AIR CARGO TRUCK SCHEDULING AT A MAJOR EUROPEAN AIRLINE

Berend Markhorst, MSc, Corresponding Author
berend.markhorst@cwi.nl

Rob van der Mei, Prof. Dr.
rob.van.der.Mei@cwi.nl

Elenna Dugundji, Dr.
elenna_d@mit.edu

Word Count: 6113 words + 1 table(s) × 250 = 6363 words

Submission Date: June 28, 2022

ABSTRACT

Air cargo has emerged as an essential pillar in the airline industry and will likely grow even more in the coming decades. Therefore, airlines need to efficiently use their resources to meet the increasing demand in volumes and speed. This study focuses on the export stream of goods, specifically on accepting deliveries at the warehouse just before departure. Most clients' trucks arrive unevenly distributed over the week. Combined with limited unloading capacity, this results in heavy congestion at peak hours, which is undesirable for the client and the airline. Currently, the airline serves the trucks roughly on a first-come-first-serve basis. However, there is room for improvement in sequencing the trucks - sticking to a traditional policy as first-come-first-serve is not beneficial with such a complex problem. This study proposes a fast heuristic that can balance the average and the spread of the waiting times, two performance measures for a schedule. The results give insight in the relation between both performance measures and enable stakeholders to make a trade-off.

Keywords: Truck Scheduling, Air Cargo, Iterated Greedy

INTRODUCTION

Nowadays, transporting air cargo, any property carried in an aircraft (*1*), is a common way of sending goods all over the world. However, it was only in the 1990s that cargo emerged as an essential pillar in the airline industry, partly caused by the rise of companies such as DHL and FedEx. The latter provided its customers in 1992 with small computers, which enabled them to track their packages, which the contemporary reader might recognize as the first steps toward track-and-trace. The growth of global (e-)commerce around the 2000s also boosted the cargo industry. First, it became common to send packages worldwide for business purposes. Later, due to companies like Amazon, air cargo became accessible to consumers. Compared to passenger transport, air cargo has experienced an even higher growth during the last 30 years (*2*). Although the growth in this industry is stagnating due to recent economic crises, experts expect it to play a more prominent role in the world economy in the coming decades. In combination with the increasing demand of customers regarding delivery speed and quality, the supply chain within this industry is under constant pressure (*3*).

In the air cargo supply chain, typically, multiple parties are involved when transporting a package from A, the *shipper*, to B, the *consignee*. Figure 1 shows a schematic representation of the air cargo supply chain. In step one, the shipper, a consumer, or a company goes to the forwarder to send a package. An example of a forwarder is PostNL. In step two, the forwarder will book some space on the airline's flight to the package's destination and deliver the package on time at the airline's warehouse. In step three, the airline packs the received goods on an unit load device (ULD), which can be seen as a big pallet for the airline industry, and puts it on the corresponding flight. In step four, having landed at the destination's airport, the airline delivers the package to a forwarder (for example UPS or DHL). In step five, these companies will cover the last part of the package's journey to the consignee. The scope of this research lies in between steps two and three.

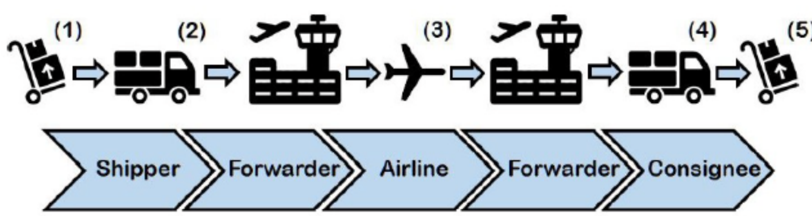

FIGURE 1: Schematic representation of the air cargo supply chain.

The Airline

The airline is one of the oldest operating airlines in the world and has been the flag carrier airline of a European country ever since its creation. Its hub is strategically placed and therefore attractive for (business) customers. The airline transports approximately 80 million passengers annually, has over 500 available planes, and has an extensive network of hundreds of destinations. Its organization can be divided into three departments: passengers, cargo, and maintenance & engineering. Although they are related to each other - passenger planes can also carry air cargo in their bellies - this research focuses primarily on cargo.

The Cargo Department

The cargo department is responsible for transporting air cargo through the airport, both on the import and export side. Although they have six dedicated cargo planes, passenger flights transport a considerable part of the cargo. Overall, the airline has 208,000 m^2 warehouse floor surface. At the airport, there are three warehouses next to each other, which figure 2 shows. Freight Building 1 is dedicated to particular goods which require high priority or special care, such as live animals, diamonds, and diplomatic mail. Freight building 2 focuses on the import side of the supply chain: it takes packages from incoming flights and puts them on trucks that deliver the goods to destinations in Europe. Finally, Freight Building 3 does the export side: trucks from a European hub deliver goods at the warehouse, which go on a flight to a non-European destination. The scope of this research lies in the third freight building.

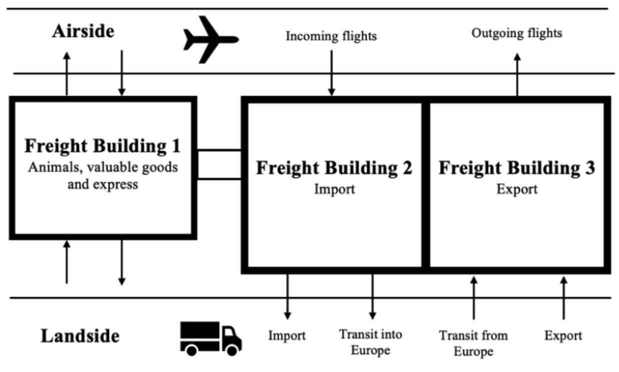

FIGURE 2: Freight buildings at the airline.

