the large problem instance FiFoPrime requires 15 seconds for a single run, where the DCOP method currently needs 30 minutes. This requires the sidenote that for the DCOP the Python implementation of the multi-agent negotiations spends 90% of the computation time in waiting for Python threads in the scheduling process, due to the fact that the DCOP implementation is executed on a single processor thread. As mentioned before the exact central method did not even get to a solution within 30 minutes for the small problem instance, therefore we are now comparing DCOP results to a more basic, but computation time efficient, central approach. The results of the two problem sets will be compared in more detail in the following two subsections.

8.3.2 Small problem set

We have used 20 tugs that have to deliver bags for 100 jobs in case of the small problem set. The results of this problem set are shown in Table 4. The DCOP-solver performs the best for most of the KPIs. The DCOP-schedule, on average, timely delivers the most bags, which is also shown in Figure 15. Even the DCOP-schedules that perform worse than the average DCOP-schedule perform better than the central scheduling methods.

For the total driving time, driving time with empty trailers and the driving time with loaded trailers, the DCOP-schedule also performs the best. The margin for the DCOP-schedule outcomes as shown in Figure 15 is small enough for the worst DCOP-schedules to outperform the central methods. Note that total driving time for DCOP is relatively not that much lower than the FiFo-methods. This is due the fact that DCOP also creates assignments for all too late bag deliveries, implying driving time for these too late deliveries is also included, where the FiFo-methods do not include these assignments. Note that DCOP method cannot incorporate hard constraints, because constraints are added by cost penalties. So there is a cost penalty for late delivery, to minimize occurrence, but it does not prevent these deliveries from being added to the schedule by the DCOP method.

The FiFoPrime-method performs the best for the total number of used trailers. This could be a result of the coupling-decoupling-method that has been implemented for this method, where coupling/decoupling is only done in case the next job requires more/less trailers.

Lastly, we see that the average train length is the longest for the DCOP-solver. This means that the tugs, on average, drive with more trailers attached. This is a measure of efficiency, as shorter trailer trains lead to more separate drives that need to be made in order to deliver the same number of bags.

Table 4: Results of simulation with 20 tugs, doing 100 jobs.

	FiFo	FiFoPrime		DCOP	
	Absolute	**Absolute**	**% of FiFo**	**Absolute**	**% of FiFo**
Bags delivered on time [#]	9216	9686	105.10	10696	116.06
Total driving time [h]	18.04	16.71	92.63	14.09	78.10
Total driving time with empty trailers [h]	9.39	8.87	94.46	7.83	83.39
Total driving time with loaded trailers [h]	7.65	7.84	102.48	6.26	81.83
Average train length [#]	2.19	2.19	100.00	3.42	156.16

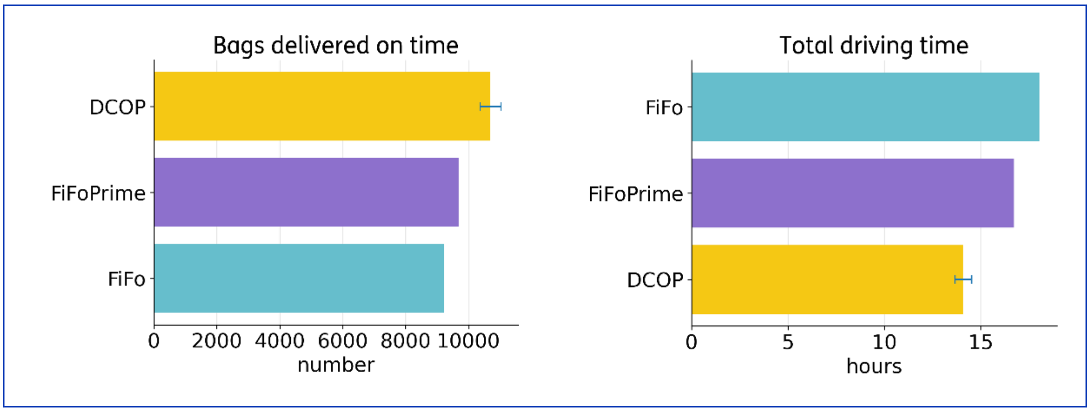


Figure 15: Comparison of the different schedulers on the "small" problem.

8.3.3 Large problem set

In the large problem set we have used 60 tugs which have to deliver batches for 500 jobs. The results of the scheduling methods are reported in Table 5 and Figure 16. In this experiment, the DCOP-solver performs best for all but one KPI - the total number of used trailers. The percentual differences compared to the baseline FiFo-results for the other scheduling methods are slightly smaller for this problem instance. Moreover, the variation in the DCOP-schedules is reduced, see also the error bar of the DCOP-outcomes in Figure 16.

The results are comparable to the results of the small problem instance, with the same ranking in methods for most KPIs. One KPI for which there is a difference in ranking of the scheduling methods is the total driving time with empty trailers. FiFoPrime now performs worse than FiFo. This can be attributed to an increased impact of the decoupling/coupling-strategy for the FiFoPrime-scheduler. Tugs keep trailers attached if they are needed for the next job, leading to an increased driving time with empty trailers compared to the FiFo-method. This does not lead to a higher total driving time for the tugs in the FiFoPrime-schedules than in the FiFo-schedules. This is an indication of the trade-off between driving with empty trailers and driving without trailers (not reported) and more efficient scheduling of the FiFoPrime method.

Table 5: Result of the simulation with 60 tugs and 500 jobs.

