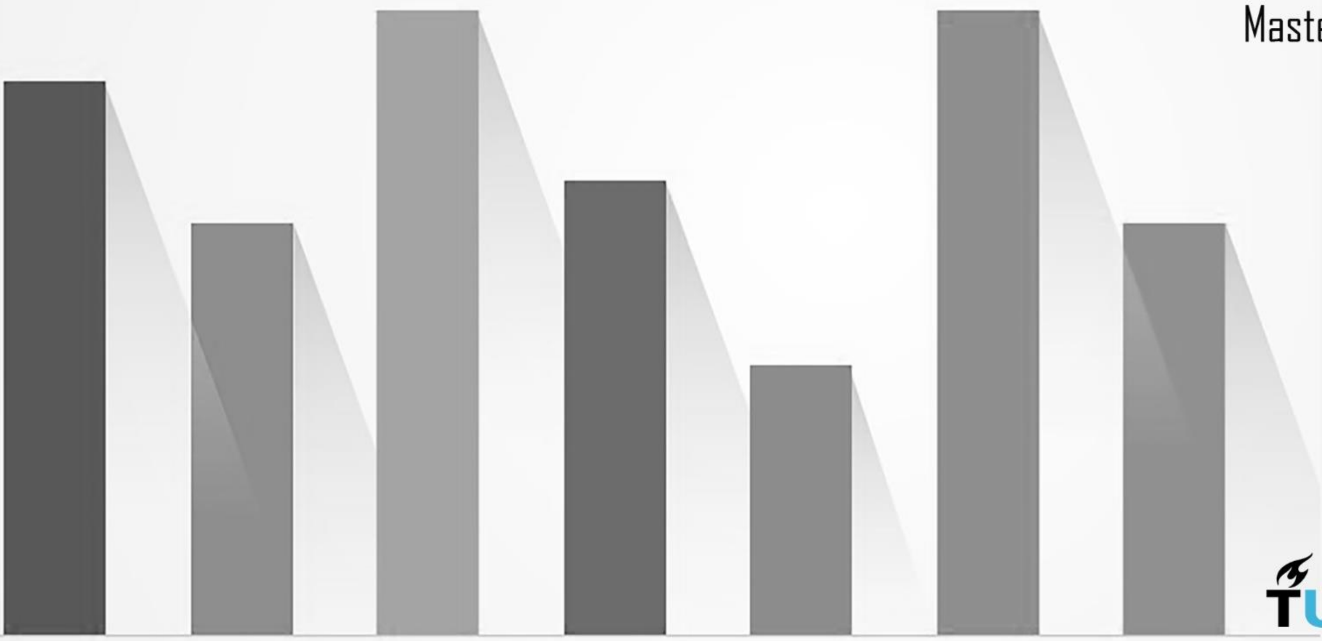
DEVELOPMENT OF AN AIRCRAFT MAINTENANCE SCHEDULE OPTIMISATION METHOD FOCUSED ON THE MINIMISATION OF WORKLOAD FLUCTUATIONS

Delft University of Technology
Faculty of Aerospace Engineering

Master Thesis

A. Conijn

2017



TU Delft

FACULTY OF AEROSPACE ENGINEERING

Department Control & Operations
Section Air Transport and Operations
Gebouw 62
Kluyverweg 1
2629 HS Delft
the Netherlands
Phone +31 (0)15- 27 82058
www.lr.tudelft.nl

Title :	Development of an aircraft maintenance schedule optimisation method focused on the elimination of workload fluctuations
Author :	Anne Conijn
Student number :	1067893
Assignment :	Master Thesis
Initiator :	Prof. dr. R. Curran, Air Transport & Operations, Faculty of Aerospace Engineering, TU Delft.
Supervisor :	Prof. dr. R. Curran, Air Transport & Operations, Faculty of Aerospace Engineering, TU Delft.
Date :	May 23rd, 2017

ABSTRACT

In aircraft maintenance, the initial maintenance schedule of an aircraft is highly dependent on the maintenance planning document. This document, set up by manufacturers, regulators and operators, contains all the requirements for the initial maintenance schedule of a specific aircraft. It serves as a reference for the development of an optimised maintenance schedule by the operator. The initial workload distributions obtained by using the task data in this document, show a high and irregular fluctuation along the operational life of the aircraft. After the application of common task packaging methods, the operator can improve the initial maintenance schedule, but workload fluctuations often persist after these measures. This fluctuation is considered to be unfavourable because of several reasons; maintenance resources are used inefficiently, workforce scheduling operations on fleet level can become very complex and maintenance delays are more likely to occur as a consequence.

In this research, a generic optimisation method is developed focused on the minimisation of the workload fluctuation resulting from the initial maintenance schedule of the aircraft. This method uses a single task reallocation process that systematically eliminates workload excesses by locally reducing task intervals. A software application was built for a fast, a more dynamic and a more customized reallocation process, using several variable inputs.

This application has shown to be very effective in eliminating the workload fluctuation along the lifecycle of an aircraft. As an indication, from the data of the maintenance planning document of an A330, the average fluctuation of the workload among the C-checks over a lifecycle of 15 years, was reduced by more than 79%. Theoretically, this minimisation of the fluctuation could lead to a workforce size reduction of 41% for the C-checks only. These figures are even higher for the A-checks of the aircraft over the same period.

Together with the simplification of the workforce scheduling activities and a more efficient usage of maintenance resources with constant workloads, these advantages are thought to outweigh the negative effects associated with the task interval reductions in the reallocation process, especially when an entire fleet of aircrafts is considered.

Through the implementation of process improvements and additional features, the efficiency and the generic capacity of this application can be further improved with the objective to rapidly generate optimal maintenance schedules that are adaptable to different operational scenarios or constraints.

AKNOWLEDGMENT

I would like to express my gratitude to Prof. Richard Curran for his support, flexibility and most of all his patience given the somewhat different circumstances I was in during this research.
Working, raising a kid and writing a thesis at the same time wasn't very easy and progress was rather slow.
I also thank Prof. W. Verhagen for giving me his knowledge on aircraft maintenance in general, and H. Lucas and E. van Veggel from the KLM Engineering and Maintenance department for providing me detailed service data and insights whenever I needed it.
Special thanks to my uncle, Hans Conijn, who helped me considerably in building the software application developed during this research.

TABLE OF CONTENT

LIST OF FIGURES & TABLES

ABBREVIATIONS & TERMINOLOGY

Abbreviations

- AFR: Average Fly Route
- AU: Average Utilisation of the aircraft (in %)
- ATA chapters: a way of categorizing the various systems on an airplane (originally created by the Air Transport Association in 1956).
- FAA: Federal Aviation Authority
- FC: Flight Cycles
- FH: Flight Hours
- HSC: Hangar Slot Capacity
- IRL: Interval Reduction Limit
- I: Interval
- MPD: the maintenance planning document
- MRBR: Maintenance Review Board Report
- MSG: Maintenance Steering Group
- mh: man hours
- T: Threshold
- TC: Type Certification

Terminology

- Absolute additional labour: the additional labour generated after optimisation. This figure is calculated by simply subtracting the total labour before optimisation from the total labour after optimisation.
- Available capacity: number of man hours that can be attributed to a new period without exceeding the HSC.
- Average excess: average amount of man hours that exceeds the HSC over the entire time span analysed.
- Letter check: base maintenance opportunity. Typically, these are the A-, B-, C- or D-checks an aircraft undergoes during its lifetime.
- Maintenance program: maintenance schedule for a specific group of components of the aircraft. The MPD is typically divided into 5 maintenance programs; the Systems and Components, the Auxiliary Power Unit, the Power Plant, the Zonal and the Structures program.
- New period: new maintenance opportunity to which a task is reallocated.
- Operational parameters/settings: the values of AU and AFR for the aircraft.
- Relative additional labour: the difference in total labour between before and after optimisation, calculated as a function of the interval reduction (see eq. 11).
- Task sequence: one occurrence of a task with a specific interval. Over the lifetime of the aircraft, a task will occur several times.
- Task occurrence: the event of a task taking place in the lifecycle of the aircraft.
- Total labour: the sum of all man hours from the tasks taking place at a specific period.
- Turn-around time: the period of time between the arrival and departure of an aircraft at the gate.
- Variable interval: the intervals that are expressed in FH, in FC or in a combination of them, and therefore depend on the operational parameters of the aircraft.

- Symmetrical tasks: identical tasks performed on each side of the plane of symmetry of the aircraft.

1 INTRODUCTION

In commercial aviation, aircraft maintenance has often been a subject of improvement studies. The large contribution of maintenance activities to the total operational costs of an airline and the scheduling challenges faced when a fleet of aircrafts and limited maintenance resources are considered, have been important drivers behind these studies.
In general, the primary objectives of aircraft maintenance optimisation consist of reducing the number and duration of operational interruptions needed for maintenance and/or increasing the efficiency of activities and resource usage on the ground. By optimising the initial maintenance schedule of an aircraft, significant gains can be obtained for these objectives.
The initial maintenance schedule for a specific aircraft is determined by its maintenance planning document (MPD). This document, developed by manufacturers, operators and regulatory agencies, serves as a reference for developing an optimal maintenance schedule for the aircraft. It contains all the required information for ensuring the airworthiness and safety of the aircraft, including the intervals with which preventive tasks need to be carried out and the man hours needed to complete each of them.
After the conversion of the task data in the MPD, the intervals of the tasks turn out to be very diverse and spread out. Therefore, if the intervals are left unchanged, the operations of the aircraft would need to be interrupted for maintenance almost daily. Fortunately, most of these intervals can be adjusted by the operator, if the initial intervals prescribed by the MPD are not exceeded. This way, the maintenance schedule can be adapted to the utilisation pattern of the aircraft and/or to the availability of maintenance resources on the ground. After generating sufficient reliability data from the aircraft operations, some task intervals may even be escalated beyond their initial value (from the MPD) but only after the approval by regulatory agencies.
The most common method used by an operator to optimise the initial maintenance schedule of an aircraft, consists of packaging tasks together based on intervals. These task packages are then carried out at periodic maintenance opportunities (i.e. letter checks or line maintenance at the gate). However, this optimisation, also called the block check method, produces a considerable fluctuation in workload among the periodic maintenance opportunities during the lifecycle of the aircraft. This fluctuation is considered very unfavourable and eliminating it from the workload distribution is thought to create substantial advantages; the average workforce size can be reduced, the maintenance resources can be used more efficiently and delays might be easier to avoid. In the fleet context where hangar slot planning and workforce scheduling can get complex, the advantages of a ‘flat’ and predictable workload distribution would have considerable benefits as well.

In this research, a generic and efficient method for the optimisation of the initial maintenance schedule of the aircraft is developed, aiming at eliminating workload fluctuations along the lifecycle of the aircraft. Using the MPD as a basis, the development of this optimisation method consists of three main steps: the task interval conversion, the generation of the initial workload distributions and reallocation of individual tasks leading to final optimised maintenance schedule (and workload distribution) of the aircraft. Each of these steps will be developed in such a way that the different inputs and parameters can easily be changed to adapt to different maintenance or operational scenarios, or to the MPD of a different aircraft.

This report contains the following chapters:

- 2. Background information: relevant information on aircraft maintenance program/schedule development.
- 3. Literature review: analysis of the relevant researches on maintenance schedule optimisations and an analysis of common optimisation practices.
- 4. Problem statement: explanation of the disadvantages of a fluctuating workload, determination of the objectives and the approach/method that will be used.
- 5. MPD analysis & preliminary workload distributions: detailed analysis and first conversion of the MPD data to obtain the preliminary workload distributions for an A330. The dependence of the workload distribution on changing operational parameters will also be highlighted here.
- 6. Generation of the initial A- and C-check workload distributions: since the preliminary workload distributions of chapter 5 do not consider the availability of maintenance opportunities whenever a task occurs, a generic model has been developed in this chapter to produce realistic initial workload distributions depending on several criteria.
- 7. Optimisation: this chapter starts with a detailed explanation of the single task reallocation principle. A manual optimisation using the single task reallocation is then carried out to analyse how the reallocation process unfolds and to determine what parameters play important roles in it. This will be used to develop an automated reallocation model in Visual Basic.
- 8. Results and Analysis: the optimised workload distributions resulting from the previous two chapters are shown and analysed in various charts.
- 9. Evaluation: this chapter determines its main advantages and shortcomings of the optimisation method developed so far. Potential improvements are suggested as a last part in this chapter.
- 10. Conclusion

2 BACKGROUND INFORMATION

To properly understand the development of aircraft maintenance schedules, this chapter summarizes the important processes, organisations and documents involved in it.

The Maintenance Steering Group

In the early days, the maintenance requirements and schedules of an aircraft were primarily determined and developed by pilots and technicians. But the introduction of commercial aircrafts required new regulations and the involvement of regulatory authorities. Regulations were put in place and maintenance schedules were developed to monitor the reliability and safety of the aircraft. The aircraft manufacturer was the main responsible for these maintenance program/schedule developments. Time limitations were defined and the entire aircraft was periodically overhauled to ensure a high level of safety. This led to the first maintenance process called Hard-Time maintenance, a process in which components are taken out of service after reaching a specified age expressed in flight hours (FH), flight cycles (FC), calendar time units (years, months, days). These components were then repaired or replaced.

In 1960, the Federal Aviation Authority (FAA) determined that a second maintenance process was necessary for ensuring high safety in commercial aviation: the On Condition maintenance process. On Condition maintenance requires components and subsystems to be regularly inspected to assure they can continue service. The main purpose of this maintenance process is to replace or repair a component before failure occurs during normal operation.

In 1968, the handbook "Maintenance Evaluation and Program Development" also referred to as the MSG-1 was developed for the B747 by the Air Transport Association's (ATA) Maintenance Steering Group (MSG). The MSG-1 process used a decision logic to develop scheduled maintenance. This program used both Hard Time and On Condition maintenance processes for the development of aircraft routine maintenance tasks.

In 1970, MSG-1 was updated to MSG-2 that introduced the third type of maintenance process: the Condition Monitoring process. Here, instead of scheduled repairs or replacements, the mechanical performance (i.e. vibration, oil consumption etc.) of the component is monitored and compared to normal operation levels. If this performance exceeds predetermined levels, the component is replaced to prevent failure in the future.

In 1978, MSG-2 was improved by the ATA to address newly developed and advanced aircrafts such as the B757 and the B767. The task force discovered shortcomings in the MSG-2 decision logic (it was not suitable for the ever-increasing complexity of aircrafts, it did not address problems related to stress tolerance and fatigue of structures, and it required many components to be tracked individually and it did not differentiate between maintenance done for safety reasons and maintenance done for economic reasons). This led to the development of the MSG-3, a new task-oriented maintenance process in which activities are assessed at system level rather than at component level. Under MSG-3, maintenance tasks are grouped into 3 maintenance programs: systems & power plant, structures and zonal.

MSG-3 is today the only maintenance methodology accepted by airworthiness authorities. It generally produces higher safety standards and it represents a more intelligent approach for the determination of maintenance tasks and intervals.

Type Certification (TC) and Maintenance Review Board (MRB)

When a new aircraft is introduced into service, all the initial scheduled maintenance requirements are determined though two different processes: the Type Certification (TC) and Maintenance Review Board (MRB) process. Through the TC process, maintenance requirements are derived to ensure that the design of the aircraft satisfies the safety levels required. These requirements are contained in the Airworthiness Limitation Section (ALS) which includes:

- The Safe Life Airworthiness Limitation Items (document on the life limited parts);
- The Damage Tolerant Airworthiness Limitation Items (ALI document for structures);
- The Certification Maintenance Requirement document (for systems);
- The Ageing Systems Maintenance document (ASM);
- The Fuel Airworthiness Limitations document (FAL).

Each of these documents needs to be approved independently.

Through the MRB process, the manufacturers, the operators, the vendors, the authorities and the industry work together in the development of the initial scheduled inspection requirements for the aircraft and the power plants. These minimum scheduled requirements are specified in the Maintenance Review Board Report (MRBR) which also results from the MSG-3 method.

There are 3 main groups involved in the process of creating and updating the MRBR:

- The Maintenance Review Board (MRB): the organisation that is responsible for the final approval of all the initial scheduled maintenance tasks for a particular aircraft. Its members are representatives of operators, manufacturers (of airframe and engine) and regulatory authorities.
- The Industry Steering Committee (ISC): its members include a select number of representatives for both operators and manufacturers. They are responsible for the activities involved in the development of scheduled maintenance, such as the determination of policies, goals etc. and the preparation of the final recommendations for the MRB.
- The Maintenance Working Groups (MWG): these include maintenance specialists from regulatory authorities, operators and equipment manufacturers. Their purpose is to apply the MSG-3 logic in order to propose maintenance tasks and intervals for a particular aircraft.

Figure 1 summarizes the different processes, organisations, documents etc. involved in the development of an aircraft maintenance program.

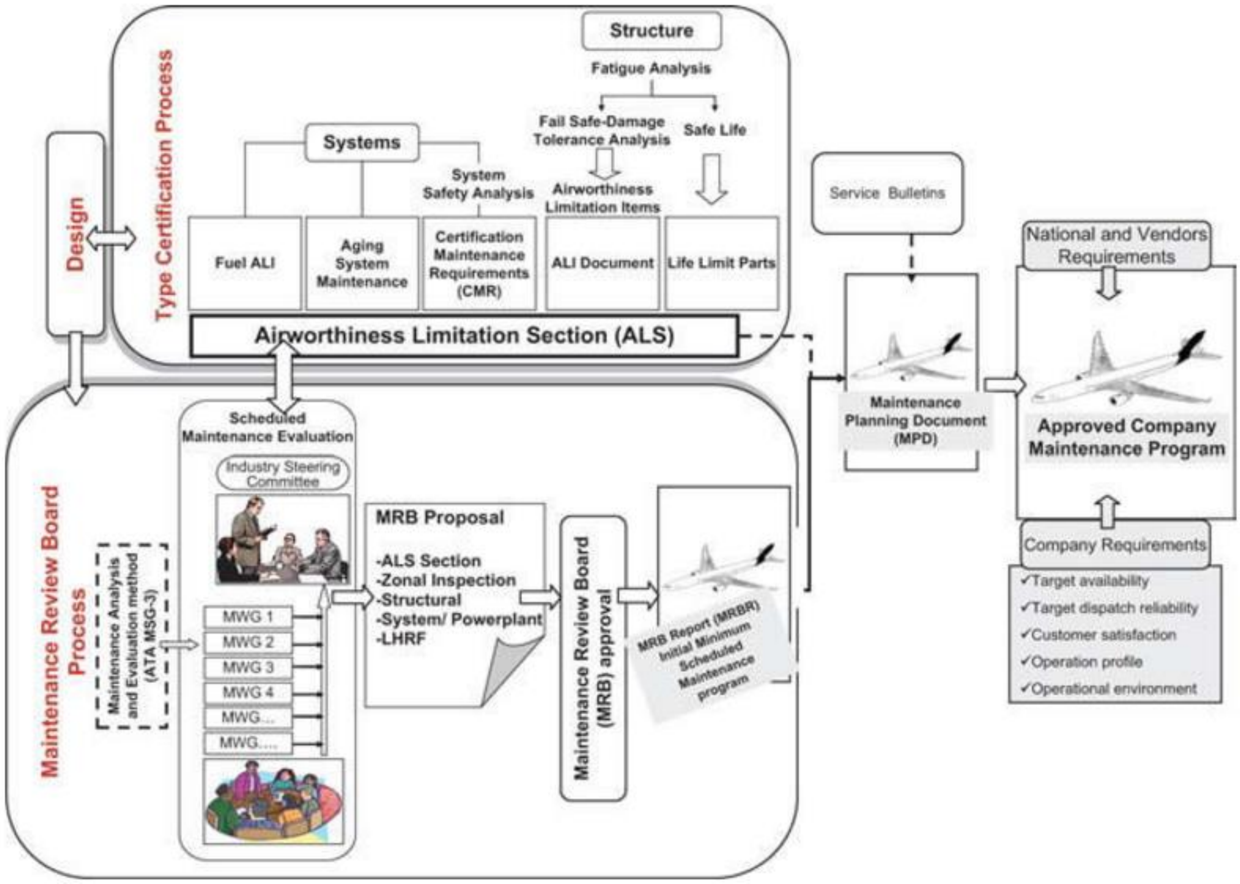


Figure 1: Process mapping of aircraft maintenance program development, A. Ahmadi et al. (2010).

Maintenance Planning Document

After receiving the final recommendations from the MRB on how the aircraft should be maintained, the manufacturer publishes the Maintenance Planning Document (MPD). This is the reference document used for the initial maintenance program of the aircraft operator. It contains all the MRB requirements found in the MRBR, as well as the mandatory scheduled maintenance requirements that can only be changed after approval of the relevant airworthy authority.

The main objective of Maintenance Planning Document (MPD) is to provide the necessary maintenance planning information for the operator to develop its own customised maintenance schedule. However, it is the final responsibility of the operator to decide when to carry out the tasks, except for those indicated as CMR or AL. Additional requirements such as service letters and bulletins and airworthiness directives, vendor manuals etc. also need to be considered. In the end, all those additional requirements make up the Operator's Approved Maintenance Program (OAMP) (see Figure 1).

Typical aircraft maintenance organisation

In a general way, aircraft maintenance can be split in line maintenance and base maintenance. Line maintenance consists of all the maintenance activities for which the aircraft does not need to be taken out of service. These tasks are typically performed each time the aircraft is at the gate during turn-around time. They mostly include general visual inspections of the exterior of the aircraft (i.e. oil leaks, wear of tires, fuselage impacts).