Freight Building 3

Between 500 and 700 trucks arrive at the third freight building every week. A truck goes through seven steps to deliver its goods at the airline. In step one, the truck enters the pre-entry area. Then, the truckers park their truck, after which they go to the documentation office for a document check. The fourth and fifth steps include entering the hub area and parking the truck. If the trucker gets a sign that it is their turn, the trucker parks at an unloading dock, and an employee of the airline starts unloading the truck. In step seven, the truck leaves. According to several domain experts at the airline, maximally, three doors can be opened simultaneously.

The Airline's Current Policy

The airline offers its customers the opportunity to book space on a flight via an online portal. Currently, the airline is trying to reward clients who book via this portal by giving them priority in their queuing policy. In addition, the airline needs to prioritize "hot" deliveries based on the latest arrival times (LAT). As a result, their current queuing policy looks at these two truck properties.

Problem Description

At Freight Building 3, trucks arrive unevenly distributed over the week. Congestion arises during peak hours (typically Tuesday and Friday nights), partly caused by the limited unloading capacity. See figure 3 for an illustration of congestion in the warehouse. This congestion increases waiting time and a decreased Flown as Planned (FAP), an important KPI for the airline that shows the ratio

FIGURE 3: Congested Freight Building 3 on a Tuesday night at 02:00.

of goods that get to the right flight at the right time. An increase of 1% in this KPI can yield higher revenues, hence the airline's interest in this problem. This research focuses on the arrival of trucks at Freight Building 3 and looks into possible sequencing methods.

Ou et al. mention that a limited number of open doors is the primary constraint during peak hours (*4*). The actual arrival times are mainly unknown to the airline, although some truckers give an expected arrival time. The clients are free to arrive at the time of their preference, usually close to the flight departure. This phenomenon gives the airline a tight time window for the operation, which can jeopardize the airline's service level. Reducing the resulting congestion is possible by planning the arrivals of trucks in a sophisticated manner since the KPIs depend on the arrival rate of the trucks (*5*). Furthermore, Chen et al. state that time slot scheduling helped alleviate peak hours in the maritime sector (*6*). There are research opportunities in the air cargo terminal operations (*7*). The authors recommend to focus on integrating operations, for example, through collaborative planning.

Figure 4 illustrates the problem. The trucks arrive in different priority lanes at the documentation: one for normal bookings and one for customers that book via the online portal. After the documentation, the trucks form a line that goes through security, after which they park their trucks near Freight Building 3. There is one big queue and three docks. The goal is to find a sophisticated sequencing method that benefits the airline and its clients. The scope of this research lies in the red marked area in figure 4 since the waiting times at the documentation are relatively negligible compared to the waiting times at Freight Building 3.

Outline

In section 3, available literature is discussed. In section 4, the data is analyzed. Section 5 provides the reader with methodologies used to solve the problem as mentioned earlier, after which the results from different scenarios follow in section 6. Finally, section 7 continues with concluding remarks and recommendations.

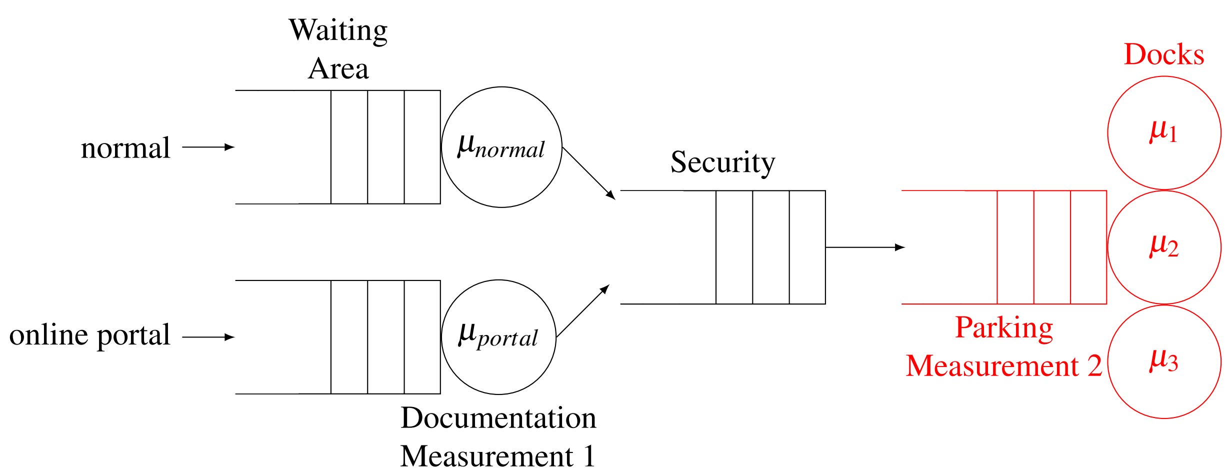

FIGURE 4: Schematic representation of the queuing system at Freight Building 3.

LITERATURE RESEARCH

The problem described in the previous section is often referred to as the Parallel Machine Scheduling Problem (PMSP) with release times and deadlines, or $Pm|r_jd_j|C_{\text{mean}}$ in Graham's notation (*8*). The parallel machine scheduling problem has been studied elaborately over the years. The addition of release times and deadlines to the original PMSP makes the problem more specific and less studied.

Classic Theories

In queuing theory, some well-known policies are easy to understand and have advantages. For this research, especially two are interesting: First Come, First Serve (FCFS) (*9*), and Shortest Job First (SJF) (*10*). Since the airline does not work with preemption, this study only considers the non-preemptive version of FCFS. As the name suggests, FCFS serves customers in order of arrival. This principle boils down to giving no one priority, which is a straightforward and fair system. However, it is not always efficient. For example, a customer with a short service time at the back of the queue must wait long before service.

As a result, the average waiting time for this policy is relatively high. SJF first serves the customer with the lowest service time, resulting in a lower average waiting time because it prioritizes small customers. One can compare this to a classic scenario in the supermarket: most people will let a fellow customer with a small number of groceries go in front if they have many items in their basket. In theory, this policy sounds simple and efficient. However, this results in longer waiting times for the larger customers and can even lead to outliers. Hence, both are not ideal options. One needs to make a trade-off between the average and the spread of the waiting times. Sticking to one policy has the benefit of clarity to customers. However, it is interesting to look into flexible policies that do not follow a specific rule but focus on increasing the service quality, such as a lowered waiting time while maintaining a relatively equal spread among the customers. Alternatively, as stated in section 1.3 of (*11*), "a more advanced schedule may be needed as a compromise between fairness and efficiency."