	FiFo	FiFoPrime		DCOP	
	Absolute	Absolute	% of FiFo	Absolute	% of FiFo
Bags delivered on time [#]	49269	55737	113.1 %	58758	119.3 %
Total driving time [h]	162.94	130.56	80.1 %	101.93	62.6 %
Total driving time with empty trailers [h]	54.12	67.80	125.3 %	50.71	93.7 %
Total driving time with loaded trailers [h]	59.55	62.76	105.4 %	51.20	86.0 %
Average train length [#]	2.04	2.13	104.4 %	2.87	140.7 %

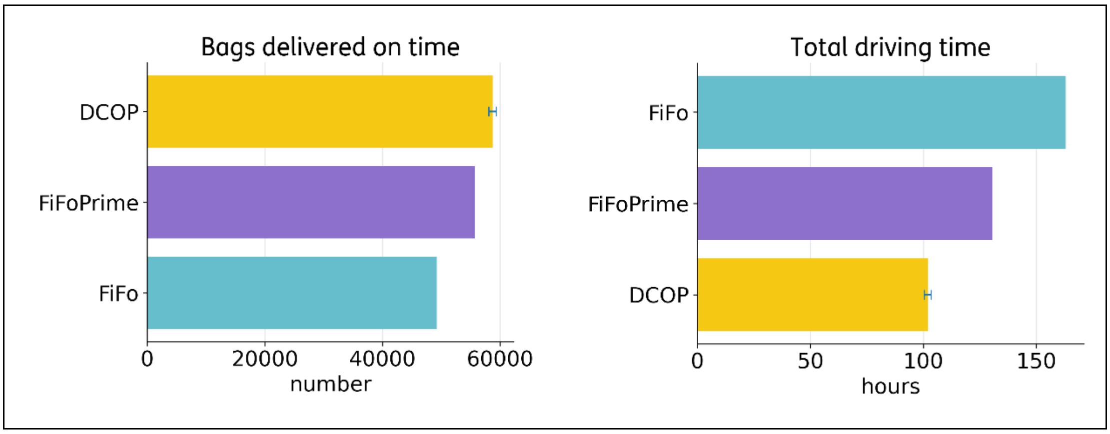


Figure 16: Comparison of the different schedulers for the large problem set.

8.3.4 Trailer train assignment

The difference in performance between the control methods is amongst others caused by the difference in trailer train formation as described in 7.2.1.1. This also results in a different use of trailer assets as different trailer assets are to be coupled and decoupled in the resulting assignments. The simulation model counts the different trailer assets that are being used and these are reported in Table 6. Dividing the number of used trailers by the delivered bags provides an indication on how efficient the tugs use the available trailer fleet.

Note that analysing the trailer usage was not a primary goal of this study, but due to the difference in control methods there is also in difference in trailer usage. For context, the total number of available trailers in the simulation is 2365, which is an estimation of the size of the trailer fleet owned by KLM. In the simulated scenarios for all three assignment methods this number of trailers is more than sufficient for the resulting schedules.

Table 6: Used trailers and trailer use efficiency for the different control methods for the large problem instance (60 tugs, 500 jobs).

	FiFo	FiFoPrime		DCOP	
	Absolute	**Absolute**	**% of FiFo**	**Absolute**	**% of FiFo**
Used trailers [#]	1739	693	39.9 %	754	43.4 %
Bags delivered on time [#]	49269	55737	113.1 %	58758	119.3 %
Trailers / Bags on time [# / #]	0.035	0.012	35.2%	0.013	36.4%

8.4 Single-fleet compared to multi-fleet

8.4.1 Multi-fleet simulation results

A medium-sized problem set has been created for the comparison between the single-fleet and multi-fleet results. This problem set consists of 20 tugs which have to deliver bags for 500 jobs. We use the smaller instance number of tugs due to computation time requirements, because of the need to do multiple simulation runs to quantify the randomness effect in the decentral approach. For the number of jobs, the larger instance set is used to intentionally provide a large set of jobs for which 20 tugs is a too small fleet to deliver all bags timely. This provides the context to analyse which control method in which is scenario is most effective in assignment of the tugs and trailers and provides a schedule which delivers the most bags timely. Thus we don't expect the tugs to deliver all jobs, but we want to quantify which method performances the best.

The same performance indicators as in the previous sections have been compared. In this section, we define the single fleet as the scenario in which the separate fleets (inbound and outbound) are separately scheduled for the separate problems. The fleet for which the jobs are scheduled consists of single fleets, hence the name.

The multi-fleet scenario refers to the case in which the two inbound and outbound fleets are combined into one joint fleet. The fleet for which both the inbound and outbound jobs are then scheduled consists of both fleets, hence the name 'multi fleet'. The two different scenarios have been created to compare the impact of fleet collaboration and asset sharing between the inbound and outbound fleets. The results are presented in Table 7 and Figure 17.

When we compare the single- and multi-fleet results for the benchmark FiFo-method, we already see improvements on the number of bags delivered on time. The other KPIs, however, have increased in a seemingly less efficient direction. This can be explained by the fact that significantly more jobs have been successfully executed, which means that the number of resources used and the number of kilometres driven increase.

Like the large problem simulation results of the central and decentral comparison, these results show that the central FiFoPrime-method outperforms the central baseline method of the FiFo-solver. This is apparent in the number of bags delivered on time, the total driving time, the number of used trailers and the average train length. The total driving time sub-KPIs of total driving time with empty/loaded trailers are higher for the FiFoPrime-schedules than for the FiFo-schedules. Combined with the drastic decrease in number of trailers used for the FiFoPrime-method, this can be explained by the decoupling/coupling-strategy for the FiFoPrime-method. The KPI for average train length is similar for the FiFo- and FiFoPrime-schedules in the single- and multi-fleet scenarios.

The FiFoPrime-method, in turn, is outperformed by the decentral DCOP-solver, with more jobs scheduled in both the single- and multi-fleet scenarios. The single fleet DCOP-schedules perform better than both the single- and multi-fleet schedules of the FiFo-solver. When comparing the DCOP-scheduler with the FiFoPrime-scheduler the on-time delivery of bags is increased with 48%. This shows that the DCOP-solver performs significantly better than the FiFo-solver. Zooming in on the DCOP-solver comparison of single fleet with multi fleet performance shows that the on-time performance of bag delivery increases by 55% in case of multi fleet collaboration compared to single fleet operations.

