Scheduling Planned Hangar Maintenance

David Lotten (2544932)

Master Thesis

Vrije Universiteit

Master Thesis

Scheduling Planned Hangar Maintenance

Author:
David Lotten

Supervisors:

Vrije Universiteit:
Rob van der Mei
Bob Rink (second reader)

Transavia:
Bram Nicolai
Robert Timmermans

Transavia Hangar 5
Halim P.K.straat 5
Schiphol-Oost

Faculty of Science
Business Analytics
De Boelelaan 1081a
1081 HV Amsterdam

Friday 28th September, 2018

Preface

This thesis concludes the final part of the master program Business Analytics. The program offers courses in a nice mixture of mathematics and computer science and many cases learning to apply both. I had a great time applying this mixture by modelling an ILP for a real life size scheduling process and the subsequent coding and testing. The best part was the process of experimenting with the final model and the heuristics, during my internship at Transavia's Fleet and Technical Services department, answering the question to what extent mathematical models can be used to create a schedule for the planned hangar maintenance for Transavia's fleet of aircraft.

I would like to thank Rob van der Mei, my VU supervisor, for his many suggestions helping me find new directions and our enjoyable bi-weekly conversations, and Bob Rink for being willing to take the role of second reader.
At Transavia I owe gratitude to Bram Nicolai for giving me the opportunity to perform this research at Transavia's Fleet and Technical Services, for deeming my weekly progress reports the most fun meetings of the week and his enthusiasm about the subject and mathematics in general and to Robert Timmermans for listening to me going on about the theoretical problems and resolving all practical problems I encountered.

David Lotten
September 2018

Management summary

Problem

Aircraft need regular maintenance to retain their airworthiness. At Transavia aircraft need to receive planned maintenance every six to eleven weeks. The current scheduling process is a time consuming, manual process involving three different departments. As a consequence, a selected schedule is made feasible by adjusting resources instead of finding a feasible schedule for a cost effective set of resources. This causes a waste of resources like hangar space and manpower.

Solution

This paper shows that mathematical modelling and heuristics can be applied effectively to provide a good schedule for planned hangar maintenance in reasonable time.

Approach

A mathematical model was created for the scheduling process which can be solved using Integer Linear Programming. The purpose of the model is to find a schedule which minimizes the different costs: labour, overhead and aircraft unavailability. Solving ILP's is NP-hard, meaning that solving realistic instances takes more time than is practical for day to day use.
Therefore heuristics were developed. The performance of these heuristics was evaluated. This was done by comparing both the objective value and the runtime of the exact solution to the objective value and runtime of the heuristics.

Results

The best heuristic creates schedules for small instances with a cost of less than 8% higher than the optimum, in less than one second. However, as the instance size increases, the heuristic performs increasingly worse compared to the optimal solution.

Conclusions and recommendations

The results are promising, but not yet good enough for practical use. The approach should be used to improve and evaluate heuristics, resulting in schedules with a near optimal use of resources, that are created in practical time.

Contents

1 Introduction

In this introduction, the background is given for this research. First the problem will be described. Then the purpose of the research described in this paper, which will be translated to a research question. Next some more detail will be given about planned aircraft maintenance at Transavia and finally the structure of the report will be given.

1.1 Problem description

This study was performed at Transavia, a Dutch low cost airline. Transavia operates a single type fleet of at most 45 Boeing 737NG aircraft during the high season. Aircraft need to receive regular maintenance. The main purpose of maintenance is to ensure the continued airworthiness of the aircraft. Aircraft manufacturers prescribe how often parts of an aircraft need to receive maintenance in a maintenance planning document (MPD). Government authorities oversee the adherence to the MPD.
Planned maintenance needs to be scheduled: planners need to decide which aircraft registration to bring in at what date and which maintenance tasks need to be performed. Scheduling planned aircraft maintenance in a cost effective way is a complex problem, even for an airline that is relatively small on a world scale, like Transavia.

Complications

The first complication comes from the fact that there are no fixed work packages: no two hangar visits are the same, in terms of number of tasks or their durations. In figure 1.1 the planned maintenance workload for five Transavia aircraft over the year 2017 is displayed based on their actual usage (the unique aircraft identifications like PHXRV are called registrations).

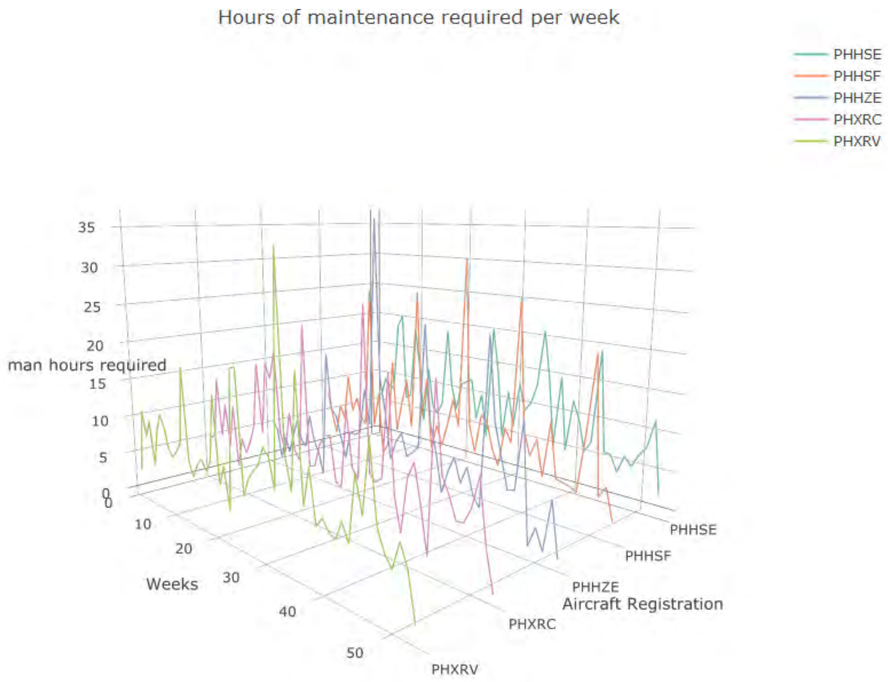

Figure 1.1: Distribution of the duration of hangar maintenance tasks

From this picture it is obvious that no clear patterns are discernible: not within one aircraft, nor between the aircraft.

The differences in what maintenance tasks become due over time is caused by several factors: the interval counts start at the delivery date of the aircraft, which is different for each aircraft. Tasks become due after a pre-specified number of cycles (one take off and one landing), calendar days or flight hours. The main driver for the planned hangar maintenance tasks for a low cost airline are the flight hours: the flight hours reach their threshold first. Flight hours depend on the destinations the aircraft flies to, whether the aircraft is being kept in reserve or not, or needs unplanned maintenance. All of these may change within the day and over the days. Furthermore, whenever a maintenance task is performed for some other reason, the interval count for that task is reset to zero. In short, every time the schedule is made or redone, the workload that has to be planned, has changed. The essence of the problem is that schedules can not be reused and once a schedule is made, it becomes outdated quickly.

The second complication follows from a combination of two things. One is that the number of possible combinations for bringing in the aircraft, the number of technicians to assign, the duration of the hangar visit are large with not enough restrictions to make choices easy. The other is that the scheduling process is currently done manually, in four stages by three different departments. As a consequence, each subsequent step in the process tries to make feasible the decisions of the previous step. It takes hours to days to create or update a schedule.

No data is available about how good or bad the current process is performing. However, several examples were given during interviews where the scheduling process resulted in poor results. Two examples are technicians going idle for a good part of a shift and an aircraft having to be brought in again after the weekend because it was impossible to finish all planned tasks in the planned time.

Costs

For a better understanding of the problem, the cost drivers need to be clear. Four cost drivers have been identified.

The first is maintenance interval loss. This measures how much of the allowed interval for performing a maintenance task is actually used. For example: if a task has to be done every ten weeks, and it is done in the fifth week, half the interval is wasted. In terms of costs, the task is done twice as often as it needs to be done. This value can be translated into a euro value, which will be explained in detail later.

The second is labour: how many technicians are necessary to perform all planned tasks? This can be easily translated into a euro value, by using the average salary as a metric.

Third is aircraft unavailability. The time an aircraft is in maintenance, it can not be used commercially, i.e. earn money by bringing passengers to their destinations. By dividing the Transavia's yearly profit by the number of flight hours over the same period this can be expressed in euros as well.

Fourth is the cost of overhead: the time and manpower required to bring an aircraft from commercial use into maintenance. It takes two hours to tow an aircraft from the platform to the hangar and another two to bring it back once maintenance is finished. This can also be expressed in euros.

1.2 Goal of the research

The purpose of this research is to develop a mathematical model. The model should be able to integrate the four steps of the process and to deliver a feasible schedule for planned aircraft maintenance in reasonable time, at optimal costs. It should return a solution that shows what aircraft to bring in at what time, how many technicians are required per shift and which tasks to perform in what order. All restrictions should be adhered to, the most important being that no maintenance is planned after the due date and it should minimize the costs.
The research question can be formulated as:

To what extent can mathematical modelling be used to create a "good" schedule for planned hangar maintenance, for the Transavia fleet of aircraft, within reasonable time?

1.3 Approach

To conduct this research the following steps were taken:

1. Conduct interviews with planners and review documentation
2. Develop mathematical model
3. Program and test model in R
4. Develop heuristics
5. Obtain numerical results
6. Draw conclusions

Structure of the paper

The remainder of this paper is divided in four parts. In chapter 2 a more detailed description will be given of the planning process, based on the interviews and documentation. In chapter 3 a detailed description will be given of the model and the validation of its implementation in R. Chapter 4 will examine the characteristics of the cost function of the model near the optimum, giving insights in the possibilities for heuristics. In chapter 5 a heuristic and several improvements on that heuristic and their performances will be described and discussed. The final chapter, chapter 6 ends with the overall conclusions and ideas for improvement that were found during the research.

2 The maintenance planning process

Transavia flies all routes from one of its home bases and back: the second leg of a route for Transavia is always back to a home base. When no special circumstances occur, like defects, or adverse weather conditions that ground an aircraft at an outstation, all aircraft end the operating day at their home bases. Transavia has its home bases in the Netherlands, in Amsterdam, Rotterdam, Eindhoven and Groningen. In Groningen no maintenance facilities are present, and in Eindhoven and Rotterdam line (platform) maintenance is possible. Transavia owns a two bay hangar, located at Amsterdam, Schiphol Airport.

2.1 Aircraft maintenance

Aircraft need to receive maintenance before the end of a prescribed maintenance interval to guarantee their continued airworthiness. Manufacturers prescribe what maintenance tasks have to be performed at what intervals. Government authorities oversee the adherence to these maintenance requirements. The intervals for the fleet Transavia operates, are determined by the Boeing Maintenance Planning Data Document (MPD). Transavia owns and leases a fleet consisting solely of Boeing 737's.

2.2 Maintenance intervals

Intervals are determined in **flight hours** (the time that elapses between a consecutive take off and landing), **cycles** (one take off and landing constitutes a cycle) or **days** (calendar days, regardless of the usage of the aircraft) or a combination of these three, whichever comes first. This gives maintenance planners freedom to tailor the maintenance schedule to the specifics of the operational needs of their airline. The specified maintenance tasks, usually visual inspections, cleaning, lubrication or replacement of a part or component, need to be performed before the end of any interval is reached.

2.3 Planned hangar maintenance definition

The prescribed maintenance can be classified in three major groups. Many tasks have a high frequency and have to be performed pre-flight, daily, weekly or monthly. These tasks do not require the special equipment or tools that are available only in the hangar. These tasks are performed on the platform. Other tasks are performed every two years or even longer, may take as long as a month to complete. These tasks are defined as heavy maintenance (HMX) and are outsourced as well as complete overhauls and structural checks of the aircraft that need to be done every 6 - 8 years. All tasks with intervals longer than the monthlies and shorter than HMX are regular "hangar" maintenance and in the scope of this research.

The hangar is located at Amsterdam Schiphol Airport. Hangar maintenance tasks have an interval of 720FH, 1500 cycles or 180 days, or longer. Like many low cost airlines Transavia makes intensive use of its fleet, operating the aircraft for up to 20 hours a day, every day. This causes the **flight hours** maintenance interval to be the first limit to be reached. Depending on the routes the aircraft are mainly used on, 720FH translates to an interval between 6 and 11 weeks. Planned hangar maintenance is done in four "hangar days": a period of 96 hours, consisting of twelve eight hour shifts, starting at Monday afternoon (15:00h) ending Friday after the morning shift at 15:00h. No planned hangar maintenance is done during the weekends.

No hangar maintenance is planned to be carried over the weekend, meaning Friday 15:00h all planned tasks need to be completed and the aircraft is returned to operations. On average, one aircraft needs to be brought to the hangar every working day and about 100 tasks need to be performed with durations ranging from a few minutes up to 40 hours. Observe figure 2.1 for a histogram of the distribution of durations of the line and hangar maintenance tasks.

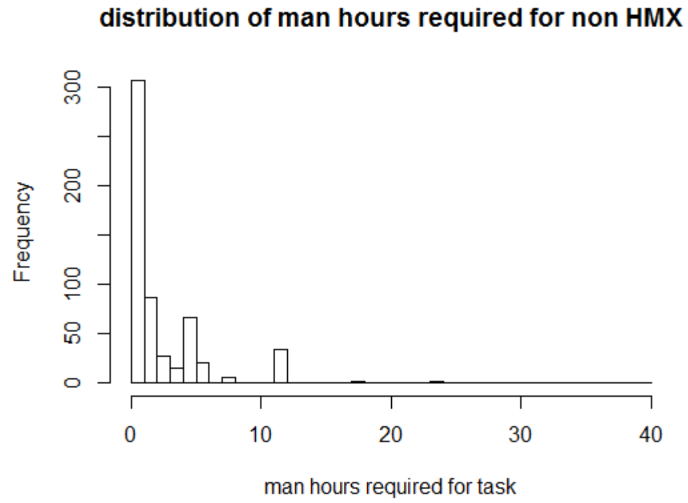

Figure 2.1: Distribution of the duration of hangar maintenance tasks

2.4 Maintenance planning process

The high level planning for how much time is available for maintenance is done two seasons in advance. Commerce plans what destinations to fly to and how often. Landing slots are scarce, so major changes in routes will not occur often. In consultation with fleet and maintenance part of the fleet is not assigned to maintenance, but to commercial operations. Currently, 96 hours per week are reserved for planned hangar maintenance. Details like which registration (i.e. the number of the specific aircraft) flies to what destination are decided much later and finalized 48 hours before operation. Assignments of registrations to a route may even be changed during operation due to unforeseen circumstances, like delays or defects.

The main steps of the process are shown in figure 2.2.

The maintenance planners first finalize their planning of which registration has to come to the hangar for planned maintenance five weeks in advance. Then this schedule is handed over to the manpower controllers who ensure enough technicians are available. After the manpower controllers, the workload controllers create the work packages: the collection of tasks cards that need to be done. One week before execution of the maintenance the workload controllers finalize the work packages, by adding known defects that were deferred. Deferred defects could be any number of small things that do not interfere with safety, like a broken light bulb in the cabin or a squeaky lavatory door. Finally, on the hangar work floor task cards are distributed among the technicians and detailed experience is added to avoid conflicting situations. Conflicting situations can occur when tasks that require power to be on are planned simultaneously with those that require the power to be off, or planning more technicians in an area than fit, for instance only two technicians can simultaneously work in a wheel well.

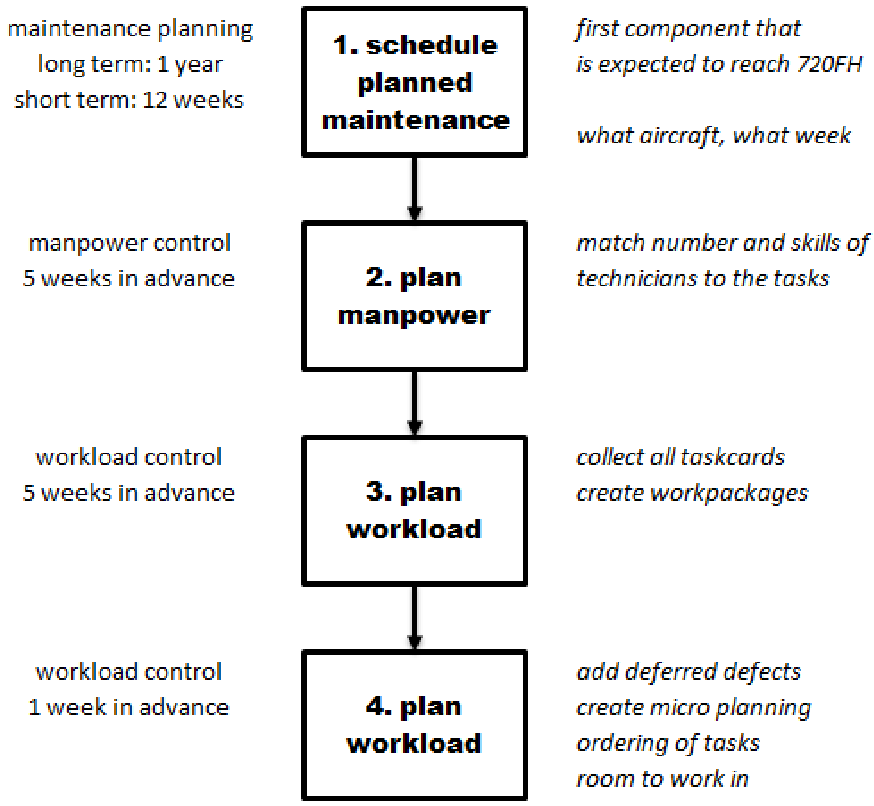

Figure 2.2: Flowchart of the scheduling process

2.5 Maintenance skills and technicians

Except for some specific tasks, that require specific certified skills to perform, most tasks may be performed by any technician. Signing off on the correct execution of a task does require specific authorizations and certifications. Transavia employs a core of certifying technicians, the formal term shortened in this document to technicians, on its payroll. They are supplemented by externally hired technicians as planned maintenance demand increases. Most technicians are deployed not in the hangar, but on the line and the two other home bases, Eindhoven and Rotterdam. On average twenty times as many aircraft are not in the hangar but in use, requiring the smaller maintenance tasks, like the preflight inspections, daily, weekly and monthly tasks. In the hangar, usually no more than twelve technicians are planned to work on the same aircraft simultaneously.