Base maintenance on the other hand, consists of all maintenance tasks for which the aircraft needs to be taken out of service. These include extensive and time-consuming activities carried out during maintenance/hangar slots. This heavier maintenance is typically organised into so called letter checks: A, B, C or D-checks. These maintenance inspections differ from each other in terms of intervals, the type of tasks carried out and the amount of labour required to complete them. Depending on the maintenance strategy of operator and the aircraft type, the intervals and workloads of these letter checks may vary.

		Interval	Location	Average duration
Line maintenance		n.a.	gate	up to 1 hour
Base maintenance	**A-check**	600 FH	hangar	10 hours
	B-check	3 to 6 months	hangar	10 to 24 hours
	C-check	18 to 20 months	hangar	Up to several weeks
	D-check	72 months	hangar	Up to 1 month

Table 1: Typical maintenance check data for the A319-21 aircrafts (Transport Studies Group University of Westminster London, 2008)

A-checks are usually carried out within several hours. Next to general inspections of the interior and the exterior of the aircraft, services checks, engine and functional checks are also included. The B-check can be considered as a more detailed A-check, with approximately of the same type of tasks. In modern maintenance programs however, the B-check is often eliminated from the MPD and incorporated into A-checks.

C-checks represent the heavier maintenance checks in the maintenance schedule of an aircraft. They can take up to several weeks. A regular C-check includes inspections of the outside and the inside of the aircraft, as well as detailed examinations of structures and load bearing components on the fuselage and wings.

D-checks are the heaviest maintenance events in the lifecycle of an aircraft. They can take up to several weeks. During these inspections, the aircraft is completely taken apart, all components are thoroughly checked, the paint is stripped and large outer panels are removed uncovering the airframe, the carrying structure, wings and other important items. In addition, systems and components are functionally checked, repaired and replaced if necessary.

These different letter checks usually take place at the maintenance facility of the operator, during so called hangar/maintenance slots; a predefined interval of time assigned to an aircraft for maintenance. In the hangar, only a limited number of aircraft can be maintained simultaneously. The efficient organisation and planning of maintenance activities and resources for an entire fleet of aircraft is therefore crucial to limit downtime and prevent unnecessary costs.

Recent trends in the aviation industry show that the letter check organisation of base maintenance activities is thought to be too rigid. That is why letter checks are gradually eliminated from maintenance programs. This way, more flexibility is given to the operator for determining the moments at which maintenance activities need to be carried out.

3 LITERATURE REVIEW ON MAINTENANCE SCHEDULE OPTIMISATIONS

For a proper understanding of the subject of this research, the important organisations, processes, documents etc., involved in the development of the maintenance program of an aircraft are initially described in this chapter. This is followed by an explanation of the problem and the formulation of the main research question. At last, the objectives and the approach of this research are defined.

Maintenance schedule optimisation has been a popular research topic in various industrial sectors. Scheduling problems can be very diverse and therefore, numerous studies and researches have been conducted in this field. Simulations and algorithms are often used to generate optimal solutions that reduce labour costs and production times.

When it comes to optimising the base maintenance schedule of an individual aircraft, two main theoretical schedule rearrangements seem to be considered; the block check approach and the equalized or phased approach.

The block check approach

The block check approach focuses on packaging tasks together according to their interval in order to attribute them to the most appropriate maintenance opportunity (the letter checks). This method offers a straightforward optimisation of the aircraft's initial maintenance schedule. The task packages are rather large (increased down time per maintenance opportunity) and the labour demand fluctuates with a regular pattern.

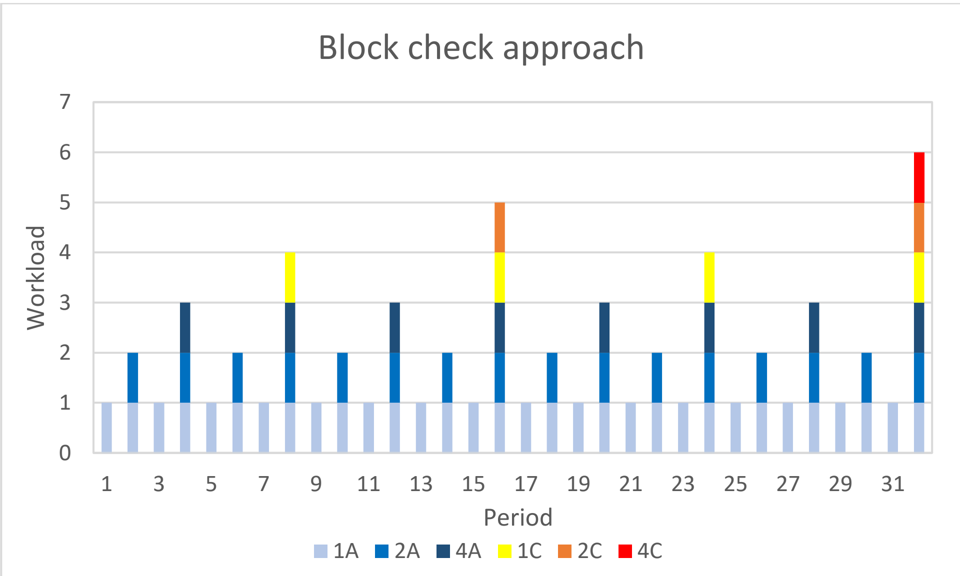


Figure 2: Illustration of the block check workload distribution

Regarding this approach, the research of A.K. Muchiri (2002) conducted at Transavia in 2002, is of particular interest. In this research, the researcher further developed the block check approach to obtain a maintenance planning and packaging method specifically adapted to a new fleet of B737 NG purchased by Transavia in 2002.

This new maintenance planning and packaging method was obtained in three phases. Tasks were first separated into line and base maintenance tasks, which were then sorted into the so-called maintenance packages based on their interval (see Figure 3). For line maintenance, tasks were also packaged according to common hangar set-up activities (the preparation work prior to carrying out the tasks).

Task Package	MPD Interval*	HV Check Notation	Boeing Man-hours	MSG-3 Category	MSG-3 Class.	Skill	ATA 100 Chapter	(MRIs)
C4000A*	4000CYC or 540 DAYS	38C4000*	1.85	5DT, 2GV	5, 6	M	20. 32	7
C4000B*	4000CYC or 540 DAYS	38C4000*	1.08	GV	0	M	52	10
C4000C*	4000CYC or 540 DAYS	38C4000*	2.49	GV	0	M	53	12
C4000D*	4000CYC or 540 DAYS	38C4000*	4.16	GV	0	M	53	9
C4000E*	4000CYC or 540 DAYS	38C4000*	0.59	GV	0	M	53	3
C4000F*	4000CYC or 540 DAYS	38C4000*	0.54	GV	0	M	53, 54	4
C4000G*	4000CYC or 540 DAYS	38C4000*	3.96	GV	0	M	55	23
C4000H*	4000CYC or 540 DAYS	38C4000*	11.21	GV	0	M	57	54
8 packages			*25.88*					*122*

Figure 3: Task packaging by interval and common set-up activities (A.K. Muchiri, 2002)

In a second phase, a detailed analysis of the actual utilisation patterns of the fleet was carried out. The objective here was to determine when potential maintenance opportunities could take place. Within Transavia, the aircraft utilisation is highly dependent on the seasons. Moreover, in low season, part of the fleet is leased to other companies. These aspects complicated the determination of a new appropriate maintenance schedule.

In a third phase, the previously defined task packages (line maintenance) and maintenance checks (base maintenance) were grouped into maintenance clusters with a specific interval and a total man hour demand. This was done with a use of a Maintenance Item Allocation Model (MIAM) which was developed with Visual Basic and Excel.

The model simulates different aircraft operational scenarios based on acquired field data and generates the due dates of the maintenance task packages.

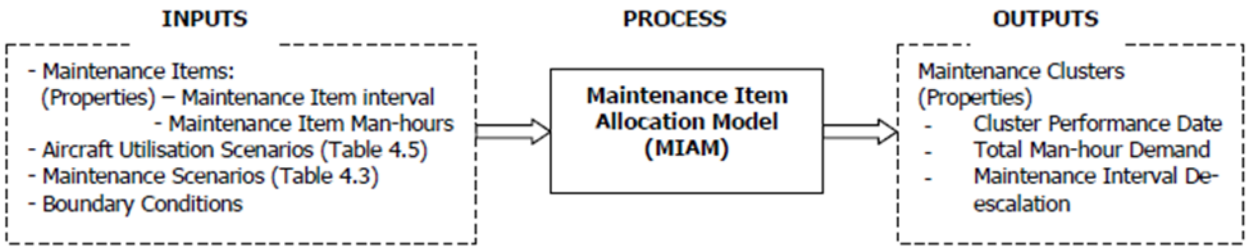


Figure 4: Clustering process description (A.K. Muchiri, 2002)

Thanks to the efficient allocation of task packages and maintenance checks, the interval de-escalation associated with the initial optimisation (intervals cannot be escalated unless this is authorized by specific authorities, see chapter 0) could be reduced, which also reduced the required labour.

The equalized approach

In the equalized or phased approach, a maintenance schedule is optimised by creating constant, and sometimes more frequent task packages. In this approach, an operator can for example spread out the tasks of a larger C-check over several preceding A-checks to balance the workload and prevent large operational interruptions.

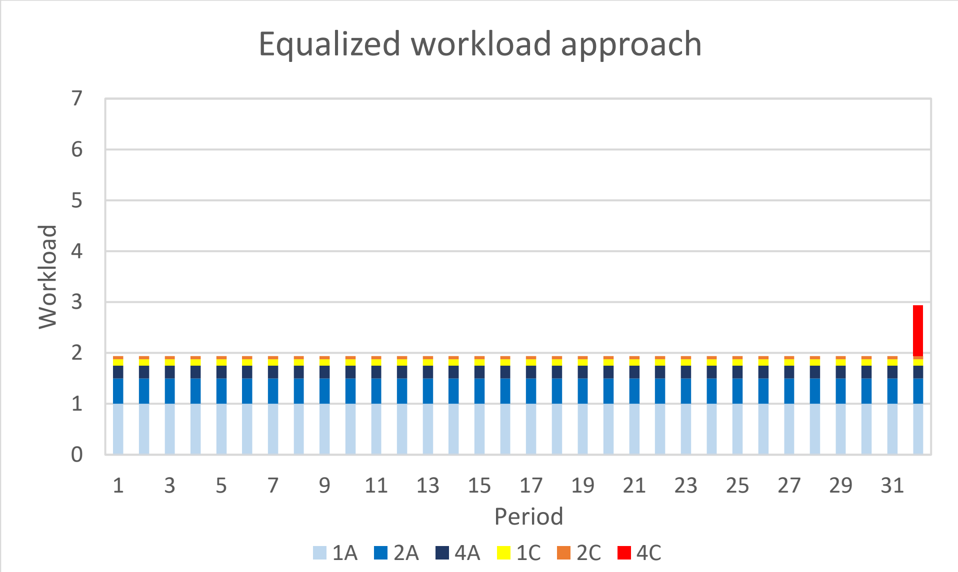


Figure 5: Illustration of the equalised workload distribution

In general, this type of optimisation offers more predictability and flexibility, but it also leads to some task intervals being reduced far beyond the optimal interval. This increases maintenance costs at the beginning of the aircraft's lifecycle.

In the early 2000s, this concept was developed in a cooperation between Airbus and Easyjet. So-called E-checks were implemented, a combination of A- and C-check tasks clustered into constant and frequent task packages. Since Easyjet was not allowed to fly at night due to noise regulations, the E-checks were carried out overnight and maintenance downtime was reduced effectively. More information on these applications or similar versions of the equalized approach was unfortunately difficult to find.

Similarly to the equalized workload approach, C. Şentürk et al., 2010carried out a research in which the objective was to reduce maintenance downtime by also spreading the workload evenly. By using an optimisation concept supported by a fuzzy analytic hierarchy process (AHP), the researchers tried to offer a scheduling solution different from the classical and rigid A, B, C, D-check base maintenance schedules, in an attempt to reduce the number of maintenance days over a cycle of 5 years. This research was conducted within the maintenance department of an operator whose airplanes were not always in service and often simply parked at an airport when they were not scheduled for operations. These frequent periods of inactivity were used by the researchers to implement the constant workload packages. The main theory of the authors was that the ground time of heavy maintenance checks could be reduced by carrying out high level maintenance on a continuous basis thanks to the advanced diagnostic prognostic systems of modern aircraft and the integrated maintenance IT solutions available today. In an attempt to further reduce the maintenance downtime, the authors stated that the selection best qualified staff for carrying out specific maintenance tasks would be important. To support this selection procedure, a fuzzy AHP model was built which used a list of criteria (licenses,

qualifications, experience etc..) and a set of weighting factors to carry out the optimal selection of the maintenance staff.
In the end, the researchers claimed that only 14 days of maintenance were required compared to the initial 29 days (see Figure 6).

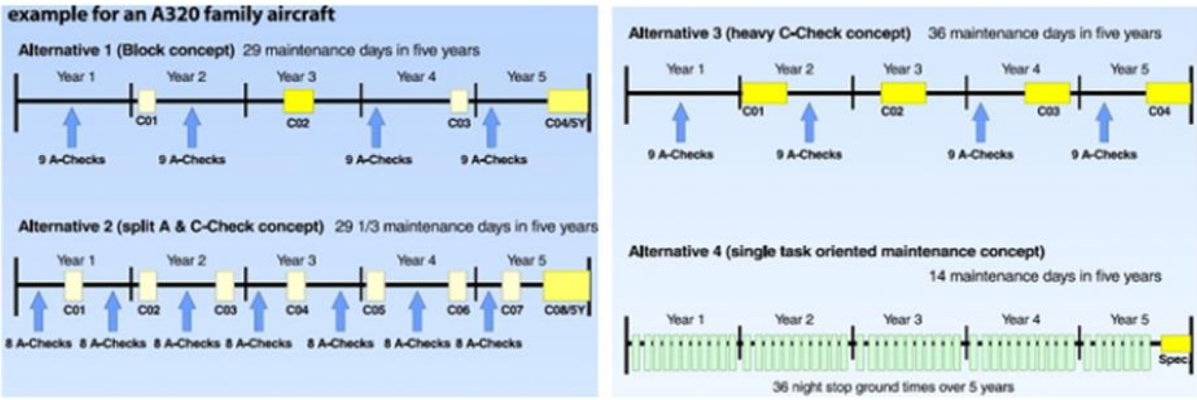


Figure 6: Illustration of the equalised workload distribution (C. Şentürk et al.,2010)

The use of algorithms to solve complex scheduling situations

Many schedule optimisation subjects encountered in the literature, involved job-shop scheduling type of problems. For most of these problems with multiple constraints and variables, simulations, software models or algorithms were used.
Even if these job-shop type of subjects differ from the maintenance scheduling subject in this report, the ways in which algorithms and simulations are used to solve these problems, might be interesting. One example of such a research in the aviation industry was one conducted by J. Beliën et al. (2012) at Sabena Technics, an aircraft maintenance firm that offers a complete range of maintenance services across Europe, including heavy maintenance (C & D-checks), light maintenance (A & B-checks) and line maintenance. The research consisted of the application of an enumeration approach based on mixed integer linear programming (MILP) to solve a workforce staffing and scheduling problem at the maintenance department of the company. This research only addressed the line maintenance part of the company's activities at Brussel's airport.
At the firm's base, the varying demand in labour - different aircrafts arriving at different moments in time with different tasks to be completed - had to be met by teams of technicians organised in two different cycles, each with different work shifts. The main objective was to build new rosters for line maintenance such that the maintenance capacity always satisfied the maintenance demand while minimizing maintenance costs. These rosters were subjected to various constraints; work shift constraints of all sorts, a maintenance demand that may not exceed the maintenance capacity in order to avoid delays and a constraint from the company's management that preferred to work with a restricted number of fixed shifts to simplify the tasks of the scheduler.

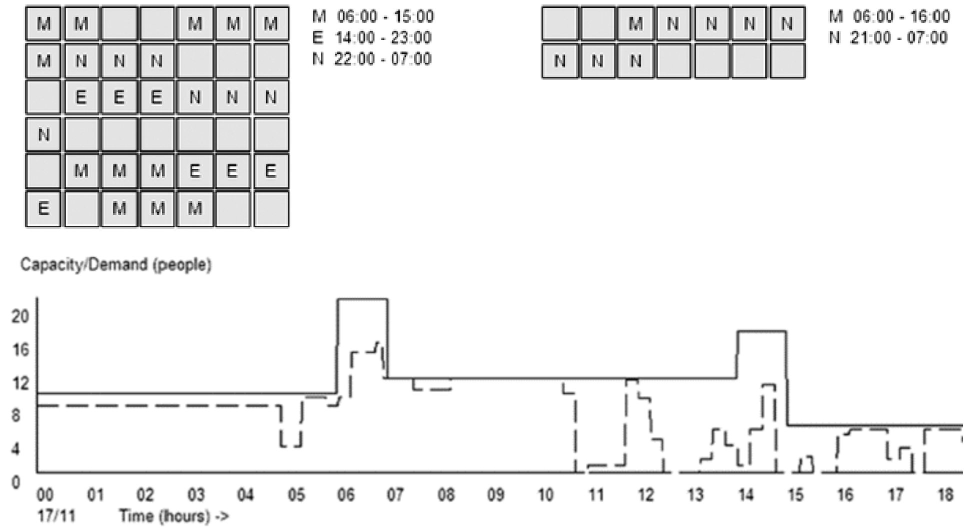


Figure 7: Illustration of the maintenance cycles/shifts and the fluctuation of maintenance demand vs. capacity (J. Beliën et al., 2012)

Although this scheduling problem differs from the one in this report, it is related to it; after the determination of the final maintenance schedule of an individual aircraft, the implementation of an adequate workforce schedule for the completion of the maintenance program will be required.
This research illustrates the potential complexity of workforce scheduling problems but it also illustrates the need for advanced problem-solving techniques, simulation, software etc. when multiple constraints and variables are involved.

Synthesis

In this literature review, the two main theoretical approaches for the optimisation of aircraft's maintenance schedule have been discussed. The block check method seems to be the most common of the two. As A.K. Muchiri (2002) demonstrated, this method can be further improved with a more accurate planning and clustering of tasks and maintenance opportunities. Even though gains were obtained, this method does not solve the problem of the workload fluctuation. Moreover, the determination of common set-up activities which are not indicated in the MPD, needs to be done 'manually'. This limits the efficient application of the model to other aircrafts with different MPD's.

The equalized workload approach seems to be a straightforward solution for eliminating the workload fluctuation. The advantages of this approach are identical to those aimed at in this research. But it seems that this method best suits the operators for which aircraft operations are frequently interrupted (these interruptions were used as maintenance opportunities). In addition, the method assumes that C-check tasks can simply be assigned to A-checks which may not always be the case since the resources available during A-checks may not be sufficient for large and complex C-check tasks. But even if this is possible, it should be noted that the great diversity of task intervals and sizes in the MPD complicates the simple and equal spreading of tasks illustrated in Figure 5. Unfortunately, this method has rarely been applied

in the field and therefore, little is known on its implementation and the potential problems associated with it.
Although the block check approach seems to be the standard when it comes to optimising the maintenance schedule of an aircraft, the equalized approach could be a promising alternative to further develop. But, considering the enormous amount of data in the MPD of an aircraft and the possible constraints and variable parameters involved in the development and application of an optimisation method, the implementation of some sort of automatization, simulation and/or algorithm will probably be required.

4 PROBLEM STATEMENT

4.1 PROBLEM DESCRIPTION & RESEARCH QUESTION

For this aircraft, base maintenance consists of A- and C-checks only since the B and D-checks have been eliminated from the MPD.
The MPD contains two types of tasks;

- the variable tasks, with intervals expressed in flight cycles (FC)and/or flight hours (FH) and therefore dependent on the utilisation pattern of the aircraft;
- the fixed tasks, with intervals expressed in either calendar units (years, months or days) or in C-checks which equals 18 months (the interval of an A-check corresponds to 600 FH, which makes it a variable interval).

Most of the tasks in the MPD are variable tasks. Since the intervals of the variable tasks are so diverse, the initial occurrence of these tasks prior to any sort of schedule rearrangement is very spread. If no interval rearrangement takes place, the operations of the aircraft will need to be interrupted almost on daily for maintenance. To avoid this, the most common schedule optimisation method used by airline operators is called the block check packaging method. This method clusters tasks together based on their interval. These task packages are then carried out during line or base maintenance opportunities.