Mathematical Modeling

A Linear Program (LP) or Mixed-Integer Linear Program (MILP) can find exact solutions to complex problems. Hence, the academic interest in this model. For example, in (*12*), the authors propose a MILP for berth allocation of vessels in the maritime sector, which translates directly to the problem in this paper. The vessels become trucks, and the berths are now docks. Note that this research has added decision variables and constraints to the original formulation.

Problem description

This problem contains two sets, namely T trucks and D docks. Also, there are two parameters: a_j and p_j are the exact arrival time and service time for truck $j \in T$, respectively. Finally, there are four decision variables. s_j and w_j are the docking time and waiting time for truck $j \in T$. x_{ij} is a binary variable that equals 1 if truck $j \in T$ gets scheduled at dock $i \in D$. $I_{ijj\prime}$ is a binary variable that equals 1 if trucks j, $j\prime$, $j \neq j\prime$ are both scheduled at dock $i \in D$ and truck j before truck $j\prime$. The objective, see (1), of the linear program, is to minimize the total sum of waiting times. Next to the four integrality constraints, see equations (8), (9), (10) and (11), there are six constraints. Equation (2) ensures that each truck gets assigned to one dock and (3) that trucks can only unload after their arrival. Line (4) puts a lower bound on the docking time of truck $j\prime$ if it unloads earlier than truck j and both trucks share the same dock. Namely, a dock can only help the next truck after completing the last truck. Equations (5) and (6) ensure that either $I_{ijj\prime}$ or $I_{ij\prime j}$ equals 1 if both trucks j and $j\prime$ are assigned to dock i. Otherwise, both decision variables equal 0. Finally, equation (7) computes the waiting time of each truck, which is defined as the difference between the docking and arrival time.

Mathematical Model

$$
\begin{aligned}
\min \quad & \sum_{j \in T} w_j && && (1)\\
\text{s.t.} \quad & \sum_{i \in D} x_{ij} = 1 && \forall j \in T && (2)\\
& s_j \geq a_j && \forall j \in T && (3)\\
& s_{j\prime} \geq s_j + p_j - M(1 - I_{ijj\prime}) && \forall j, j\prime \in T, i \in D \quad \text{s.t.}\, j \neq j\prime && (4)\\
& I_{ijj\prime} + I_{ij\prime j} \leq \frac{1}{2}\left(x_{ij} + x_{ij\prime}\right) && \forall j, j\prime \in T, i \in D \quad \text{s.t.}\, j < j\prime && (5)\\
& I_{ijj\prime} + I_{ij\prime j} \geq x_{ij} + x_{ij\prime} - 1 && \forall j, j\prime \in T, i \in D \quad \text{s.t.}\, j < j\prime && (6)\\
& w_j \geq s_j - a_j && \forall j \in T && (7)\\
& x_{ij} \in \{0, 1\} && \forall j \in T, i \in D && (8)\\
& I_{ijj\prime} \in \{0, 1\} && \forall j, j\prime \in T, i \in D \quad \text{s.t.}\, j \neq j\prime && (9)\\
& w_j \in \mathbb{R}_+ && \forall j \in T && (10)\\
& s_j \in \mathbb{R}_+ && \forall j \in T && (11)
\end{aligned}
$$

Disadvantages

This method requires a considerable amount of decision variables. In the case of 200 trucks and three docks, realistic numbers for a peak hour on Friday night at the airline, 120,400 decision variables are needed. The majority of these are $I_{ijj\prime}$. One can imagine that this slows down a solver on a regular laptop, mainly because it involves binary variables. Furthermore, besides the integrality constraints, there are 80,000 other constraints. Therefore, solving such an instance to optimality requires significant time and computation power.

Column generation

Column Generation (CG) is an efficient algorithm for solving large linear programs and is based on the idea that many LPs are too large to consider all the variables explicitly. Usually, one starts with a subset of variables and iteratively extends this set with the most promising variables. This procedure stops when adding new variables does not improve the objective. Ideally, it uses only a tiny fraction of the variables. The algorithm divides into two problems: the master-problem and the sub-problem. The first is the original LP which considers a subset of variables. The latter looks for and adds the most promising variables to the master-problem. For a more detailed explanation, see (*13*).

In (*14–16*) column generation is used to solve parallel machine scheduling problems. Despite looking very promising for solving big instances, CG has only been described in (*16*) to solve instances with three machines and with a maximum of 50 jobs. In this study, a schedule for approximately 200 trucks arriving within 24 hours should be made. Therefore, CG becomes overly computationally expensive. Furthermore, CG is useful when there are many variables compared to the number of constraints, which is not the case in the MILP of (*12*). For both reasons, this method is considered unsuitable for this research.

Meta-heuristics

MILPs are usually computationally expensive. Therefore, one prefers to use heuristics in practice. Regular heuristics are problem-dependent. However, meta-heuristics are problem-independent and applicable to many different problems. Well-known examples are genetic algorithms and simulated annealing.

Genetic Algorithm

A Genetic Algorithm (GA) is a search heuristic inspired by Charles Darwin's Theory of Evolution and applicable to different complex problems. According to the principle of Survival of the Fittest, the individual solutions in a generation keep improving until convergence. Parent selection, crossover, and mutation play an important role in this process. See (*17*) for a more detailed explanation.

Reeves et al. propose a method to use GA to solve a sequencing problem (*18*). Each genome represents an allocation of trucks to docks. Each cell in the genome represents a truck and corresponds to the dock to which the truck gets assigned. However, how should one find a sequencing schedule for each dock individually in that case? The available literature did not answer this question. Section 5 elaborates on this topic regarding this specific question.