2.6 Maintenance data

All the maintenance data is stored in an application called TRAX. All tasks in the maintenance program are registered in TRAX, as well as each task card, which contains detailed procedures and working instructions how to perform all maintenance tasks. Also the status of each task is kept in TRAX. Information about aircraft usage, the Flight Hours and Cycles, is updated in TRAX almost real time: every time an aircraft releases the brakes, takes off or lands, comes to a full stop at the gate or platform an automated signal is send by the aircraft, via a system called ACARS, which is within minutes uploaded in TRAX.

3 Model

In the previous chapter background and detail was given about the maintenance process and the way it is scheduled. That information is now used in this chapter to formulate a model that serves the purpose of this research. First an overview is given of literature on the modelling of aircraft maintenance scheduling. Next, in section 3.2 the model that is used in this research is described, with its cost function and the model assumptions. Finally, a description is given of how a sanity check was done to verify the correct numerical outcomes of the model.

3.1 Literature on aircraft maintenance scheduling

Airline maintenance scheduling is a much studied topic. Much of this research is focused on finding a flight schedule such that the aircraft are at a maintenance location when maintenance is due: the aircraft maintenance routing problem (AMRP), see [2] or [9]. Finding the minimum number of maintenance bases for given routes is the topic of [5] and finding the best locations for maintenance bases is described in [4].

Many papers, even more recent ones, assume maintenance intervals to be grouped by the manufacturer in A, B, C and D checks: maintenance processes that have to be performed after a predetermined number of days. In [8] the authors give an excellent summary of shortcomings of earlier work: *"For the most part, the theoretical literature and common AMRP practices have focused on two policy types. The first ignores the maintenance requirements of individual aircraft and only considers generic maintenance routes, e.g., cyclic or n-day rotation policy [16], [18], [21], [24], [26]. The second schedules the periodic maintenance activities and plans the route of individual aircraft on a day-to-day basis [15], [25]; Keysan et al., 2010; [14], [23], a policy favoring "feasibility" over "optimization". The majority of past studies simplify maintenance requirements by only considering the more frequently occurring maintenance tasks (type A). In practice, each aircraft has many maintenance tasks, with over 50 different checks, which must be done on a regular basis during the life cycle."*

In short none of the literature found about the topic was still up to date, applicable to Transavia or took in to account other cost factors than used in this paper. For instance, the objective in [4] is to minimize inventory costs of the maintenance facilities, and in [9] the objective is to minimize the collective costs of all tasks in an "A" or "B" check, which they assign a fixed price per city and aircraft as well as the cost of assigning different aircraft types to specific routes.

3.2 Model description

In this section the model that creates a schedule for the planned maintenance for a fleet of aircraft will be made, based on the information in chapter 2.

A schedule needs to be created for a time horizon of T time units. The time units t are numbered $1, 2, ..., T$. In the model the set of time units t in the time horizon T are denoted by $\mathcal{T} = \{1, 2, ..., T\}$, to make it possible to write $\sum_{t \in \mathcal{T}}$ instead of having to write $\sum_{t=1}^{t=T}$. The time unit t can be any size, like minutes or hours.

The time units in the set $\mathcal{T}$ are also subdivided in shifts and weekends. The number of shifts is denoted by $SHIFTS$. The shifts s are numbered $1, 2, ..., SHIFTS$. All shifts are of equal length denoted by t_per_shift. Let $\mathcal{S}$ denote the set of shifts s, $\mathcal{S} = \{1, 2, ..., SHIFTS\}$ and let $shift_s$ denote the set of time units t in shift s. Weekends are assumed to have length of 1 time unit, since no planned maintenance is ever scheduled in the weekends. Let w_end denote the set of t's that indicate the weekend.

To be able to distinguish between the costs at day time and night time, shifts can be either day shifts or night shifts. Let day denote the set of time units t in the day shifts and $night$ the set of time units in the night time.

In table 3.1 an example is given for a time horizon $T = 17$ where $w\_end = \{17\}$, the number of shifts $SHIFTS = 4$, the shift lengths are $t\_per\_shift = 4$, the day shifts $day = \{shift_1, shift_3\}$ and $night = \{shift_2, shift_4\}$ and $shift_1 = \{1, 2, 3, 4\}$ and $shift_2 = \{5, 6, 7, 8\}$, etc.

s	t				shift type
1	1	2	3	4	day
2	5	6	7	8	night
3	9	10	11	12	day
4	13	14	15	16	night

Table 3.1: Example of the subdivisions of the time horizon T

The size of the fleet of aircraft is K aircraft; so aircraft are numbered $1, 2, ...K$. Let $\mathcal{K} = \{1, 2, ..., K\}$ denote the set of aircraft k. Each aircraft has a list of N tasks i, numbered $1, 2, ..., N$. The tasks in the list for each aircraft can be different. Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of tasks i.

There are as many line places as there are aircraft, and there are h hangar places. In total M is the number of maintenance locations, where $M = K + h$ and the locations are numbered $j = \{1, ..., M\}$ and let $\mathcal{M} = \{1, 2, ..., M\}$ denote the set of maintenance locations.
The duration of each task is assumed to be deterministic and the duration of task i is denoted as D_i.
The due dates for each task can have a different value for each aircraft. The due date for task i of aircraft k is denoted by $\delta_{k,i}$, and these are given. The due dates are assumed to be deterministic and have to lie within $\{1, 2, ..., T\}$. As some tasks need more than one technician to perform it, man_req_i is defined as the number of technicians required to perform task i. Each task i has a given prescribed maintenance interval I_i. The assumption is made that $I_i > T$: the model does not handle tasks that repeat within the time horizon T. Maintenance can be done on the line (at the platform) if that is allowed or in the hangar which is always allowed, for each task this is given, let:

$$L_i = \begin{cases} 1 & \text{if task } i \text{ is allowed to be scheduled at the platform} \\ 0 & \text{else} \end{cases}$$

task i may be performed on the line $L_i = 1$ or not: $L_i = 0$.

Also define the decision variable:

$$x_{k,i,j,t} = \begin{cases} 1 & \text{if for aircraft } k \text{ task } i \text{ is scheduled at location } j \text{ at interval } t \\ 0 & \text{else} \end{cases}$$

and the support variables:

$$s_{k,i,j,t} = \begin{cases} 1 & \text{if for aircraft } k \text{ task } i \text{ is being executed at location } j \text{ at interval } t \\ 0 & \text{else} \end{cases}$$

$$b_{k,j,t} = \begin{cases} 1 & \text{if aircraft } k \text{ is located at location } j \text{ at interval } t \\ 0 & \text{else} \end{cases}$$

$$Z_{k,j,t} = \begin{cases} 1 & \text{if aircraft } k \text{ was at location } j \text{ at } t-1 \text{ and not at } t \\ 0 & \text{else} \end{cases}$$

Also define a support variable:

$$m_{j,shift_s}$$

The variable $m_{j,shift_s} \in \mathbb{N}$ is the number of technicians needed at shift $shift_s$.

Also some physical constraints and company policies that form constraints exist:

1. no task may be scheduled after its due date;
2. at one location there can be a most one aircraft at the same time;
3. one aircraft can be at most at one location at the same time;
4. no planned hangar maintenance may be performed during the weekend;
5. all tasks for all aircraft need to be scheduled during the time horizon.

Also the cost functions need to be defined. There are functions for the four identified costs: *maintenance interval loss, overhead, labour* and *unavailability.*

Maintenance interval loss costs

Maintenance interval loss costs, are the costs for performing a maintenance task earlier than its due date. The moment the task is done, the interval for that task is reset. So each time unit the task is performed before its due date, is wasted. This waste is what needs to be calculated. Note that the maintenance interval I_i for task i may be much larger than the length of the schedule time horizon. The interval for task i may, for example, be 120 time units and the due date $\delta_{k,i}$ for task i for aircraft k in the schedule time horizon, can be 12 time units. This means task i for aircraft k was reset (last performed) 108 time units before the start of the schedule time horizon. The interval I_i is given, the due date, $\delta_{k,i}$ is given and t depends on the decision that is made. To calculate the correct costs, it is necessary to determine how much more often the task would be performed, than the minimum of exactly once, if it scheduled exactly on the due date. $I_i - \delta_{k,i}$ is how much time has passed since the last reset of the interval. For example: if the interval is 10, and the due date is 8, then 2 time units have passed since the last reset of task i of aircraft k. Added to that is the decision t when the task i for aircraft k is scheduled. Suppose that $t = 5$ then the task is scheduled at 70% of the interval:

$$\frac{I_i - \delta_{k,i} + t}{I_i} = \frac{10 - 8 + 5}{10} = 0.7$$

If the task is scheduled at 70% of its allowed interval, it is done $\frac{1}{0.7} \approx 1.4$ times instead of once in the same time horizon. So the waste, i.e. the cost of maintenance interval loss in this case is 0.4 times the cost of performing the task. So after dividing by I_i on both sides of the fraction we get:

$$\frac{1}{1 - \left(\frac{\delta_{k,i}+t}{I_i}\right)} - 1$$

It is assumed that a only fixed amount for labour costs is calculated for performing a task. The cost of replacing parts is assumed to be negligible. Thus the outcome has to be multiplied by the cost of labour C_1, times the duration of the task D_i and the number of technicians, man_req_i to perform the task. This results in the following statement for the *interval loss*:

$$C_1 * \left(\frac{1}{1 - \frac{(\delta_{k,i}-t)}{I_i}} - 1\right) * D_i * x_{k,i,j,t} * man\_req_i$$

Overhead (or transition) costs

Overhead cost is more easily explained: every time the aircraft is moved to or from a maintenance location, $Z_{k,j,t} = 1$ is added to the costs, times some weight C_2 and a weight that depends on the location that is moved to or from: c_j. The necessity of the latter weight can be understood when considering that the overhead for maintenance at the line is on average 1 hour, while the total overhead for hangar maintenance is 4 hours. This results in the following statement for the overhead costs:

$$C_2 * Z_{k,j,t} * c_j$$

Labour costs

The third cost factor is labour. Some weight C_{3shift_s} that depends on the shift in order to differentiate between night and day shift. The support variable $m_{j,shift_s}$ calculates the maximum number of technicians that were deployed during a shift. In the cost function we need to multiply this by the length of the shift, resulting in the following term for the *labour costs*:

$$C_3(shift_s) * m_{j,shift_s} * t\_per\_shift$$

Unavailability costs

The fourth cost factor is unavailability. This is some weight, $C_4(t)$ that depends on the time of day, applied to the duration that the aircraft are in planned maintenance. The t's that define the night rate for unavailability need not be equal to t's that are in the night shifts:

$$C_4(t) * b_{k,j,t}$$

The weight is variable over time, to make a distinction between the unavailability costs at night (lower) and day.

The data for a fleet of aircraft and their maintenance tasks that become due in the time horizon, and given the time horizon and its subdivisions in shifts and weekends as well as available maintenance locations need to be given. It is then possible to assign the tasks for each aircraft, to a maintenance location at a specific time. The objective is to minimize the sum of the costs for *maintenance interval loss*, *overhead*, *labour* and *unavailability*, while adhering to the mentioned constraints. A schedule with minimal costs can be obtained, by solving the integer linear program (ILP) on the next page. An example of how the model works, is given at the bottom of page 15.

$$\text{minimize} \quad \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\sum_{t\in T} C_1 * \Big(\frac{1}{1-\frac{(\delta_{k,i}-t)}{I_i}} - 1\Big) * D_i * x_{k,i,j,t} * man\_req_i \tag{3.1}$$

$$+ \sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{M}}\sum_{t\in T} C_2 * Z_{k,j,t} * c_j \tag{3.2}$$

$$+ \sum_{j\in\mathcal{M}}\sum_{shift_s\in\mathcal{S}} C_3(shift_s) * m_{j,shift_s} * t\_per\_shift, \tag{3.3}$$

$$+ \sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{M}}\sum_{t\in T} C_4(t) * b_{k,j,t} \tag{3.4}$$

$$\text{subject to} \quad s_{k,i,j,t} = \sum_{q=\max\{1,t-D_i+1\}}^{t} x_{k,i,j,q} \quad k\in\mathcal{K}, \quad i\in\mathcal{N}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.5}$$

$$b_{k,j,t} \leq \sum_{i\in N} s_{k,i,j,t}, \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.6}$$

$$Nb_{k,j,t} \geq \sum_{i\in N} s_{k,i,j,t}, \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.7}$$

$$\sum_{j\in\mathcal{M}} b_{k,j,t} \leq 1, \quad k\in\mathcal{K}, \quad t\in\mathcal{T} \tag{3.8}$$

$$\sum_{k\in\mathcal{K}} b_{k,j,t} \leq 1, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.9}$$

$$Z_{k,j,t} \leq b_{k,j,t} + b_{k,j,t-1}, \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.10}$$

$$Z_{k,j,t} \geq b_{k,j,t} - b_{k,j,t-1}, \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.11}$$

$$Z_{k,j,t} \geq b_{k,j,t-1} - b_{k,j,t} \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.12}$$

$$Z_{k,j,t} \leq 2 - b_{k,j,t} - b_{k,j,t-1}, \quad k\in\mathcal{K}, \quad j\in\mathcal{M}, \quad t\in\mathcal{T} \tag{3.13}$$

$$\sum_{t\in W\_end} s_{k,i,j,t} = 0, \quad k\in\mathcal{K}, \quad i\in\mathcal{N}, \quad j\in\mathcal{M} \tag{3.14}$$

$$m_{j,shift_s} \geq \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}} s_{k,i,j,t} * man\_req_i, \quad j\in M, \quad t\in shift_s, \quad shift_s\in\mathcal{S} \tag{3.15}$$

$$\sum_{t\in\mathcal{T}}\sum_{j\in M} x_{k,i,j,t} = 1, \quad k\in\mathcal{K}, \quad i\in\mathcal{N} \tag{3.16}$$

$$\sum_{t=\delta_{k,i}+2-D_i}^{\mathcal{T}} x_{k,i,j,t} = 0, \quad k\in\mathcal{K}, \quad i\in\mathcal{N}, \quad j\in\mathcal{M} \tag{3.17}$$

$$\sum_{t\in\mathcal{T}} x_{k,i,j,t} \leq L_{i,j} \quad k\in\mathcal{K}, \quad i\in\mathcal{N}, \quad j\in\mathcal{M} \tag{3.18}$$

$$x_{k,i,j,t}, \quad s_{k,i,j,t}, \quad b_{k,j,t} \text{ and } Z_{k,j,t} \in \{0,1\}, \quad m_{j,shift_s} \in \mathbb{N} \tag{3.19}$$

Explanation

There is one decision variable, $x_{k,i,j,t}$ which is the decision to schedule task i, for aircraft k on location j at time t, or not.
The support variable $s_{k,i,j,t}$ keeps track of whether task i scheduled for aircraft k at location j at time t is still being performed at some time q or not. This relation between the support variable $s_{k,i,j,t}$ and the decision variable $x_{k,i,j,t}$ and the duration of the task D_i is given in equation (3.5).

The second support variable, $b_{k,j,t}$ keeps track of the location of the aircraft. This serves two purposes. The *first purpose* is to model the physical constraints mentioned earlier: no more

than on one aircraft can occupy some maintenance location at some time and one aircraft can not be in maintenance at more than one location on the same time.
The *second purpose* is to calculate the unavailability costs: if an aircraft is in maintenance at some time t, it is not available for commercial use. The relation between $s_{k,i,j,t}$ and $b_{k,j,t}$ is given in equations (3.6) and (3.7). These ensure that $b_{k,j,t} = 1$ when one or more tasks are being performed for aircraft k at location j at time t and 0 else.

The third support variable, $Z_{k,j,t}$ keeps track of location changes of the aircraft. Whenever an aircraft changes location, overhead (or transition) costs need to be calculated. Since $b_{k,j,t}$ is a binary variable, equation (3.4) for $Z_{k,j,t}$ can be written as

$$Z_{k,j,t} = |b_{k,j,t} - b_{k,j,t-1}|.$$

Check: if both $b_{k,j,t}$ and $b_{k,j,t-1}$ are equal to either 0 or 1 for some k, j, t there was no change to or from that location, for that aircraft, from time $t-1$ to time t, so $Z_{k,j,t} = 0$. If either $b_{k,j,t} = 1$ or $b_{k,j,t-1} = 1$, but not both, aircraft k did change location and then $Z_{k,j,t} = 1$.
The absolute value in the equation, which makes it a non linear equation, can be made linear by writing this as four equations to define the XOR relation: $Z_{k,j,t} = b_{k,j,t}$ xor $Z_{k,j,t} = b_{k,j,t-1}$, which are specified in equations (3.10) - (3.13).

The last support variable m_{shift_s} calculates the shift sizes. The size of a shift is equal to the maximum number of technicians that were deployed during any time unit of that shift. The maximum can be calculated by a linear equation by making the variable m_{shift_s} greater than or equal to the sum over all aircraft k and all tasks i of the number of tasks being performed $s_{k,i,j,t}$ multiplied by the technicians required man_req_i at location j during all the t in some shift $shift_s$. Since, this is a minimization problem, no value greater than the maximum will be chosen. This is specified in equation (3.15).

Equations (3.17), (3.8), (3.9), (3.14) and (3.16) model physical and company restrictions 1 through 5 described earlier on page 12.

The resulting integer linear problem has 2KNMT+2KMT+M*SHIFTS variables and KNMT+6KMT+KT+2MT+2KNM+KN constraints.

Example

To illustrate the model, an example is given. Table 3.3 gives the data for three aircraft with ten tasks each. The time horizon T is 25. Given are $SHIFTS = 6$ shifts, of $t\_per\_shift = 4$ time units length. Time unit $t = 25$ indicates the weekend w_end. There is one hangar space and three line spaces available. The values for the costs for this example are given in table 3.2.

Table 3.2: Given costs for the example

Cost factor	**day**	**night**	**either**
C_1 interval loss			1.2
C_2 overhead			5
C_3 labour	1	1.2	
C_4 unavailability	7	4.5	
c_1 overhead hangar			1
c_2 overhead line			0.25

The optimal schedule is visualized in figure 3.1.