Table 7: Results of the single- and multi-fleet scheduling.

	FiFo		FiFoPrime		DCOP	
	Single	Multi	Single	Multi	Single	Multi
Bags delivered on time [#]	14248	22414	16920	27380	25042	38838
Total driving time [h]	64.97	82.47	62.76	82.29	73.51	79.29
Total driving time with empty trailers [h]	32.86	37.91	32.37	40.84	35.37	36.82
Total driving time with loaded trailers [h]	27.98	34.50	30.39	41.46	38.13	42.46
Average train length [#]	1.49	1.57	1.52	1.60	3.75	3.39

For all scheduling methods, the multi-fleet scenario allows for an increase in total driving time compared to the single-fleet scenario. This increase is the strongest for the central scheduling methods. The increase from single-fleet to multi-fleet scheduling in the other driving time KPIs is similar for each of the scheduling methods. The increase of total driving time with empty trailers is the strongest for the FiFoPrime-scheduler. This can be explained by the fact that the FiFoPrime-method only couples or decouples trailers in case the number of trailers needed for the next job does not match the number of trailers needed for the current job. This means that the tug often drives towards the next job with a set of empty trailers. This behaviour is also part of the DCOP-scheduler, but is not part of the FiFo-scheduler.

Lastly, we report on the average train length. This is, as in the other problem sets, the highest for the DCOP-schedules. Interestingly, the average train length slightly increases for the central scheduling methods, while the DCOP-schedules have shorter trains on average for the multi-fleet problem set compared to the single-fleet problem set. This can be an indication of the many jobs that are delivered in the DCOP-multi-fleet schedule, which means that there might be more jobs at approximately the same time that need to be executed. This could indicate that the scheduling preference for the DCOP-scheduler lies with the execution of jobs and less with the average train length. In other words, the DCOP-scheduler prefers the execution of as many jobs as possible over maximising the average train length. This is also in line with the fact that the DCOP-scheduler explicitly steers on the execution of jobs and does not explicitly focus on maximising the train lengths.

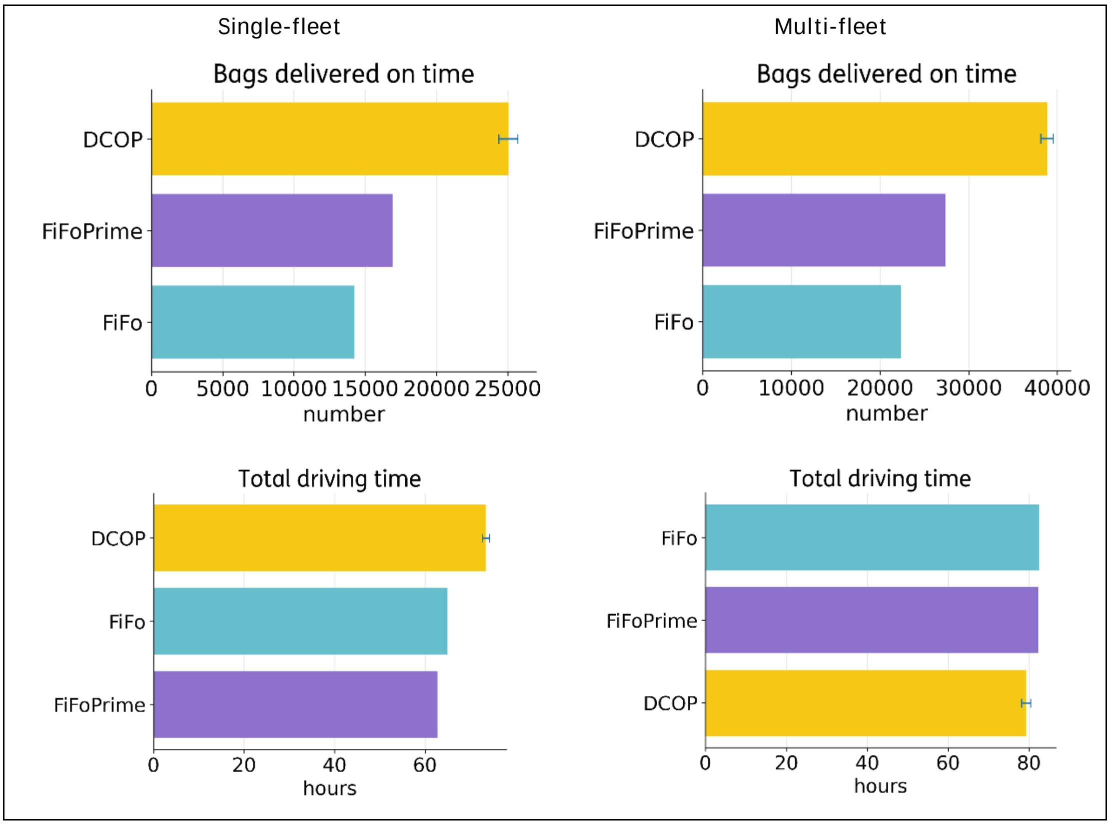


Figure 17: Comparing single-fleet and multi-fleet scheduling.

8.4.2 Multi-fleet performance improvement

The previous section provides the absolute numbers resulting from the simulation. From these results we have calculated the fleet efficiency by looking at the average number of bags delivered per tug per hour to give a more direct answer to the question on what the gain of asset-sharing in the multi-fleet is. The bags per tug per hour are presented per control method and for single- and multi-fleet scenarios in the table below. Comparison of the single- with multi-fleet scenario's shows a tug efficiency increase of 23 to 44 %. Implying that multi fleet asset sharing can reduce the requirements for tug and tug drivers significantly.

Table 8: Increase in tug effectivity from single- to multi-fleet in number of bags delivered per hour for the three control methods. .