Simulated Annealing

Another heuristic is simulated annealing (SA). Liu et al. and Gorissen et al. describe how to use SA to minimize the makespan of the m-machine and n-job flow shop sequencing problem (*19, 20*). It helps to find an approximate global optimum and is known for using a parameter called *temperature*. SA works as follows: the temperature decreases from an initial value to zero. The algorithm selects a neighboring solution at each iteration and jumps to it based on a temperature-dependent probability. Because the temperature is relatively high initially, it is possible to escape local optima in that optimization phase. Hence, exploration takes place in the beginning, whereas later exploitation occurs. Similar to GA, a benefit of this method is that it does not require a tailored algorithm to produce reasonable solutions.

Iterated Greedy

However, GA and SA usually require substantial computation power. In most cases, "tailored algorithms are more efficient and more effective than simulated annealing" (*19*). For this reason, both GA and SA are unsuitable for this research. However, Iterated Greedy (IG), another meta-heuristic, was still considered for this research because of its simplicity, speed, and high performance.

IG is closely related to SA: both can use a temperature parameter. IG is SA with specific settings: both fall under stochastic local search. The authors in (*21*) explain the principle of this method, which is the current state-of-the-art for scheduling problems. IG is a search method based on two pillars: the *partial deconstruction* of a complete candidate solution and a *subsequent recon-*

struction. First, a constructive method produces an initial solution. Greedy construction heuristics are usually fast and perform better than randomly initialized solutions. Additionally, seeding the next step with a greedy solution can speed up the minimization process considerably. Then, the algorithm enters a loop, which stops after a specific stop criterion, and consists of four iterative steps. First, a few components of the candidate solution are removed. This happens either randomly or according to a specific method. Second, the partial solution is rebuilt into a new candidate solution. An optional third step consists of applying local search to this solution, for example, swapping components of the solution around. The last step is the acceptance criterion. This study focuses on a minimization problem. Therefore, solutions with a lower objective value are always accepted. However, to escape local optima, it is sometimes desirable to accept worse solutions. One usually uses the Metropolis criterion to make the trade-off between accepting or rejecting such solutions (*21*):

$$\exp\left(\frac{s-s'}{T}\right) \tag{12}$$

where s' and s are the objective values of the new solution and the solution, respectively, and T is the temperature. The resulting value can be put in a Bernoulli distribution to determine whether a worse solution should be accepted. To summarize, the pseudo-code of IG is shown in algorithm 1. IG has beneficial characteristics. It is simple and only uses a few primary parameters.

Algorithm 1 Iterated Greedy

```
1: Solution from greedy construction heuristic
2: while Stop criterion is not met do
3:     Partial solution after destruction
4:     Candidate solution after construction
5:     Candidate solution after local search
6:     Acceptance criterion chooses between old and candidate solutions
7: end while
```

Similar research

Casteren follows a similar pattern and concludes that an LP is a suitable method for sequencing trucks (*5*). The author researched the effect of a slot allocation policy with prioritization for trucks that were booked via an online system. Although the results on small problem instances look promising, the high computation times prevent the model from being used in practice. Another disadvantage is that the author did not test it on realistic problems. For future research, the development of a heuristic was recommended. The author suggests to study whether that method works well on more extensive and realistic problem instances.

DATA

This section gives a description and insights into the data used. The research primarily uses one data set, containing information about roughly 50,000 trucks that arrived between December 28th 2020 and January 2nd 2022. Furthermore, it consists of 28 attributes, which the Appendix describes more elaborately, section **??**.

Data Quality

In general, the quality of the data is reasonable. Most of the values are entered automatically via the airline's internal system. However, some parts of the data seem erroneous. For example, according to the column "Actual weight of the AWB", 2,996 trucks carry a total load weighing less than 5 kg, and 7,556 trucks more than 10,000 kg. Also derived attributes, such as "Service Time (minutes)" and "Waiting Time (minutes)," contain outliers: 2,577 trucks require more than 2 hours of unloading time, and 9,971 trucks are waiting more than 5 hours. Also, some data is not included. For example, the airline does not store data about the trucks' docking time, which would be useful for applying queuing models. Additionally, the data did not include which trucks were booked via the online portal.

Data Wrangling

The data does not have specific attributes for the unloading and waiting time. However, subtracting the start and finish time of unloading yields the unloading time. Subtracting the actual arrival time from the start of unloading gives the waiting time. Finally, the difference between the flight departure and the arrival time results in the time a truck arrives in advance. Note that negative values for these derived attributes should not be possible.

Data Exploration

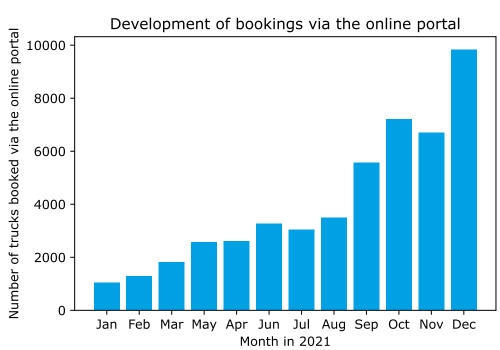

FIGURE 5: Development of portal-bookings.

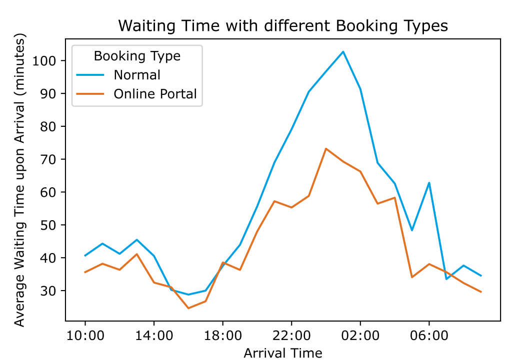

FIGURE 6: Waiting time for (non-)portal trucks.