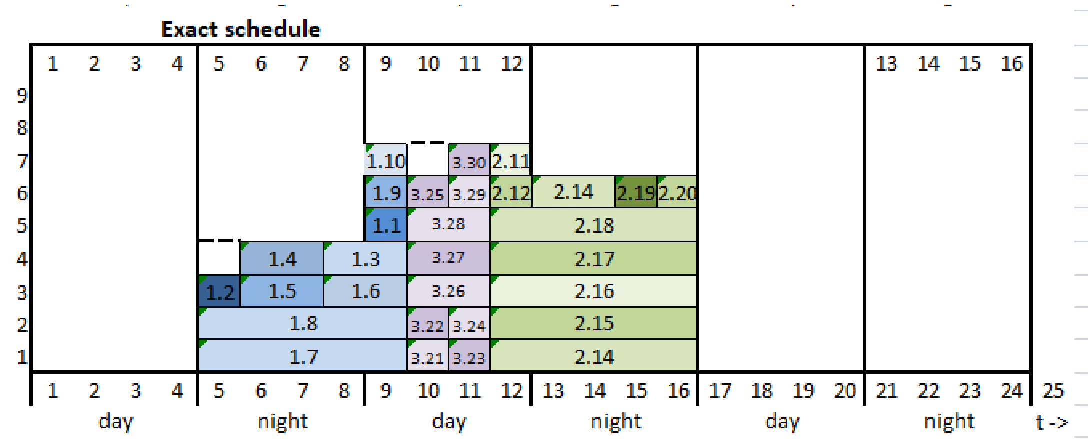

Figure 3.1: Solution schedule of the example.

The time units t in the time horizon T are on the x-axis, as well as the indication whether a shift is a day or night shift. The y-axis indicates the shift size. The rectangles are the tasks. The numbers in the rectangles indicate the aircraft number followed by the task number as given in table 3.3, e.g. 1.1 is aircraft 1, task 1. The colours have no specific meaning, but the same basic colour was used per aircraft. The width of the rectangle is the given duration D_i of the task and the height the given required number of technicians man_req_i.

Table 3.3: Given data for the example

aircraft k	taks nr i	due date $\delta_{k,i}$	man_req_i	line allowed? L_i	Duration D_i	interval I_i
1	1	16	1	0	1	160
1	2	16	1	0	1	160
1	3	16	1	0	2	160
1	4	16	1	0	2	160
1	5	16	1	0	2	160
1	6	16	1	0	2	160
1	7	16	1	0	5	160
1	8	16	1	0	5	160
1	9	20	1	1	1	120
1	10	20	1	1	1	120
2	11	16	1	0	1	160
2	12	16	1	0	1	160
2	13	16	1	0	2	160
2	14	16	1	0	5	160
2	15	16	1	0	5	160
2	16	16	1	0	5	160
2	17	16	1	0	5	160
2	18	16	1	0	5	160
2	19	20	1	1	1	120
2	20	20	1	1	1	120
3	21	16	1	0	1	160
3	22	16	1	0	1	160
3	23	16	1	0	1	160
3	24	16	1	0	1	160
3	25	16	1	0	1	160
3	26	16	1	0	2	160
3	27	16	1	0	2	160
3	28	16	1	0	2	160
3	29	20	1	1	1	120
3	30	20	1	1	1	120

The solution of the ILP for this example instance is a vector of length 2KNMT+2KMT+M*SHIFTS = 864.

3.3 Validation experiment of the implementation

In this section the method and results of validating the implementation of the model in the code are described. The Integer Linear Program that was formulated in the previous section, was coded in 'R', the open source statistical software and solved using the Gurobi solver. Both choices are a matter of personal preference, but Gurobi being commercial grade and free for students did influence the decision.
Both the model and the code are subject to human error. To validate both, a series of experiments was conducted. First a description of the instance is given and the set up of the experiments. Then the results of the validation experiments are displayed, supported by the resulting schedule when an interesting change occurred. Each schedule and its correctness are commented on directly.

Set up and data for the validation experiments

The basic idea for the validation was to create one small, simplified instance, in order to be able to easily verify the resulting solutions by hand. Then for that one instance, each of the cost parameters in turn were varied and the solution checked for correctness. If there is a mistake in the design of the model or the coding in R, the presence of the error could be found easily.

The described set up means all parameter values were kept fixed throughout the experiments except the values for the cost factors, *interval loss costs* ($C1$), *overhead costs* ($C2$), *labour costs* ($C3(shift_s)$) and *unavailability costs* ($C4(t)$). The value for each of the cost factors was varied stepwise from zero to one, while keeping the sum of the cost factors at one. This resulted in a series of 44 experiments where the weights of all costs were varied relative to the other cost values. To clarify the set up, an example is given in table 3.4 of the cost factor parameter settings is given for the eleven experiments varying of cost factor $C1$, interval loss costs. The

	experiment number										
cost factor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
C1	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
C2	0.33	0.30	0.27	0.23	0.20	0.17	0.13	0.10	0.07	0.03	0.00
C3	0.33	0.30	0.27	0.23	0.20	0.17	0.13	0.10	0.07	0.03	0.00
C4	0.33	0.30	0.27	0.23	0.20	0.17	0.13	0.10	0.07	0.03	0.00

Table 3.4: Cost parameter set up for varying C1

costs were set up **time independent**, so no distinction was made between day shift and night shift costs or day or night unavailability costs. This reduced the number of tests and the correct performance of the model for these aspects was easily checked separately. In the remainder of this section the time indices $shift_s$ and t are left out in the notation of $C3(shift_s)$ and $C4(t)$.

Instance data

A simplified, small instance was examined for validating the ILP and the code, using the following data:
Table 3.5 gives the data for $K = 2$ aircraft k, with $N = 2$ tasks i each. The time horizon T is 18. Given are $SHIFTS = 4$ shifts, of $t\_per\_shift = 4$ time units length. The set of time units $\{9, 18\}$ indicate the weekend w_end. There is one hangar space, location 1 and one line space, location 2 available. Several tests were also performed with $K = 1$ and $K = 3$ and other data values adjusted accordingly, but not as extensively.

Table 3.5: Given data for the example

aircraft k	task nr i	due date $\delta_{k,i}$	man_req_i	line allowed? L_i	Duration D_i	Interval I_i
1	1	8	1	1	5	40
1	2	17	1	1	3	40
2	1	8	1	1	5	40
2	2	17	1	1	3	40

The task durations, D_i were chosen such that both tasks can be performed sequentially in one week. The interval length for the tasks, I_i was chosen at a value of 40 arbitrarily, any value greater than 18 sufficed.
The due dates of task i of aircraft k were chosen such that for both aircraft the same tasks are due with the same due date. With this choice of due dates we would like to observe that if the weight of the costs of interval loss is increased enough, relative to the other costs, the

model schedules task 2 separate from task 1. The setting for $L_{i,j}$: whether task i is allowed to be done at location j, was set to 1 for all tasks: all tasks may be done at either locations, but only one (cheaper) line space is made available. The factor for overhead costs for the transfer to the hangar was set four times as high as the overhead at the platform, so $c_j = (1, 0.25)$, for $j = 1, 2$.

Results and discussion varying interval loss costs (C1)

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.2: Schedule $C1 = 0, \ C2 = C3 = C4 = 0.33$

C1 = 0.0

Since no cost is incurred for advancing a task before its due date, the model is expected to minimize the sum of the other three costs. Both aircraft have the lowest possible unavailability costs of 5 time units each. Both are moved to and from a location once. As a consequence of task 1 being due at $t = 4$ for both aircraft and only one spot being available at location 2 it is not possible to decrease the overhead costs by performing both maintenance jobs at location 1. Since task 1 takes longer than 1 shift, two shifts of at least one technician are needed. A lower value for the labour cost could be achieved by scheduling task 2 directly after completion of task 1. This would result in 2 shifts of 1 technician for both aircraft, equals $2 \times (1 + 1) = 4$ shifts of one technician, with a labour cost of $4 \times 0.33 = 1.33$, a reduction of 0.66. But, this would increase the unavailability from 5 intervals to 8 intervals for both aircraft, adding 0.33×2 aircraft $\times\, 3$ intervals $= 2$ to the unavailability costs. Meaning the given solution is correct.
Both tasks 1 could be planned starting anywhere on $t \in \{1, 2, 3\}$ and task 2 anywhere within the span of task 1 for the same costs, as interval loss costs are zero.

C1 = 0.1

C1=0.1	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.3: Schedule $C1 = 0.1, \ C2 = C3 = C4 = 0.30$

Increasing C1 to a value above zero, relatively small compared to the other cost values, will force the scheduling of the tasks as close to the due date as possible, but will not change the preference given to reducing the unavailability and not having to move the aircraft around more than the minimum.

$\mathbf{C1 \in \{0.2, 0.3, 0.4, 0.5\}}$

The total costs keep decreasing but the schedule does not change: planning task 2 at its due date would increase both the unavailability costs as the overhead costs and at their current weight these still exceed the interval loss costs.

C1 interval loss costs = 0.6

C1 = 0.6	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
loc 2 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0

Figure 3.4: Schedule $C1 = 0.6,\ C2 = C3 = C4 = 0.133$

At these weight settings for the costs, it is cheaper to schedule one task 2 at location 2 as close as possible to its due date, at $t = 15$, because the decrease in interval loss costs exceeds the increase in unavailability and overhead costs. Check: Using the formula in equation (3.1) and the given parameters, the interval loss costs for either task 2 can be calculated. These are 0.682 at $t = 2$ and 0.095 at $t = 15$: a decrease of rounded 0.588.
Scheduling task 2 at location 2 at $t = 15$ increases the overhead costs by an extra move into maintenance and one out of maintenance at the platform: $2 \times 0.133 \times 0.25 \approx 0.067$. Labour costs do not change: the technician performing task 2 is either needed in the second shift or in the fourth shift. The unavailability of the aircraft increases by 3 time units, adding $0.133 \times 3 = 0.4$ to the unavailability costs. So 0.588 cost reduction $> 0.4 + 0.067$ cost increase. The other task 2 is not scheduled at $t = 15$. The maintenance location is occupied by the other aircraft. The task could be scheduled in the hangar at $t = 15$, but for that location the increase in overhead would be $2 \times 0.133 \times 1 \approx 0.267$ instead of 0.067 for location 2. So if we increase the interval loss weight slightly further at the expense of the other weights, we would expect the solver to schedule task 2 of both aircraft at time $t = 12$.

C1 = 0.7

First is shown that the expectation is correct that task 2 of the second aircraft is scheduled at location 2, but at time $t = 12$. This does not happen at $C1 = 0.7$ but briefly at an interval around $C1 = 0.655$. A little further increase in the weight of the interval loss costs at the

C1 = 0,655	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Figure 3.5: Schedule $C1 = 0.655,\ C2 = C3 = C4 = 0.115$

expense of the other combined weights will lead to a solution with as little interval loss costs as possible and scheduling all tasks as close to their due dates as the other constraints permit. Further increase of the interval loss weight will not change the resulting schedule, only the value of the objective function will decrease, until it reaches its minimum value, since all tasks are scheduled as close to their due dates as possible and all other weights are zero.

C1 = 0.7	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Figure 3.6: Schedule $C1 = 0.7,\ C2 = C3 = C4 = 0.10$

Results and discussion varying Overhead costs (C2)

With overhead costs at zero, the availability costs dictates the outcome, making sure both tasks 2 are not scheduled outside the boundaries of tasks 1. The interval loss pushes the tasks 2 for both aircraft as close as the other constraints allow to their due dates.

C2 = 0.0

C2 = 0,0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.7: Schedule $C2 = 0.0,\ C1 = C3 = C4 = 0.333$

There is no reason for the model to change the schedule at any increase of the weight of $C2$, since the minimum unavailability solution leads to the minimum overhead costs solution. Only when $C2$ reaches 1.0 the other weights are zero. Then any schedule adhering to the other constraints that ensures both tasks for each aircraft are performed at the same location without interruption (overhead) has the optimal objective value. Differences are caused by the order in which the solver solves finds the optimal value, see figure 3.8.

C2 = 1	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.8: Schedule $C2 = 1.0,\ C1 = C3 = C4 = 0.0$

Results and discussion varying labour costs (C3)

C3 = 0.0

We would expect the model, again, to keep the unavailability costs as low as possible, scheduling the tasks as close to their due dates as possible, without affecting the unavailability and without

C3=0,0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.9: Schedule $C3 = 0.0,\ C1 = C2 = C4 = 0.333$

causing any location changes. Increasing the weight that is given to the labour cost, at some point we would expect the model to plan both tasks sequentially for both aircraft, giving precedence to labour over availability, but still avoiding location changes. Finally, when the labour cost weight is 1 and the other three are zero, any schedule with both tasks for one aircraft scheduled sequentially, resulting in two shifts of 1 technician per location results in the lowest objective value.

When the labour costs weight is increased to 0.6 it is cheaper to create a schedule that minimizes the labour costs, by spreading the work over two shifts with size of one technician. This is true for both aircraft, since this does not change the transfer costs. It is cheaper to perform task 1 at time $t = 1$ and tasks 2 at $t = 6$ than to move task 2 up to $t = 1$. Leaving task 1 scheduled at $t = 4$ results in minimal interval loss costs for task 1.

C3 = 1.0

When all costs are zero except $C3$ the model performs as expected, the constraints are adhered to, but any ordering of the tasks is equivalent.

C3 = 1.0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.10: Schedule $C3 = 1,\ C1 = C2 = C4 = 0.0$

Results and discussion varying unavailability costs (C4)

C4 = 0.0

C4 = 0,0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.11: Schedule $C4 = 0.0,\ C1 = C2 = C3 = 0.333$

When the unavailability is unimportant the model gives precedence to labour and transfer costs. It was already shown that, all weights being equal, the interval loss weighs the least, explaining the schedule that was created. Increasing the weight of the unavailability, the model is expected to plan task 2 inside the duration of task 1.

C4 = 0.1

C4 = 0,1	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16	t=17	t=18
loc 1 a/c 1 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 1 a/c 2 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.12: Schedule $C4 = 0.1,\ C1 = C2 = C3 = 0.30$

When some weight is given to the unavailability costs, the model finds the optimum by planning task 2 somewhere within the duration of task 1, giving precedence to the availability over labour cost. That can be explained because the unavailability increases with steps of 1 times its weight. With this instance the unavailability needs to increase with 3 units times its weight to reduce labour cost with one. Scheduling a task 1 time unit earlier than it is due is only a number between one and zero times its weight.
Within any solution that has the smaller task(s) inside the span of the largest task, the interval loss is minimized. This is expected to remain so, until the weight of the costs other than $C4$ are set to zero. At that point any solution is acceptable as long as task 2 is planned within the boundaries of task 1 for each aircraft and the other constraints are adhered to. The costs will steadily increase, as the schedule does not change, while the unavailability can not go below 5 time units per aircraft, since that is the longest task.

Conclusion

From the results of these experiments we can conclude that the model behaves as expected and for all instances the coded model returned the optimal schedule and calculated the objective values correctly.

4 Characteristics of objective values near the optimum

Solving an ILP is known to be NP-hard. For small instances the optimal solution can be obtained using a solver. However, for more realistic instances, that are not even as large as realistic instance would be for Transavia, it will take more time than is acceptable for day to day use to find an optimum, even using a commercial grade solver like Gurobi and maybe cloud computing power.

An attempt was made to create a more realistic instance, for a fleet of 5 aircraft, 30 tasks per aircraft, a T of 15 weeks (2520 hours) and 7 locations. While filling the first constraint (simple sparse) matrix in R with these variables, progress was observed and timed. The time needed to fill only the first matrix was estimated to take 300 hours.

Thus heuristics are needed to make the model usable. In order to find heuristics, it is important to know if the optimum is a single point with a value far lower than solutions that are less good, or perhaps many solutions exist with a value equal to or near the optimum. If a solution schedule, that differs slightly from the optimum schedule, results in an objective value that is much worse, it is important to find the optimum or a solution very close to the optimum. If on the other hand, a solution some distance from the optimum, still gives an objective value that differs a little from the optimum value, or even better if many solutions exist with the same objective value as the optimum, there is room for applying a heuristic that finds an acceptable value. The definitions of "worse" and "a little" should be determined by the business and would be some percentage of the optimal value.
Research question is:

To what extent do the cost functions of the model show characteristics that suggest the possibility of finding a good heuristic?

In this chapter a small instance is presented and all solutions for this instance are calculated and plotted, for different settings of the cost factors. The results are plotted and discussed. Then the conclusion is drawn whether for the problem at hand, good heuristics are likely to exists.

4.1 Experimental set up to obtain the characteristics

In order to observe the behaviour of the objective function near the optimum, a new instance was created. Table 4.1 gives the data for two aircraft, $K = 2$, with two tasks $N = 2$ each. The time horizon T is 9. Given are $SHIFTS = 2$ shifts, of $t\_per\_shift = 4$ time units length. Time unit $t = 9$ indicates the weekend w_end. There is one hangar space and one line space available. The cost for line maintenance (location 2) is set at 0.25 times the cost of hangar maintenance (location 1): $c_1 = 1; \quad c_2 = 0.25$.

Table 4.1: Aircraft and task data

aircraft k	task nr i	due date $\delta_{k,i}$	man_req_i	line allowed? L_i	Duration D_i	Interval I_i
1	1	5	1	0	2	160
1	2	8	1	1	1	120
2	3	5	1	0	2	160
2	4	8	1	1	3	120

As an example a schedule that was found with optimal value for this instance, using the regular values for *interval loss*, *overhead*, *labour* and *unavailability*, as were given in table 3.2 is presented in figure 4.1. This gives an impression of the small size of the instance and the reader may be able to imagine how alternative solutions can be created by exchanging tasks or sliding them along the time axis.

normal costs	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9
loc 1 a/c 1 task 1	1	1	0	0	0	0	0	0	0
loc 1 a/c 1 task 2	0	1	0	0	0	0	0	0	0
loc 1 a/c 2 task 3	0	0	1	1	0	0	0	0	0
loc 1 a/c 2 task 4	0	0	1	1	1	0	0	0	0
loc 2 a/c 1 task 1	0	0	0	0	0	0	0	0	0
loc 2 a/c 1 task 2	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 3	0	0	0	0	0	0	0	0	0
loc 2 a/c 2 task 4	0	0	0	0	0	0	0	0	0

Figure 4.1: Optimal schedule with regular costs

The set up for the experiment is as follows: for this instance, each of the four cost factors, interval loss, overhead, labour and unavailability, were one after the other set to 1 and the other three to 0, see table 4.2. The time indices of C_3 and C_4 are removed again in the notation, since the value is set to 1 or 0 for all shifts and time units. Adding the differentiation in these cost factors would not have changed the general outcome or conclusions of the experiments.