	FiFo		FiFoPrime		DCOP	
	Single	Multi	Single	Multi	Single	Multi
Bags delivered on time [#]	14248	22414	16920	27380	25042	38838
Total driving time [h]	64.97	82.47	62.76	82.29	73.51	79.29
Bags delivered on-time per hour	209.6	271.8	269.6	332.7	340.6	489.8
% increase bags per hour from single to multi-fleet		*29,7 %*		*23,4 %*		*43,8 %*

9 Conclusions and discussion

9.1 Conclusions

Combining the knowledge from literature and insights in airside processes simulation results leads to the following conclusions and answers on the posed research questions. First we answer the main question and subsequently the sub questions.

What type of control (central/decentral/hybrid) is most suitable for managing a system of multiple fleets of baggage tugs, taking into account specific features of unmanned vehicles?
The developed decentral scheduling (DCOP) method outperforms the greedy heuristics that represents central decision making. For the two central decision-making methods that have successfully been implemented, the FiFoPrime-method performs best. The central MILP-approach, implemented in solver software PULP in Python has not made it to the final list of evaluated solvers, due to the extremely high computation times. The MILP-solver should be able to find an optimum, if it is given enough computation time. This means that we could not compare the results of the implemented methods to optimal schedules. We have, however, implemented two types of central scheduling methods in order to create a benchmark schedule and to show the improvement potential. The developed methods can control fleets of manned and unmanned vehicles, more on autonomous vehicles is elaborated below at the question on the differences posed by unmanned vehicles.

The performance of the decentral control methodology is dependent on the setup and calibration of the multi-criteria objective functions of the agents, as discussed in section 7.2.2. Setting the right objective is always critical for decision making algorithms, but in the decentral setup this is even more challenging compared to the central situation due to the complex interaction between agents within multi agent systems.

To conclude this answer, our report indicates that the decentral scheduling method performs the best when taking both computation times and KPI-results into account. With a 6 to 10% increase in on-time delivered bags when comparing DCOP to FiFoPrime in the initial single fleet results and 48% increase in on-time delivered bags in multi-fleet results. The increase was higher in the multi-fleet scenario due to larger, more challenging, problem with more baggage transport jobs.

What are the differences between central and decentral planning of a (baggage) fleet?
Central planning of a baggage fleet requires less communication between vehicles. The communication that is needed for decentral applications provides a computational challenge in the current Python setup, which is a single server simulation environment. However, this can be resolved by distributing the problem over separate entities of resources. The decentral communication and optimisation method will generally outperform the computation time of exact central methods, especially when applied to larger problem sizes.

In the decentral planning setup, information will be shared with all vehicles in the problem. This does not necessarily mean that all information is shared with all vehicles. Information on the preference of each vehicle will only be shared with other vehicles if they are in each others neighbourhood, i.e. if they are interested in the same jobs. General information about jobs will be shared with all vehicles.

In the central planning setup, the central, “control tower”, entity will have full information. This means that the central entity should be able to find an optimal solution if it is given enough runtime. Because of the incomplete information in the decentral setup, this same guarantee on the optimality cannot be given. It is, however, not the case that the decentral setup cannot find an optimal solution at all. This is still possible, but there are no guarantees.

What are the differences in organising and controlling a manned and unmanned fleet?
A manned and unmanned fleet differ in vehicle properties, such as driving speed and ability to handle dynamic and potentially unexpected situations. This could become apparent through restrictions on the maximum driving speed for unmanned vehicles, because of their different ability to handle unexpected situations. These properties are inputs for the performance levels of the vehicles and need to be known and incorporated in the scheduling problem.

However, a vehicle is a vehicle, from the plain perspective of the scheduling algorithms. Characteristics such as driving speed are inputs for the algorithms, but the characteristics that define the difference in manned and unmanned vehicles do not influence the way vehicles are defined for the algorithms. These properties are known in advance and taken as given in order to define the problem constraints. Therefore the developed control methods can also schedule fleets with unmanned vehicles, however due to a lack of information on unmanned vehicles characteristics this has not be done in this study.

What are the effects of sharing resources (vehicles and drivers) amongst fleets?
The effects of sharing resources has been discussed in section 8.3.4. In this study, we have investigated two separate fleets (incoming and outgoing) of KLM Ground Services. We have defined two different fleet setups to investigate this impact: (1) a single-fleet scenario in which fleets are separate entities and (2) a multi-fleet scenario in which all resources are shared and the resource pool for the entire problem consists of all resources combined. Inbound and outbound jobs and resources are separated for the single-fleet problem instances: inbound jobs can only be handled by inbound tugs and similarly for outbound jobs and tugs. The inbound and outbound jobs are combined into one pool of jobs for the multi-fleet scenario, in which every tug (both inbound and outbound) can execute any available job, irrespective of its direction.

The analysis has shown that 55% more orders can be delivered in time in the case of the multi-fleet scenario. In absolute terms, the DCOP-solver has performed best and shows the most improvement in bags that are delivered in time from single- to multi-fleet. The other two solvers have also improved in the number of bags that are delivered in time. Logically, the total driving time and number of used trailers have also increased for each multi-fleet scenario compared to their single-fleet equivalent. This is to be expected, with the strong increase in jobs that can be executed in time.

The concept of merging the fleets is not novel to KLM Ground Services. Additionally, it is relevant to note that there are factors complicating the merge of inbound and outbound fleets into one combined pool of tugs. The main reason for this is overall scheduling complexity of dynamic airport processes and an increase in complexity when ingoing and outgoing are combined. Handlers should weigh the benefits of the number of bags delivered in time against the costs of making each driver employable in both fleets.

9.2 Discussion

The conclusions are based on the knowledge attained from and experiments with the developed simulation model. The choices that have been made in this research and limitations to available data and insights have room for improvement and further research. In this section we will discuss possible steps for future research.