Online Portal

The airline offers its customers the opportunity to book space on a flight via an online portal. This enables the customers to provide the airline with some essential details: the *volume*, *contents* and *weight* of the freight, but also the estimated time of arrival at the warehouse. In practice, the data regarding these timeslots contains too many errors to draw conclusions on the certainty of truck arrivals. In 2021, roughly 8,000 trucks were booked via the online portal, which amounts to 16%

of all trucks that arrived in that year. The stakeholder of the airline in this project, mentioned an estimated ratio of 75% in the middle of 2022. This seems reasonable because this ratio is growing by the day, as can be seen in figure 5. Additionally, the airline is prioritizing portal-trucks in their current docking policy in order to incentivize sharing information. From the data, see figure 6, it can be seen that portal trucks have considerably lower waiting times during peak hours.

Waiting Time

The waiting time of a truck depends heavily on its arrival time, which is shown in figure 7. This figure shows data aggregated over all weekdays. Hence, the pattern might differ per day. Typical peak hours are between 7:00 PM and 3:00 AM. On average, a truck waits 50 minutes before unloading. However, the deviation is considerable: during the day, the waiting time is below 30 minutes, but at night, it can amount to almost 80 minutes.

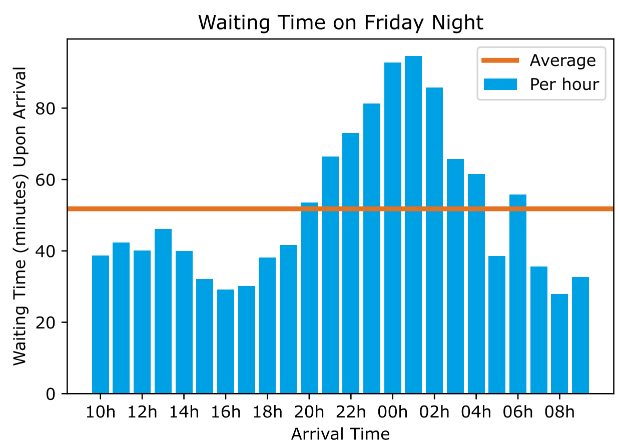

FIGURE 7: Waiting time upon arrival of a truck.

METHODOLOGY

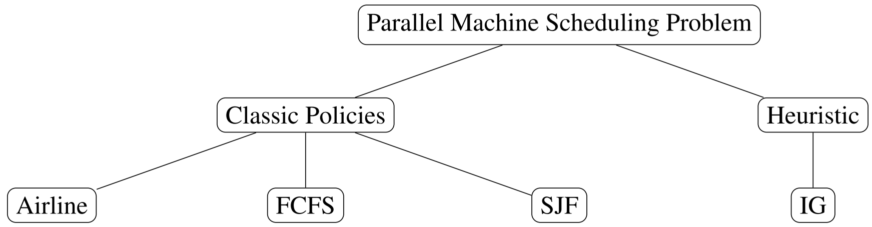

FIGURE 8: Overview of used methods in this study.

This section will explain the experimental setup. Figure 8 shows an overview of all used methods in this study. The top of the figure shows the problem to be solved: PMSP. Since the export supply chain contains many random factors, a reliable analysis requires a sufficient amount of data or multiple simulations. This study focuses on the moment of the week with the highest traffic load: from Friday 10:00 am to Saturday 10:00 am. The data set contains only 52 Fridays; hence simulations are needed. Section 5.1 addresses the methodology behind the simulations. Following, the diagram splits into two sections: one category for classic policies (section 5.2) and one for heuristics (section 5.3). The third level of the diagram displays four methods (Airline, FCFS, SJF, and IG) that deal with the deterministic version of the PMSP.

Simulation

Original Data

As mentioned earlier, Friday night is the busiest moment in the week at the airline. That part of the week was used for detailed analysis of congestion at peak hours, a moment at which the airline's scheduling problem are best visibile. A realistic arrival schedule of trucks requires four attributes per truck:

- **Arrival Time**: time a truck arrives at the airline's documentation office.
- **Service Time**: time required for a truck to unload at a dock, including docking and unloading time.
- **Time in Advance**: time in advance a truck arrives before the corresponding flight departure.
- **Online Portal**: binary variable which equals 1 if the truck was booked via the online portal and 0 otherwise. In current practice, 75% of the trucks come via the online portal at the airline.

A fifth characteristic is the *number of trucks per hour*. In queuing theory, it is common to assume that the arrivals occur according to a Poisson process. This process implies that the number of trucks counted in a certain interval has a Poisson distribution and that the inter-arrival times follow an exponential distribution. These two conditions have been checked for each hour on a Friday night via QQ-plots and visual inspection (figure 9 and 10). The Poisson distribution aligns with the number of truck arrivals per hour. Additionally, the inter-arrival times follow an exponential distribution. Hence, a Poisson arrival process was considered to be valid in this study.

Simulated Data

Trucks were independently created based on random draws from distributions obtained from pooled data. The number of trucks per hour came from a Poisson distribution, the service time from an

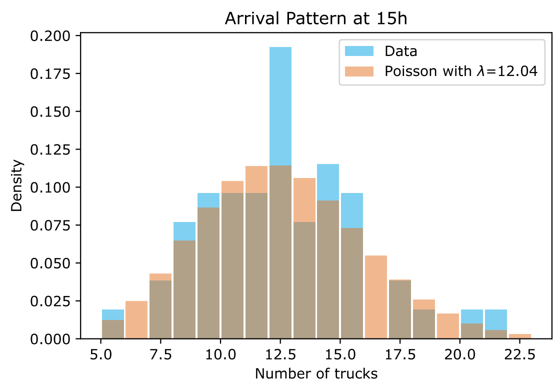

FIGURE 9: Bar plot of historical arrivals at Friday 15:00.