Table 4.2: Cost factors set up

Set up	**experiment**			
Cost setting:	1	2	3	4
C_1 **interval loss**	1	0	0	0
C_2 **overhead**	0	1	0	0
C_3 **labour**	0	0	1	0
C_4 **unavailability**	0	0	0	1

Then the objective values for all possible solutions were calculated. The objective values were arranged in a non-decreasing order and plotted. For each setting of the cost factors, this process was repeated. This was done by creating a vector for each possible solution, feasible and infeasible, and then adding this solution as a restriction to the model. The model restricted to the one solution was entered into the solver. The number of possible solutions for this instance is $16^4 = 65,536$: each of the four aircraft/task combinations, can be scheduled on 2 locations and 8 time intervals (8 because no due date is higher than 8). To limit the number of calculations, part of the infeasible solutions were removed beforehand, by limiting the t's for each aircraft/task combination to only the allowed start time. This results in $2 \times 4 \times 2 \times 8 \times 2 \times 3 \times 2 \times 6 = 12,288$ possible solutions: aircraft 1 can be scheduled on 2 locations, but only starting on $t \in \{1, 2, 3, 4\}$ since its duration is 2 and it is due on $t = 5$. Task 2 for aircraft 1 can be scheduled on two locations, but is allowed to start on $t \in \{1, 2, 3, 4, 5, 6, 7\}$, since it has a duration of 1 and has to be finished no later than $t = 8$, etc.

Obviously not all infeasible solutions were removed this way: it is easily seen that task 1 for aircraft 1 can only be scheduled on location 1, as it is not allowed to be scheduled on the line (line = 0). However, the objective was not to rebuild the model, but to quickly reduce the number of solutions that the objective value is calculated for. The testing of the feasibility of the solutions was left to the solver, which - if feasible - also calculated the objective value.

4.2 Results of the experiments

In this section all solutions are plotted for the given instance with each of the cost factors set at one and the others at zero. The resulting plots are commented on directly after the plot is given. The overall conclusion of this chapter will be given in section 4.3.

Interval cost (C1) at 1.0

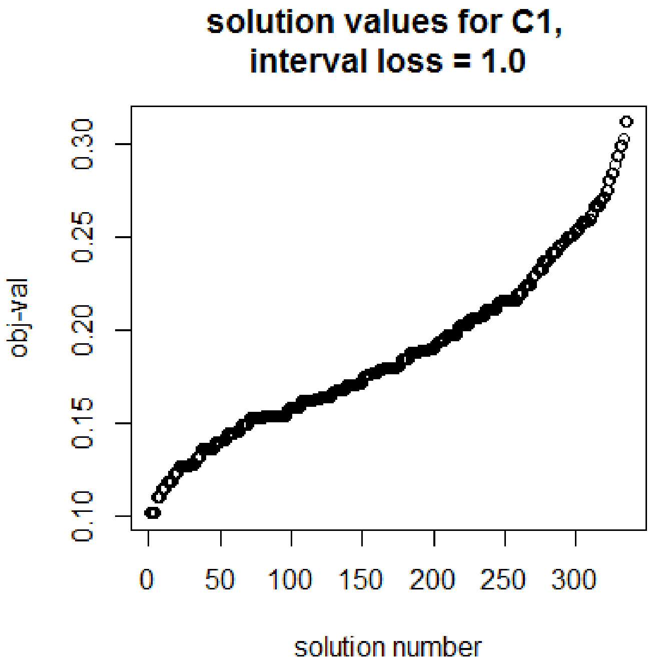

Figure 4.2: Behaviour of interval loss costs (C1)

The optimal value is not zero, since not all tasks can be scheduled on their due dates. We can observe a rather smooth line with steep gradient near the optimal solution. The left most dot is the optimum since the objective is to minimize costs. The first 50 worse than optimal solutions are within 50% increase of the optimum: the best solution has value 0.10 and the 50^{th} is still under 0.15. For such a small instance the increase is large.

Conclusion C1

Overall, this is not a desirable result: good heuristics can only be found if their are many solutions equivalent to the optimum. For the whole model, this result means that good heuristics may exist, if the other cost factors show better properties and if the interval loss costs $C1$ are a relatively small contributor to the total costs.

Overhead costs (C2) at 1.0

Smooth equivalent sections of objective values can be observed. Again the optimal value is not zero, since all tasks have to be scheduled and the costs for moving into and out of a maintenance location are incurred at least once er aircraft. About 40 solutions can be observed that are equivalent to the optimum for this instance: all solutions that require two hangar visits and no more.
Then a jump is made as one line task is scheduled on the line, then another jump for two hangar and two line tasks etc. The number of possible jumps depends on the number of aircraft, the

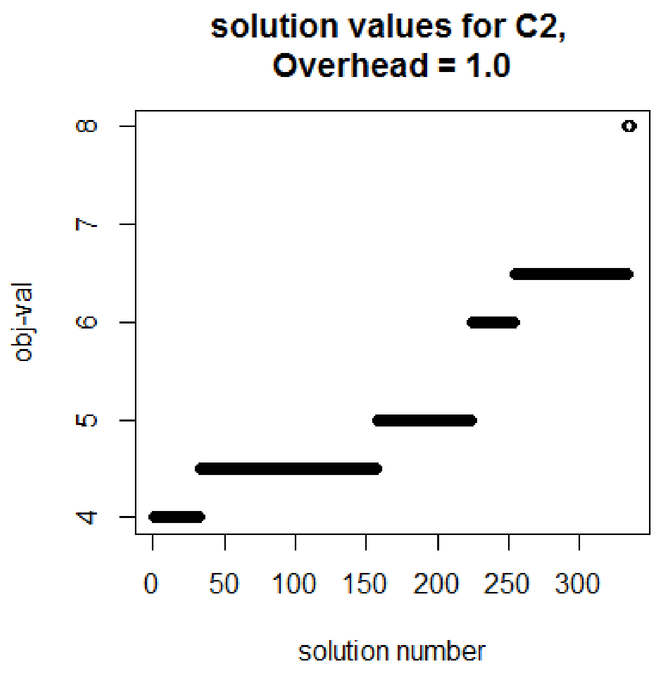

Figure 4.3: Behaviour of the overhead costs (C2)

number of tasks per aircraft and how many of those tasks can be done on the line.

Conclusion C2

Conclusion is that a many solutions may be found equivalent to the optimum for the overhead costs, as long as the solution found is on the first section. This is a promising result for being able to find heuristics.

Labour costs (C3) at 1.0

Again equivalent graph sections of the objective value can be observed. Also many solutions are visible equivalent to the optimum. The horizontal sections can be understood as each represents all permutations of solutions with the same number of shifts of equal size or some fraction of the shifts with a multiple number of technicians equal to the inverse of that fraction. The shortness of the section of optimal solutions can be explained by the durations given for this instance: for aircraft 2 the tasks do not fit one shift if scheduled sequentially. Thus, all optimal solutions are feasible permutations of all tasks of both aircraft planned sequentially, using one technician for each shift.

Any solution using two technicians at the same interval leads to worse solutions for this instance. A worse solution jumps at least the cost of one shift length, which can also be understood as technicians are hired for the entire length of a shift, so either increasing the number of technicians in a shift by one or increasing the number of shifts by one, using one technician, increases the objective value by the cost of one shift length times the cost of labour.

Conclusion C3

Conclusion is that many solutions may be found with the same objective value as the optimum. This is also a promising result for finding good heuristics.

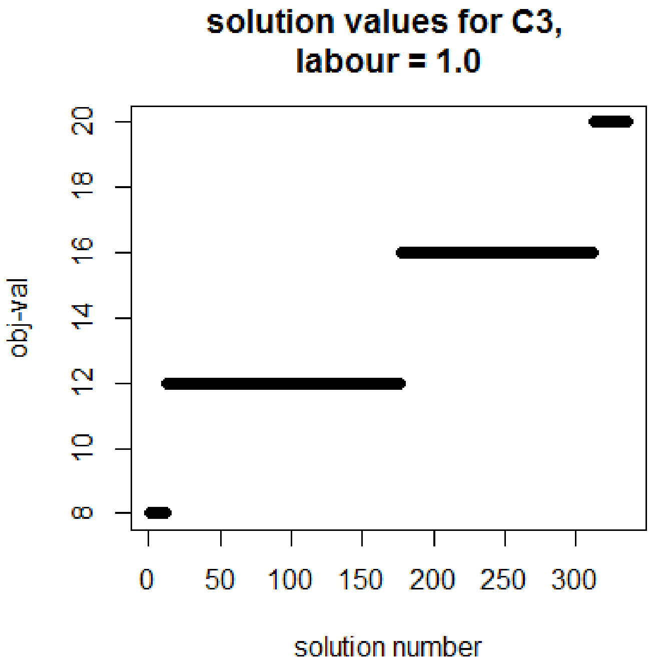

Figure 4.4: Behaviour of the labour costs (C3)

Unavailability costs (C4) at 1.0

Again smooth equivalent sections of graph can be observed, with jumps to a series of worse solutions that are equivalent to each other. An objective value of 5 is the minimum as the maximum duration is 2 for aircraft 1 and 3 for aircraft 2. Jumps are of size one, as each worse solution adds one time interval to the total unavailability of both aircraft, until no tasks are planned at the same time interval for either aircraft.

The observation that relatively few solutions were found equivalent to the optimum, is not specific for this instance, but something that will happen in general: the number of permutations the way the tasks can be scheduled at equal costs, will increase as the length of the longest duration of one task increases.

In figure 4.6 an illustration is given of all possible solutions for some instance of three tasks, of durations $4, 2, 2$. From this it can easily be understood how much the number of possible solutions would increase if the allowed width was not 4 but 5 time units.

Conclusion C4

It can be concluded that for unavailability in absolute terms many solutions equivalent to the optimum can be found. Solutions further from the optimum increase the cost by at least the unavailability costs of one time interval.

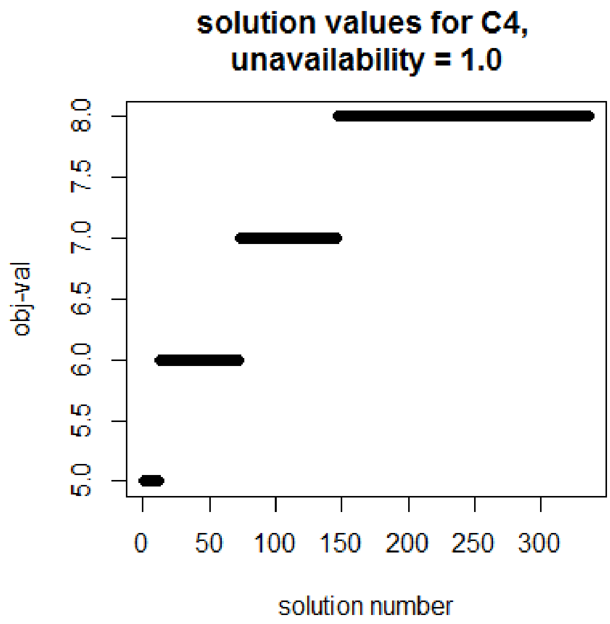

Figure 4.5: Behaviour of the unavailability costs (C4)

4.3 Discussion and conclusion characteristics

First conclusion is that many solutions do exist that are equivalent to the optimum, for each cost function, except the interval loss.

Of interest in this instance are the conflicting interests between the unavailability and the labour costs: all solutions that find optimal labour costs, distribute the tasks over eight sequential time intervals, which is the worst solution for the unavailability costs. Many more instances can be conceived where this is the case. In general, an good solution takes both these factors into account, finding a balance, depending on the difference in actual weights that labour and unavailability have.

Overhead and interval loss have opposite objectives: interval loss gives the lowest costs when all tasks are planned as close to their due dates as possible. This may, in the worst case, result in as many maintenance visits as there are tasks and thus a maximum of overhead costs. The opposite is true for overhead: one visit for all maintenance tasks for one aircraft scheduled in one visit, minimizes the overhead costs. The weights of the actual costs determine which should be given prevalence when balancing these costs.

Unavailability and overhead have similar objectives: stacking all maintenance on top of the longest task results in the lowest unavailability costs for one aircraft: this also results in the lowest overhead costs. This leads to a possible conclusion that clustering the tasks may lead to good results.

In summary, it appears that good solutions can be found. Many solutions may exist with objective values close to or equal to the optimum.

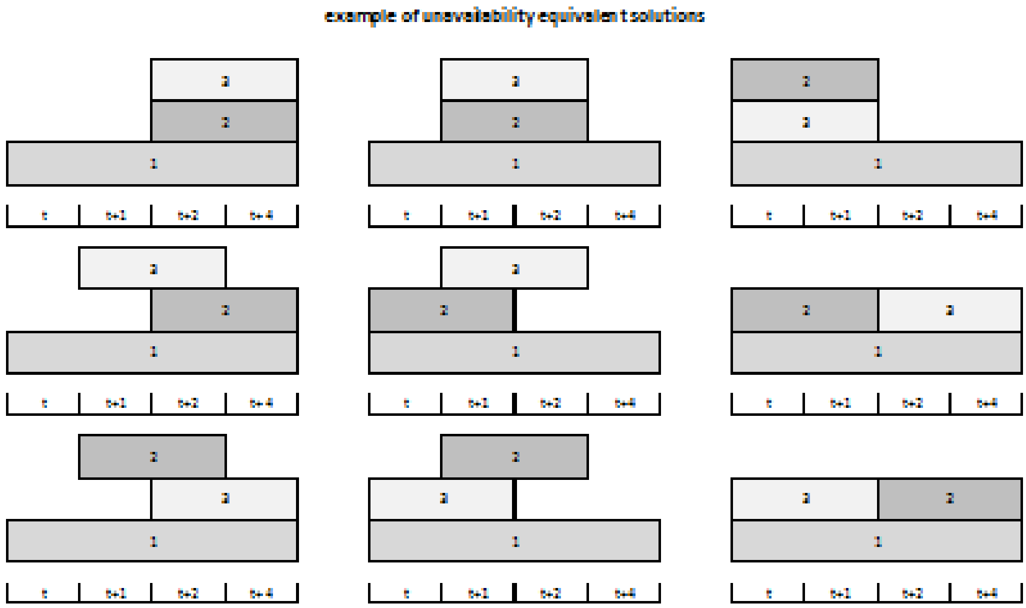

Figure 4.6: Equivalent solutions for unavailability, durations 4,2,2

5 Heuristics

In Chapter 4 it was seen that solving an ILP is NP-hard. It was also shown to be impractical to attempt to find an exact solution even for smaller than realistic instances. This was due to the long run times, both to set up the model and to run the solver. Analysing the different cost functions gave insights in their characteristics and the possibility to find heuristics. This leads to a new research question:

How to develop a heuristic that returns schedules with "acceptable" objective values and run times as instance sizes increase?

In this chapter the main idea of using bin packing as a heuristic will be explained. Then the bin packing problem, some notation and bottom right bin packing as a heuristic for clustering maintenance tasks will be explained. This is followed by the experimental set up for the evaluation of the performance of the heuristic. After that the results for bottom right bin packing will be shown and discussed. Then three improvements on that heuristic will be explained and evaluated in the same manner. This chapter is concluded with the overall conclusions drawn from the results.

5.1 Main idea of the heuristic and experiments

The main idea of the heuristic is reducing the search space, by clustering the tasks per aircraft and due date. This may still result in a good result. Finding the optimal schedule for the clusters of tasks per aircraft and due date, are then the new problem for the exact algorithm to solve. Based on the findings, the heuristic will be further improved.

Bin packing

Clustering the tasks per aircraft and due date is a lot like two dimensional bin packing, which is a much studied topic and is about efficiently solving the problem of placing a collection of rectangles in to a minimum number of bins. When there is one bin with fixed width and an open end the objective is to minimize the height of the highest placed rectangle, the problem is also called the strip packing problem [7], [6]. Unfortunately, 2D bin packing problems are also NP-hard to solve. However, many efficient heuristics have been developed and the upper bounds of their performance compared to the optimum have been proven.
Since the problem is so well studied and described, similar notation is used in many documents and similar notation will be introduced in this paper. This notation is inspired by [1] but with a different figure. Using this notation the applicability to the problem of clustering maintenance tasks will be explained.

Notation

Consider a rectangle, or bin, R, that is open ended on the top side and has width w, (see figure 5.1) and a collection of rectangles organized in a list of length n: $L = (p_1, p_2, ..., p_n)$. Each rectangle is defined by an ordered pair $p_i = (x_i, y_i), \quad 1 \leq i \leq n$, where x_i corresponds to the horizontal dimension of the rectangle and y_i to the vertical dimension.

The objective is to pack each rectangle of L into R such that the height h, measured from the bottom of the bin to the top of the highest rectangle, is minimal. Rectangles may not overlap. Considered are orthogonal and oriented packing only. Orthogonal means that the edges of the rectangles have to be parallel to the edges of R. Oriented means that the given orientation of the rectangles has to be respected: the given order in the ordered pair (x_i, y_i) is strict. Rotations of 90 deg are not allowed.

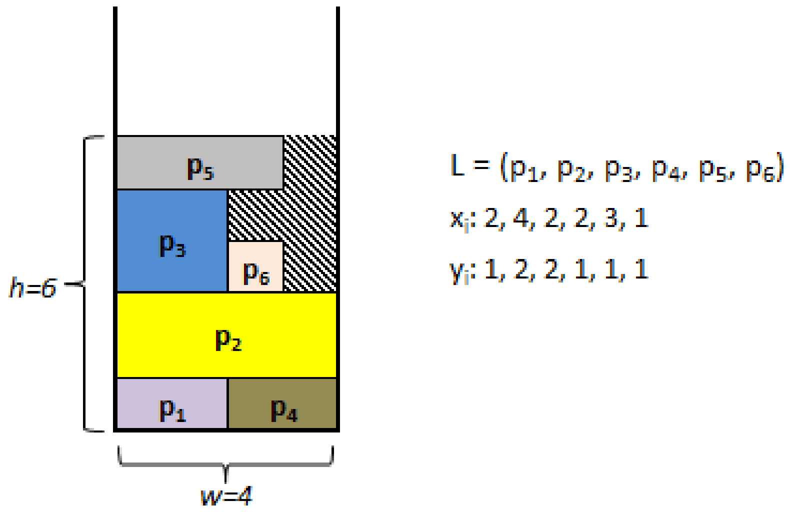

Figure 5.1: Two dimensional packing

Figure 5.1 gives an example solution for the following model instance: the length of the list $n = 6$, the rectangles in the list L are $p_1 = (2,1), p_2 = (4,2), p_3 = (2,2), p_4 = (2,1), p_5 = (3,1), p_6 = (1,1)$ and the width $w = 4$. The resulting height $h = 6$.