A first improvement would be to compare the performance of our models with real world schedules. Due to the sensitivity of information in the actual schedules, we have not been able to compare with the scheduling results of BagPro. As this is a central model, this would give more insight on the performance of the decentral scheduling method. Moreover, it would allow us to make realistic checks on our models. Related to this, an addition of a central scheduling method would be interesting. As the mixed integer linear programming method is too computationally heavy, future research could investigate the performance of commercial MILP-solvers or central metaheuristics such as Simulated Annealing, Genetic Algorithms and Tabu Search.

The analysis in this research could also be extended by including dynamics in the simulation, such as disturbances and rescheduling during the model runs. This will create a simulation environment that better resembles the real-world situation, in which rescheduling happens often. This would also allow for more insights in the difference in performance for the central and decentral methods. Decentral methods have as benefit that they are quick, while central methods usually attain better results due to their higher level of information availability. While computation time may not be an issue for day ahead scheduling, it becomes an increasingly important factor when rescheduling during the day. Rescheduling during the day needs to be done quickly in order to maintain the same level of performance as before the rescheduling and prevent the mismanagement of jobs. The preference in the balance between performance and computation time may shift in case of dynamic scheduling with disturbances, and hence, a shift in the "best" computation method may occur. Moreover, this may also lead to the investigation of hybrid control methods. These can be implemented on multiple levels. Firstly, a combination of central and decentral scheduling methods could be used when scheduling both a day in advance and during the day. The day ahead schedule could be made by a central method, ensuring a high level of performance, while rescheduling during the day may be done with a decentral method, ensuring a relatively short computation time. In addition, a hybrid form of control can also be implemented as way to allocate certain decisions to specific control methods. Include a hybrid form of control where straightforward decisions are made local on a decentral level and decisions which require negotiation are solved on a higher, more central, level. For example all decisions that do not lead to decision conflicts can be directly assigned, but in case of decision conflict the decision-making choice could be provided to a central planner to make the call.

The decentral control method could be improved by a more extensive analysis of the impact of changes in the utility functions on the performance. This could also be done in cooperation with handlers, in order to make the utility function as realistic as possible.

References

[1] G. Zhang, L. Gao en L. Shi, „An effective genetic algorithm for the flexible job-shop scheduling problem," *Expert Systems with Applications,* vol. 38, nr. 4, pp. 3563-3573, 2011.

[2] M. R. Garey, D. S. Johnson en R. Sethi, „The Complexity of Flowshop and Jobshop Scheduling," *Mathematics of Operations Research,* vol. 1, nr. 2, pp. 117-129, 1976.

[3] F. Baita, R. Pesenti, W. Ukovich en D. Favaretto, „A comparison of different solution approaches to the vehicle scheduling problem in a practical case," *Computers & Operations Research,* vol. 27, nr. 13, pp. 1249-1269, 2000.

[4] W. Guo, P. Xu, Z. Zhao, L. Wang, L. Zhu en Q. Wu, „Scheduling for airport baggage transport vehicles based on diversity enhancement genetic algorithm," *Natural Computing,* vol. 19, nr. 4, pp. 663-672, 2020.

[5] J.-Q. Li, P. B. Michandani en D. Borenstein, „The vehicle rescheduling problem: Model and algorithms," *Networks,* vol. 50, nr. 3, pp. 211-229, 2007.

[6] G. L. Pingen, C. R. Van Ommeren, C. J. Van Leeuwen, R. W. Fransen, T. Elfrink, Y. C. De Vries, J. Karunakaran, E. Demirovic en N. Yorke-Smith, „Talking Trucks: decentralized collaborative multi-agent order scheduling for self-organizing logistics," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2022.

[7] G. Lodewijks, Y. Cao, N. Zhao en H. Zhang, „Reducing CO_2 Emissions of an Airport Baggage Handling Transport System Using a Particle Swarm Optimization Algorithm," *IEEE Access,* vol. 9, pp. 121894-121905, 2021.

[8] Y. Du, Q. Zhang en Q. Chen, „ACO-IH: An improved ant colony optimization algorithm for Airport Ground Service Scheduling," in *2008 IEEE International Conference on Industrial Technology*, 2008.

[9] K. Sörensen, „Metaheuristics - the metaphor exposed," *International Transactions in Operationsal Research,* vol. 22, nr. 1, pp. 3-18, 2015.

[10] C. J. Van Leeuwen en P. Pawelczak, „CoCoA: A non-iterative approach to a local search (A)DCOP solver.," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 31, nr. 1, 2017.

[11] P. J. Modi, *A multi-objective optimization approach to the ground handling scheduling problem,* 2014.

[12] K. Hirayama en M. Yokoo, „Distributed partial constraint satisfaction problem," in *Principles and Practice of Constraint Porgramming-CP97*, G. Smolka, Red., Berlin, Heidelberg, Springer Berlin Heidelberg, 1997, pp. 226-236.

[13] T. Grinshpoun, T. Tassa, V. Levit en R. Zivan, „Privacy preserving region optimal algorithms for symmetric and asymmetric DCOPs," *Artificial Intelligence,* 2019.

[14] F. Fioretto, E. Pontelli en W. Yeoh, „Distributed Constraint Optimization Problems and Applications: A Survey," *Journal of Artificial Intelligence Research,* vol. 61, pp. 623-698, 2018.

[15] D. Pisinger en S. Í. Hjøllum Rude, „Advanced algorithms for improved baggage handling," *Journal of Airport Management,* vol. 14, nr. 3, pp. 280-290, 2020.

[16] M. Frey, C. Artigues, R. Kolisch en P. Lopez, „Scheduling and planning the outbound baggage process at international airports," in *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, 2010.

[17] A. Abdelghany, K. Abdelghany en R. Narasimhan, „Scheduling baggage-handling facilities in congested airports," *Journal of Air Transport Management,* vol. 12, nr. 2, pp. 76-81, 2006.