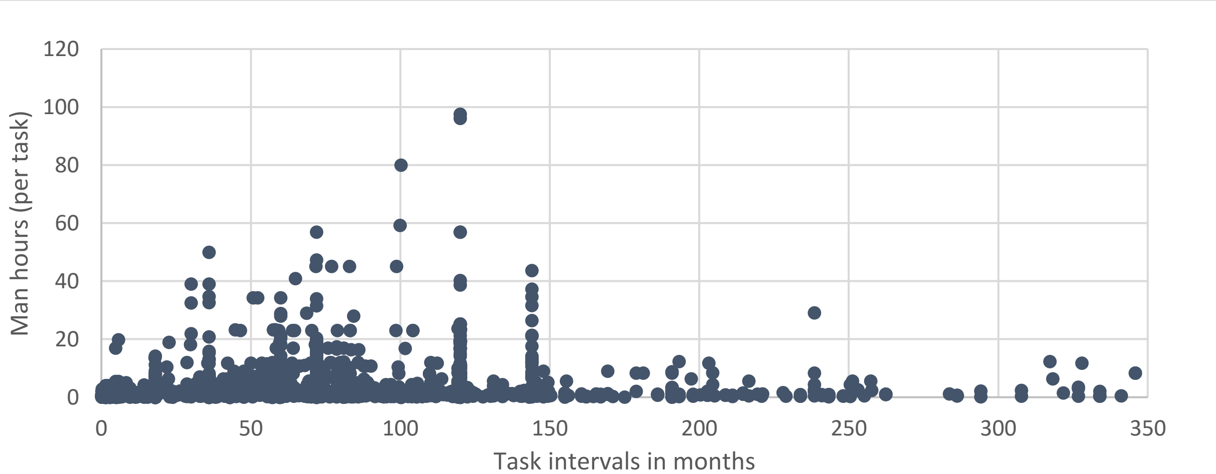


Figure 13: The spread of tasks resulting from the MPD data (A330)

This relatively simple method limits the number of unnecessary operational interruptions in between maintenance opportunities. This method can be further improved as in the research conducted by (Muchiri, 2002) in which maintenance task packages were not only based on intervals but also on common required activities and where a detailed analysis of aircraft utilisation and hangar slot availability resulted in optimal intervals for these task packages.
The disadvantage of the block check packaging method is that it creates task packages with sometimes strongly varying workloads; the workload fluctuates from one task package to the other along the lifecycle of the aircraft.

The chart below represents the workload fluctuation for the A-checks of a single aircraft, resulting from the MPD data. The variable tasks have been allocated to the closest preceding A-check.

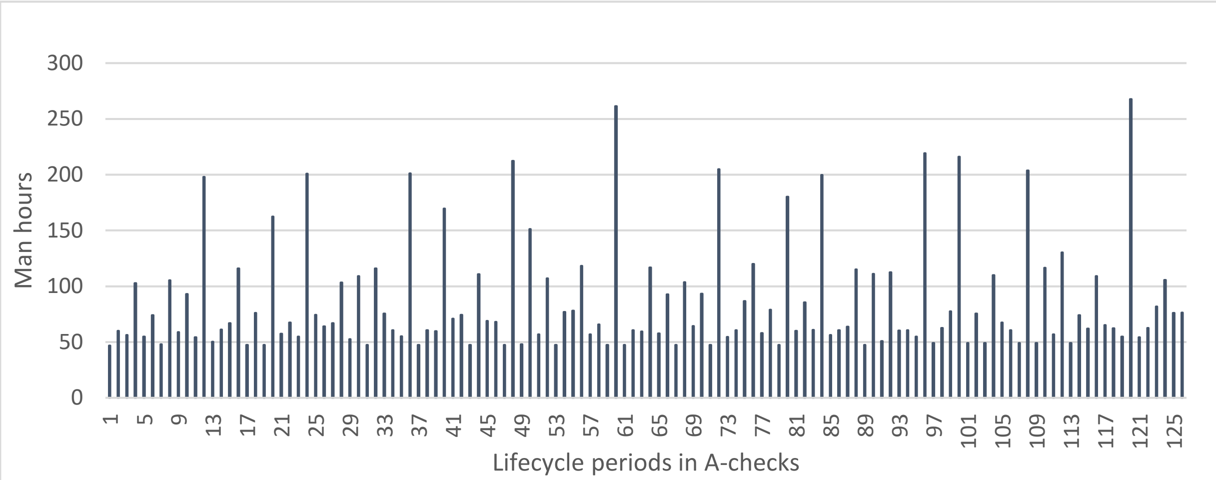


Figure 29: Initial A-check workload distribution over 15 years (A330)

This considerable workload fluctuation is considered to be unfavourable because of several reasons. The simple illustration of Figure 8 helps to better understand the problem.
In this illustration, four aircrafts of the same type have been allocated to different maintenance slots for the completion of a planned A-check. For this example case, the following assumptions are made:

- All aircrafts have an identical operational pattern and therefore an identical workload fluctuation during their lifecycle (see Figure 29);
- The aircrafts have been introduced into operations at different moments in time;
- Each slot has an identical maximum duration of 24 hours.

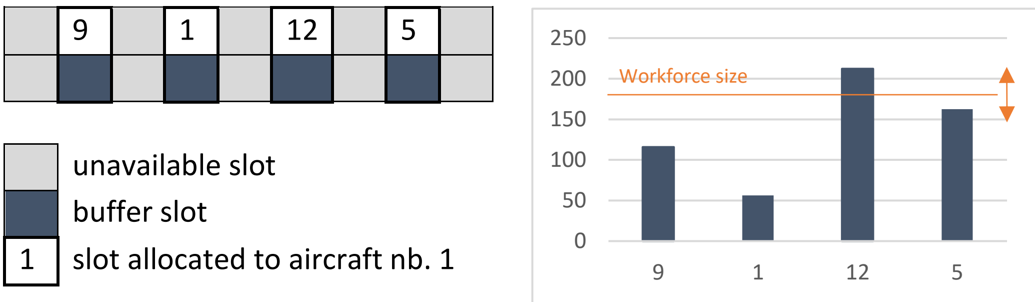


Figure 8: Illustration of the allocation of maintenance slots to aircrafts and the corresponding workload variation between each A-check slot

Aircraft nb. 9, 1, 12 and 5 are scheduled for their 16th, 2nd, 48th and 20st A-check respectively. Due to the workload fluctuation illustrated in Figure 8 and the different moments at which the aircrafts have been introduced into operations, the size of the workload can vary considerably, from one slot to the other. Ideally, the required size of the workforce would be adjusted to the varying workload. This is however considered to be unrealistic, especially when unpredicted delays and the slot allocation changes that result from it are considered. The workforce scheduling organisation becomes even more complex when an entire fleet of several hundred aircrafts is considered.

On the other hand, having a constant workforce size that can cope with the largest A-check encountered during the lifecycle of the aircrafts, is also very inefficient and costly due to the underutilization of the available workforce when the workload requirement is low.

A constant workload distribution is therefore believed to be solution for these inconveniences. Since it is possible to modify the interval (only by reducing it) of many tasks in the MPD, it should be possible to optimise the maintenance schedule of the aircraft and reduce or even eliminate the workload fluctuation along the lifecycle of the aircraft. However, the enormous number of tasks in the MPD, the dependence of task intervals and letter check intervals (A-checks) on the operational parameters of aircraft and the repetitive character of the tasks, may form considerable challenges. Moreover, reducing task intervals generates more labour costs on the long term since it increases task occurrence frequencies.

To conclude, the research question can be formulated as follows:

How can a generic and efficient schedule optimisation method and model be developed, focused on the minimisation of the maintenance workload fluctuation along the lifecycle of an aircraft?

In the current research, the term optimisation is used to designate the process of efficiently improving the maintenance schedule of an aircraft by reallocating tasks and with the objective to minimise workload fluctuations.

4.2 OBJECTIVES & APPROACH

The primary objective of this research is the development of a maintenance schedule optimisation method focused on the minimisation of the workload fluctuation. This will be achieved by a task reallocation process carried out through interval modification. The MPD of the aircraft is used as a basis reference for all the initial data of the initial maintenance schedule of the aircraft.

As a secondary objective, this optimisation method needs to be efficient and eventually easily applicable to any aircraft and any operational scenario. To achieve this, the following aspects need to be considered:

- A generic method that can produce and optimising different maintenance schedules according to specific conditions/criteria;
- Limitation of the possible cost increments caused by interval reductions. Reducing the interval of a task increases the occurrence frequency of this task and therefore increases maintenance costs.
- Automation of the optimisation process to cope with the large amount data contained in the MPD;

The approach for the development of this optimisation consists of three main steps. At first, a detailed analysis of the MPD is carried out to investigate the data it contains and in particular the data on tasks. The initial intervals of tasks are calculated and preliminary workload distributions are generated.

As a second step, a generic model is developed through which different initial and realistic workload distributions are obtained. In this model, tasks can be separated or clustered according to specific criteria after which they are automatically allocated to the most appropriate maintenance opportunity. These workload distributions will then be used for the final optimisation process.

As a final step, the optimisation is carried out through a systematic single task reallocation process. This process intends to flatten the initial workload distribution obtained in the previous step. It is first carried out manually to analyse how the process unfolds and to determine what parameters play important roles in the process. A software application has been developed for a quicker and more efficient reallocation.

The reallocation process will be carried out with different settings. All relevant data such as the overall increase in labour caused by the interval reductions, will be tracked and analysed in parallel.

Figure 9 below illustrates the three main steps taken in this research.

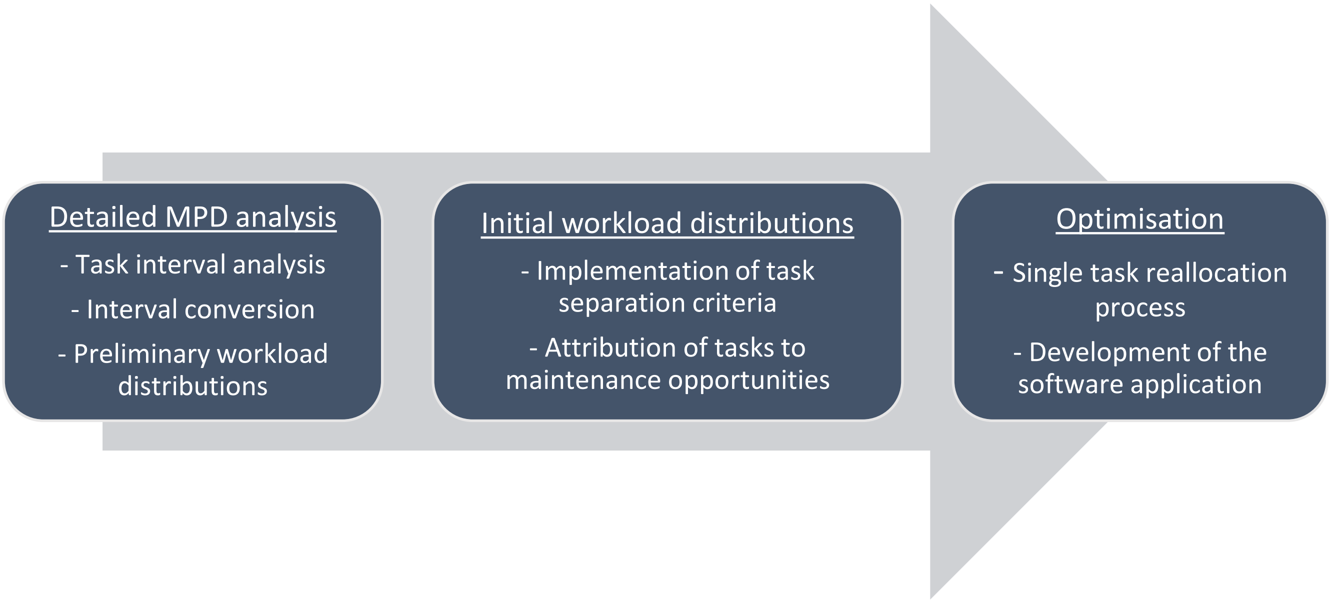


Figure 9: The three main process steps in this research

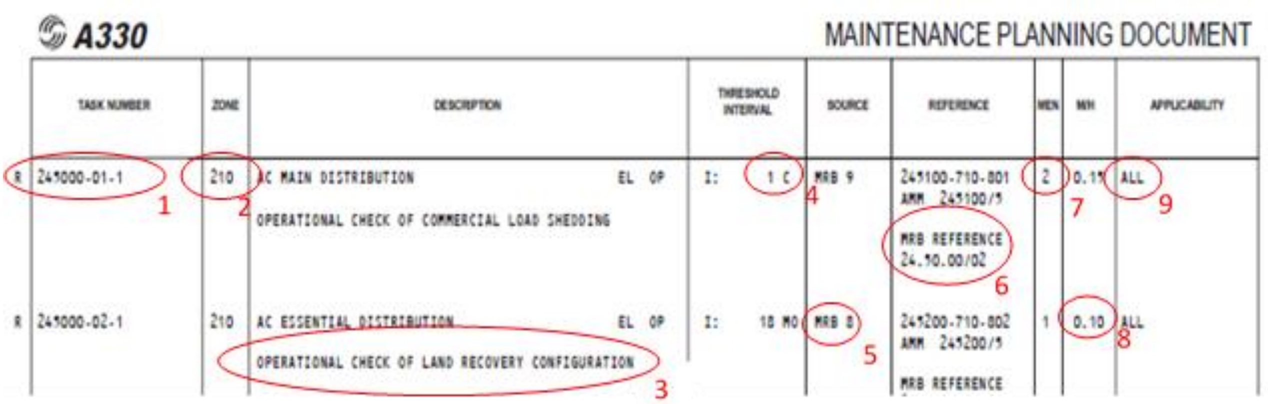
5 MPD ANALYSIS & PRELIMINARY WORKLOAD DISTRIBUTIONS

In this research, the workload distributions that result directly from the MPD, prior to any sort of rearrangement/optimisation, are called the preliminary workload distributions. To generate these preliminary workload distributions, the task data in the MPD needs to be analysed and converted. After obtaining these distributions, the need and the potential of an optimisation will become more obvious.
At first, the important data in the MPD is analysed explained in this chapter. The principles of the conversion of task intervals into months are explained in a second phase. At last, the workload distributions for each maintenance program are presented, as well as the effects of input variations on the distributions.

5.1 THE MPD DATA

As indicated earlier, the MPD in this research contains all the data required for the initial scheduled maintenance program of the A330-200. The data is arranged into six maintenance program groups: the Systems & Components, the APU, the Power Plant, the Zonal, the Structures and the Time Controlled Items programs. In this research, special attention is paid to the maintenance task and flight envelope data since it leads directly to the conversion of the task intervals into months.

5.1.1 The maintenance tasks


A330 MAINTENANCE PLANNING DOCUMENT

	TASK NUMBER	ZONE	DESCRIPTION			THRESHOLD INTERVAL		SOURCE	REFERENCE	MEN	MH	APPLICABILITY
R	245000-01-1	210	AC MAIN DISTRIBUTION OPERATIONAL CHECK OF COMMERCIAL LOAD SHEDDING	EL	OP	I:	1 C	MRB 9	245100-710-801 AMM 245100/5 MRB REFERENCE 24.50.00/02	2	0.15	ALL
R	245000-02-1	210	AC ESSENTIAL DISTRIBUTION OPERATIONAL CHECK OF LAND RECOVERY CONFIGURATION	EL	OP	I:	18 MO	MRB 8	245200-710-802 AMM 245200/5 MRB REFERENCE	1	0.10	ALL

Figure 10: Task related information (MPD A330)

In the MPD, the task information includes the following specifications:

1. Task number: indicates the ATA chapter to which the task belongs, the sequence number of the task (for tasks related to the same hardware) and the applicability index;
2. The zone: indicates the position where the tasks needs to be accomplished;
3. The task description: indicates the type of task to be accomplished. There are 11 types of tasks used in the MPD, each with its own specific code (see Figure 11);

TASK CODE	DEFINITION	TASK CODE	DEFINITION
CHK	Check for condition, leaks, circuit continuity, Check fluid reserve on item, Check tension and pointer, Check fluid level, Check detector, Check charge pressure, Remove and check, Check torque, Leak Check	OP	Operational check
DS	Discard	RS	Remove for restoration
DI	Detailed Inspection	SDI	Special detailed inspection
FC	Functional check	SV	Drain, Servicing, Replenishment (fluid change),
GVI	General Visual Inspection	VC	Visual Check
LU	Lubrication		

Figure 11: Task codes defining the type of the tasks (MPD A330)

4. The interval (I): indicates the interval of the task in one of the various times units (FH, FC, letter checks, calendar units or a combination of them). The interval column can also include a threshold (T) which determines the first accomplishment of the task after which its regular interval is applied;
5. The source: specifies the source document from which the task data is derived;
6. The reference: indicates the reference documents for the accomplishment of the task;
7. The number of men required: indicates the minimum number of workers needed to complete the task;
8. The man hours needed; indicates the number of man hours needed to complete the task;
9. The applicability: indicates the applicability of the task to the aircraft; "-200" or "-300" indications refer to the A330-200 and A330-300 respectively and "ALL" refers to all A330 types of aircraft.

5.1.2 The aircraft utilisation envelopes

Commercial aircrafts are built for a specific operational range. With this information, the manufacturer indicates the conditions for the most efficient utilisation of the aircraft, both technically and financially. The aircraft utilization envelopes in the MPD expresses these operational limits in a maximum and minimum number of FC and FH per 600 FH and per year. The MPD only stays applicable when the aircraft is operated within these limits.
Figure 12 shows the section in the MPD where is this information is indicated.

A330-200

With interval defined as 600FH:	Flight cycle envelope	=	33 FC to 200 FC
	Calendar envelope	=	33 days to 109 days
Annual utilisation (12 months):	Flight hour envelope	=	1667 FH to 5667 FH
	Flight cycle envelope	=	367 FC to 1100 FC

Figure 12: Flight envelope data (MPD A330)

For the A330-200, the annual utilization of the aircraft has a lower limit of 1667 FH and an upper limit of 5667 FH. This indicates that, for a correct interpretation of the data in the MPD, the aircraft needs to be used for a minimum of 4.56 FH/day and a maximum of 15.52 FH/day (the remaining time is used for maintenance, turnaround, line maintenance etc.). Similarly, the aircraft should operate between a minimum of 1 FC/day and a maximum of 3FC/day.

$$4.55 < \mathrm{FH_{day}} < 15.52 \qquad \text{(eq. 1)}$$
$$1 < \mathrm{FC_{day}} < 3 \qquad \text{(eq. 2)}$$

where FH_{day} is the average flight hours per day and FC_{day} the average flight cycles per day.

5.1.3 The different intervals

In the MPD, task intervals are expressed either in FH, FC, years (YE), months (MO), days (DY), letter checks (A or C) or in a combination of them. This diversity in interval units is explained by the fact that the different components on an aircraft wear in different ways. As such, the wear of tires is only influenced by the number of take offs and landings (FC)and the deterioration of compressor blades in the engines depends on the amount flight hours (FH). For the intervals expressed in calendar units, components susceptible to corrosion for example can be thought of.
For most of the tasks, the intervals are expressed in a combination of units, for example in a '8000 FH or 18 months' interval. In this case, the interval corresponds to whichever interval is reached first which depends on the operational pattern of the aircraft. Nevertheless, the tasks in the interval are very diverse and spread out. Figure 13 is a good illustration of the spread-out character of tasks in the MPD.

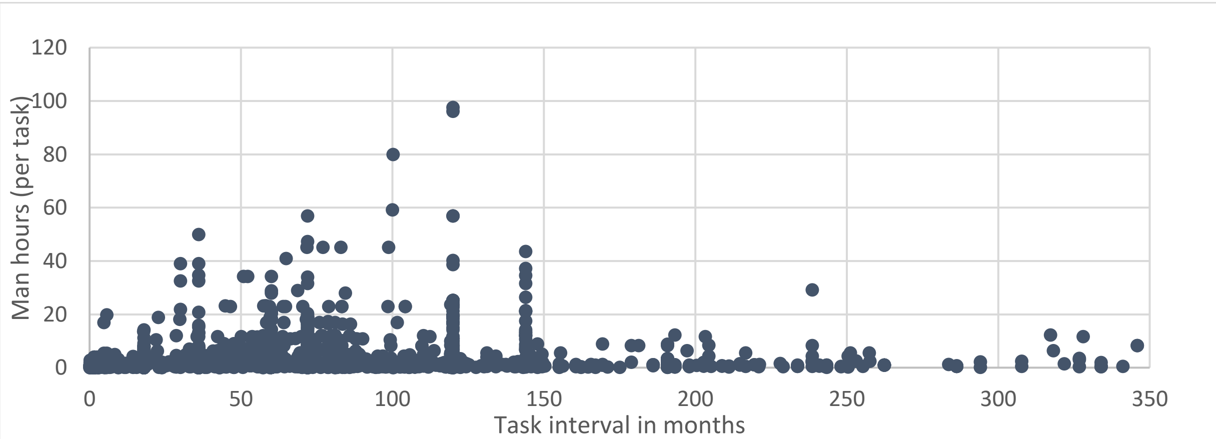


Figure 13: The spread of tasks resulting from the MPD data (A330)

The interval of the maintenance opportunities in the MPD are also expressed in different units:

- A-Check : 600 FH
- C-Check : 18 months
- 2C-Check : 36 months
- 4C-Check : 6 years
- 8C-Check : 10 years

In the MPD and more specifically in the Structures Program, most of the intervals include a threshold designated by the letter T. This threshold indicates the first accomplishment deadline of the task, after which the regular interval (I) is applied.