The applicability to clustering maintenance tasks is not hard to see. To cluster the tasks for one aircraft some bin width w has to be decided on: that represents the duration of the hangar visit for that aircraft.
The list of tasks corresponds to the list of rectangles. The length in the horizontal (x) orientation of the rectangle is the given duration of the task. The length in the vertical orientation (y) is the given number of technicians required to perform the task.

Bottom left bin packing

Many algorithms and meta heuristics, like genetic algorithms or simulated annealing to efficiently pack the tasks into the bin, can be found in literature. We decided to apply an algorithm already described in the 1980's by [1], even though algorithms with better ratios of the height to the optimum, like the 1.5 described in [3] can be found. The prime reason for using BL-packing is that that is easily explained (in a business environment) and implemented. When the list of tasks is ordered correctly, specifically by decreasing width, the authors of [1] have proven that the resulting height of the bin is no worse than three times the optimum.

The algorithm they called "bottom-up, left-justified (or simply BL)": the rectangles p_i are taken in order from the list L and are placed in the lowest possible location and then left justified in that vertical position of R. The packing in figure 5.1 is an example of BL-packing. As a matter of personal preference, in this application, right justification was chosen (after this: BR). It is easily seen that this has no real consequence for the result of the packing.

If the heights of the rectangles are not all the same size, another decision to be made is how to order the list L with regard to the rectangles of the same width, but different heights. Initially the choice is made for non-increasing height.

The final decision to make is the choice for the bin width w. The initial choice is to set the bin width w equal to the maximum width of the rectangles in L.

The following pseudo code describes the placing algorithm: First for the coding of the algorithm the height of the bin R is set at the worst case height: the sum of the height of all rectangles in L: $\sum_{i=1}^{i=n} y_i$. This is reset to the top y-coordinate of the highest placed rectangle, after the clustering is complete.

```
while the are rectangles to place
set placed = FALSE
    loop over the y coordinates of the bin from 1 to its height
     loop over the x coordinates of the cluster from its width to 1
         if the current coordinate is empty
             if the rectangle width fits the bin width left of the current x coordinate
                 if there is no overlap left or above with already placed rectangles
                     place the rectangle
                        set placed = TRUE
               stop checking the current row if the rectangle width will not fit any more
               stop checking the current row if the rectangle was placed
        stop checking for the current rectangle if the rectangle was placed
```

5.2 Experimental set up

The purpose of the experiments is to evaluate how well the heuristic performs in comparison to the exact solution. The idea is to create task lists for a fleet of aircraft, that have to be scheduled before the due dates. For each instance, the exact solution will be calculated solving the ILP. Next, the heuristic will be applied on the same instance, clustering the tasks, thus creating a much smaller instance for the ILP to solve. This is repeated for up to 100 unique instances and then the average objective value and average run time will be compared. Also the resulting schedules will be analysed and the worst performing schedules will be used to improve the heuristic.

Realistic instances can not be used, because of the run time of solving the ILP for the full instance. Small instances will be used, with an acceptable run time. The instance size will be increased, so conclusions can be made about the performance on larger instances.

In this section first the settings of the cost values throughout the experiments will be explained. These were set to fixed values in realistic proportions to one another. Next the argumentation is given for choosing the other parameter settings, followed by a table, giving an overview.

Values for the cost functions

The costs are set to fixed values approximating the proportions of the realistic values. The costs that were derived are described per cost factor. The values were obtained from an interview with the manager of Fleet and Technical Services, and analysis of the data in TRAX. The derived values for the cost factors are presented in table 5.1 below.

Interval loss (C1)

The costs of interval loss can be best explained by three examples: if maintenance is done exactly on the due date, interval loss is zero, if maintenance with an interval equivalent to four weeks and it is done after two weeks, the loss is 50% and if it is done daily, the loss is close to 100%. The exact formula can be found in the cost function of the model, presented earlier. Only the labour cost of the maintenance is taken in to account when determining $C1$. This means the aspect of replacing or repairing is left out of the calculation of the costs. Of all the 1,100 tasks in the MMA for hangar maintenance only 11 require a part to be replaced, which also are not very valuable systems.

Overhead (C2)

Overhead costs for hangar maintenance are four hours of unavailability: two hours towing the aircraft from the platform to the hangar and two hours back. During this period no commercial operations are possible, but also only limited maintenance is possible: two technicians are required, that can perform some operational and functional checks and preparations like opening access panels. Also one tow truck driver, costing €60 per hour adds to the costs. One hour of unavailability is counted as €400 per hour. Distinction between night and day is not made for overhead costs. One hangar visit costs €1600+€480+€240=€2,320 per visit. Line maintenance has one hour of overhead and no tow truck driver is needed. The costs for one line maintenance turn is estimated at €400+€120=€520, roughly a quarter of a hangar visit each time.

Labour (C3)

One technician on average costs €60 an hour. During the night there is a 20% increase in this rate, so €72 per hour.

Unavailability (C4)

Aircraft unavailability is estimated to be €400 per hour for Transavia. There is a four hour period during the night shift when no flights are scheduled, so during the night the costs for unavailability are on average €200 per hour.

Values used

All realistic values were divided by 60. To reduce the weight of the overhead because shift lengths, day lengths and week lengths are shorted, overhead costs were set at an arbitrary 5. This is large enough value to give the overhead costs a realistic heavy weight. Below in table 5.1 an overview is given:

	day	night
C1, interval loss	1,2	1,2
C2, overhead	5	5
C3, labour	1	1,2
C4, unavailability	7	4,5

Table 5.1: Overview of the costs used

These values for the cost factors remained unchanged for all experiments.

General set up and sampling

Task sets were generated varying the numbers of aircraft K in the fleet, the length of the time horizon T, the number of shifts $shifts$ in the time horizon and the shift-lengths t_per_shift and durations of the tasks D_i and the number of tasks i in the task lists.
Parameter values that do not influence the outcome of the experiment when varied, i.e. due dates $\delta_{k,i}$, the maintenance interval value I_i, whether the task may be done on the platform or not L_i, were kept fixed. Changing these parameter values, will result in a different schedule, in

the sense that the end date and objective value will change, but it will essentially result in the same ordering of the tasks per aircraft and overall schedule. The parameter I_i specifically was kept fixed to keep the interval loss costs fixed as well.
Two clusters of tasks were generated: a minimum of four tasks with one due date and two small tasks with a later due date. All choices were made to resemble the proportions of the real data.

The length of the time horizon T was set at three values: 17 and 25. The values were kept small to limit calculation times. The larger value was chosen to accommodate for larger fleet sizes and to observe the effect of increasing the time horizon on the runtime.
The number of aircraft K was limited to 1, 2 or 3, to limit the runtime of the exact solution. Tasks were divided in sets with two due different due dates. This was done to resemble the observation from the real data where there are a large number of tasks with an interval of 720 flight hours that come due every six to eleven weeks and some separate tasks with due dates between 720 and twice that value.
The number of tasks N to be scheduled in the first set of tasks was varied between 4 and 24 with steps of 2, to limit the number of runs and to limit the instance size with regard to the total experimentation times. Durations D_i of the tasks were limited to 1 through 5. The purpose was to have tasks exceeding the shift length, like in reality. The value of the number of technicians required to perform a task were also limited to small deviations of 1, as in reality most tasks are 1 man tasks.

Table 5.2 gives the general values used for the experiments. Whenever different values were used this is stated and explained.

General			First set					Separate tasks				
K	M	T	N	$\delta_{k,i}$	D_i	I_i	L_i	N	$\delta_{k,i}$	D_i	I_i	L_i
1	2	17	4 - 24	12	1 - 5	160	0	2	16	1	120	1
2	3	17	4 - 24	12	1 - 5	160	0	2	16	1	120	1
3	4	25	4 - 24	12	1 - 5	160	0	2	16	1	120	1

Table 5.2: Parameter values used for the experiments

The durations for the experiments reported in this paper were sampled without replacement from all $\binom{N+K-1}{K}$ possible combinations to draw K from N, where K is the number of tasks needed and N maximum duration a task can have. The sample size for 4 tasks and a maximum duration of 5 is thus limited to $70 = \binom{5+4-1}{4}$. Drawing from the unique combinations of six durations for three aircraft will also not yield 100 samples per aircraft. In general the experiments, were performed 100 times with unique combinations of durations per aircraft. The number of technicians set to one and higher in the same distribution as in reality.

All experiments were performed on the same HP ProBook 450 G2 lap top computer, Intel(r) Core(TM) i7-4510U CPU @ 2.00GHz 2.50GHZ with 8GB RAM and with no other applications open during the experiments.

The code to create the model to feed to the solver and the heuristics were written using R software as a matter of personal preference and the solver Gurobi was selected, because it offers commercial grade performance and is free to use for students.

Procedure description

Figure 5.2 gives an overview of the procedure that was followed for the experiments. For lengths of the task lists N from 4 through 24 step 2, all possible unique combinations of durations were

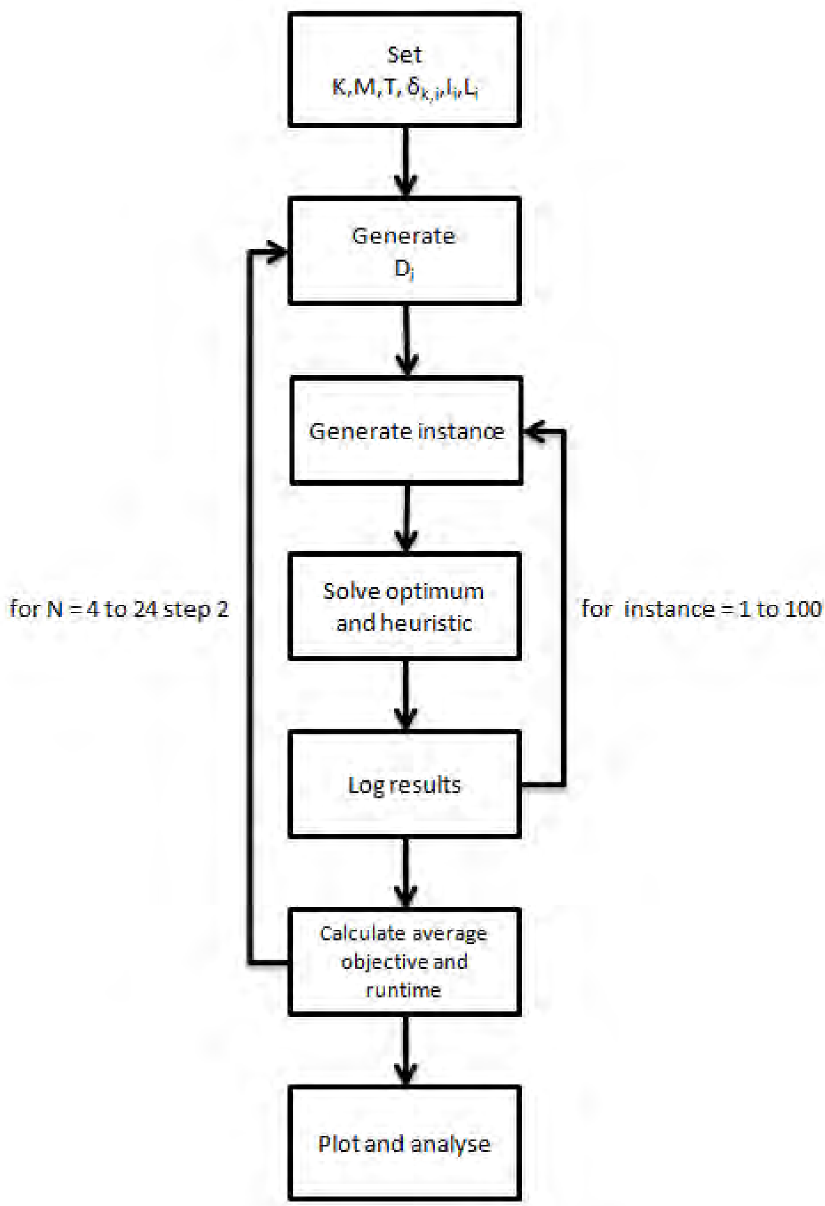

Figure 5.2: Procedure of the experiments

generated. These were shuffled randomly. From these combinations 100 durations D_i were selected per aircraft, or less if there the number of combinations was less. This was done using `sample` from base R with `replace = FALSE`.

In the analysis of the results, schedules are visualized by placing tasks in a two dimensional grid. The x-axis of the grid is the time axis, divided in time units t by the grid. Day and night shift subdivisions are indicated. The y-axis is the labour costs in technicians. A task is depicted as a rectangle and has a duration as its width and a number of technicians required for the task as its height. Colours have no meaning, and are only used to easier distinguish between aircraft and when the same task is depicted in a set of schedules it will have the same colour. Tasks may be numbered if it helps to explain. In figure 5.3 the graphical representation of a task of duration 3 time units requiring two technicians is shown.

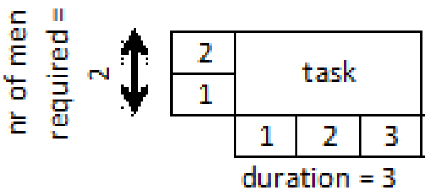

Figure 5.3: graphical representation of a task

In figure 5.4 an empty schedule of 17 time units is depicted. There are four time units in a shift, whether it is a day shift or a night shift is indicated and there are four shifts. Time unit 17 in this case is the weekend. No planned maintenance is ever scheduled in the weekend. Dashed lines outline the shift sizes created by the displayed schedules. Then for each combination of

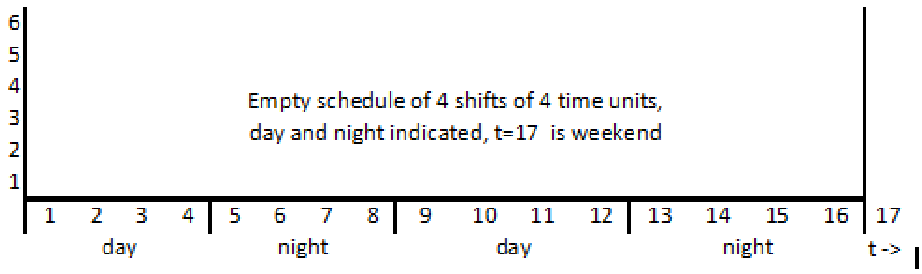

Figure 5.4: graphical representation of an empty schedule

durations per aircraft, an instance was created. One such instance was given as an example earlier in table 3.3.

The instance was then solved by the solver. The resulting schedule was also given earlier in figure 3.1. The same data was given to the heuristics function. The new task set with clustered tasks from the heuristics function was then solved by the solver. For that given example, the clustered data generated by the basic bottom right heuristic, is shown in table 5.3.

Table 5.3: Task info generated by the basic heuristic

k	i	$\delta_{k,i}$	man_req_i	L_i	D_i	I_i
1	91	16	4	0	5	160
1	92	20	2	1	1	120
2	93	16	6	0	5	160
2	95	20	2	1	1	120
3	95	16	6	0	2	160
3	96	20	2	1	1	120

The resulting heuristic schedule is given in figure 5.5.

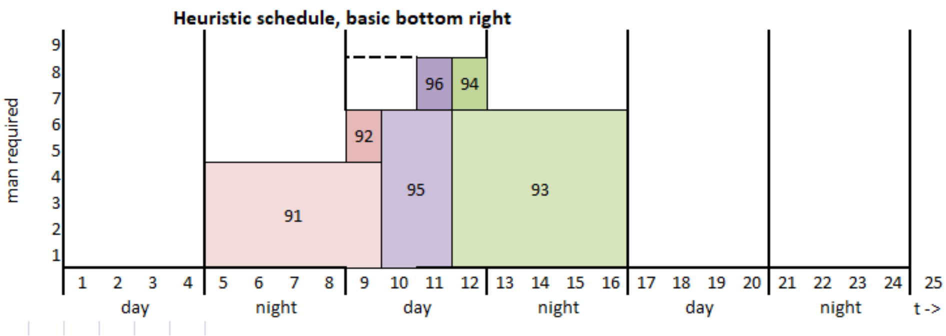

Figure 5.5: Basic heuristic example schedule

The objective values and run times of both the exact solution and the heuristic were saved for each instance. For each instance also the performance of the heuristic relative to the optimal solution was calculated. The heuristic object value was divided by the exact objective value.

$$\text{relative objective value} = \frac{\text{objective value of the heuristic}}{\text{optimal objective value}}$$

The total runtime of the heuristic was divided by the runtime of the exact solution.

$$\text{relative runtime} = \frac{\text{runtime heuristic function} + \text{runtime solver for clustered tasks}}{\text{runtime solver optimal solution}}$$

This process was repeated for all instances and summary statistics calculated. Run times for the solver were obtained by using the run time the Gurobi solver returned as part of the solution. The system time was used to log the start and end time of the heuristics function, which was then added to the solver runtime for the smaller instance of clustered tasks the heuristic created. Then the whole process was repeated for all different numbers of tasks per cluster in the sequence of 4 to 24 step 2.

The mean difference in objective value, the mean difference in run time and both the minimum and maximum difference in objective value were used as statistics to compare the performance.

During the execution .RDS files were made of the durations input of each instance and the results list. From this any instance could be found and recreated for analysis.

5.3 Results of the basic bottom right heuristic

In this section the performance of the basic bottom left heuristic is compared to the exact solutions. First the performance plots for the generated instances are presented and then discussed. Next the outcome of the analysis of the worst performing instances and the possible improvements to the heuristic that follow from the analysis are presented.

Results basic bottom right heuristic

All instances were created with the parameter sets as described in table 5.1 and table 5.2. For the basic heuristic the tasks were clustered per aircraft, per due date using the described bottom right, oriented orthogonal packing algorithm, with

1. the tasks ordered in non-increasing width
2. then the tasks with equal widths are ordered in non-increasing height
3. an open ended bin is used
4. the bin width chosen equal to the duration of the longest task

The results of the experiment for one aircraft is shown in figure 5.6 and for two aircraft in figure 5.7. In the figures the performance of the heuristic, relative to the exact solution is plotted.