[18] C. Malandri, M. Briccoli, L. Mantecchini en F. Paganelli, „A Discrete Event Simulation Model for Inbound Baggage Handling," in *International Conference on Air Transport - INAIR 2018*, 2018.

[19] T. Barth en D. Pisinger, „Scheduling of outbound luggage handling at airports," in *International Conference on Operations Research*, Zurich, Switzerland, 2011.

[20] M. Frey, F. Kiermaier en R. Kolisch, „Optimizing Inbound Baggage Handling at Airports," *Transportation Science,* vol. 51, nr. 4, pp. 1210-1225, 2017.

[21] S. M. A. P. Van Loenhout, *Optimizing Outbound Baggage Handling at KLM,* Delft University of Technology, 2019.

[22] S. Padrón Astorga, *A multi-objective optimization approach to the ground handling scheduling problem,* Universitat Autònoma de Barcelona, 2014.

[23] T. C. Barth, J. Timler Holm en J. Lindorff Larsen, „A model for transfer baggage handling at airports," DTU Management Engineering, 2013.

[24] T. Clausen, *Airport Ground Staff Scheduling,* Technical University of Denmark, 2010.

[25] J. Rood, *Computations of the Demand for Ground Resources at KLM,* University of Twente, 2018.

[26] J. P. Cavada en C. E. R. P. A. Cortés, „A workforce planning and allocation model for the outbound baggage loading area at Santiago International Airport," *Information Systems and Operational Research,* vol. 58, nr. 3, pp. 537-559, 2020.

[27] J. O. de Filho, Z. Papp, R. Djapic en J. Oostveen, „Model-based design of self-adapting networked signal processing systems," in *Proceedings of the International Conference on Self-Adaptive and Self-Organizing Systems*, Philadelphia, 2013.

[28] J. Forrest, T. Ralphs, S. Vigerske, H. G. Santos, L. Hafer, B. Kristjansson, E. Straver, M. Lubin, S. Brito, M. Saltzmann en B. Pitrus, *coin-or/Cbc 2.10.11,* 2023.

[29] C. J. van Leeuwen en P. Pawelczak, „Hybrid DCOP Solvers: Boosting Performance of Local Search Algorithms," in *International Workshop on Optimization in Multiagent Systems (OptMAS)*, Stockholm, 2018.

[30] D. D. Sprenger, R. W. Fransen, R. G. Hoogendoorn, J. C. Van Meijeren en B. S. De Jong, „WP4: Impact of multi-fleet truck scheduling. KIP Self Organising Logistics," TNO, 2023.

[31] R. Fransen, D. D. Sprenger, W.-J. Lee en J. v. Meijeren, „Performance of Self-Organizing Logistics: a Practical," in *International Physical Internet Congres*, Athens, 2023.

[32] R. Fransen, L. Sprenger en H. Onverwagt, „Multi-fleet Talking Trucks user requirements and algorithm options. KIP Self Organising Logistics," TNO, The Hague, 2023.

Signature

TNO ⟩ Mobility & Built Environment ⟩ The Hague, 9 October 2025

Jan Hoegee
Research manager

Ruben Fransen
Author

Appendix A
Appendix

A.1 Subprocesses in air plane turnaround

A.1.1 Ground handler

The ground handler has a broader portfolio of activities around the airplane turnaround than only the baggage handler. In general, a ground handler is responsible that the following processes are executed during airplane turnaround:

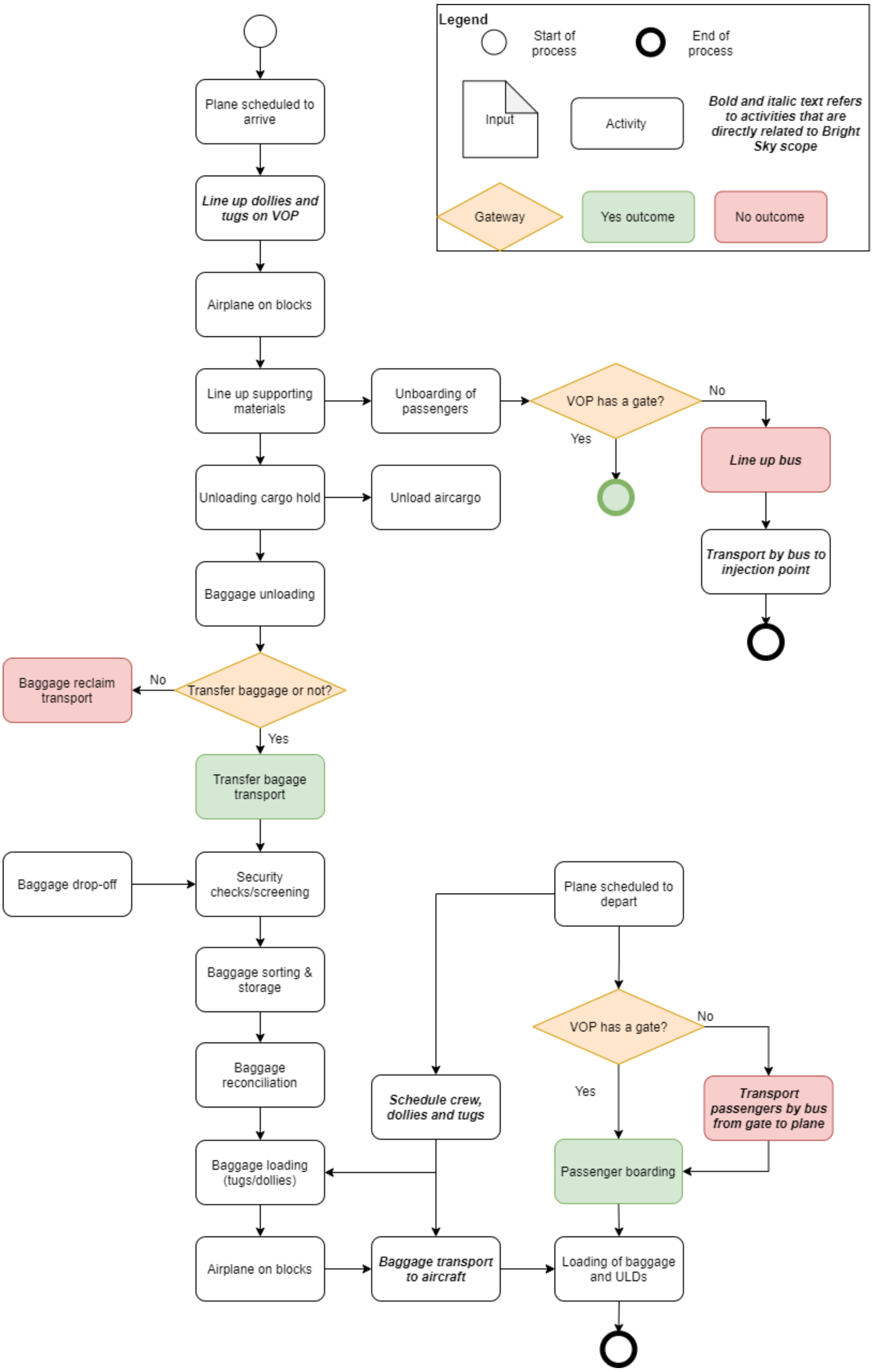
1. Baggage and goods (un)loading
2. Refueling
3. Maintenance
4. Waste collection
5. Cabin cleaning
6. Catering services