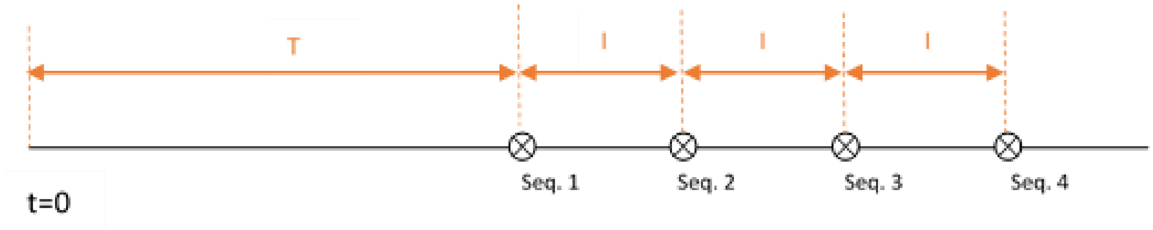


Figure 14: Illustration of task occurrences for tasks with a threshold (T) and an interval (I)

For the structures program in the MPD, task intervals and thresholds are sometimes indicated as SI or ST respectively. This relates the sampling program which is designed to detect systematic deterioration caused by the environment or fatigue. This program monitors the oldest aircrafts in the fleet. The tasks in the structures program fall into 3 categories:

- 100 %: fatigue or corrosion tasks for which high fatigue sensitivity and/or accidental damage is likely are not selected for sampling. Therefore, only the 100% threshold/interval applies.
- 100% + sampling: tasks having a medium fatigue sensitivity are selected for sampling. These tasks have a sampling and a 100% threshold/interval. All aircraft not inspected as sampling aircraft have a 100% threshold/interval.
- Corrosion tasks for which damage is not likely, are selected for sampling and have a sample threshold/interval. Again, all aircrafts not inspected as sampling aircrafts have a 100% threshold/interval.
- Sampling: tasks with a low fatigue sensitivity are selected for sampling. These tasks have a sampling threshold/interval only. These are applicable to the sampling aircrafts only.

  In case of findings, the operator is supposed to take appropriate measures to adapt the maintenance program for that task.

For the 100% + sampling tasks, whether to use the sampling values or the 100% values is explained by the next figure.

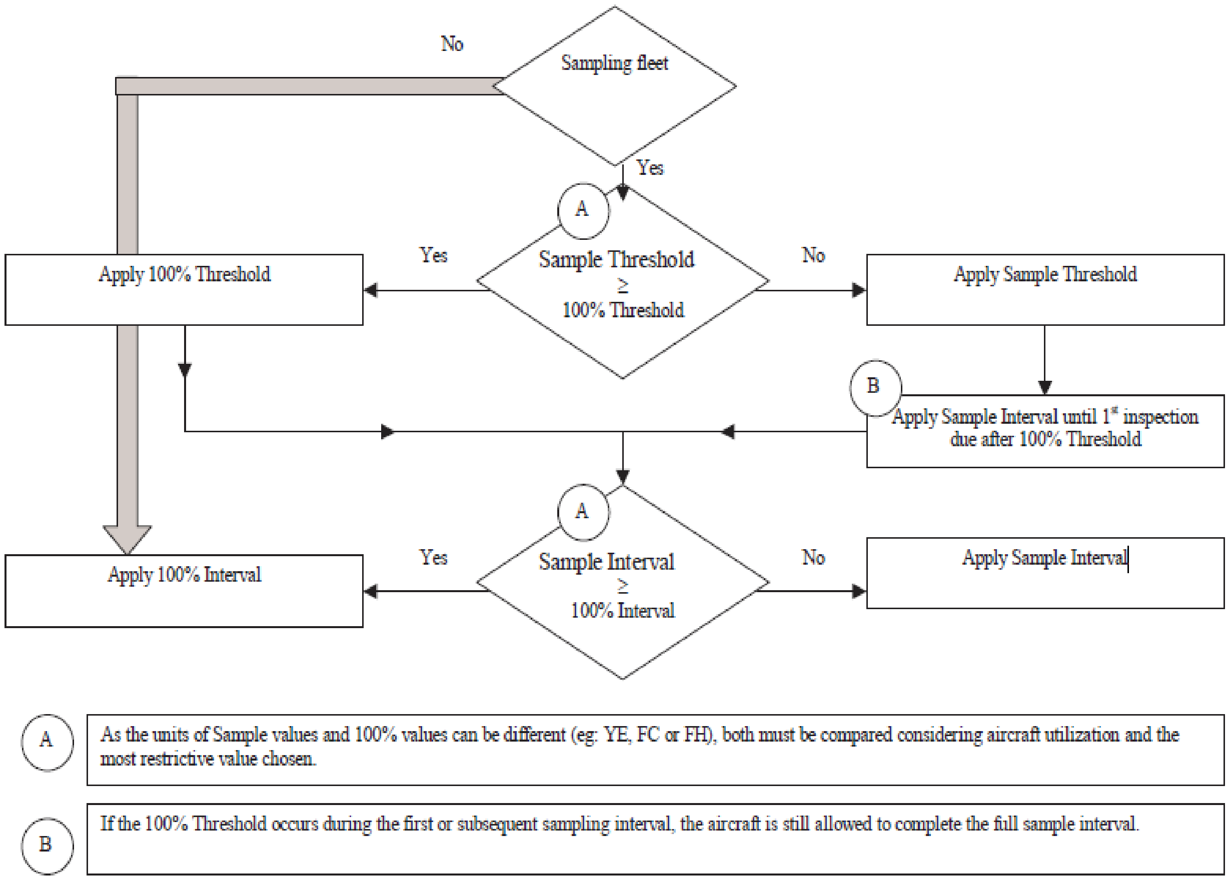


Figure 15: Decision tree for the application of sample or 100% interval/threshold values (MPD A330)

5.1.4 Missing data and considerations

For some tasks in the MPD, the information on intervals or man hours is either not available, to be determined or needs to come from external documents.

In the interval column, this information is often indicated as:

- NT (Note): this means that the operator should refer to the note at the end of the task description;
- VR (vendor recommendation): interval value dependent on task vendor recommendation;
- NR (national requirement): task known as being subject to national regulatory requirement;
- EC (engine change): task should be accomplished at next engine change opportunity.

In the man hour column, this information is usually indicated as:

- N/A: this stands for not applicable, e.g. shop maintenance;
- TBD: To Be Determined indicates that the number of man hours for this task still needs to be defined.

Note: the man hours indicated in the MPD only include 'on aircraft' activities and therefore, they do not include:

- Non-routine work, e.g. repair, troubleshooting, shop overhaul;
- Preparatory work such as aircraft cleaning, positioning work stands, connecting ground power cables;

- One-time actions, e.g. de-greasing, stripping, painting;
- Embodiment of modifications, cabin (galley, lavatory, furnishings) refurbishment;
- Non-productive time, e.g. shift-change, set-up of tools, waiting for sealant or paint drying;
- Planning and establishment of procedures.

The missing information has been neglected primarily because this research focuses mainly on the development of an optimisation methodology. Besides, this missing information would have been very difficult to find because of the unavailability of the required documents. The amount of missing information is the largest in the Power Plant and APU programs.

It is important to note that the information contained in the MPD only stands for the 'on-aircraft' scheduled maintenance activities. The amount of labour indicated for each task in this document, does not include the labour required for 'off-aircraft' activities such as component repairs of refurbishments.

5.2 MPD TASK DATA CONVERSION

For the generation of the preliminary workload distributions, the MPD task data needs to be converted such that the corresponding labour (man hours) can be attributed to the correct months. As explained earlier, most of the task intervals are expressed in FC, FH or in a combination of them. To convert these intervals into months, it is necessary to calculate the average number of FC and FH an aircraft is carrying out over a period of time (days, months or years). To do so, two operational parameters are introduced: the average fly route (AFR) and the average utilisation (AU). The AFR of the aircraft is required to determine the average number of flight cycles per day and the AU is used to compensate for changes in the average utilisation of the aircraft caused by seasonal changes, delays etc. FC_{day} and FH_{day} from eq. 1 and 2 are related to AFR and AU by the following two equations:

$$\left((\mathrm{FH}_{\mathrm{day}})_{\max} \times \mathrm{AU}\right) = \mathrm{FH}_{\mathrm{day}} \qquad \text{(eq. 3)}$$

$$\left(\mathrm{FH}_{\mathrm{day}}\right) / \mathrm{AFR} = \mathrm{FC}_{\mathrm{day}} \qquad \text{(eq. 4)}$$

where $(\mathrm{FH}_{\mathrm{day}})_{\max}$ is the maximum amount of flight hours possible per day (see eq. 1), AFR is the average fly route (in hours) of the aircraft and AU is the average utilisation of the aircraft (in %). Since the value of $(\mathrm{FH}_{\mathrm{day}})_{\max}$ is known (see eq. 1), it can be concluded that FH_{day} depends on the AU only, whereas FC_{day} depends of both AU and AFR.

Given a particular value of AU, the aircraft utilisation envelope discussed in 5.1.2 will determine the theoretical maximum and minimum values of AFR: 13.96 and 4.65 FH.

Once the specific AFR and AU values are set, FC_{day} and FH_{day} will lead to the simple conversion of the intervals into months.

For the intervals expressed in a combination of units, this conversion needs a little more work. An example case for the interval 8000 FC or 50000 FH is used to illustrate how the conversion works.

In Figure 16 below, line A and line B illustrate two operational scenarios with an identical AU of 100% but with a different AFR. For scenario A, an AFR of 5.55 FH leads to a FC/FH coefficient of 0.18 and the limit of 8000 FC is reached before the limit of 50000 FH. According to eq. 3 and 4, the aircraft performs a maximum 2.8 FC/day. The limit of 8000 FC is reached after approximately 94 months (7.8 years).

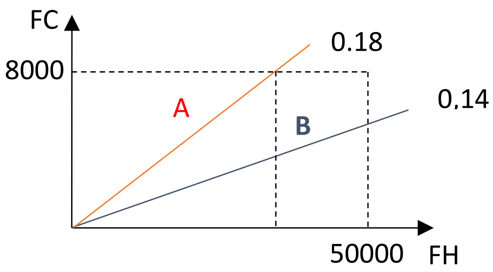


Figure 16: Interval calculation for intervals expressed in a combination of units

For scenario B, an AFR of 6.25 FH leads to a FC/FH coefficient of 0.14 and the limit of 50000 FH is reached before the limit of 8000 FC. With 15.52 FH/day (see eq. 1), 50000 FH is reached after 105.5 months (8.8 years). Figure 17 summarizes the way the MPD data and the inputs are used by the conversion model to obtain the initial scheduled workload distribution.

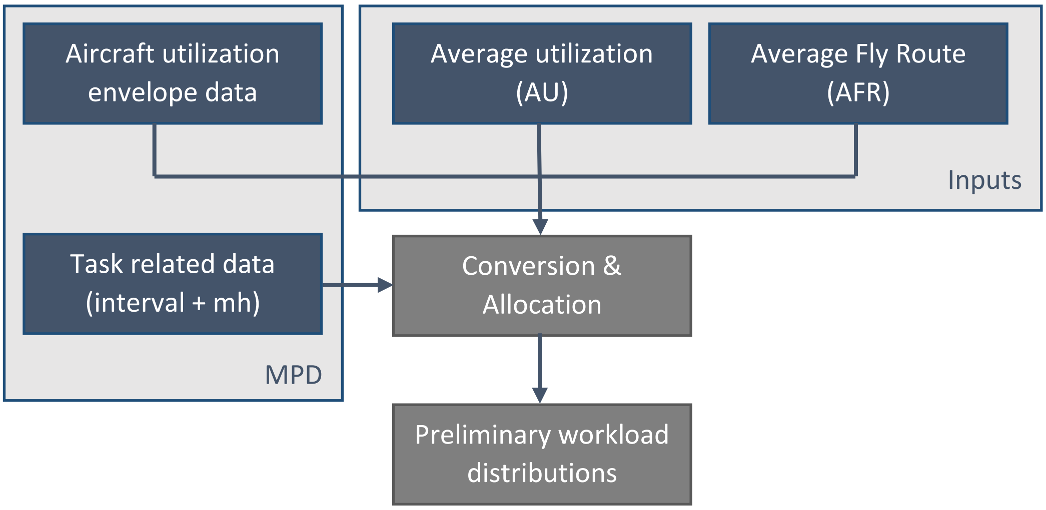


Figure 17: The different elements contributing to the task interval conversion

The aircraft utilisation envelope data is first combined with the AU and the AFR of the aircraft to calculate the numbers of FH/month and FC/month. These values are then used to convert all task intervals into months. The conversion model assigns all corresponding labour (man hours) to the correct months. The man hours are then summed up for each month and this directly results into the preliminary workload distributions for the different maintenance programs.

5.3 PRELIMINARY WORKLOAD DISTRIBUTIONS & ANALYSIS

From this chapter onwards, the AFR will be set to a value of 10 FH and the AU will be set to90% to account for potential unexpected delays. The workload distributions will be generated for a lifecycle of 15 years. Workload distributions per maintenance program.

5.3.1 Systems & Components program

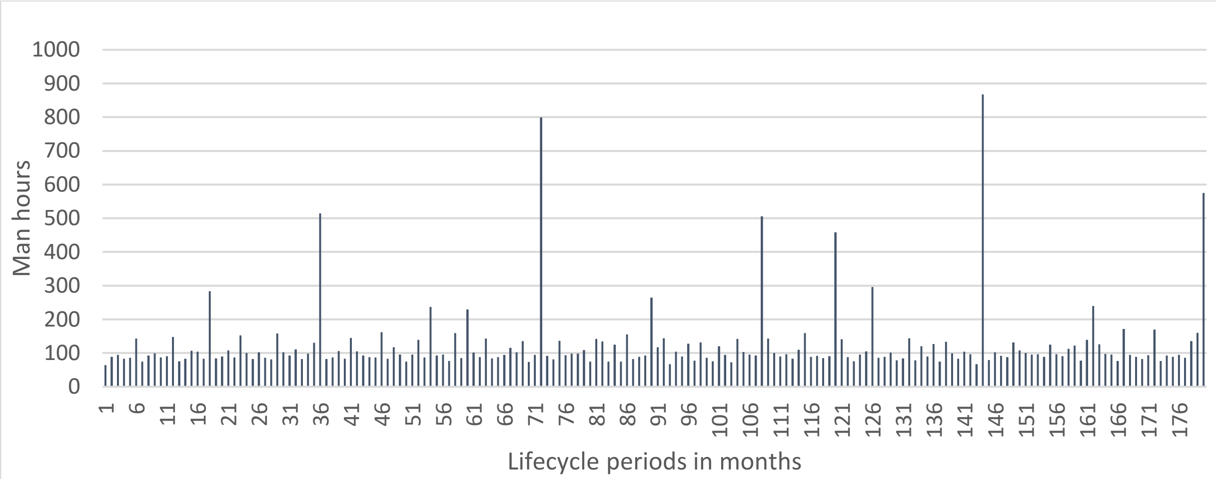


Figure 18: Preliminary workload distribution for the Systems & Components program

In the Systems and Components program, most of the tasks are related to aircraft systems such as the hydraulics, pneumatics, electrics and components such as mechanical actuators etc. In this maintenance program, most of the task intervals are expressed in FH, in letter checks (multiples of A's or C's) or in the combination of FH and months. This means that, according to eq. 1, changes in the AU of the aircraft will affect this workload distribution the most. In this chapter of the MPD, only a negligible amount of data was missing.

The distribution clearly shows the peaks in labour corresponding to the C-checks, and the more constant monthly amount of labour that seems to be fluctuating between approximately 100 and 150man hours and that corresponds to the more frequent A-checks and line maintenance tasks. The C-checks workload peaks seem to have a regular pattern of 200-500-200-700 mh for the 1C, 2C, 3C and 4C-checks respectively. Also note the peak of labour at the 120st month, which occurs just in between 6C and 7C. Even though for the A330-200 program the D-check was eliminated from the maintenance program, this labour peak at 120 still seems to correspond to it.

5.3.2 APU program

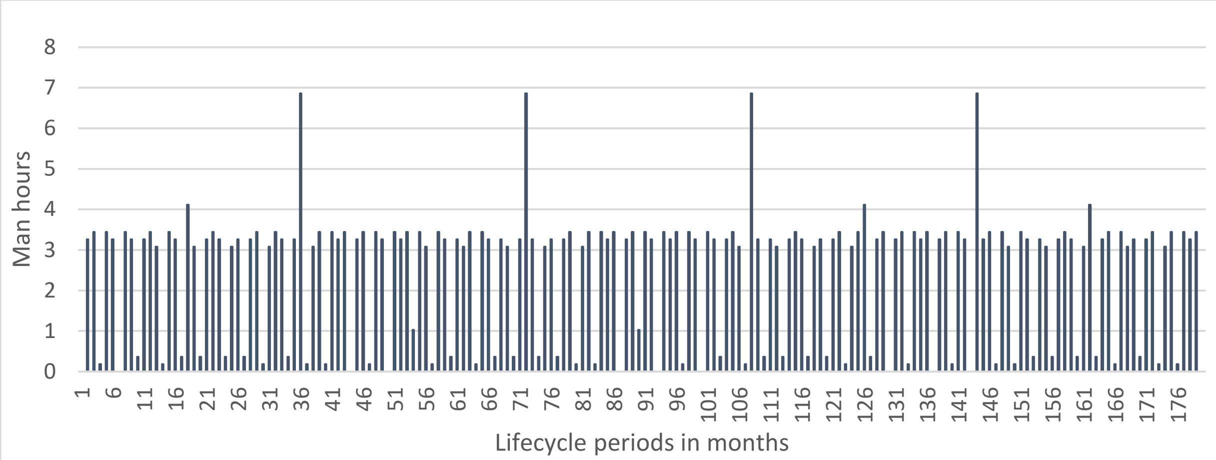


Figure 19: Preliminary workload distribution for the APU (Auxiliary Power Unit) program

The APU program includes all the maintenance tasks required for the Auxiliary Power Unit. It is the smallest of all six MPD maintenance programs with only 9 tasks for a total of just 6.86 man hours. A total of six tasks have an interval of 1C or 2C and one task with an interval of 1A. The APU workload distribution shows a large fluctuation in amplitude but overall, the maintenance tasks of this chapter just contribute a minimum to the total workload distribution for the aircraft.

5.3.3 Power Plant program

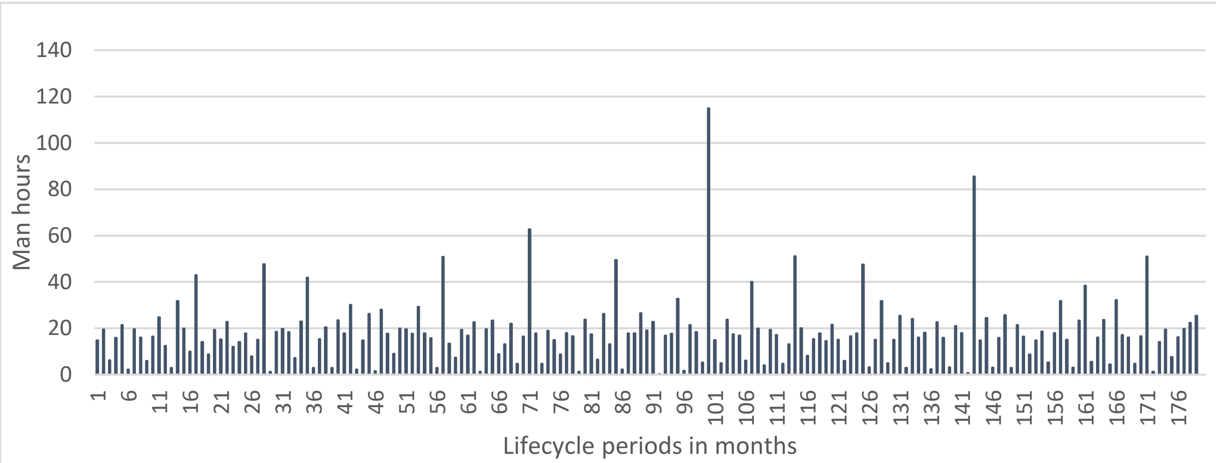


Figure 20: Preliminary workload distribution for the Power Plant program

The Power Plant program contains the maintenance tasks related to the engines of the airplane. Most of the tasks in this maintenance program are expressed in FH, FC or in a letter check (A- or C-check). But tasks intervals are also expressed in a combination of FH and FC or in a combination of A's or C's and FH. This chapter includes some of the largest maintenance tasks of the MPD (some up to 97 man hours involving up to 10 technicians). One of these consists for example of discarding the life limited parts of the compressor module of the engine, or of a detailed inspection of the IP compressor rear stub shaft splines (4200 FC).

However, most of the intervals for these large tasks are not indicated in this MPD and refer to other documents such as the Engine Shop Manual (GE and P&W) or the Time Limits Manual (RR), which are the engine manuals delivered by the manufacturers. Since accessing these documents is too difficult, these tasks are simply neglected as explained in 5.1.4.

Again, this distribution shows workload peaks at the C-checks. Considering the large tasks mentioned previously, it is easily understandable that these tasks need to take place during the heavier base maintenance checks (C-checks) instead of light A-checks where time and resources maybe insufficient.