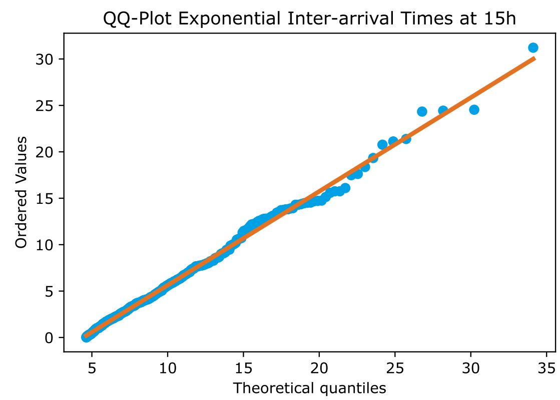

FIGURE 10: QQ-plot of historical inter-arrival times at Friday 15:00.

exponential distribution, the time in advance from a gamma distribution, online portal from a Bernoulli distribution, and the arrival time from a uniform distribution. This process repeats for every hour in the simulation. This study used the airline's Friday night data to generate the distributions mentioned earlier. Since the data only contained the unloading time and not the remaining overhead time required by a truck for docking, 15 minutes were added to the exponential distribution of the service time as a mock overhead time parameter. This manipulation was necessary to obtain an average waiting time of 50 minutes for the airline's current policy, which resembles the average waiting time in the data.

Variation in Service Time

As part of the parameter analysis, which section 6 elaborates on, the service times' variability was manipulated in order to study its influence on the solutions. The arrival schedules that the simulations produced remained the same. However, new service times were inserted, which were drawn from a gamma distribution with an adjusted coefficient of variation. The values of the parameters, shape-parameter k and scale-parameter θ, for the gamma distribution were obtained by using the following equations for the mean and variance, respectively.

$$\mu = k\theta \tag{13}$$

$$\sigma^2 = k\theta^2 \tag{14}$$

Rewriting both equations yields $k = \frac{\mu}{\theta}$ and $k = \frac{\sigma^2}{\theta^2}$. Combining these equations gives $\theta = \frac{\sigma^2}{\mu}$. To move k to the left side of the equation, $\theta = \frac{\mu}{k}$ and $\theta^2 = \frac{\sigma^2}{k}$ were used, which results in $k = \frac{\mu^2}{\sigma^2}$. By changing the standard deviation in both k and θ, the coefficient of variation can be artificially modified.

Classic Policies

For FCFS, all trucks were sorted on their arrival times. Then, the algorithm iterated over the trucks and assigned them to the queue with the least congestion at that moment. The airline's current policy is similar to FCFS. All trucks get a priority score based on LAT and online portal. In case of an equal priority score, the airline serves the truck with the earliest arrival time. Contrastingly, SJF sorts trucks based on service time in ascending order. If a server is idle, it serves the first truck in the queue.

Heuristics

The benefit of an exact model, such as a MILP, is that it should hypothetically produce an optimal solution. However, a well-known characteristic of LPs is their high computation times. Usually, a trade-off must be made between the computation time and the quality of the solution. In reality, most stakeholders prefer high-speed algorithms that produce near-optimal solutions. Such algorithms are generally called heuristics.

Iterated Greedy

Different meta-heuristics were considered. As mentioned in section 3, Iterated Greedy is a state-of-the-art method for the PMSP. IG is a search method based on two pillars: the *partial deconstruction* of a complete candidate solution and a *subsequent reconstruction*. Because of its simplicity (both for implementation and explanation), limited computation time, and high performance, this study implements IG. The following enumeration elaborates on the different components of the algorithm.

- **Representation**: a job sequence represents each machine. For example, the following is a 3-machine 10-job solution representation: [1,2,3,4,5], [6,7,8,9], [10]. This representation is suitable for IG because it makes swapping and inserting trucks simple.
- **Initial construction heuristic**: first, the algorithm sorts the trucks at arrival time. Then, the algorithm iterates over the ordered trucks and assigns each truck to the dock with the shortest completion time. Although this method is greedy, it usually performs well.
- **Destruction**: two options are considered. First, the algorithm removes d random trucks from the sequence. Second, it removes the sequence's d trucks with the longest waiting time. A disadvantage about the latter is that it might cause cycling. Therefore, it was not used in the experiments.
- **Construction**: the d removed trucks are inserted into the solution individually. The algorithm chooses the best option for every truck out of all possible positions in the sequence. The algorithm exclusively considers insertion points within a few hours around the truck's arrival time for computation purposes.
- **Local search**: the algorithm removes a random truck from the sequence. Then, it chooses the first position that gives a lower average waiting time (the First Improvement Method). This process repeats until failure.
- **Acceptance criterion**: currently, the algorithm only accepts improvements. However, the Metropolis criterion is useful because it enables IG to escape from local minima.
- **Objective function**: the goal of IG is to optimize both the average and the spread of the waiting times in a schedule. The spread is measured with the standard deviation of the waiting times. A small deviation is desirable. The average and the standard deviation of the waiting times need to be minimized. Therefore, the following objective function is suggested:

  $$\text{score} = \alpha\gamma\bar{W} + (1-\alpha)\sigma \tag{15}$$

  where α represents the weight, varying between 0 and 1, of the average waiting time and γ the scaling-factor that ensures an equal order of magnitude for the average and spread of the waiting times. Finally, σ represents the standard deviation of the waiting times.

Different terms are used in the literature about meta-heuristics for the PMSP than in this thesis. Therefore, the table 1 serves as a translation to bridge the terminology gap. Minimizing the

TABLE 1: Terminology translation.

Original	Scheduling
Dock	Machine
Truck i	Job j
Service time p_i	Processing time p_j
Arrival time a_i	Release date r_j
Start time x_i	Start time s_j
Waiting time $w_i = x_i - a_i$	No scheduling terminology
No original terminology $C_i = a_i + p_i$	Completion time $C_j = s_j + p_j$
Truck Unloading Deadline $c_i + a_i - L$	Due date d_j
No original terminology $F_i = x_i - a_i + p_i$	Flow time $F_j = C_j - r_j$

average waiting time is not a common objective in the scheduling literature. Most studies minimize the total flow time, which is an equivalent term because:

$$\min \sum_{i \in T} (x_i - a_i) / |T| = \min \sum_{i \in T} (x_i - a_i) \tag{16}$$

$$= \min \sum_{i \in T} (x_i + p_i - a_i) \tag{17}$$

$$= \min \sum_{i \in T} (C_i - a_i) \tag{18}$$

$$= \min \sum_{i \in T} F_i \tag{19}$$

$$\tag{20}$$

where all equalities hold by simple manipulation of constant values (which do not affect minimization).