A.1.2 Air Traffic Control

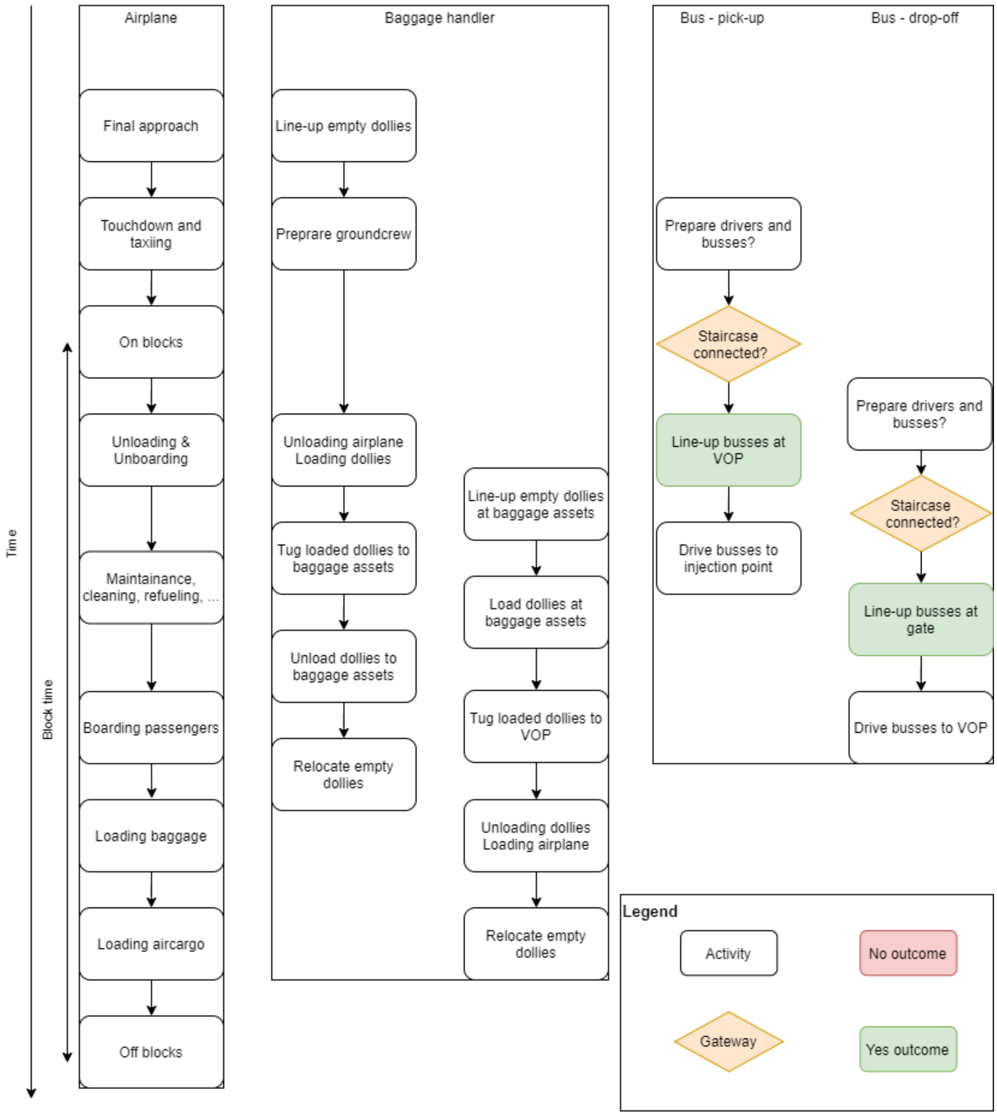
After all turnaround processes have been achieved it is not certain that the plane can directly depart afterwards. As the airplane needs lift off clearance from the Air Traffic Control, Lucht Verkeersleiding NederLand (LVNL). This is an additional actor to actors mentioned in chapter 4, but as it is not relevant in the baggage handling process it is left out of scope of the simulation model:

LVNL ensures safe flight, take off and landing procedures. The air traffic control center (LVNL) assigns landing and departure slots on the air strips to the flights. Their main goal is safety and no accidents (can) occur. With their decision they impact the arrival and departure time of flights and they do not play a direct role in the turnaround process; they merely define then when the process starts and ends. Therefore, they are very far away from the baggage transport process and are considered out of scope for the remainder of this report.