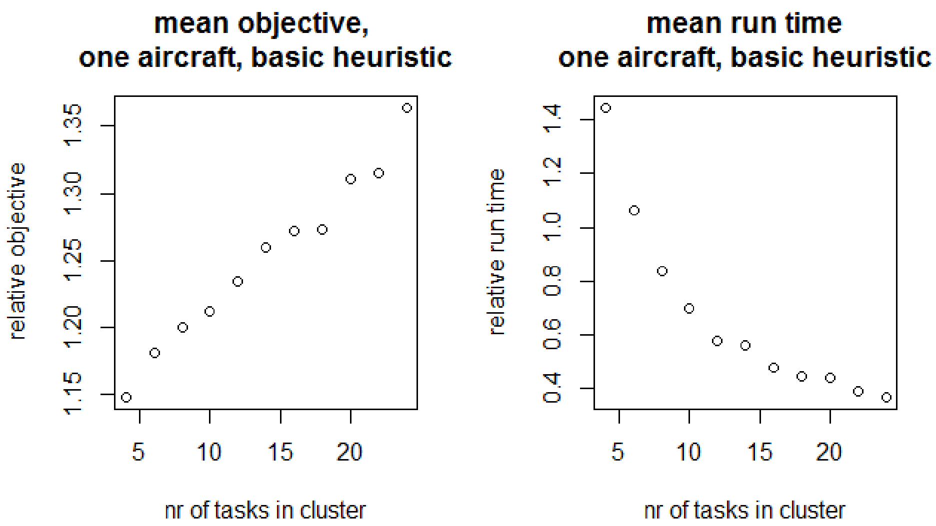

Figure 5.6: Basic heuristic, one aircraft

The plot of the three aircraft experiment was left out, as it did not provide new information.

Discussion of the basic bottom right heuristic results

The plot of the mean objective for both one and two aircraft show two characteristics: first is that the performance of the heuristic deteriorates as the number of tasks per aircraft increase. The almost straight line of the mean objective indicates a possible linear relation between instance size and relative objective value of the heuristic. Second observation is that the relative value is large in both cases. It is greater than 10% worse than the exact solution in all cases.

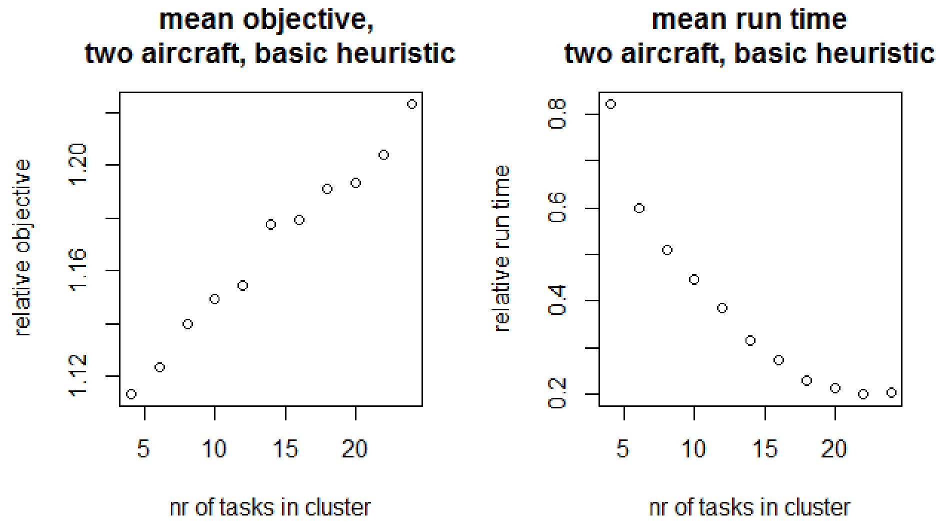

Figure 5.7: Basic heuristic, two aircraft

The plot of the mean difference in runtime shows a decrease. Positive observation is that for two aircraft the runtime of the heuristic is almost half that of the exact solution for larger instances. It also can be observed that for one aircraft the runtime of the heuristic exceeds the runtime of the exact solution for the two smallest instances.

Analysis of the worst performing instances

For all instances two clusters of tasks are created by using the two due dates; the basic heuristic clusters tasks based on their due dates. A detailed inspection of the results obtained for the cluster of size 4 is given below. The objective was to schedule four tasks with a due date $\delta_{k,i} = 12$ and with durations D_i ranging from 1 to 5 time units and two tasks i with a due date $\delta_{k,i} = 16$. For $k = 1$ all 70 $\binom{n+k-1}{k}$ unique combinations of 4 numbers from 1 to 5 are in the sample, in random order.

In figure 5.8 a plot of all results is displayed. Instance 63, showing the worst relative performance of the heuristic is examined:

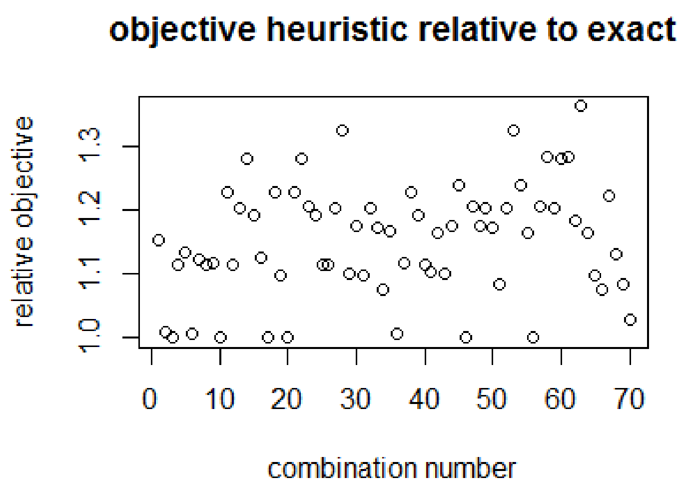

Figure 5.8: Relative performance of the heuristic for all combinations of 4 durations of 1 to 5

Table 5.4: Task info of instance 63

aircraft k	task nr i	due date $\delta_{k,i}$	man_req_i	line allowed? L_i	Durations D_i	interval I_i
1	1	12	2	0	3	160
1	2	12	1	0	4	160
1	3	12	1	0	4	160
1	4	12	1	0	5	160
1	5	16	1	1	1	120
1	6	16	1	1	1	120

The data for instance 63 are as shown in table 5.4.

Below in figure 5.9 the resulting schedules are shown: on top the optimal solution and on the bottom the BL-clustering.

Two possible improvements for the heuristic can be seen: the heuristic creates a 5x5 task (#91), causing the need for two shifts of 5 technicians, where the exact solution is able to restrict the 5 technicians to a single shift and lets only the task with duration 5 run over into the the day shift. Also the heuristic adds the clustered single tasks to the left, causing one extra hour of unavailability. This is an extreme example of a possible improvement that can be be observed in many instances.

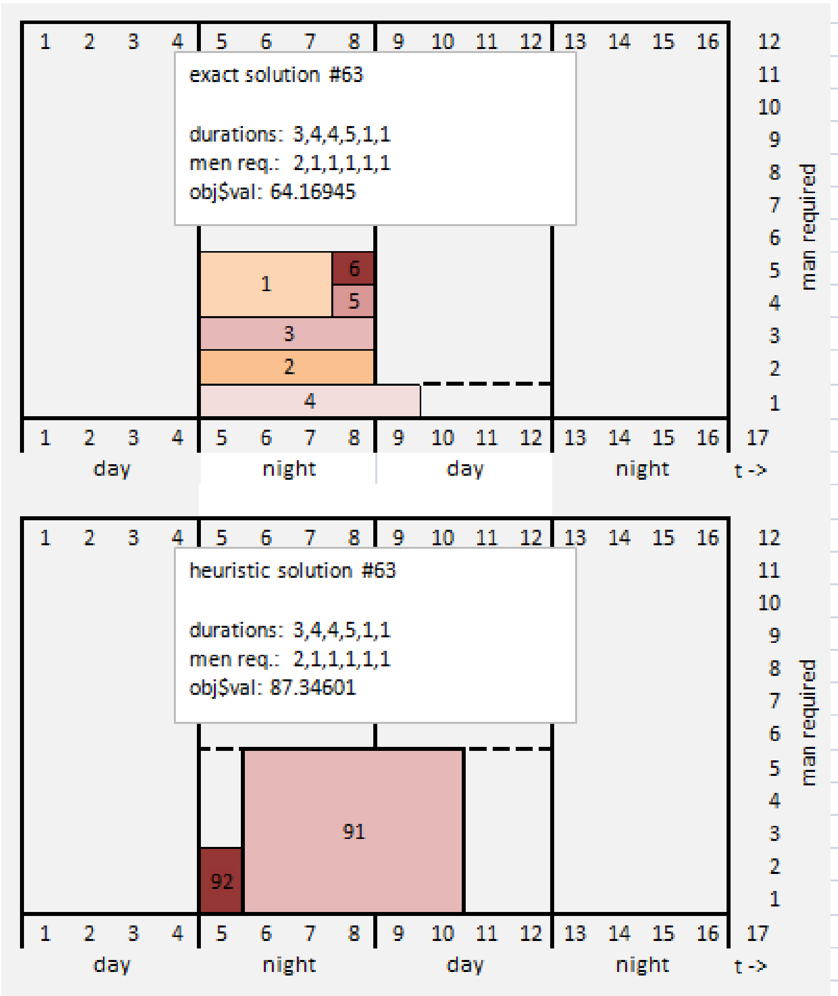

Figure 5.9: schedules from exact and heuristic solution for instance 63

For two aircraft, for 4 tasks in the main cluster the schedules in figure 5.10 the solution is shown where the heuristic performed worst. The worst cases for all fleet sizes show similar problems.

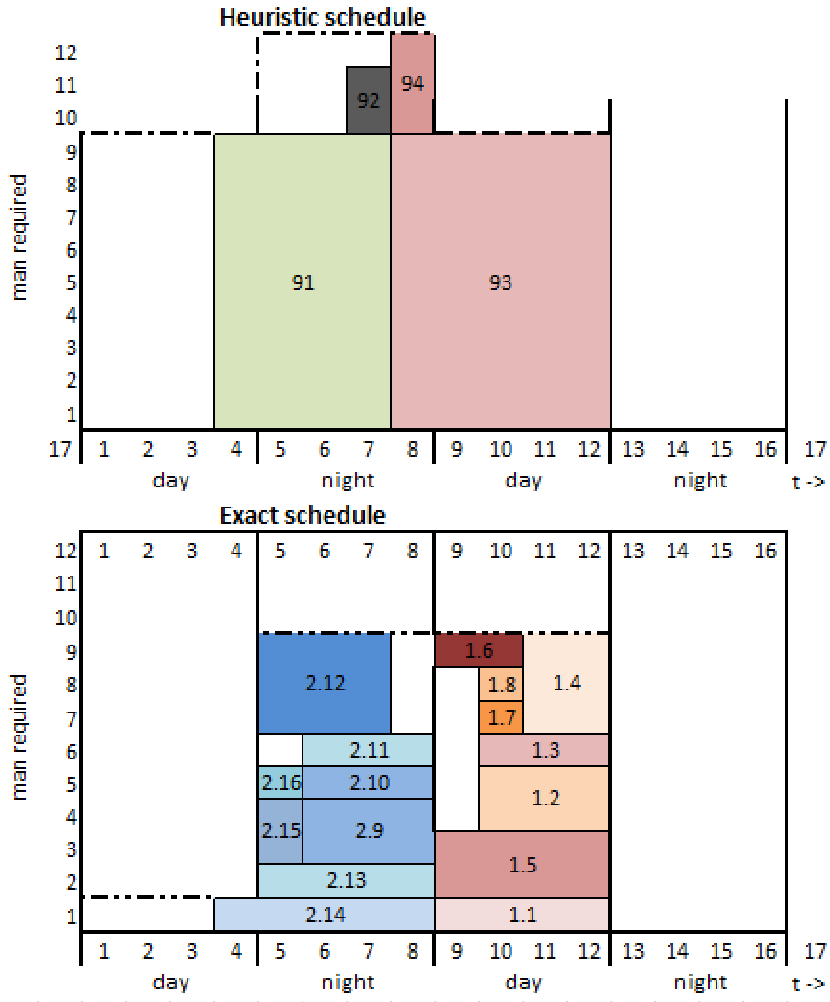

Figure 5.10: Schedules from exact and heuristic solution for two aircraft

The first problem is that clustering by due date $\delta_{k,i}$ results in two clusters, which the solver tries to schedule in one hangar visit. No instances were encountered were the exact solution scheduled the tasks clusters separately. This also follows from the cost structure.

The second problem in the worst cases is that when there is at least one task with a duration exceeding the shift length, this results in the heuristic schedule using two equally sized consecutive shifts, where the exact schedule can split this in a larger shift and a small shift. This knowledge will be used for improving the heuristic.

5.4 Best of two secondary orderings

Before applying the knowledge of analysing the worst performing heuristic solutions, in this section an insight that was obtained earlier is described. First the improvement is explained, then the results are demonstrated.

Explanation of the best of two secondary orderings

After the basis heuristic of bottom right bin packing was formulated and tested, a first possible improvement to the basic heuristic was observed. This also followed from the literature mentioned earlier. The insight is that secondary ordering matters: an ordering that works for one instance may be worse for another. So the first improvement that was added, was to also use non-decreasing height as the secondary ordering, record both objective values and use the best solution. It can be checked that non-decreasing height as a second ordering used in the left packing gives a worse solution for the instance in figure 5.11. Note that this is the same set of rectangles P_i as in figure 5.1, but only the ordering of the list L changes.

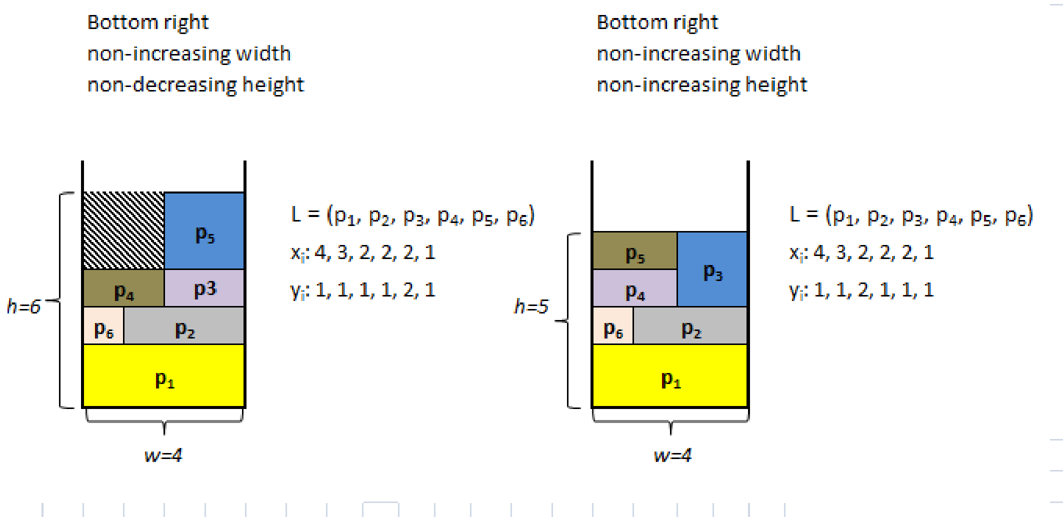

Figure 5.11: Secondary ordering matters

Results of adding best of two secondary orderings to the heuristic

Choosing the best of two secondary orderings was added to the heuristic. The same parameter set and experimental set up was used to evaluate the the influence of this addition to the heuristic on the objective value and run time. Only the result plot for $K = 2$ is displayed, as the other plots show similar results.

Figure 5.12 suggests a near linear increase as the number of scheduled tasks increase. This is as expected as we did not introduce any real extra intelligence and the run time shows a decrease as instance size increase.

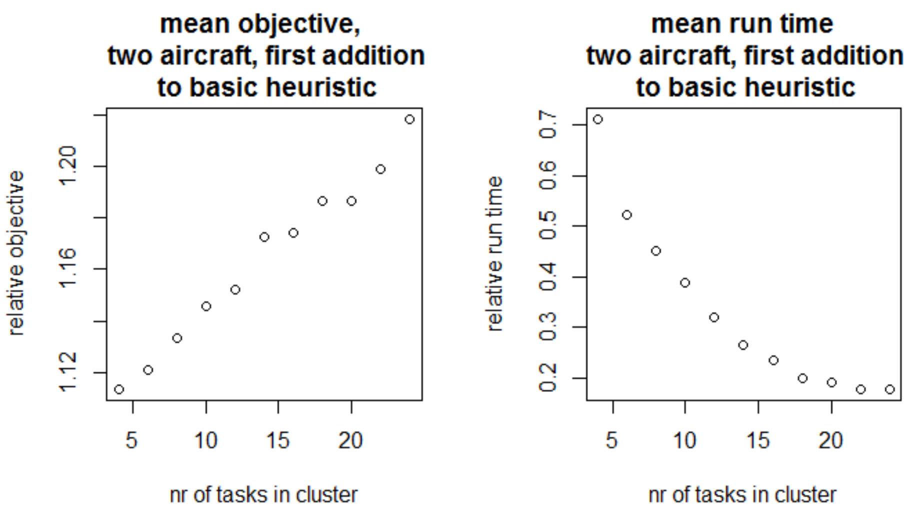

Figure 5.12: Best of secondary orderings

To better show the influence of this addition to the heuristic, one results plot is shown in figure 5.13. This shows the results, using an identical set of instances for the basic heuristic and taking the best of the secondary orderings. The results are plotted in the same plot, with the same axes. It is clear that the objective values of the first addition are never worse but occasionally better. The run time shows the same shape of an exponential decrease, but all values are clearly higher than for the basis heuristic. Since each ordering is done twice, but not all other steps of the heuristic are done twice, this is within expectation.