5.3.4 Structures program

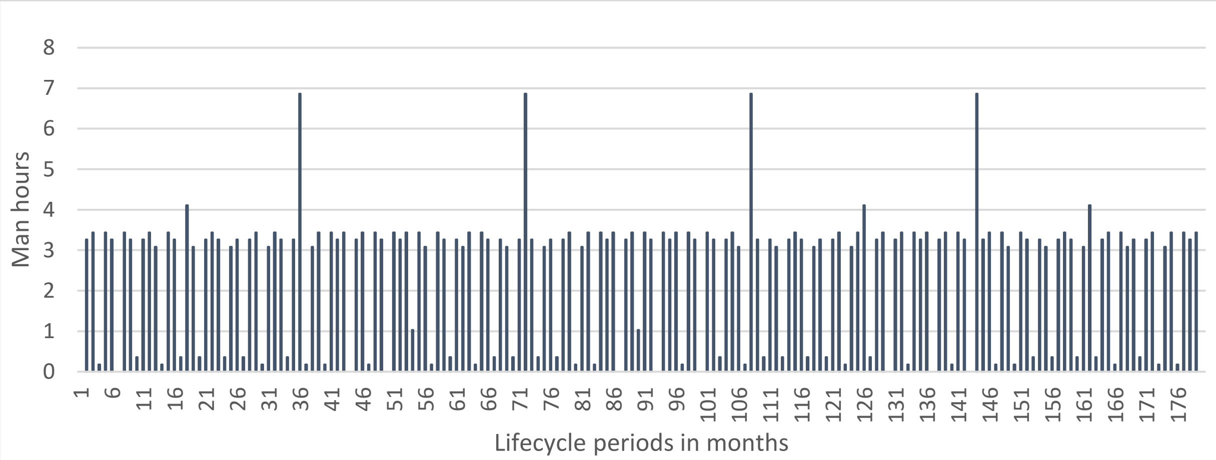


Figure 21: Preliminary workload distribution for the Structures program

The Structure program covers all structural items in the landing gears, fuselage, wings, nacelles, doors etc. The tasks consist of detailed inspections (DI), special detailed inspections (SDI), sometimes using non-destructive testing methods (NDT). For most of them, the aircraft needs to be overhauled and partly or entirely dismantled to access the items to be inspected. Fatigue and corrosion play important roles in this maintenance program.

Most of the tasks in this maintenance program have an interval combined with a threshold. Most of the intervals and thresholds are expressed in a combination of different units (FC, FH and months).

The chart shows a rather irregular workload distribution with a very large peak at 6 years. This is caused by the fact that many tasks with 'combined unit' intervals have an identical of 50.000 FH interval which, with an AU of 90% corresponds to 120 months approximately. Most of the required labour seems to be carried out only after about 6 years which is explained by the use of thresholds in the intervals. This suggests that the structural deterioration of an aircraft is not very significant during the first 6 years.

5.3.5 Zonal Program

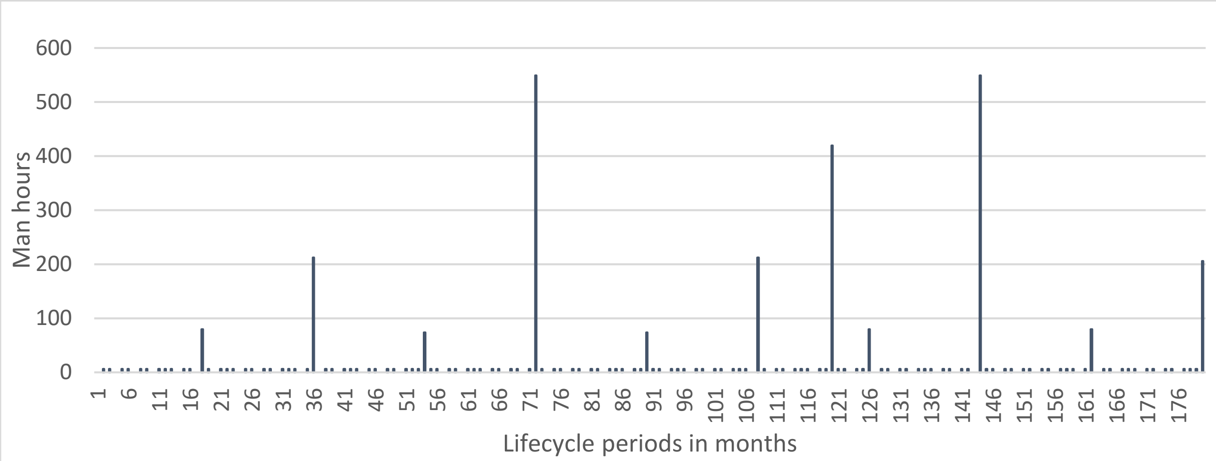


Figure 22: Preliminary workload distribution for the Zonal program

The Zonal maintenance program is a special chapter in the MPD. It provides information for the general visual inspections (GVI) needed for each zone of the aircraft. Systems and power plant installations are inspected for safety and general condition, but the structure is inspected for general condition only. The different zones of this maintenance program are: the lower and upper halves of the fuselage, the stabilizers, the tail unit, the power plant, the nacelles and pylons, the wings, the landing gear including its doors and the doors of the fuselage. All the tasks in this maintenance program are expressed in a letter check, mostly in C-checks and multiples. This explains the very regular pattern of the workload distribution with a 75-200-75-540 mh pattern for the C-checks, as well a large labour peak at 120 months that probably corresponding to the former 2D-check.

5.3.6 Total maintenance workload distribution

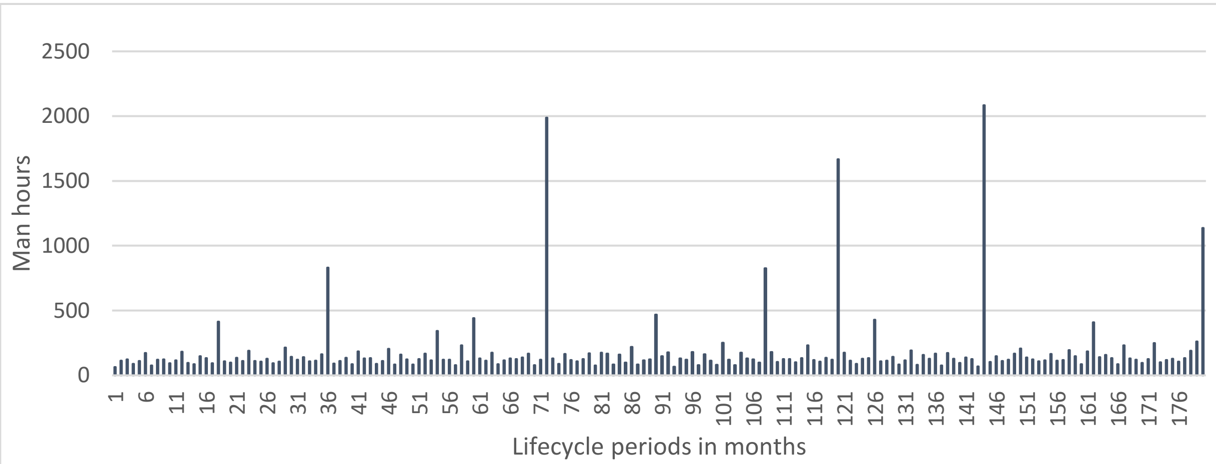


Figure 23: Total preliminary workload distribution (all programs combined)

The chart of Figure 23 represents the total workload distribution, all maintenance programs included. The workload peaks at the C-checks are clearly visible and show a logical pattern. The other workload peaks are those taking place at months 60 and 120 (probably the former

D-checks). The rest of the labour, as indicated earlier for the Systems & Components workload distribution, corresponds the maintenance carried out during A-checks and at the gate (line maintenance). This amount of labour seems to be relatively constant for each month along the distribution.

5.3.7 Effects of a different operational pattern

For all distributions analysed in this chapter, two constant values for AFR and AU have been used as inputs. These two values have been set to 10 hours and 90% respectively. As explained in 5.2, reducing the AFR of the aircraft increases FC_{day}.

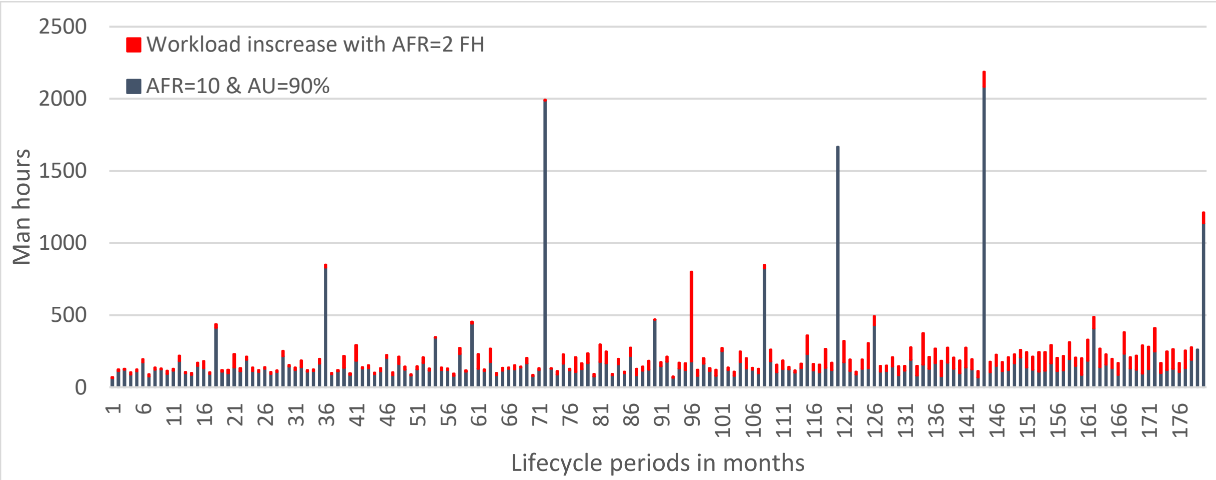


Figure 24: Changes in the preliminary workload distribution when AFR = 2 FH

Since many tasks, especially in the Structures program, has intervals (partially) expressed in FC, it could be interesting to see how much the workload distribution is influenced by this variation in AFR. In Figure 24, the red sections indicate the increase in workload caused by a reduction of the AFR length from 10 hours to 2 hours. An AFR of 2 hours is exaggerated since with an AU of 90%, the minimum value of AFR would equal 4.65 flight hours (see eq. 1), but this value better emphasizes the changes caused in the distribution.

In a similar way, Figure 25 shows the variation in workload caused by an AU variation from 90% to 50%. Note that the red sections in Figure 25 represent reductions of the workloads instead of increments.

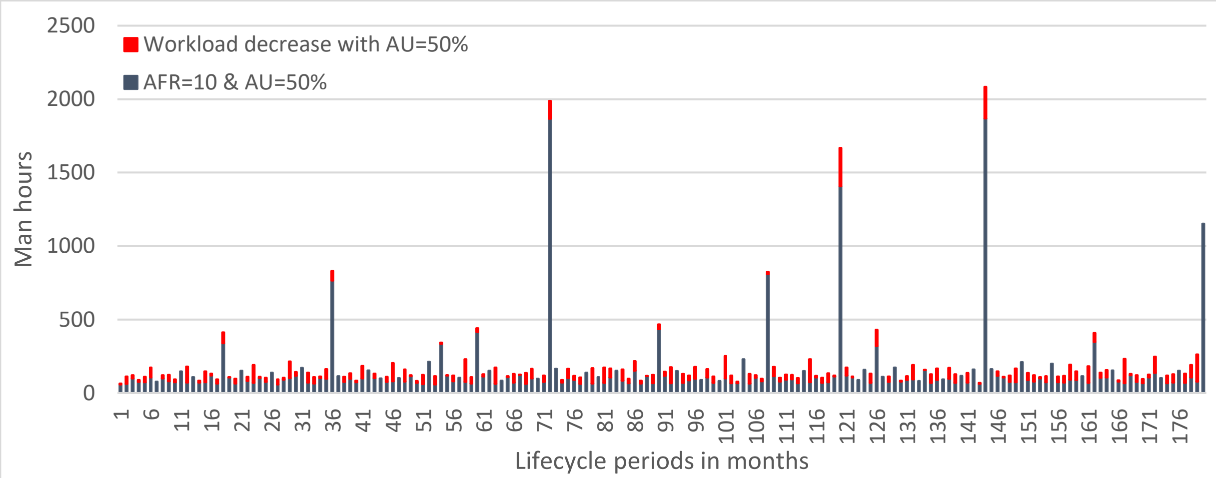


Figure 25: Changes in the preliminary workload distribution when AU = 50%

The largest changes in both distributions are caused by the large amounts of tasks with identical specific intervals. These tasks have often intervals expressed in two units of which one is identical for all of them. For example, let us assume that T_A has an interval of 5000 FC or 18 months and T_B has an interval of 5000 FC or 20 months. With the initial settings of AFR and AU, T_A would take place at 18 months and T_B at 20 months. Decreasing the AFR increases FC_{day} which in turn increases the FC/FH slope (see principle of Figure 16) up to such a level that the interval of both tasks is now equal to 5000 FC. When this occurs for a large number of tasks, large changes such as at month 96 in Figure 26, take place.

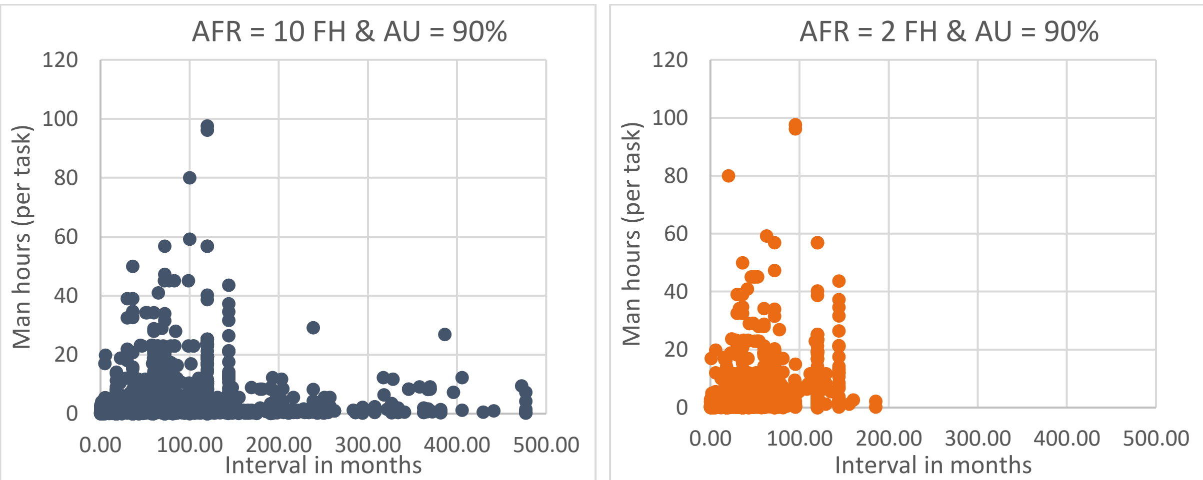


Figure 26: Variation of task intervals with 2 different values of the AFR

Figure 26 illustrates how a part of the tasks is affected by a reduction in AFR. Reducing the AFR increases the number of FC carried out per day or per month. Consequently, the intervals expressed in FC are reduced (the number of months corresponding to the interval is smaller). Since smaller intervals increase the task occurrence frequency, the total labour required for that task over a period increases as well.

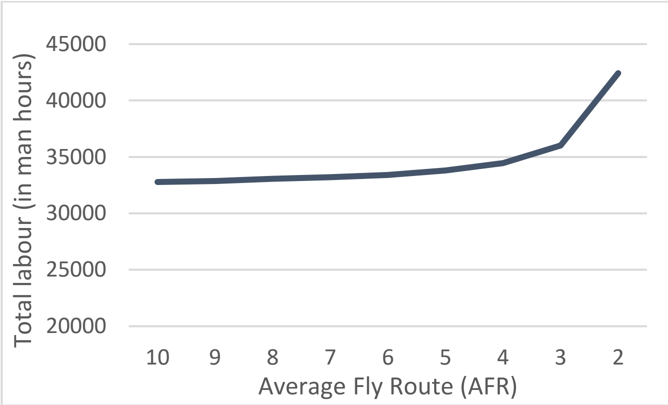


Figure 27: Variation of the total labour with a decreasing AFR

Notice the sharp increase in total labour when the AFR reaches values below 4 hours.

6 GENERATION OF THE INITIAL WORKLOAD DISTRIBUTIONS

The objective of the model developed in this chapter is to generate various initial workload distributions that will be used in the following optimisation chapter. In the previous chapter, the preliminary workload distributions were generated by simply allocating tasks to the corresponding months in the lifecycle of the aircraft. But in reality, an appropriate maintenance opportunity may not be available every month. For the base maintenance tasks for example, the maintenance opportunity with the smallest interval (the A-check = 600 FH), has an interval that corresponds to 1.43 months with an AFR of 10 and an AU of 90%.

The model developed in this chapter will generate realistic workload distributions, called initial workload distribution, by automatically allocating tasks to the closest preceding maintenance opportunity. But instead of only using the interval to determine whether a task needs to be allocated to an A-check, a C-check or to gate maintenance, additional task criteria will be used in this model. These criteria will favour the optimisation in the next chapter, but they will also increase the capacity of the operator to modify and adapt the maintenance schedule of the aircraft to the maintenance organisation/resources on the ground. For example, if the tasks belonging to the structures program are better carried out during C-checks because of specific resources (special equipment, experts etc.) that are not available during regular A-checks, this generic model offers the possibility to allocate these tasks to C-checks only. Thanks to the detailed task data in the MPD, this distinction and separation of tasks according to different criteria is relatively easy to achieve.

Next to the capacity to separate tasks from each other, the model will also present the option to allocate all base maintenance tasks to the A-checks. This way, the equalized approach discussed in the literature review can be obtained and assessed after the reallocation process in chapter 7.

6.1 TASK SEPARATION CRITERIA

Three main task separation criteria have been determined for this generic model: the interval of the task, the size of the task and the MPD program to which the task belongs. Additional separation criteria such as the type of the task or the ATA chapter to which the task belongs, would have led to a more specific separation of the tasks but unfortunately, the MPD data had to be entered manually in Excel. Entering all the data would have costed too much time. Nevertheless, this does not affect the principle and the potential of this generic model.

6.1.1 Task interval

Line maintenance tasks will not be considered for optimisation since the benefits of optimising the line maintenance schedule is expected to be very negligible; only 31 of the more than 2300 tasks in the MPD are line maintenance tasks. Together, they account for a total of only 20.3 man hours. These tasks have an interval that is smaller than one A-check (< 600 FH).

As seen earlier, the intervals used in the MPD are very diverse in size and intervals. For the tasks whose intervals are expressed in letters (A or C), the A-/C-check separation has already been done. For all other tasks, those with an interval smaller than 18 months cannot be attributed to a C-check. These tasks are considered to be exclusively A-check tasks.

$$\text{If } 600\text{ FH} \leq I_{task} < 18\text{ months} \rightarrow \text{A-check task} \qquad \text{(eq. 5)}$$

whereI_{task} is the interval of the task.

As it will be explained in the next section, tasks with I_{task}> 18 months will not necessarily be attributed to a C-check. It is important to note that the task interval criterion of eq. 5 will overrule any following criterion.

6.1.2 Task size

If it is decided to separate the total workload distribution into an A- and a C-check distribution, a separation criterion based on the size of the task may be required. As Figure 13 showed, some task sizes are very large (over 97 mh). If these large tasks are allocated to the A-check distribution, the labour peaks created may be difficult or impossible to eliminate by the reallocation process in the next chapter. Therefore, a size limit in man hours needs to be determined above which tasks will automatically be attributed to a C-check.
For the generation of the initial A-/C-check workload distributions, since the task density seems to be largest under 12 man hours approximately, a size limit of 12 mh is selected.

$$\text{If } MH_{task} > 12\text{ mh} \rightarrow \text{C-check task} \qquad \text{(eq. 6)}$$

whereMH_{task} corresponds to the size of the task in man hours.

It is important to note that the size criterion of eq. 6 will not overrule the interval criterion of eq. 5. The spread chart beneath graphically indicates how the tasks are separated according to the task interval and size criteria.

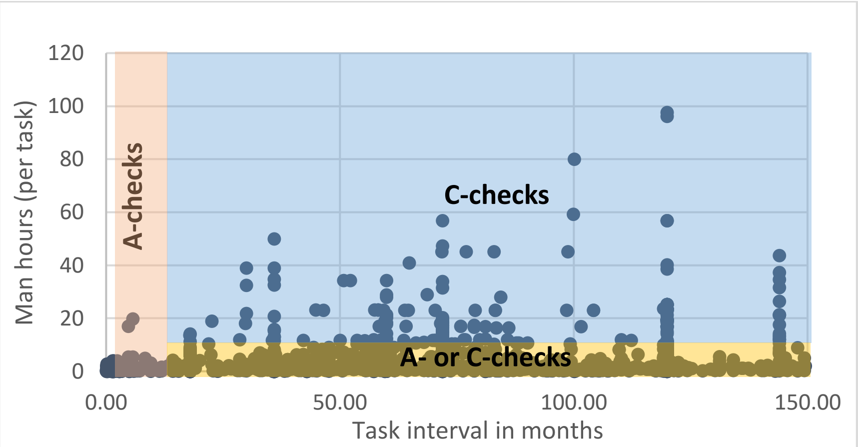


Figure 28: Visualisation of task separation (x-axis enlarged)

As shown, the yellow section includes tasks of both types. This is explained by the fact that some tasks in this area, with less than 12 man hours, have an interval expressed as a 'C'.