RESULTS

Figure 11 shows the trade-off between the average and the standard deviation of the waiting times for one instance. The x-axis represents the average, and the y-axis displays the spread of the waiting times. The figure shows that SJF has a low average and high spread of the waiting times and that FCFS has a high average and a low spread of the waiting times. Therefore, FCFS can be found on the right bottom and SJF on the left top of the figure. There is a trade-off between the spread and the average waiting times. As explained in section 5, IG minimizes both. With weight α, the importance of either the average or the spread can be adjusted. In the figure, several data points are plotted to show IG solutions. Each point denotes one solution of IG with a different value for α. With $\alpha = 0$, IG minimizes the waiting times spread, resulting in a point on the right bottom of the figure. As α increases, the points move to the top left of the figure. The points form a hockey-stick shaped pattern. As α increases, a decrease in the average waiting time must be compensated with a large increase in the spread of the waiting times. The airline is represented by one data point, combining both portal and non-portal customers. Due to the airline's prioritization policy, portal customers have a substantially lower average waiting time than non-portal customers. However, for the sake of simplicity, this was not included in the figure.

Figure 11 is based on one regular instance with a normal variability in the service time. The experiment was repeated on the same instance but with adjusted service times. Figure 12 is based on an instance with constant service times (i.e., with no variability) whereas figure 13 is based on an instance of which the squared coefficient of variation in service time amounts to 19.65. The aforementioned hockey-stick shape disappears in figure 12 and becomes clear in figure 12. In addition, FCFS and SJF are the same in figure 12, as well as IG with $\alpha \in \{0.4, 0.6, 1\}$. Figure 13 shows that IG can produce solutions with a lower average waiting time than SJF.

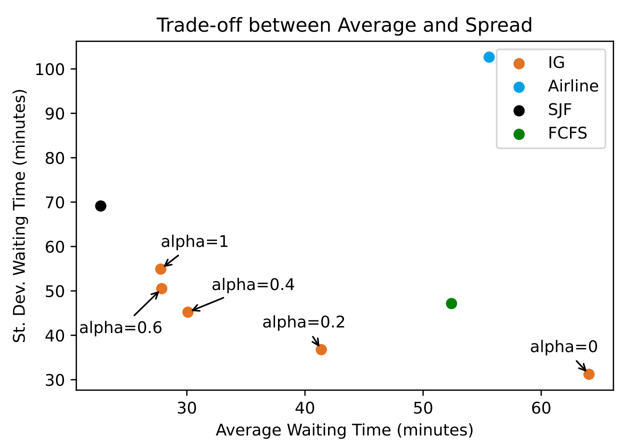

FIGURE 11: Run of IG with different weights on one instance with regular variability in service time.

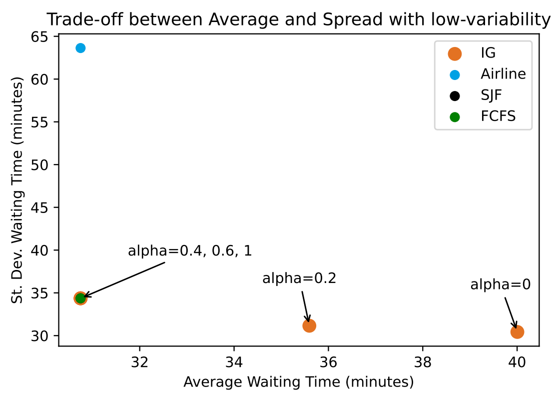

FIGURE 12: Run of IG with different weights on one instance with no variability in service time.

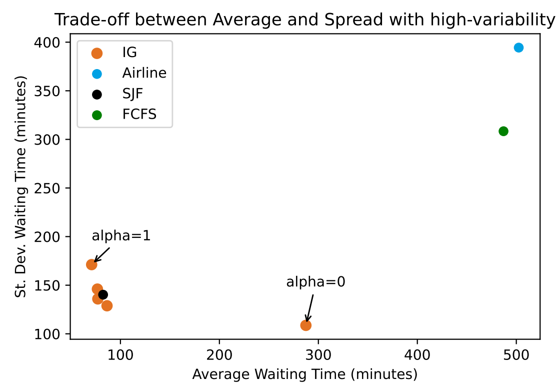

FIGURE 13: Run of IG with different weights on one instance with high variability in service time.

CONCLUSION

Discussion of the Results

Figures 11, 12 and 13 show that the shape of the hockey-stick becomes more evident when the variability in the service time increases. When this variability is increased, the trucks can be divided into two groups regarding their service times: mice and elephants. In case of no variability, corresponding to figure 12, every truck has a constant service time. In that case, SJF and FCFS are equal. Additionally, the hockey-stick is less evident because there are no large customers (elephants) that can be sacrificed for the smaller customers (mice). Figure 13, with considerable variability in service time, shows some elephants that can be sacrificed for the mice. However, this method can only be applied to a certain extent. After $\alpha = 0.5$, the elephants need to compensate (i.e., wait) considerably to lower the average waiting time. This phenomenon results in unfair schedules.

Figure 11 shows that IG never reaches the average waiting time of SJF, even though α equals 1. This is likely caused by the IG getting trapped in a local minimum. Since the current implementation of IG is simplistic and does not accept worse solutions to escape from a local minimum, it is predictable that IG cannot find the global minimum.

In theory, SJF is expected to produce the lowest average waiting time. However, figure 13 shows that IG can outperform SJF in terms of average waiting time. This could be due to starvation, which happens if many short jobs arrive sequentially (*22*). In that case, a truck with a long service time has to wait almost indefinitely in case of an SJF policy. Because the variability in the service times in that experiment was considerably large, starvation could explain the performance of IG.