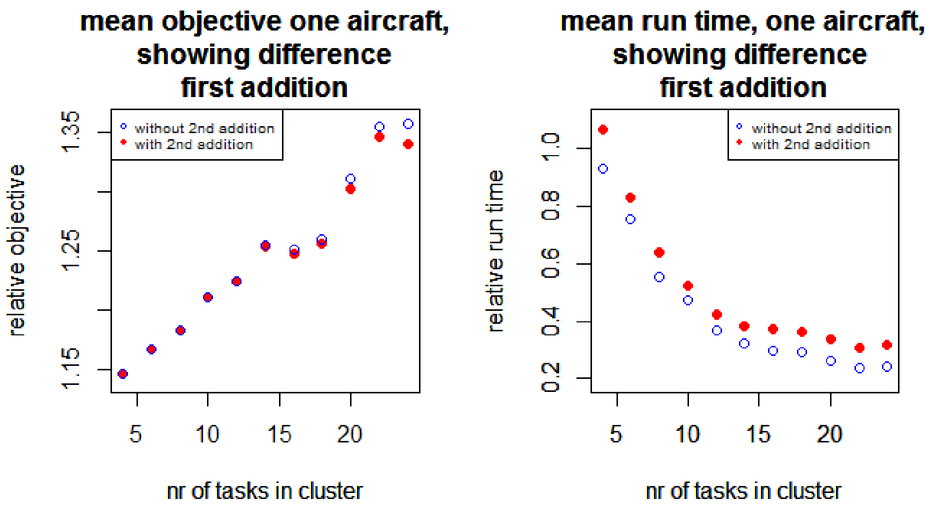

Figure 5.13: The benefit of taking the best of two secondary orderings

One new interesting worst case came forth from the analysis of the worst cases: the bin width w is chosen to be equal to the maximum of the durations D_i in the task list L. When the maximum duration is shorter than the shift length t_per_shift, all tasks are stacked in a narrow cluster. This leads to a shift size almost twice as high as necessary in the case shown in figure 5.14. This will later be used to adapt the heuristic.

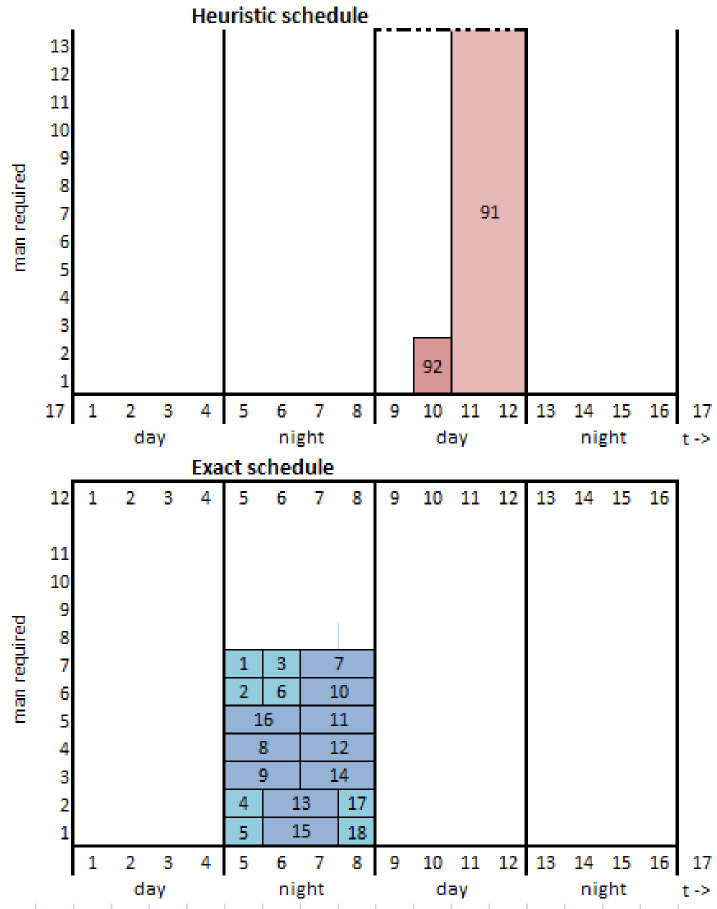

Figure 5.14: schedules from exact and heuristic solution for one aircraft with many short tasks

5.5 Clusters per aircraft

The basic heuristic clusters all tasks, per due date, using bottom right, orthogonal, oriented, non-increasing width, non-increasing height. The first improvement was to also apply non-decreasing height as a second ordering and choose the best solution. Based on the results of the basic heuristic, as a second improvement, the clusterings were made not per due date, but per aircraft: so all tasks in the planning horizon for each aircraft are given to the heuristic to cluster. This section describes the adjustments made to the heuristic and the results that were obtained.

Explanation of the adjustment

Clustering tasks per aircraft regardless of due date instead of clustering by aircraft and due date is self explanatory. However, one complication had to be resolved before clustering tasks per aircraft instead of per aircraft and due date. This is the question what due date to use for the new cluster of tasks. Up until now the clustering was per due date, so no problem arose. In order to solve this problem, the earliest due date $d_{k,i}$ was used for the new task that is created when clustering all tasks that are in the task list of aircraft k. This ensures that no tasks in the cluster are scheduled after their due dates. For the correct calculation of the objective value both the due date $\delta\_cost_{k,i}$ and the interval $I\_cluster_i$ of the cluster were recalculated by weighting them:

$$I\_cluster_i = \frac{\sum_{i=1}^{i=N} I_i * man\_req_i * D_i}{\sum_{i=1}^{i=N} man\_req_i * D_i}$$

$$\delta\_cost_{k,i} = \frac{\sum_{i=1}^{i=N} \delta_{k,i} * man\_req_i * D_i}{\sum_{i=1}^{i=N} man\_req_i * D_i}$$

The parameter $I\_cluster_i$ simply replaces the parameter I_i and is written as I_i for the clusters as well: it is always clear when a clustering of tasks is described and it simply replaces the the old interval. The parameter $\delta\_cost_{k,i}$ is added to the task lists, since the original minimum due date of the clustered tasks needs to be retained as well.

Results of clustering per aircraft

Below the result plots for $K = 2$ and $K = 3$ are shown in figure 5.15 and figure 5.16. Also a plot is added for one aircraft, where for one identical dataset, first only the first addition, i.e. selecting the best result of two secondary orderings was executed and next the clustering per aircraft, called second addition was executed. The results are in the same plot, different colours, with the second addition the dot is filled.

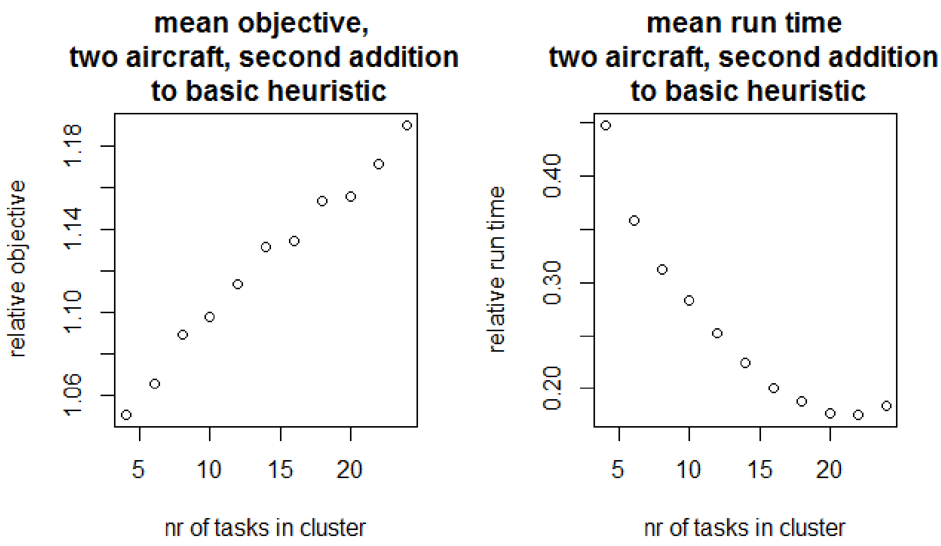

Figure 5.15: Clustering per aircraft, two aircraft

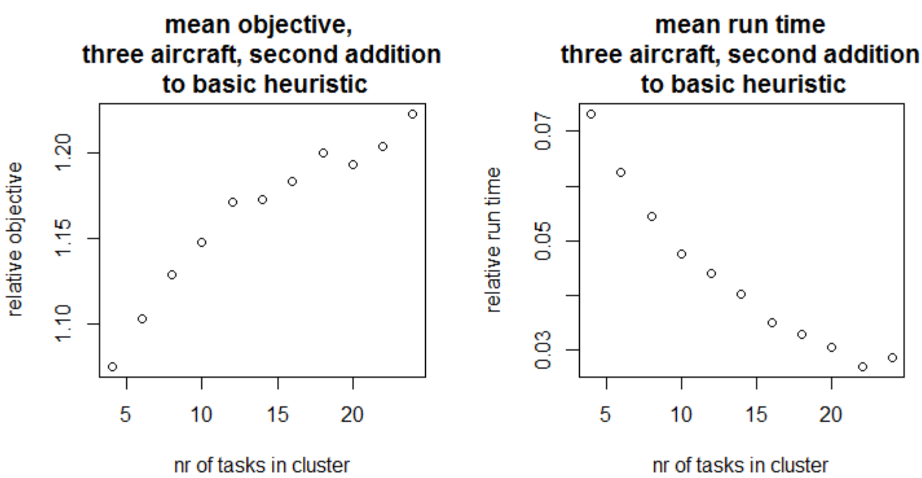

Figure 5.16: Clustering per aircraft, three aircraft $T = 25$

Discussion clustering per aircraft

First thing that is noticed in figures 5.15 and 5.16 is that the results still show a linear increase as instance sizes increases. The run time plots keep showing a decrease as instance sizes increase. The gradient of the increase in relative objective values has decreased compared to the basic heuristic: 0.14 per 24 as compared to 0.19 per 24. A slight increase in the run time can be observed in the last point of figure 5.16, for the 24 tasks in a cluster: this is likely to be caused by repeatedly checking how far the running of the instances had advanced, as this took hours. The influence of scrolling is predominantly absorbed by the exact solution's runtime. Another observation that can be made is that increasing the time horizon over which the tasks need to be scheduled, shows the superiority in run time of the heuristic compared to the exact solution.

Finally, in figure 5.17 it can be seen that for an identical dataset the objective values of clustering per aircraft are better than those without. Also, very interesting, but not surprising, is that adding the second addition improves the run times. This can be explained by the fact that one new clustered task per aircraft is given to the solver to schedule; without the clustering per aircraft the solver has the two clustered tasks per aircraft and per due date to schedule, creating a lot more possibilities to check over the time horizon.

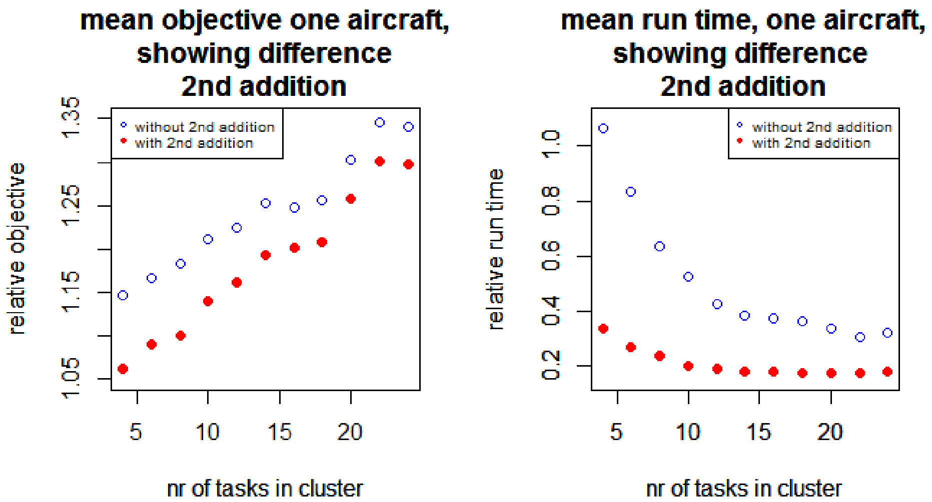

Figure 5.17: Comparison of first addition with and without second addition

5.6 Double bin width or split cluster

In this section a third improvement to the heuristic, based on the findings of the experiments is explained, the results are presented and discussed.

Explanation of the improvement to the heuristic

The third improvement consists of a choice between one of two elements, or not to apply either. The first element is to apply some balancing between unavailability and labour costs if that leads to lower costs given the values of the costs for unavailability and labour. The aim is to solve situations as depicted in figure 5.14. The second element is to split the clustering of tasks in one wide cluster the size of the duration of the longest task and another the size of a multiple of shift lengths, if the duration of the longest task exceeds the duration of a shift length. The aim is to improve the situations as shown in figure 5.10. The second element is only applied if the first is not, so it is possible that the third improvement is not used for some aircraft.

Double bin width

In words, this part of the heuristic solves the following equation: if the cost of interval loss plus the cost of labour of the current bin width is larger than the cost of interval loss plus the cost of labour for twice the current bin width, then double the bin width. Before the optimal placing of the clustered tasks is solved by solving the ILP, some values are unknown and assumptions have to be made. Firstly, it is unknown whether the cluster will be placed in a day shift of night shift of both. Secondly, the height of the resulting cluster is unknown. The first assumption is that the cost for unavailability is the mean of the day and night unavailability costs. So, assume the cost for unavailability

$$\overline{C4} = \frac{C4(day) + C4(night)}{2}.$$

In this equation *day* is the set of all t in the day shifts and night the set of t's in the night shift as defined earlier, making $C4(day)$ the parameter value for the unavailability costs in the day time and $C4(night)$ the parameter value for the unavailability costs in the night shifts. The second assumption is that we can calculate the optimal height h_{opt} of the bin given the current width w as follows:

$$h_{opt} = \frac{\sum_{i \in \mathcal{N}} D_i * man\_req_i}{w}.$$

Labour cost is the number of shifts, times the number of technicians per shift, the "height" of the cluster. For the "height" we can take h_{opt} calculated earlier. The number of shifts can be obtained by taking the ceiling of w divided by t_per_shift. For shortness of notation t_per_shift is renamed L in these equations:

$$\lceil \frac{w}{L} \rceil.$$

Finally, we assume the average of the labour cost $C3(shift_s)$ for day and night is also correct a correct value: $\overline{C3}$. Then, if

$$w * \overline{C4} + h_{opt} * \lceil \frac{w}{L} \rceil * \overline{C3} * L > 2 * w * \overline{C4} * \frac{h_{opt}}{2} * \lceil \frac{2 * w}{L} \rceil * \overline{C3} * L,$$

then the bin width is doubled. This option is only applied if the original bin width is less than one shift length, to prevent infeasible solutions. If doubling the bin width is not a better solution, the algorithm checks the second element.

Split cluster

The second element is only called if the first element of this third addition to the heuristic is not. In other words: it checks whether any duration exceeds a multiple of shift lengths. If there are one or more, it splits the task list in two clusters: one cluster with bin width equal to the longest task, and the other (if there are any) with bin width equal to the multiple of shift lengths that was exceeded.
So again the maximum of the durations D_i in the task list with $i \in \{1, ..., N\}$ tasks is called w, the bin width, and the number of time units per shift we call L. Then if $\lfloor \frac{w}{L+1} \rfloor > 0 \wedge N > 1$ then two clusters of tasks are created: one with with bin width w, containing all tasks with $D_i > \lfloor \frac{w}{L+1} \rfloor * L$ and the second with bin width $\lfloor \frac{w}{L+1} \rfloor * L$ containing all tasks with $D_i \leq \lfloor \frac{w}{L+1} \rfloor * L$.

Results of adding double bin width or split cluster

The third addition to the heuristic was to apply balancing between between labour cost and unavailability costs if that is advantageous, or double de bin width if that is advantageous else do nothing other than the first and second additions. In figures 5.18, 5.19 and 5.20 the performances of the heuristic relative to the exact solutions are plotted.

In the result plots in figures 5.18 and 5.19 for two aircraft several things stand out: all objective values are well below 10%, and the linear relation between instance size and objective value is there, but for smaller instances the gradient seems to be almost zero. Starting with instances

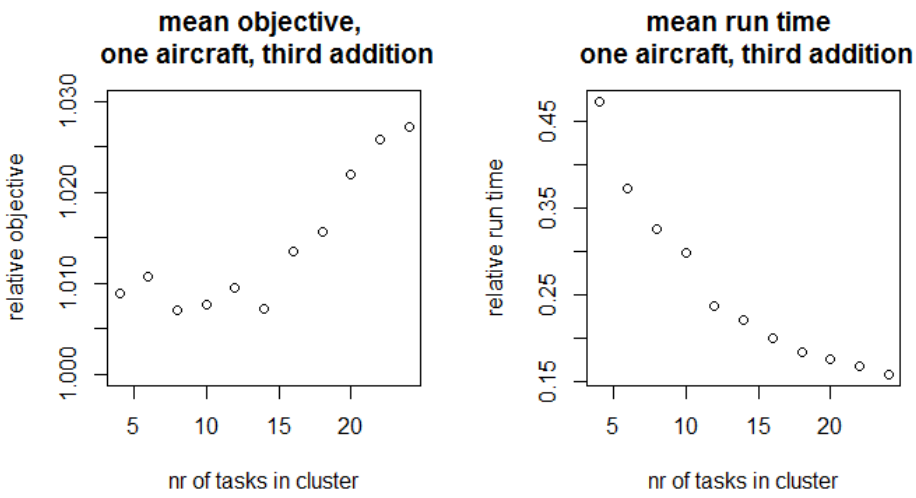

Figure 5.18: Third addition to the heuristic, one aircraft

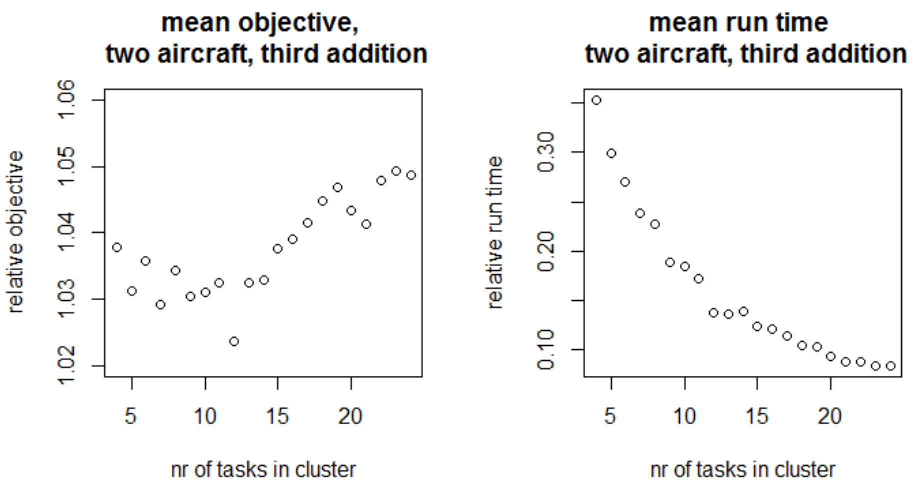

Figure 5.19: Third addition to the heuristic, two aircraft

sized 16, the linear relation with the instance size reappears.