6.1.3 Task category

The third and last separation criterion corresponds to the MPD program to which a task belongs. The MPD program of a task determines to a certain extend the type of maintenance resources (tools, processes, jigs etc.) needed during that task. This separation criterion may be useful if it is preferred to carry out the tasks of a specific maintenance program during either an A- or a C-check. Again, this criterion will not overrule the task size and interval criteria. For example, if it is decided to allocate the tasks of the structures program to the C-checks, only those with I_{task}> 18 months will be selected. The tasks with I_{task}< 18 months cannot be allocated to C-check tasks regardless of their size or MPD program.

6.1.4 Summary of the task separation criteria

Table 2 summarizes the separation criteria with which the tasks are attributed to A- or C-check.

	A-checks	**C-checks**
Task interval	I_{task} < 18 months ; A's	I_{task} > 18 months ; C's
Task size	MH_{task} < 12 mh	MH_{task} > 12 mh
Maintenance program	Systems & Components APU Power plant Zonal	Structures

Table 2: Summary of the task separation criteria applied

6.2 TASK ALLOCATION TO MAINTENANCE OPPORTUNITIES

After the separation of A- and C-check tasks, the tasks with a variable interval, which depend on the operational parameters of the aircraft, need to be adjusted for a correct allocation of tasks to maintenance opportunities. This attribution is done automatically by rounding down the interval to the closest preceding maintenance opportunity (intervals cannot be increased). To make sure that every single task is attributed correctly to a maintenance opportunity, the value of the thresholds needs to be rounded down similarly.

Instead of being represented per month, the workload distributions will now be represented per maintenance opportunity (A- or C-check) instead.

This attribution of tasks to maintenance opportunities can lead to significant initial interval reductions for the C-check tasks, since a maintenance opportunity only takes place every 18 months. If for example a C-check tasks initially takes place with an interval of 45 months, this task will now have an interval of 36 months, which corresponds to a 2C interval. This interval reduction will lead to an increase in labour which that is expected to be more important for the C-checks than for the A-check workload distribution. This aspect will be considered in the following chapters.

6.3 INITIAL WORKLOAD DISTRIBUTIONS

With the application of the different task separation criteria of 6.1, the A- and C-check workload distributions over a period of 15 years have been generated.

6.3.1 Initial A-check workload distribution

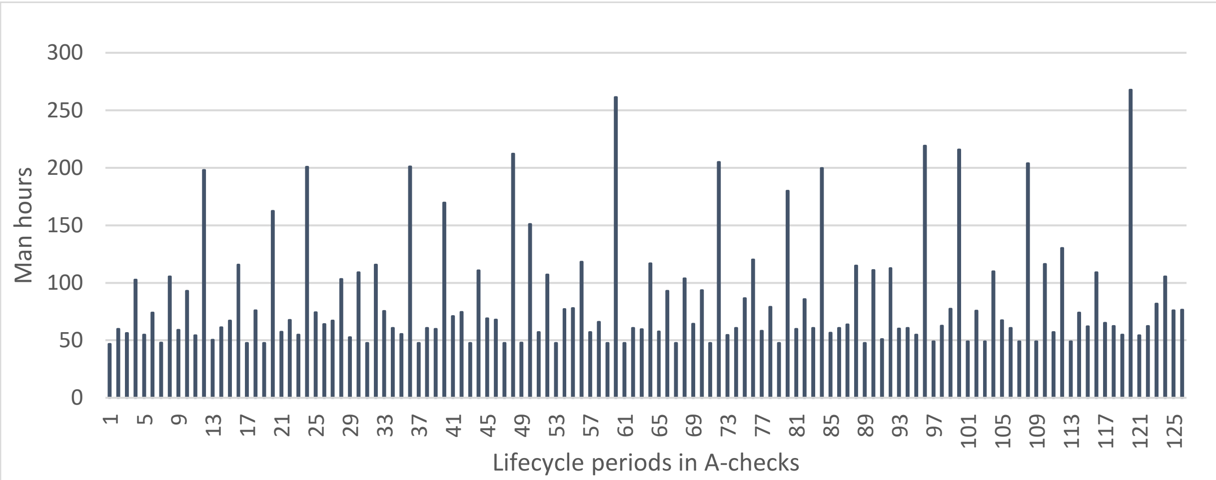


Figure 29: Initial A-check workload distribution over 15 years (AFR = 10 & AU = 90%)

Note that the units of the x-axis are expressed in A-checks instead of months. Since an A-check corresponds to 1.43 months with the current operational parameters, 126 A-checks corresponds to 15 years approximately. The intervals of the tasks will from now on be expressed in A or multiples.

The A-check workload distribution of Figure 29 shows a high and irregular fluctuation among the different A-checks. This fluctuation is partly explained by the repetitive character of A-check tasks. Although the occurrence of the workload peaks seems to be relatively constant (every 4months approximately), the amplitude of these peaks varies considerably (between roughly 100 and 260 mh).

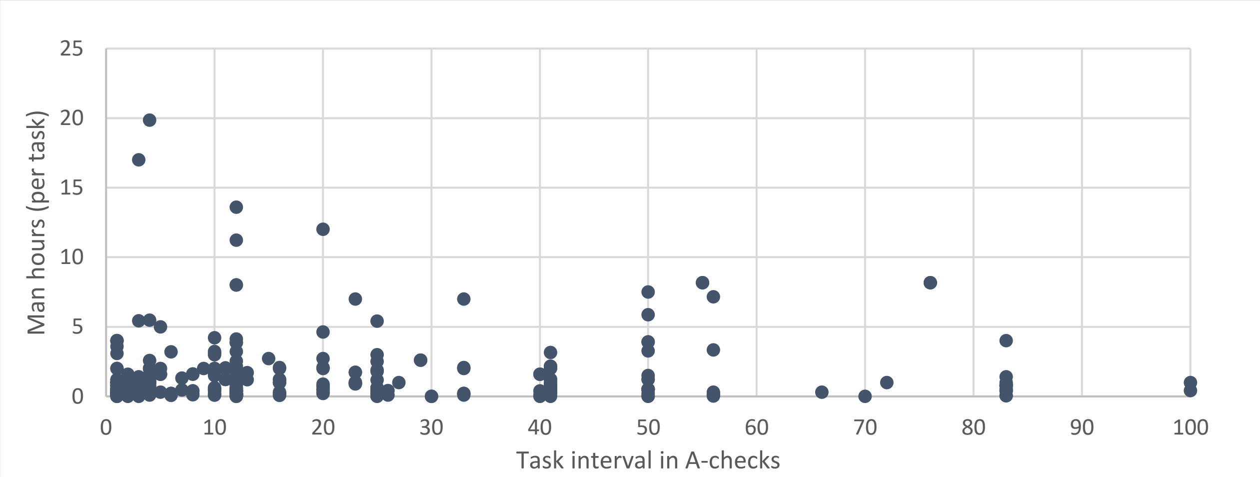


Figure 30: Spread of A-check tasks with the x-axis in A-checks instead of months

Figure 30 represents the spread of the A-check tasks. Note that the x-axis is in A-checks instead of months. Every task has properly been attributed to an A-check opportunity. Only three tasks with more than 12 man hours could not be attributed to the C-check distribution because of I_{task}< 18 months.

Table 3 shows relevant data for the A-check distribution.

	Value
Max. workload (mh)	267.83
Min. workload (mh)	46.73
Average workload (mh)	86.22
Mean absolute deviation (mh)	36.34
Number of tasks involved	382
Average interval (A's)	18.92
Largest task size (mh)	19.86
Average task size (mh)	1.26
Total labour over 126 A-checks (mh)	10863.60

Table 3: Relevant data of the A-check workload distribution over 15 ye.

Some of these values will be of interest during the reallocation process. One of these values is the value of the total labour generated over the entire lifecycle period. The reallocation, and the task interval reduction associated with it, increases the total of labour over the 140 periods. This increase of labour is thought to be the largest disadvantage of the reallocation process. Tracking this value may therefore be important.

Another interesting value is the average task size (in mh). The smaller the task, the easier it is to reallocate it to another period without creating labour excesses. As shown in Table 3, the average task size is very small, which seems promising for the reallocation.

6.3.2 Initial C-check workload distribution

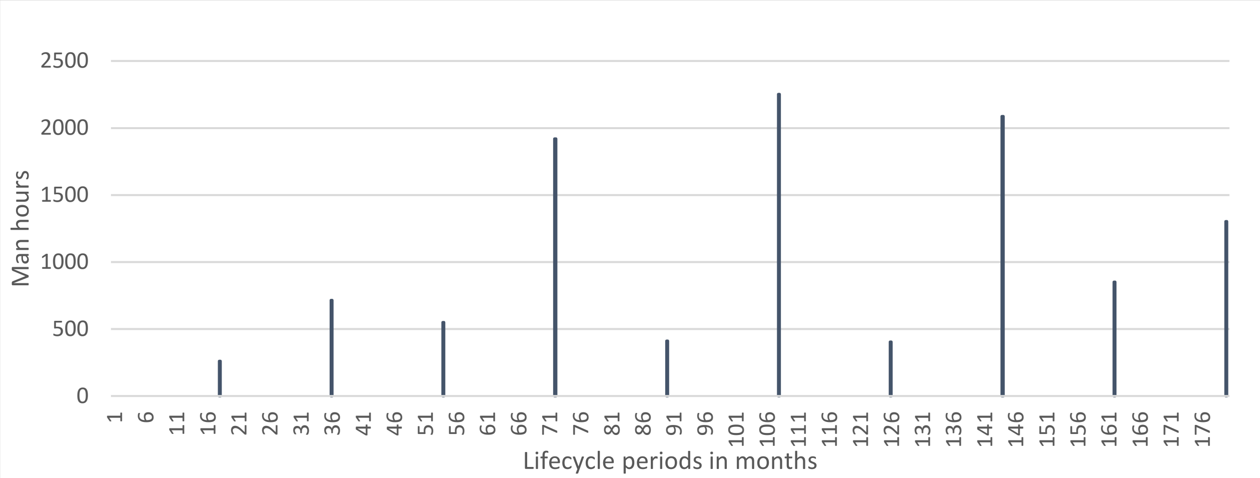


Figure 31: Initial C-check workload distribution over 15 years (AFR = 10 & AU = 90%)

Note that the x-axis for the distribution above is now expressed in months since a C-check interval corresponds to 18 months (see 4.1.3). For this workload distribution, the workload fluctuation is considerable as well, although a much larger workload during the 4C or an 8C-check was expected. Moreover, some of the large tasks that had 60 and 120 months intervals (probably the former D-checks) have now been attributed to the 3C- (54 months) or the 6C-check (108 months).

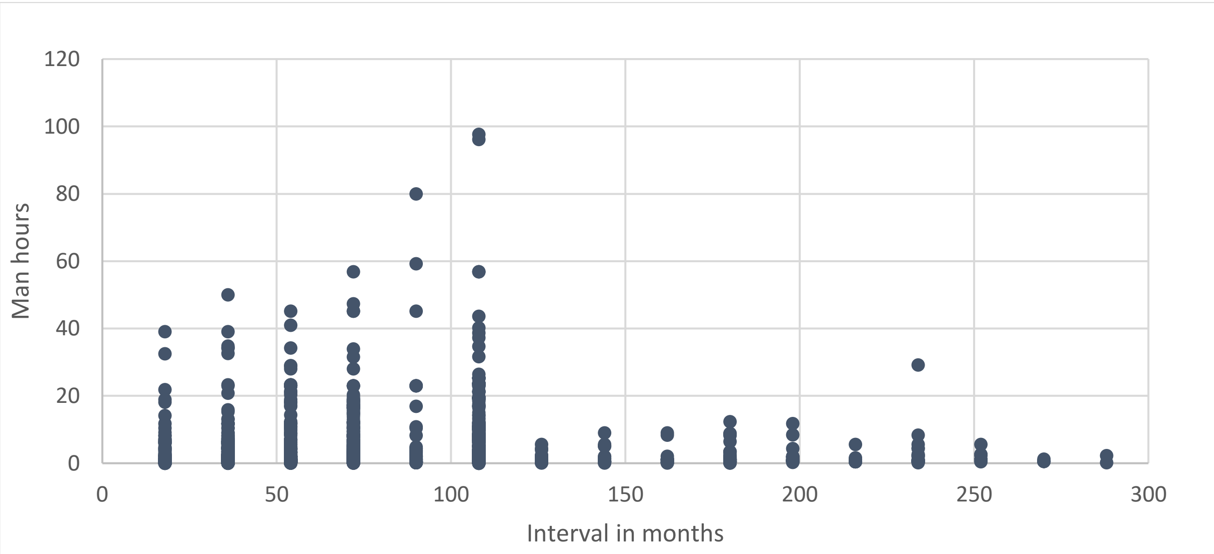


Figure 32: Spread of C-check tasks

As shown in the spread chart of Figure 32, all the C-check tasks have been properly allocated to a C-check. Notice that a considerable number of tasks in the C-check workload distribution is smaller than 12 mh. These are the tasks that already had an interval expressed in ‘C’s’. These were therefore not allocated to an A-check. Table 4 shows the relevant data for the initial C-check distribution.

	Value
Max. workload (mh)	2252.71
Min. workload (mh)	260.84
Average workload (mh)	1075.54
Mean absolute deviation (mh)	652.93
Number of tasks involved	1815
Largest interval (months)	468
Average interval (months)	87.18
Largest task size (mh)	97.6
Average task size (mh)	4.41
Total labour over 180 months	10755.42

Table 4: Relevant data for the initial C-check workload distribution over 15 ye.

An important value in this table is the largest task size of 97.6 man hours. This value might complicate the reallocation process later on as explained earlier. This also applies to the average task size of 4.4 mh which does not seem high but which is almost four times higher than for the A-check workload distribution. The number of tasks involved is also much higher for the C-checks.

These initial workload distributions are converted into task lists in .csv format such that the data corresponding to these distributions can be read by the software application that carries out the reallocation process (see chapter 7.3). Task lists contain the interval, the size (in mh) and the MPD maintenance program for each task.

7 OPTIMISATION

The optimisation is based on a process that focusses on flattening the workload distribution by a systematic reallocation of individual tasks. The process uses the initial A- and C-check workload distributions obtained in chapter 6 and is carried out by locally reducing the intervals of individual tasks.

After a more detailed explanation of the reallocation process, a manual optimisation is carried out to determine the constraints and essential parameters involved in this process. Considering the time-consuming aspect of this manual optimisation and because of the need to actively track several decision variables for an efficient optimisation, a software application to automatically and dynamically carry out the optimisation, is developed. This Visual Basic application generates the optimised workload distributions while different inputs and parameters can be used.

7.1 TASK REALLOCATION PRINCIPLE

The fluctuations in the workload distributions are eliminated by locally reducing the interval of a task to allocate it to another maintenance opportunity where the workload is lower. For the A330-200, the base maintenance opportunities considered are either A- or C-checks. Each 'bar' in the workload distributions represents an accumulation of different tasks with different intervals, all taking place at the same maintenance opportunity. For example, task A with an interval of 2 months, task B with an interval of 3 months and task C with an interval of 6 months all take place at month 6. If there is a workload peak at month 6, one or more of these tasks can be reallocated.

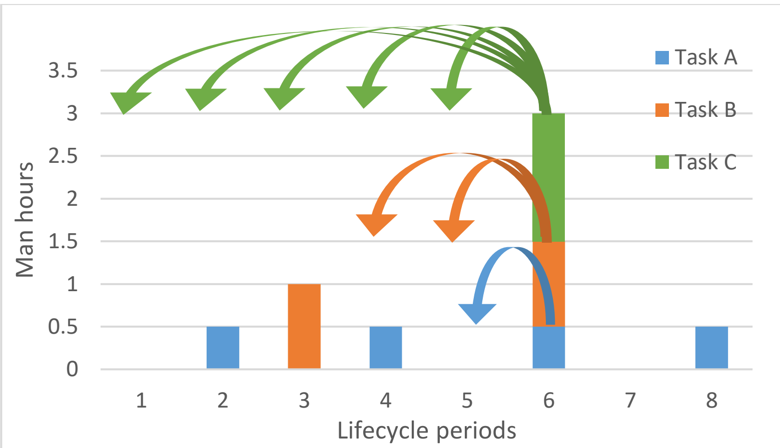


Figure 33: Illustration of task reallocation possibilities

It is important to understand that in the reallocation process, reducing an interval locally does not imply reducing the intervals between all the sequences of the reallocated task. If task B from the previous example is taken, the task sequences will normally take place at months 3, 6, 9, 12, 15 etc. Reducing the interval locally means for example that sequence nb. 2 takes place at month 4 instead of 6. The interval is thus only reduced between sequence nb. 1 and 2 and task B would now take place at months 3, 4, 7, 10, 13 etc. (see Figure 34).

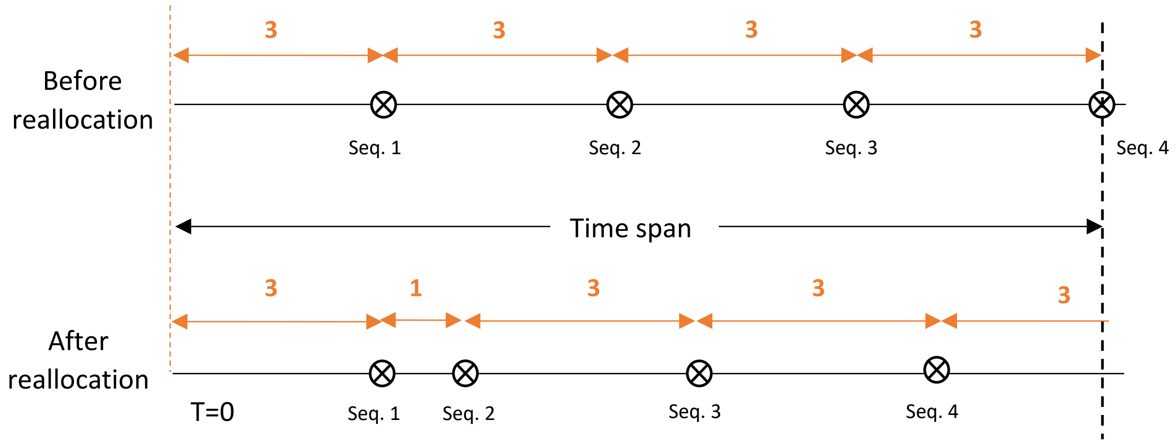


Figure 34: Reducing an interval locally

Note that, even if the reallocation of a task reduces a workload peak at one point, it could as well create other peaks further down the distribution since all the following sequences of the task are shifted as well by the reallocation. Therefore, the maintenance opportunity to which a task is reallocated needs to be chosen adequately.

Also, even if locally reducing an interval limits the increase in labour compared to a method that would reduce every interval between the task sequences, labour costs will inevitably increase the long term. It is thought however, that these costs are negligible compared to the benefits of a constant workload distribution.

7.2 MANUAL OPTIMISATION

A manual optimisation is carried out at this stage to evaluate the feasibility and the potential of the reallocation process, but more importantly to determine the important constraints and parameters involved in it. This will help defining the requirements of the automated reallocation model (the software application, see 7.3).

In this chapter, only the A-check distribution is optimised as a test. The manual optimisation takes place with the use of task interval and task sequence matrices built in Excel. By manually changing the task sequence numbers, tasks intervals are locally reduced to make the task occur at a desired period. With the matrices, all subsequent intervals of the same task are maintained.

7.2.1 Process & key values

For the manual optimisation, the following terminology is used to designate the different parameters and variables involved:

- Period: the A-check number at which a task takes place;
- New period: the A-check to which a task is reallocated;

- Hangar slot capacity (HSC): the limit value (in man hours) above which the task reallocation needs to take place.
- Excess: the amount of man hours exceeding the predefined HSC at a specific period;
- Available capacity: amount of man hours that can be attributed to a new period (available capacity = HSC – actual workload at new period);
- Interval reduction limit (IRL): the maximum interval reduction (in %) applied during the reallocation process. This value will hold for all tasks and will be rounded up where necessary.

Two inputs will be used in the manual reallocation process: the Hangar Slot Capacity (HSC) and the Interval Reduction Limit (IRL).

The HSC corresponds to the maximum workforce size (in mh) available during a hangar slot. For every A-check where the initial workload exceeds the HSC, a reallocation of tasks will be required.

The IRL is an input that limits the interval reduction associated with the reallocation process. Without it, it would be possible to reduce the initial interval of a task by a maximum of 100%. Although an IRL of 100% increases the capacity of the reallocation process to flatten the initial workload distribution, it can also significantly increase the total labour required over the lifecycle of the aircraft and may also lead to excessive task repetitions.

At the A-check where the workload exceeds the HSC, the tasks with the largest interval are the first to be reallocated. Larger intervals offer more flexibility to the reallocation process since the number of possible new periods is higher. Moreover, these tasks have a smaller occurrence frequency and therefore, the effects of the reallocation on the rest of the distribution are smaller.

During the process, the variation of the average excess and total labour will be tracked to observe the efficiency of each reallocation step. For the average excess, the following formula is used:

$$\text{Avg. excess} = \frac{\sum_{n=1}^{126}(\mathrm{MH}_n - \mathrm{HSC})}{126} \qquad \text{(eq. 10)}$$

where MH_n is the number of man hours at period n.