Limitations of the Research

The most significant limitation of this research is that it tries to compare apples to oranges. The airline's current policy, FCFS, and SJF are online queuing policies: they update the planning every time a new truck arrives. However, IG is an offline model. It makes one plan in advance which does not change. Because the basic policies have the flexibility of updating the schedule, it is difficult to compare them with the model proposed by this research.

The advantage of the airline's current policy, FCFS, is its clarity. All customers know in advance what they can expect: trucks are served in order of arrival, except for portal and late trucks. The system proposed in this study can be seen as a black box by the clients. The customers do not know how the truck sequencing is determined. In other words: the policy cannot be explained simply in a few sentences.

Key Insights

The PMSP in the supply chain of the airline is far from trivial. Hence, applying a basic sequencing model as first-come-first-serve is not desirable for this operation. This study has shown that it can be beneficial to differ from FCFS by prioritizing small customers, even if they are not physically present yet at the warehouse. As a result, sometimes a dock might be held idle while waiting for such a customer, which can be considered counter-intuitive. Nevertheless, the results in section 6 have shown a reduction in both the average and spread of the waiting times. Hence, there are several opportunities to improve the current policy. It should be noted that the information provided by the forwarders can be considerably more accurate, which benefits the planning quality. If this is not possible, the airline's current performance is reasonable compared with other classic queuing policies.

Future Research

This research focused on offline scheduling methods. However, it may be useful to explore the possibilities of online scheduling. This research did not implement uncertainty in the heuristic, whereas it is also possible by using SimOpt (*23*). SimOpt optimizes the average scenario, making the schedule robust against fluctuations in the input parameters. Finally, it is useful to study at which local minimum IG ends and how IG can be adjusted to find a better optimum.

REFERENCES

1. Allaz, C., *History of air cargo and airmail from the 18th century*. Google Consultant, 2005.
2. Kupfer, F., H. Meersman, E. Onghena, and E. Van de Voorde, The underlying drivers and future development of air cargo. *Journal of Air Transport Management*, Vol. 61, 2017, pp. 6–14.
3. Ramachandran, G. and S. Tiwari, Challenges in the air cargo supply chain. *Communications of the ACM*, Vol. 44, No. 6, 2001, pp. 80–82.
4. Ou, J., V. N. Hsu, and C.-L. Li, Scheduling truck arrivals at an air cargo terminal. *Production and Operations Management*, Vol. 19, No. 1, 2010, pp. 83–97.
5. Casteren, L. v., *Truck arrival scheduling for air cargo terminals: Modelling a slot allocation policy to alleviate congestion and improve performance*. Ph.D. thesis, University of Groningen. Faculty of Economics and Business, 2018.
6. Chen, G., K. Govindan, and Z. Yang, Managing truck arrivals with time windows to alleviate gate congestion at container terminals. *International Journal of Production Economics*, Vol. 141, No. 1, 2013, pp. 179–188.
7. Feng, B., Y. Li, and Z.-J. M. Shen, Air cargo operations: Literature review and comparison with practices. *Transportation Research Part C: Emerging Technologies*, Vol. 56, 2015, pp. 263–280.
8. Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of Discrete Mathematics*, Elsevier, Vol. 5, 1979, pp. 287–326.
9. Schwiegelshohn, U. and R. Yahyapour, Analysis of first-come-first-serve parallel job scheduling. In *SODA*, Citeseer, 1998, Vol. 98, pp. 629–638.
10. Arpaci-Dusseau, R. H. and A. C. Arpaci-Dusseau, *Operating systems: Three easy pieces*. Arpaci-Dusseau Books LLC, 2018.
11. Drozdowski, M., Classic scheduling theory. In *Scheduling for Parallel Processing*, Springer, 2009, pp. 55–86.
12. Eisen, H. E., J. E. Van der Lei, J. Zuidema, T. Koch, and E. R. Dugundji, An Evaluation of First-Come, First-Served Scheduling in a Geometrically-Constrained Wet Bulk Terminal. *Frontiers in Future Transportation*, 2021, p. 23.
13. Desaulniers, G., J. Desrosiers, and M. M. Solomon, *Column generation*, Vol. 5. Springer Science & Business Media, 2006.

14. van Den Akker, J. M., J. A. Hoogeveen, and S. L. van de Velde, Parallel machine scheduling by column generation. *Operations Research*, Vol. 47, No. 6, 1999, pp. 862–872.
15. Chen, Z.-L. and W. B. Powell, Solving parallel machine scheduling problems by column generation. *INFORMS Journal on Computing*, Vol. 11, No. 1, 1999, pp. 78–94.
16. Van Den Akker, J., J. Hoogeveen, and J. W. van Kempen, Using column generation to solve parallel machine scheduling problems with minmax objective functions. *Journal of Scheduling*, Vol. 15, No. 6, 2012, pp. 801–810.
17. Whitley, D., A genetic algorithm tutorial. *Statistics and Computing*, Vol. 4, No. 2, 1994, pp. 65–85.
18. Reeves, C. R., A genetic algorithm for flowshop sequencing. *Computers & Operations Research*, Vol. 22, No. 1, 1995, pp. 5–13.
19. Van Laarhoven, P. J., E. H. Aarts, and J. K. Lenstra, Job shop scheduling by simulated annealing. *Operations Research*, Vol. 40, No. 1, 1992, pp. 113–125.
20. Ishibuchi, H., S. Misaki, and H. Tanaka, Modified simulated annealing algorithms for the flow shop sequencing problem. *European Journal of Operational Research*, Vol. 81, No. 2, 1995, pp. 388–398.
21. Stützle, T. and R. Ruiz, Iterated Greedy. *Handbook of Heuristics*, 2018, pp. 547–577.
22. Jabbour, R. and I. H. Elhajj, SAF-PS: Starvation avoidance for priority scheduling. In *2008 5th International Multi-Conference on Systems, Signals and Devices*, IEEE, 2008, pp. 1–6.
23. Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, Vol. 2, 2015, pp. 62–72.