The plot for three aircraft, figure 5.20 has lower relative objective values than the the previous addition's results plot in figure 5.16, but again clearly has a linear relation with the instance size. The gradient of the plot is 0.03 over the 24 increments.

The run time plot for two and three aircraft show the same pattern and values as the previous additions, which would indicate the third addition does not slow down processing significantly.

To give the reader some insight in the absolute run times: the run time of the heuristic for "nr of tasks in cluster" = 24 never exceeded 1 second, while the runtime of the exact solution had a minimum of 11 seconds and a maximum of 137 seconds. When running $K = 2$ and $T = 33$ the mean runtime of the heuristic also did not exceed 1 second, while the runtime of the exact solution averaged 3 minutes, with a maximum of over 9 minutes

Finally, for one and two aircraft, using identical instances, the plots of the relative objective values and run times are displayed in one figure using the same y-axis scale. In figure 5.21 the results are displayed for one aircraft; in figure 5.22 for two aircraft. In these plots the change in

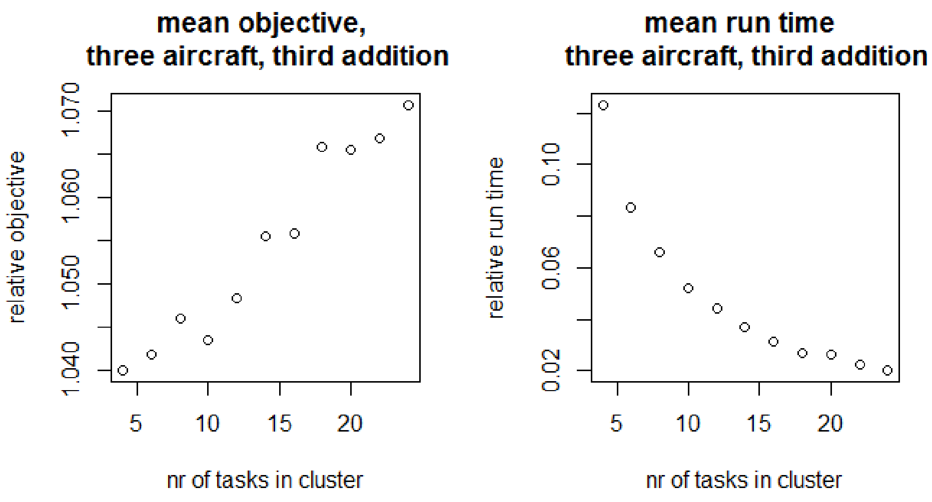

Figure 5.20: Third addition to the heuristic, three aircraft $T = 25$

the results of each addition to the basic heuristic is made visible and it clearly shows how much better adding “double bin width or split cluster” performs compared to the earlier versions of the heuristic.

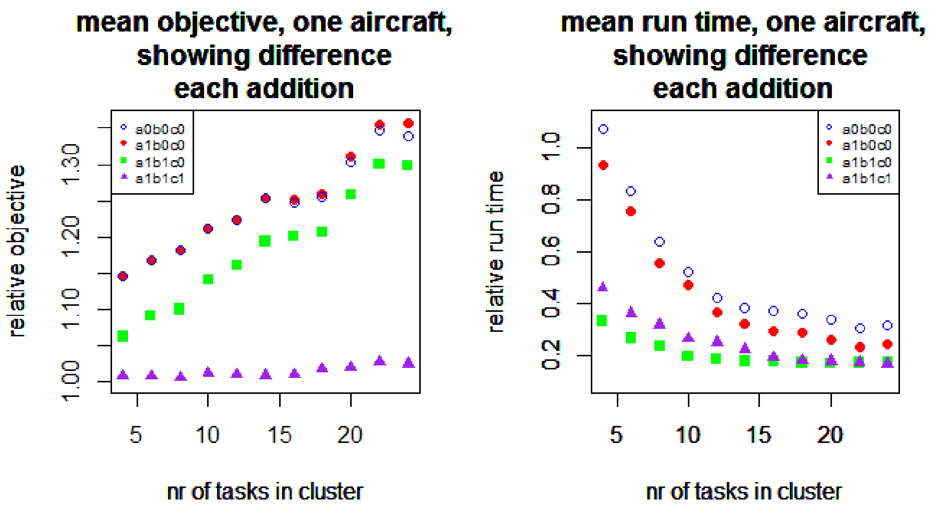

Figure 5.21: Overview of the results of each addition for one aircraft

Discussion

For larger instances “double bin width or split cluster” still shows some challenges that need to be solved. The increase of the objective value relative to the exact solutions, can be explained by figure 5.23. Here the schedules created by the exact solution and the heuristic are displayed next to each other. Comparing these schedules a challenge can be observed:
The heuristic comes with a very good solution for each of the aircraft separately, but, since no rules were added, yet, for how to distribute the schedules over the shifts (or over the week or the entire time horizon), it fails to better use the unused time of the second shift. Another example demonstrating the increase of the relative errors of the heuristic can be seen in figure 5.24, where the solutions for three aircraft separately are good, but the heuristic does not take advantage of the free space in the shifts and creates a whole extra shift as compared to the heuristic. Note that this is the same example that was used in chapter 3. in figure 3.1

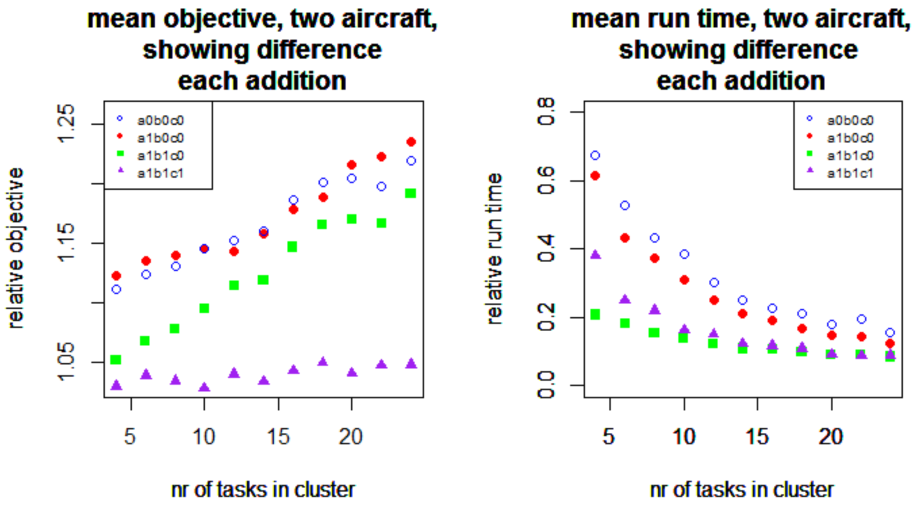

Figure 5.22: Overview of the results of each addition for two aircraft

5.7 Conclusion

Each addition to the basic heuristic shows an improvement. The third addition to the heuristic, "double bin width or split cluster" shows promising results. For the tested instances, the objective value is less than 8% worse than the exact solution. However, the gradient of the linear relation to the instance size may still be to steep for the heuristic to perform well for realistic instances.

The analysis of the poor performing instances showed that the heuristic performs as expected and mimics the behaviour of the exact solutions well for the tasks of individual aircraft, but needs to also take into account the whole time horizon. That would be an important next step to perform constantly well also for increasing instance sizes.

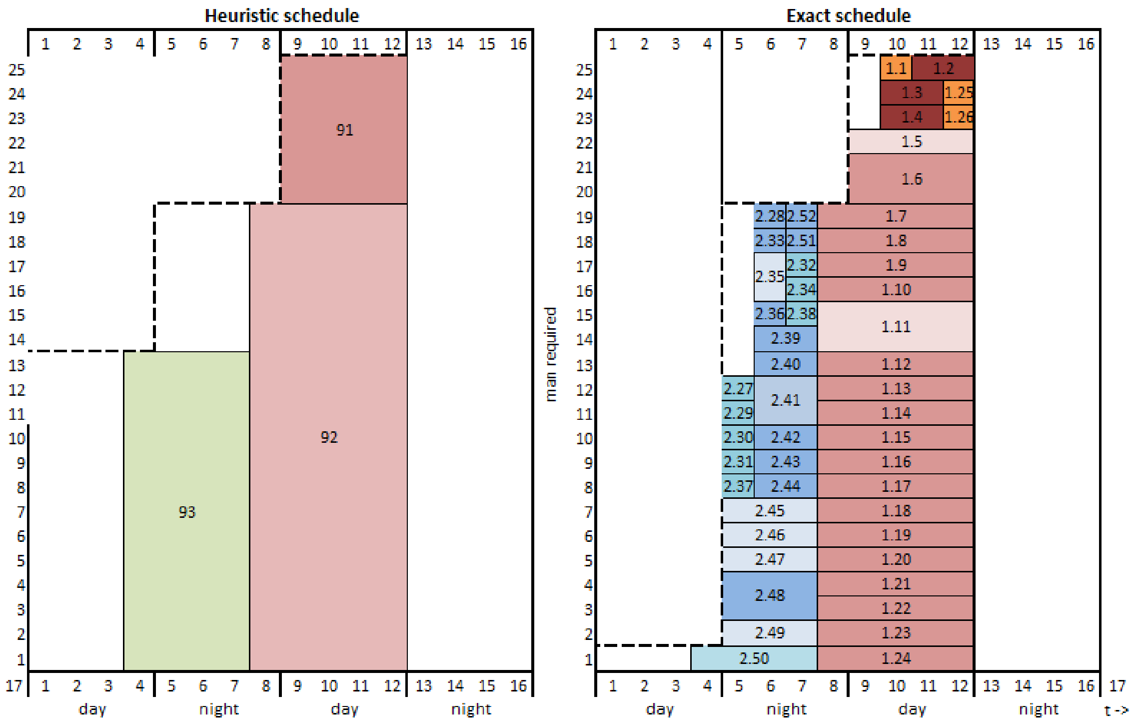

Figure 5.23: Two schedules demonstrating the inability of the third addition to look over the shifts

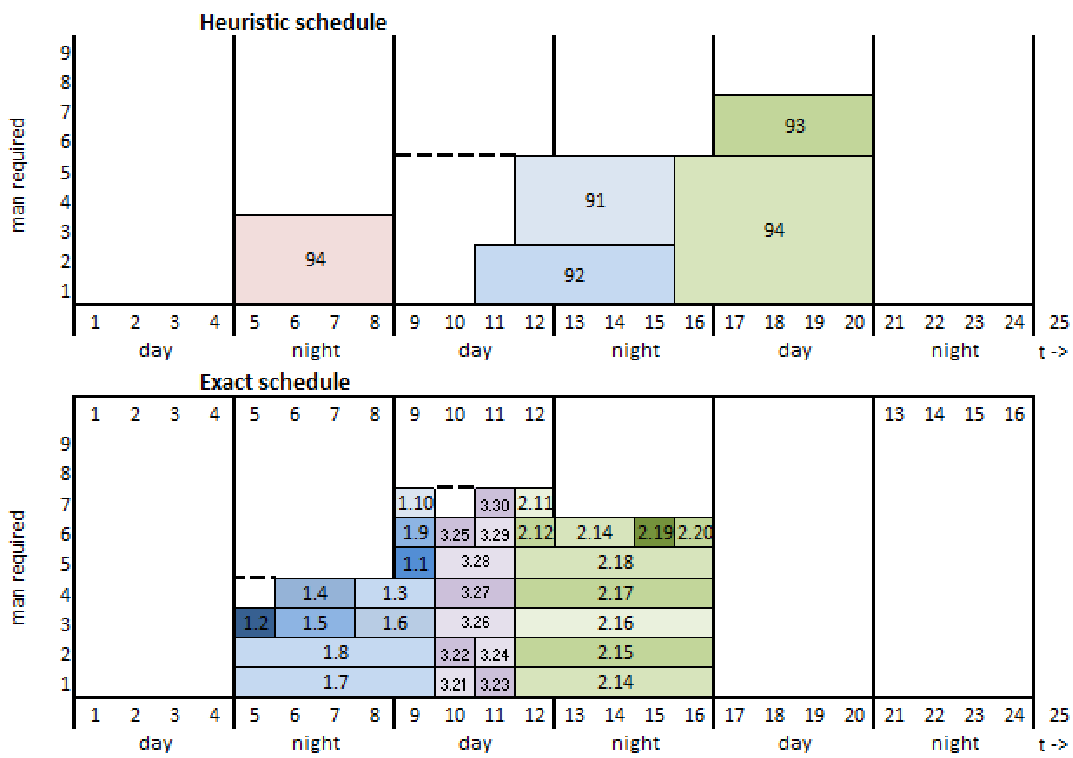

Figure 5.24: Challenge that remains to be solved

6 Overall conclusion and possible future research

In summary, mathematical modelling, of which the optimum can be found by solving an ILP, can in theory be used to find an optimal schedule for the planned maintenance of a fleet of aircraft. Main practical limitation is the run time of both the model and its preparation. It was shown that the run time does improve by reducing the number of tasks that need to be scheduled per aircraft by applying clustering.
The results obtained are good, but not yet good enough. The gradient of the objective value of the heuristic relative to the exact solution needs to be reduced even further to be useful for larger, realistic instances.

In the following paragraphs some ideas will be described, that could improve performance as well as improve practical applicability. Some of these ideas described have been conceived and partially modelled and coded during the research.

Restricting the number of technicians in a shift

Modelling and programming a restriction that limits the number of technicians per shift to a certain bandwidth limits the number of solutions. Adding a penalty to the cost function of the ILP for deviating from a certain number of technicians is a possible alternative. Achieving the same limitations with the heuristic may prove a little more complicated. A possible solution could be to iteratively vary the bin width of the schedule per aircraft. This could then be extended to testing several combinations of bin widths over a period of a week or the whole time horizon.

Exchanging technicians between aircraft

The current model regards two hangar maintenance spots as two separate locations, without the possibility of exchanging technicians between the two aircraft in the hangar. If for one aircraft some tasks still need work and more technicians are present than needed, the model leaves them idle until the end of the shift. A better model needs to be able to schedule these technicians on another aircraft in the same location (i.e. hangar or platform), possibly incurring a small penalty for each time a technician changes aircraft. An even better model should be able to consider sending a technician from the hangar to the line or vice versa at higher costs. How this could be incorporated in the heuristic is not immediately clear. Point of interest: this is currently not considered during the planning stage in practice either. During the execution of the maintenance, technicians may change aircraft in the hangar when tasks for one aircraft run out and another aircraft still needs work.

Dealing with realistic time horizons

It was shown that as the time horizon increases, the relative improvement of the run time of the heuristic decreases. To be able to deal with realistic time horizons the heuristic still needs to be improved on this aspect. One major cause is that tasks are measured in hours (in reality in fractions of hours), where the overall planning is in shifts and days, causing the time horizon, measured in hours, to become too large. Several possible solutions can be tested, for example, only scheduling the first week or two weeks in detail and for a longer time horizon only establishing that the maintenance that needs to be performed will fit that time frame.

Restricting the number of technicians per zone in aircraft

This was already modelled in an earlier version of the model. At that point in time, it was decided to stop adding the smaller details to the model and move on with the evaluation of the model and the research into the behaviour of the cost functions, because of time constraints to this project. The constraints needed for the exact model were already modelled; it will be interesting to see how that can be added to the heuristic.

Consider all aircraft in the time frame

The final challenge to solve with a heuristic that was discovered, is that the heuristic does not consider other aircraft that need to be scheduled. A first step could be to take the longest duration from the task list of each aircraft and check if they all fit the time horizon and how much time intervals are left. The excess time intervals could be added to the bin widths of aircraft that have the highest number of man hours, and then again fill up the weeks more evenly. The search space of this improvement could be limited by also implementing the bandwidth for the number of technicians per shift.

7 References

[1] Brenda S Baker, Edward G Coffman Jr, and Ronald L Rivest. "Orthogonal packings in two dimensions". In: *SIAM Journal on computing* 9.4 (1980), pp. 846–855.

[2] Mehmet Başdere and Ümit Bilge. "Operational aircraft maintenance routing problem with remaining time consideration". In: *European Journal of Operational Research* 235.1 (2014), pp. 315–328. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2013.10.066. URL: http://www.sciencedirect.com/science/article/pii/S0377221713008953.

[3] Edward G Coffman Jr et al. "Performance bounds for level-oriented two-dimensional packing algorithms". In: *SIAM Journal on Computing* 9.4 (1980), pp. 808–826.

[4] Thomas A Feo and Jonathan F Bard. "Flight scheduling and maintenance base planning". In: *Management Science* 35.12 (1989), pp. 1415–1432.

[5] Ram Gopalan. "The Aircraft Maintenance Base Location Problem". In: *European Journal of Operational Research* 236.2 (2014), pp. 634–642. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2014.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0377221714000095.

[6] Eva Hopper and Brian CH Turton. "A review of the application of meta-heuristic algorithms to 2D strip packing problems". In: *Artificial Intelligence Review* 16.4 (2001), pp. 257–300.

[7] Mitsutoshi Kenmochi et al. "Exact algorithms for the two-dimensional strip packing problem with and without rotations". In: *European Journal of Operational Research* 198.1 (2009), pp. 73–83.

[8] Nima Safaei and Andrew K.S. Jardine. "Aircraft routing with generalized maintenance constraints". In: *Omega* 80 (2018), pp. 111–122. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2017.08.013. URL: http://www.sciencedirect.com/science/article/pii/S0305048316309410.

[9] Chellappan Sriram and Ali Haghani. "An optimization model for aircraft maintenance scheduling and re-assignment". In: *Transportation Research Part A: Policy and Practice* 37.1 (2003), pp. 29–48.