To track the increase in labour, simply counting the total number of man hours over the total of 126 months may not be accurate. The illustration of Figure 34 helps to understand why. After the reallocation, the 2nd sequence of task B was reallocated to period 4. In absolute terms, over the entire time span, the task occurs 4 times in both before and after the reallocation. But it would not be correct to assume that the interval reduction after the task reallocation does not add additional costs. The loss of interval caused by the reallocation is therefore converted into an increase in labour for which the following formula is used:

$$\text{relative additional labour (in mh)} = \left(\frac{I_{initial} - I_{new}}{I_{initial}}\right) \times MH_{task} \qquad \text{(eq. 11)}$$

where$I_{initial}$ corresponds to the interval before reallocation, I_{new} corresponds to the interval after reallocation and MH_{task} is the number of man hours of the task.
On the other hand, the relative additional labour as calculated in eq. 11 may not reflect the absolute increase of labour costs spent on the aircraft over a specific time span. This strongly depends on the time span over which this increase of costs is calculated, as illustrated in Figure 35. Here, the time span goes from T=0 to T=10.5 instead of from T=0 to T=12.

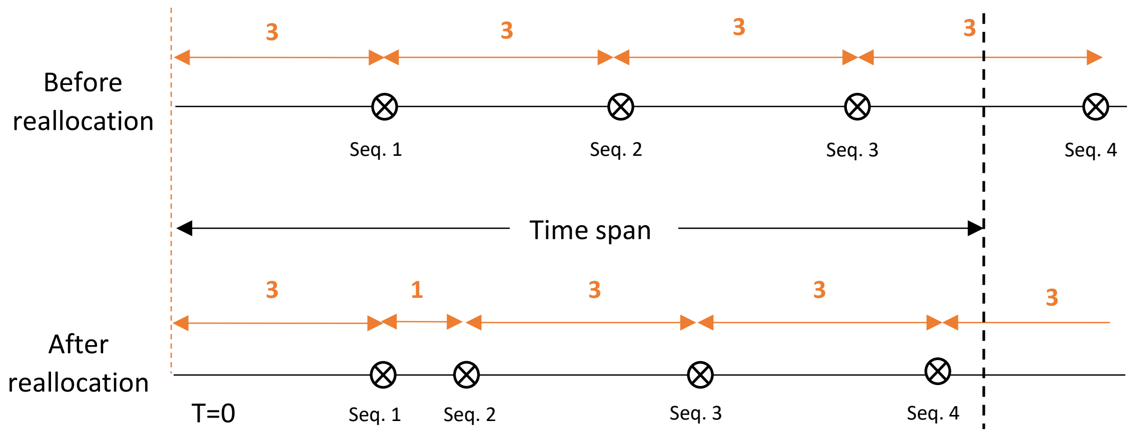


Figure 35: Absolute labour before and after a task reallocation over a specific time span

According to eq. 11, the relative additional labour after the reallocation would correspond to 2/3 x MH_{task} (interval between seq. 1 and 2 reduced by 2/3) whereas the absolute additional labour over the same time span corresponds to 1 x MH_{task} (see Figure 35).
The relative and absolute additional labour are two different ways of quantifying the additional costs involved in the task reallocation process. The relative additional labour takes the interval loss into account and is thought to be a more accurate quantification of labour costs over an infinite time span. The absolute additional labour is a more direct quantification of labour costs and considers simply the difference in labour spent on the aircraft over a bound time-span.
The manual optimisation is carried out with a HSC of 87 mh which almost corresponds to the mean of the initial distribution. The IRL is set to 50% to limit interval reductions. Both values have been chosen to properly see their limiting effect on the reallocation process.

7.2.2 Result analysis

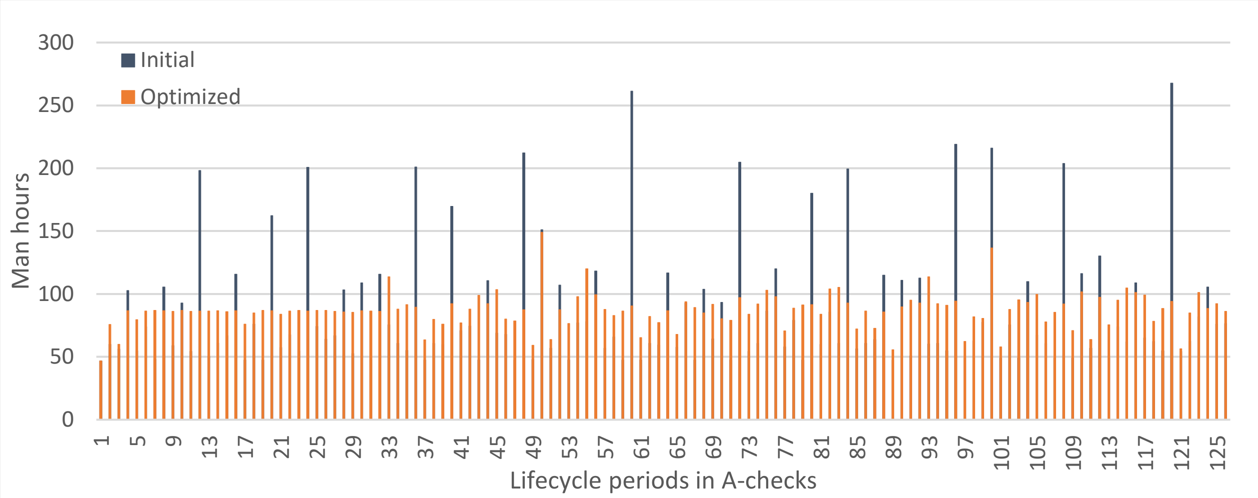


Figure 36: Initial vs optimised A-check distribution

The manual optimisation has been carried out over the first 32 periods. It required 10 steps, each one corresponding to an excess that was eliminated. Due to the IRL of 50% and the very low HSC (almost equal to the mean of the distribution), continuing the reallocation process while staying under the HSC of 87 mh was impossible beyond period 32. With a larger IRL, the available capacities at the first four periods might have been used more effectively.
Table 5 presents the relevant data for this manual optimisation.

	Before optimisation	After optimisation
Max. workload (mh)	267.83	149.01
Min. workload (mh)	46.73	46.73
Average workload (mh)	86.22	86.88
Mean absolute deviation (mh)	36.34	8.95
Nb. of reallocations	-	142
Average interval reduction (%)	-	34
Total workload over 126 A-checks (mh)	10863.60	10947.39
Absolute additional labour (%)		0.77
Relative additional labour (%)	-	0.86

Table 5: Relevant data for the manual optimisation of the initial A-check workload distribution

For this process, 142 task reallocations have taken place for which the initial task intervals have be reduced on average by 34%.
By only observing the first 32 periods, the workload fluctuation seems to have been eliminated effectively and the workload at each period (except the first four) equals almost exactly the HSC. This last aspect results from the large amount of very small tasks in the A-check distribution, that enabled to perfectly fill the available capacities at the new periods.
Another interesting observation is that, by only optimising the first 32 periods, the workload fluctuation has been reduced considerably over the rest of the distribution as well. This is due

to the repetitive character of the tasks; for example, by eliminating an excess at the 4th period, there is a high chance that excesses will also be eliminated at periods 8, 12, 16 etc. In Figure 37, the variations of the relative additional labour and the average excess have been plotted against the reallocation steps taken.

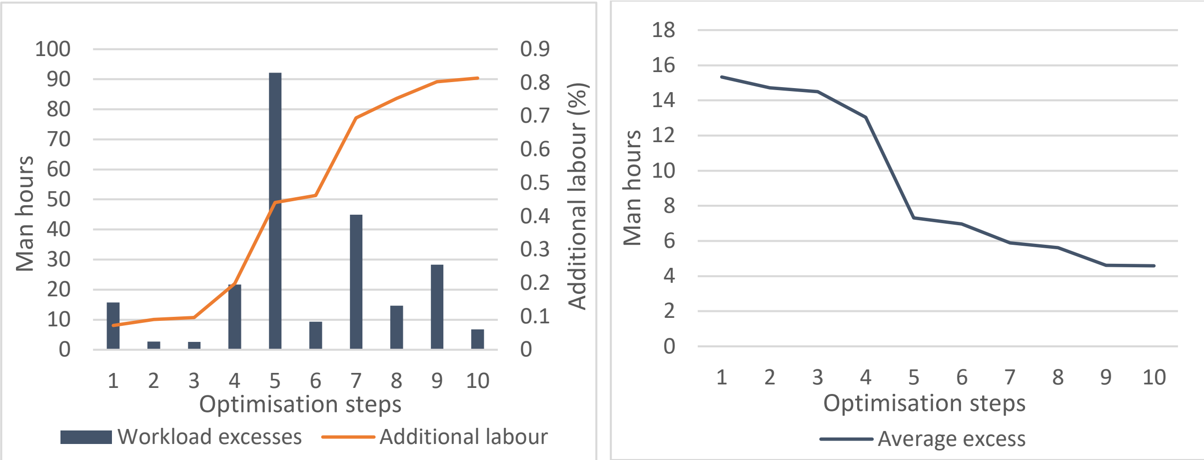


Figure 37: Increase of the relative additional labour and decrease of the average excess per reallocation step

The average excess has been reduced from 15.33 to 4.58 mh while only 0.86 % relative additional labour has been generated over the entire timespan (126 periods). From the left chart above, the correlation between the increase in the relative additional labour and the eliminated workload excesses is clearly shown. At step 4 and 7, excesses of 92 and 45 man hours were eliminated which caused sharp increases in the relative additional labour. The chart on the right shows the evolution of the average excess, which is also correlated with the workload excesses encountered.
An important observation is that the last 5 steps have only managed to decrease the average excess by 2.72 mh (25%) whereas the relative additional labour increased by 45%. This suggests that relatively inefficient reallocations were carried out during these steps. Since each step in the process presents different reallocation options, different task reallocations may improve the optimisation process along these last 5 steps. In fact, by monitoring the evolution of the average excess during the reallocation process, it should be possible to determine the optimal reallocations.

In the manual optimisation, three values have played the most important roles:

1. The HSC; ideally, this value should be as low as possible because it corresponds to the amount of labour per A-check. Whether it is possible to keep the workload under the HSC depends however on the interval reduction limit as well.
2. The IRL; a large IRL could considerably increase the total labour but it also limits reallocation possibilities and therewith the possibility to achieve a low HSC.
3. The average exceeding value; the more it decreases, the more efficient the reallocation.

These values will be used as key values in the development of the software application in the next section.

7.3 SOFTWARE ASSISTED REALLOCATION

The manual optimisation has shown to be a time-consuming process through which nonetheless promising results were obtained. However, the development of a software application is indispensable to further develop, test and evaluate the reallocation process.
In this chapter, the requirements and the process flow for this software application built in Visual Basic will first be defined and explained. The actual use of the software and its interface will then be illustrated. Finally, the testing procedure explains the way in which the optimisation will be run.

7.3.1 Requirements

Quick and automatic reallocation process

The manual optimisation process of 7.2 showed that it was impossible to continue the reallocation process beyond the 32nd period without changing the values of the HSC and the IRL.
Considering the different initial workload distributions that may need to be optimised and the very large and varied amount of data that needs to be processed, the automatization of the reallocation process is absolutely required. To achieve this, the software application needs to automatically detect the periods at which the workload exceeds the HSC, after which it selects and reallocates tasks to more appropriate new periods. These new periods will be the preceding periods at which the workload is lowest.
With the use of a software application, the process of selecting and reallocating tasks can be carried out in a matter of seconds for the entire workload distribution. Different optimisations can thus easily be carried out with different HSC and IRL values, and an optimal solution can be quickly be found.

Inputs & outputs

In a first stage, the application needs to read the initial workload distributions generated in 6.3. This is done using task lists that correspond to the initial workload distributions and contain the interval and number of man hours for each task, including the maintenance program to which they belong. Since the software application is built in Visual Basic and since the initial task lists in chapter 6 were generated in Excel, these tasks lists will be read by the application as .csv files.
Just as for the manual optimisation, the HSC and the IRL will be used as process inputs in this program. These two inputs are dependent on each other in the attempt to efficiently eliminate the workload fluctuation of the entire distribution (see manual optimisation). The HSC will be entered in mh and will thus serve as a limit value above which the reallocation is triggered. Since the task intervals differ in length, the IRL is expressed in % and therefore, a rounding factor needs to be applied to assure a correct reallocation.

Regarding the outputs, the program will generate the initial and the optimised distribution in the form of task matrices in Excel. These are easily converted to the initial and optimised workload distributions and maintenance schedules.

Implementation of decision values

During the manual optimisation, several values such as the average excess and the relative additional labour were monitored to observe the effects and the effectiveness of each reallocation step. But these values were passively tracked after each task reallocation.

As explained in 7.1, reallocating a task to another period might locally reduce a workload peak but it can also create other peaks further down the distribution and even increase the average excess over the entire lifecycle. To avoid this, the software application should have the capacity to carry out each reallocation based on how it affects the average excess or the relative additional labour (see Figure 38).

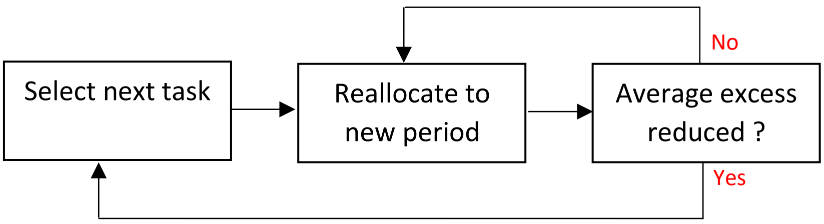


Figure 38: Dynamic reallocation with the average excess as a decision value

Interface

In order to read the specific task lists (.csv files corresponding to the different initial workload distributions) and to enter specific inputs such as the HSC and the IRL, the software application requires an interface through which the reallocation process can be launched, monitored and completed.

7.3.2 Process flow

The process used by the software application is very similar to the process used during the manual optimisation. The only difference is that average excess measured after each reallocation will now determine if the actual reallocation of a task is efficient or not. If not, another task or another new period will be found for a better reallocation.

The process diagram of Figure 39 shows how the application is supposed to carry out the optimisation step by step.

The reallocation process will work its way up the lifecycle of the airplane, from period 1 to the last period. Each excess encountered will be eliminated by a single task reallocation. The process uses 4 main steps, of which 2 are calculation steps (step 1 and 4) providing a ‘yes/no’ result leading separate process paths:

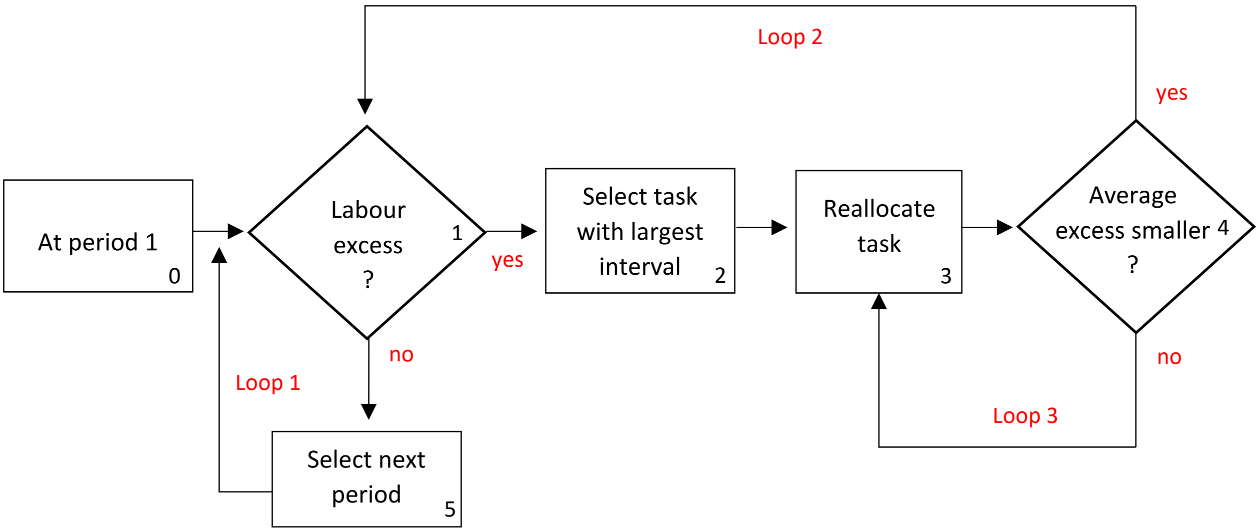


Figure 39: Process flow for the software assisted reallocation

- Step1: this step calculates if there is a workload excess at the first period selected in the process. If this is the case, the program continues with step 2. If there is no excess, the next period is selected.
- Step 2: the tasks with the largest interval taking place at that the selected period will first be selected for reallocation.
- Step 3: the task selected in step 2 is reallocated to anew period where the workload is lowest. However, two parameters are influencing the selection of this new period:
    - The interval reduction limit: this limit is set at the start of the reallocation process and corresponds to a percentage of the initial task interval. A task will be reallocated to a period where the labour is minimal but not beyond this limit;
    - The available capacity at the new A-check: a task can only be reallocated to a preceding period if the available capacity at the new period is sufficient.
- Step 4: calculates the variation of the average excess labour for the task reallocation of step 3. If the average excess is decreased, loop 2 reinitiates the whole process starting at step 1. If on the other hand the average excess is increased, loop 3 reinitiates the reallocation by trying out another new period where the labour is minimal. Step 4 is executed again until the average excess is reduced.

To ensure the continuation of the reallocation process, the program uses three loops that repeat themselves;

- loop 1 selects a period and determines if there is a workload excess. It repeats itself until a workload excess is found;
- loop 2 repeats the reallocation of tasks until the workload excess is eliminated;
- loop 3 repeats the reallocation of one single task until the average is reduced.

7.3.3 The software application in practice

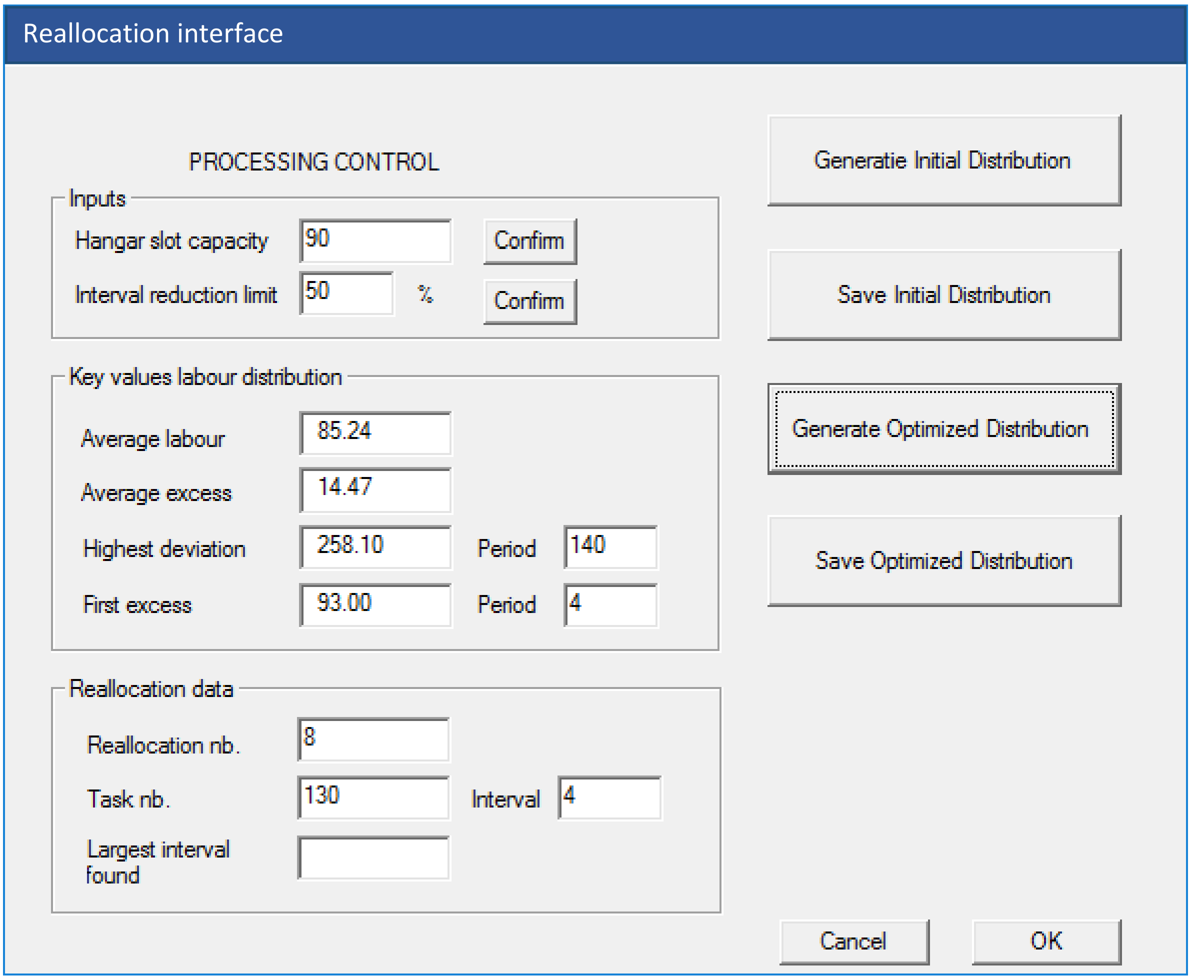


Figure 40: Interface window of the software application that carries out the reallocation process

The task lists generated in 6.3 in .csv format are first opened and read by the application. The process interface is then opened where the HSC and the IRL values are put in and confirmed. By pressing the 'Generate initial distribution' button, the initial arrangement of tasks is generated and key values such as the mean, the average excess, the highest deviation, the first excess etc. are displayed in the interface. The initial distribution can then be saved with the button 'Save Initial Distribution'. This leads to the starting point of the reallocation process. The 'Generate Optimised Distribution' button executes the optimisation either task by task or all at once by keeping the button pressed. Reallocation data such as the number of reallocations or the period (of the workload excess) where the process finds itself in, is displayed.