A.1.3 Overview of turnaround process and role of baggage handler and busses



Activities per stakeholder in turnaround process.



A.2 Data format example


```
experiment_name: problem1

# The input file where this problem was derived from, only for traceability
input_data_file: schiphol_data_20191008_a.pkl
problem_generation_timestamp: "2022-09-14T10:04:21Z"

# Fleet information, define the fleets of tugs for different types of operation
fleets:
 inbound:
 tugs:
  - tug_id: tug_1234
  driver: autonomous  # The driver type may impact the break times, driving style, reliability
  fuel: electric   # Fuel type defines where it may charge, but also can be used to take into account
emission behaviour
  driving_time_left: 7200 # The driving time left (in seconds) is a simplified model of the fuel tank /
battery size and efficiency, and thus the action radius. For now this ignores the amount of cars, weights
and driving speed. If needed we can elaborate this if we want to make the model more precise. The unit is
seconds, which corresponds to the unit of the driving times between locations.
  max_driving_time: 8000 # Maximum driving time (action radius) on a full fuel tank / battery.
  max_train_length: 4  # Maximum amount of trains this tug can pull, a simplification of the maximum pull
weight, if needed we can make this more precise.
  location: basementA  # Location of the tug at the start of the simulation. The string is a reference to
the location further down in the problem file
  - tug_id: tug_abcd
  driver: human
  fuel: diesel
  driving_time_left: 6000
  max_driving_time: 10000
  max_train_length: 6
  location: fuel_station1
 outbound:
 empty:

# Trailers are the empty cars on which we can load baggage or containers. They can be loaded or empty,
and have a physical location which may of course change over time.
trailers:
 - trailer_id: trailer1
 # max_weight: 123
 type: luggage      # The type of trailer defines whether it can be loaded with separate luggage items,
or a single container.
 load: basementA_to_KL1212_batch1  # This trailer has a load at the start of the problem definition
 location: basementA     # The location defines where the trailer is
 - trailer_id: trailer2
 type: luggage
 load:          # An empty trailer is denoted as one where the load is null
 location: basementC
 - trailer_id: trailer3
 type: container
 load:
 location: trailer_park

# Flight information contain information about deadlines and locations where the luggage has to go to or
come from.
flights:
 - flight_id: KL1211
 type: inbound       # Type of flight denotes wether the flight is inboud or outbound
 arrival_time: "2022-08-31T11:00:00Z" # Time when the flight is (expected) to arrive at the VOP to be
unloaded
 unload_deadline: "2022-08-31T12:00:00Z" # For an inbound flight: this is the time when the flight has to
be fully unloaded
 VOP: VOP123        # The location where the flight is arriving
 priority_level: 40      # A priority can be set to influence the planning. Here a number between 0 and
100 is suggested, but might as well be 0-1
 - flight_id: KL1212
 type: outbound
 departure_time: "2022-08-31T13:00:00Z"
 load_deadline: "2022-08-31T12:30:00Z" # For an outbound flight: this is the time when the loading has to
be finished
 VOP: VOP123
 priority_level: 50

# The luggage of the flights that need to be transported from one place to another, separated into jobs
jobs:
 - job_id: basementA_to_KL1211_firstclass_transfer
```

```yaml
 flight: KL1211
 end_time: "2022-08-31T12:00:00Z"
 origin: basementA      # The origin of the luggage for this job, where it need to be loaded on the
trailer
 destination: VOP123     # The destination of the luggage for this job, where it is unloaded.
 job_class: first class    # The class of the job may influence the priority
 service: transfer      # The service type influences the priority, but also to where the loading times
or locations should point to.
 batches:         # A job is divided into batches. One batch can be loaded onto one trailer, hence the
amount of batches indicate how many trailers are needed.
  - batch_id: basementA_to_KL1211_batch1
  type: luggage       # The type of batch corresponds to whether this batch is "loose" luggage, or a
single container. This should correspond to the type of trailer
  available_time: "2022-08-31T10:00:00Z" # Different batches of the job will have a different time at
which they are ready for picking up. This is useful information for a planner as it may choose to pick up
a single batch, or wait for multiple and take them in a single drive.
  nr_of_bags: 10       # Integer denoting number of baggage items in batch
 - job_id: basementA_to_KL1212_economy_reclaim
 flight: KL1212
 end_time: "2022-08-31T12:30:00Z"
 origin: basementA
 destination: VOP123
 job_class: economy
 service: reclaim
 batches:
  - batch_id: basementA_to_KL1212_batch1
  type: luggage
  available_time: "2022-08-31T10:00:00Z"
  nr_of_bags: 10
  - batch_id: basementA_to_KL1212_batch2
  type: container
  loading_time: 60      # The loading time (in seconds) of a batch can override the "default" loading
time that is specified in the locations file.
  available_time: "2022-08-31T10:05:00Z"
  nr_of_bags: 1
  - batch_id: basementA_to_KL1212_batch3
  type: container
  loading_time: 60
  available_time: "2022-08-31T10:10:00Z"
  nr_of_bags: 12
  - batch_id: basementA_to_KL1212_batch4
  type: luggage
  available_time: "2022-08-31T10:15:00Z"
  nr_of_bags: 13
  - batch_id: basementA_to_KL1212_batch5
  type: luggage
  available_time: "2022-08-31T10:20:00Z"
  nr_of_bags: 14
  - batch_id: basementA_to_KL1212_batch6
  type: luggage
  available_time: "2022-08-31T10:25:00Z"
  nr_of_bags: 15

locations: !include locations_schiphol.yaml  # The locations object is loaded from another file, since
this file is expected to be quite big, and does not change a lot between problems.
```

Mobility & Built Environment

Anna van Buerenplein 1
2595 DA Den Haag
www.tno.nl

innovation
for life