The program gives an end signal when the optimisation is either completed or interrupted due to an excess that cannot be eliminated due to an inappropriate HSC or/and IRL value.

The final optimised workload distribution can then be saved with the 'Save Optimised Distribution'. A task matrix is then obtained in Excel just as for the initial distribution. The final workload distribution charts can easily be generated in with this matrix.

8 RESULTS & ANALYSIS

The different optimisations carried out with the software application have the objective to find the lowest possible values for HSC and IRL for a constant workload distribution over a time span of 15 years.
As observed in the manual optimisation, the values of the HSC and the IRL can obstruct the optimisation process. The optimisation runs will therefore be performed for decreasing values of HSC and for each of these, the IRL will be decreased step by step as well. In the end, finding a minimal HSC value will determine the constant size of the workforce required for each A- or C-check.
The AFR and AU will be set to the constant values of 10 FH and 90% respectively. The automatic reallocation will be carried out for the A- and C-check workload distributions over a period of 15 years (126 A-checks/10 C-checks or 180 months). It will also be attempted to optimise one single workload distribution that includes both A- and C-checks to evaluate the feasibility and the consequences of the equalized method and/or the 'E-check' approach applied by Easy-Jet in the early 2000s.
For each optimisation run, the final workload distribution will be plotted and analysed. Key values such as the total labour, the relative and the absolute additional labour and the number of reallocations required will be presented in a table. Only the most relevant optimisation runs will be analysed and presented in this chapter.
For the next subchapters, the blue distribution represents the initial workload distribution. The orange distributions represent the various optimised workload distributions for which different values of the HSC and IRL have been used.

8.1 A-CHECK WORKLOAD DISTRIBUTION OPTIMISATIONS

<u>HSC = 120 mh & IRL = 100%; 10%</u>

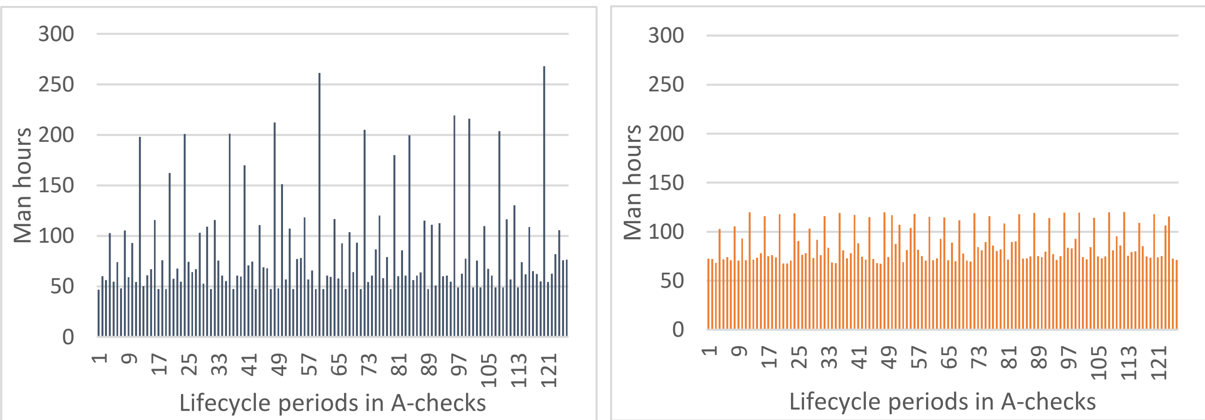


Figure 41: Initial vs. optimised A-check workload distribution with HSC=120 mh & IRL=100%

In this first optimisation, a HSC of 120 mh has been chosen with an IRL of 100%. As observed in Figure 41, this optimisation could be carried out over the entire time span of the workload

distribution. This was predictable since the mean of the initial distribution is much lower than 120 mh (86 mh). Therefore, sufficient available capacity was found at the new periods to reallocate even the largest tasks in this distribution.
In fact, the HSC was large enough to carry out the optimisation of the entire distribution even with an IRL of only 10%. The difference between the use of an IRL of 100% and 10% is shown in Figure 42.

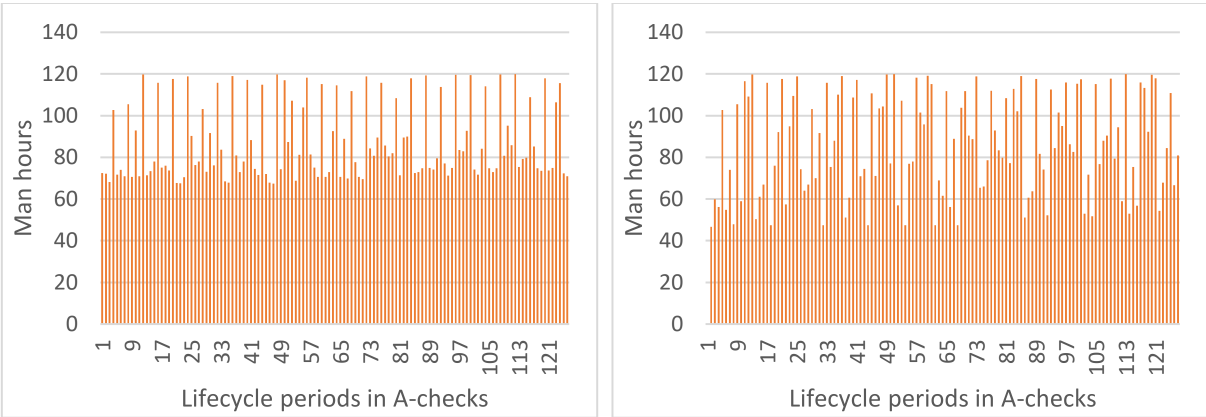


Figure 42: A-check workload distributions optimised with IRL=100% (left) and IRL=10% (right)

In the chart on the right, it can be observed that, instead of reallocating tasks up to period 1, tasks have now only been reallocated to the few periods preceding an excess. This is well illustrated by the workload peak at period 12, where tasks have been reallocated only to period 11 instead of up to period 1. Table 6 presents some important values for both optimisations (IRL 100% & 10%) for the time-span of 15 years (126 A-checks).

	Initial	**IRL = 100%**	**IRL = 10%**
Mean absolute deviation (mh)	36.34	15.51	21.49
Number of reallocations	-	144	153
Total labour (mh)	10863.60	10971.46	10871.89
Absolute additional labour over 126 A-checks (%)	0	0.99	0.07
Relative additional labour (%)	0	1.11	0.24

Table 6: Optimisation data for the A-check workload distribution (HSC=120 mh & IRL=100% ; 10%)

Table 6 shows the effectiveness of the IRL on the reduction of the relative additional labour; it has been reduced by 78%. On the other hand, the higher value of the mean absolute deviation for an IRL of 10% indicates that the workload fluctuates more heavily with an IRL of 10%.

HSC = 100 mh & IRL = 100%;25%

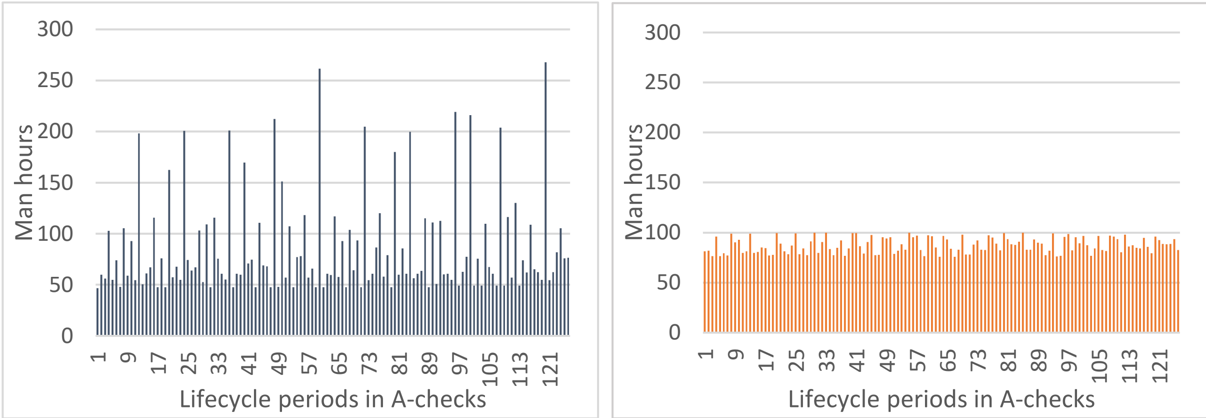


Figure 43: Initial vs. optimised A-check workload distribution with HSC=100 mh & IRL=100%

In this optimisation, an HSC of 100 mh was used with an IRL of 100%. Again, the HSC value turns out to be high enough to carry out the optimisation over the entire time span. But the additional labour generated is expected to be higher since more reallocations are required to eliminate the workload peaks.

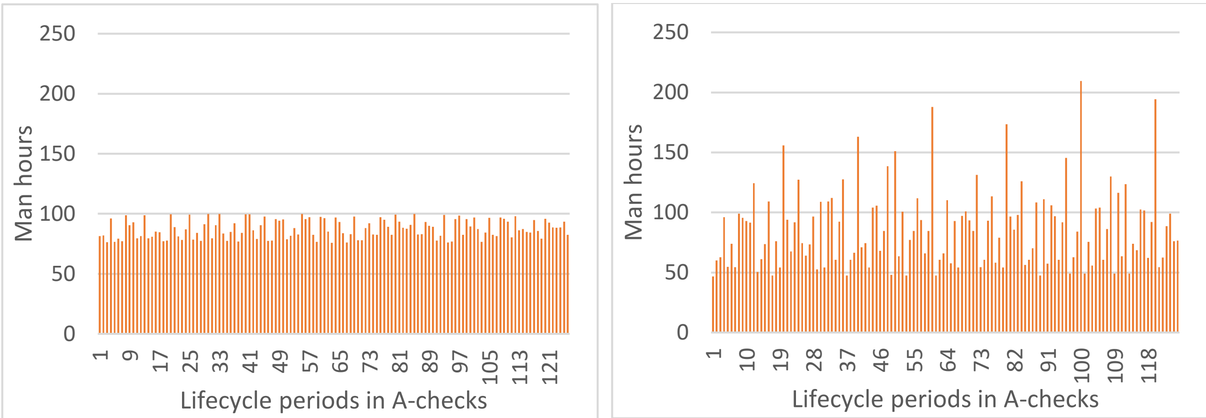


Figure 44: A-check workload distributions optimised with IRL=100% (left) and IRL=25% (right)

The chart on the right of Figure 44 shows the limiting effect of IRL on the reallocation process. With an IRL of 25%, the optimisation cannot be carried out beyond period 12. Although it seems that there is enough available capacity between periods 1 and 7, the reallocation could not continue due to a large task of 19 mh with an interval of 4, that was too large to be reallocated to from period 12 to 11 (with an IRL of 25% and an interval of 4, the task can be reallocated by a maximum of 1 period). Table 7 presents the relevant data for both these optimisations.
Note that for an HSC with an IRL of 100%, the additional labour is already higher than for a HSC of 120 mh. This is explained by a lower HSC for which more reallocations are required.

	Initial	IRL = 100%	IRL = 25%
Mean absolute deviation (mh)	36.34	6.74	24.44*
Number of reallocations	-	182	61*
Total labour (mh)	10863.60	11030.09	10863.60*
Absolute additional labour over 126 A-checks (%)	-	1.53	0*
Relative additional labour (%)	-	1.61	0.33*

Table 7: Optimisation data for the A-check workload distribution (HSC=100 mh & IRL=100%; 25%) (: optimisation not completed)*

<u>HSC = 90 mh & IRL = 100%</u>

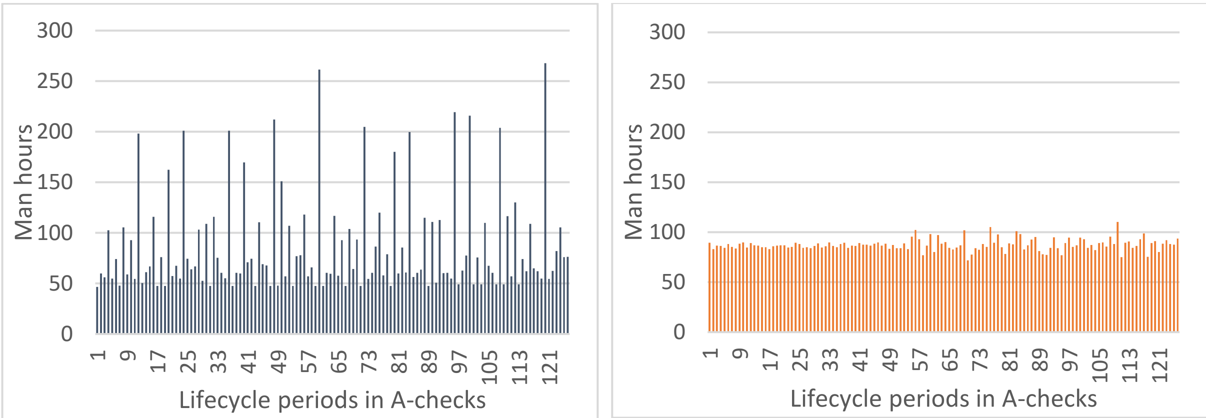


Figure 45: Initial vs. optimised A-check workload distribution with HSC=90 mh & IRL=100%

Figure 45 shows that the optimisation with a HSC of 90 was almost possible. However, the optimisation could not be continued beyond period 55. Nevertheless, similarly to the phenomenon observed in the manual optimisation, the remaining part of the distribution has been significantly flattened, with an average excess of only 2.7 mh. In this case, further reducing the IRL is not interesting.

	Initial	IRL = 100%
Mean absolute deviation (mh)	36.34	4.15*
Number of reallocations	-	201*
Total labour (mh)	10863.6	11034.19*
Absolute additional labour over 126 A-checks (%)	-	1.57*
Relative additional labour (%)	-	1.62*

Table 8: Optimisation data for the A-check workload distribution (HSC=90 mh & IRL=100%) (: optimisation not completed)*

HSC = 94 mh & IRL = 100%

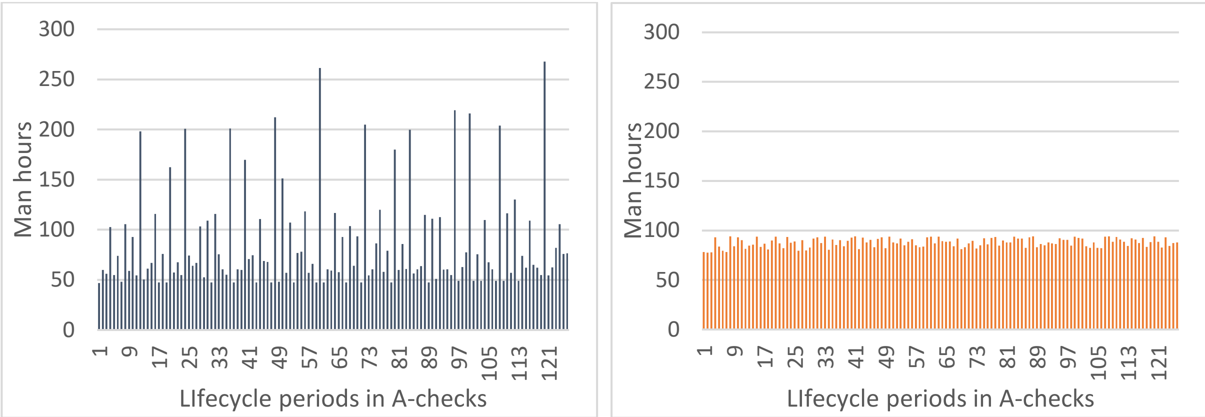


Figure 46: Initial vs. optimised A-check workload distribution with HSC=94 mh & IRL=100%

After several optimisations, the HSC turns out to have a minimal achievable value of 94 mh with an IRL of 100%. The key values of this workload distribution are found in Table 9 below.

	Initial	**IRL = 100%**
Mean absolute deviation (mh)	36.34	3.89
Number of reallocations	-	270
Total labour (mh)	10863.6	11065.12
Absolute additional labour over 126 A-checks (%)	-	1.86
Relative additional labour (%)	-	1.97

Table 9: Optimisation data for the A-check workload distribution (HSC=94 mh & IRL=100%)

If the A-check slot duration is known, this smallest possible value of the HSC indicates the theoretical maximum size of the workforce that is required to complete the scheduled maintenance of all A-checks over a period of 15 years.

8.2 C-CHECK WORKLOAD DISTRIBUTION OPTIMISATIONS

HSC = 1400 mh & IRL = 100%; 50%

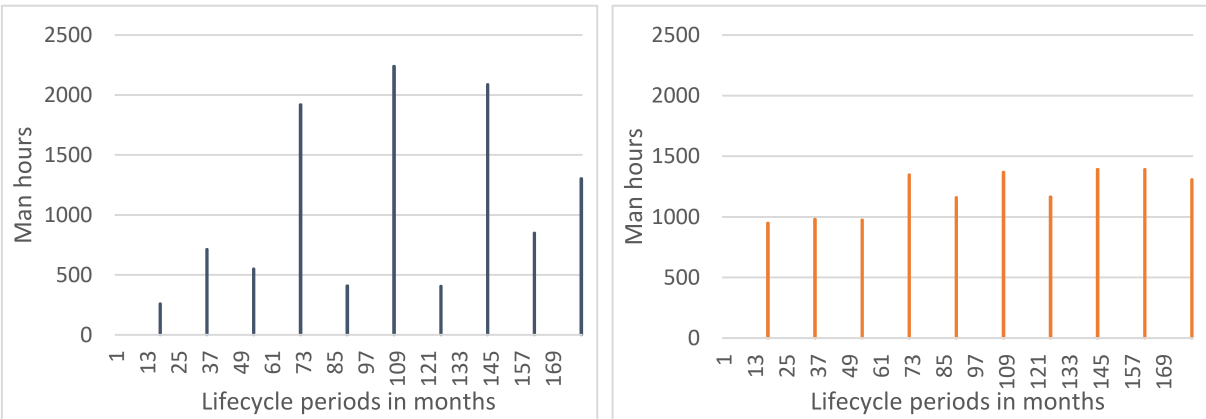


Figure 47: Initial vs. optimised C-check workload distribution with HSC=1400 mh & IRL=100%

Since the average task size is much larger for the C-checks, obtaining a highly constant workload distribution is expected to be more difficult. Therefore, a first optimisation run with a HSC value of 1400 mh was chosen, which is largely above the mean of the distribution (1074 mh). Figure 47 above shows that all excesses have been eliminated successfully.

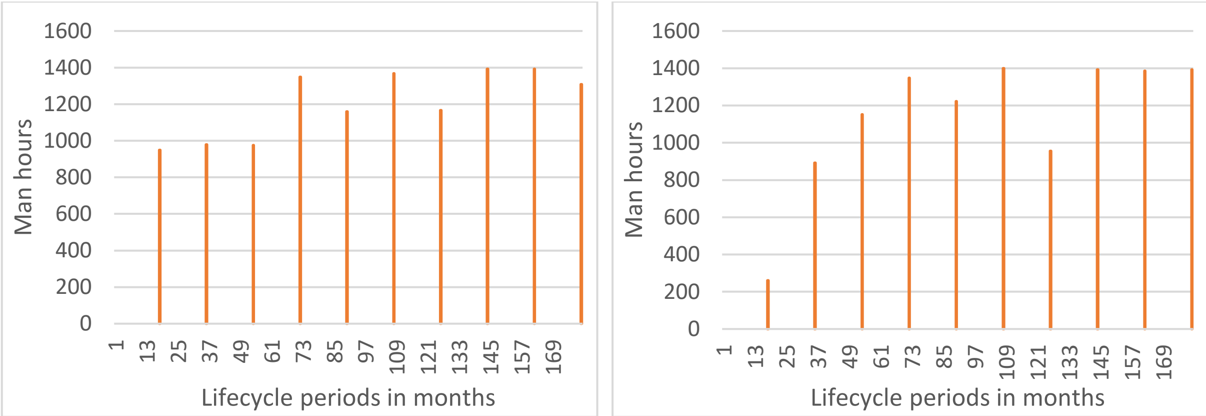


Figure 48: C-check workload distributions optimised with IRL=100% (left) and IRL=50% (right)

The effects of a smaller IRL (50%) are clearly visible in Figure 48 ; similarly to the A-check optimisations of Figure 42, the mean absolute deviation of the optimisation with an IRL of 50% is higher, but the additional labour is considerably lower (see Table 10), by approximately 50%.

	Initial	**IRL = 100%**	**IRL = 50%**
Mean absolute deviation (mh)	651.09	158.12	262.08
Number of reallocations	-	331	298
Total labour (mh)	10743.7	12040.5	11401.25
Absolute additional labour over 180 months (%)	-	12.07	6.12
Relative additional labour (%)	-	10.67	5.21

Table 10: Optimisation data for the C-check workload distribution (HSC=1400 mh & IRL=100% ; 50%)

HSC = 1200 mh & IRL = 100%

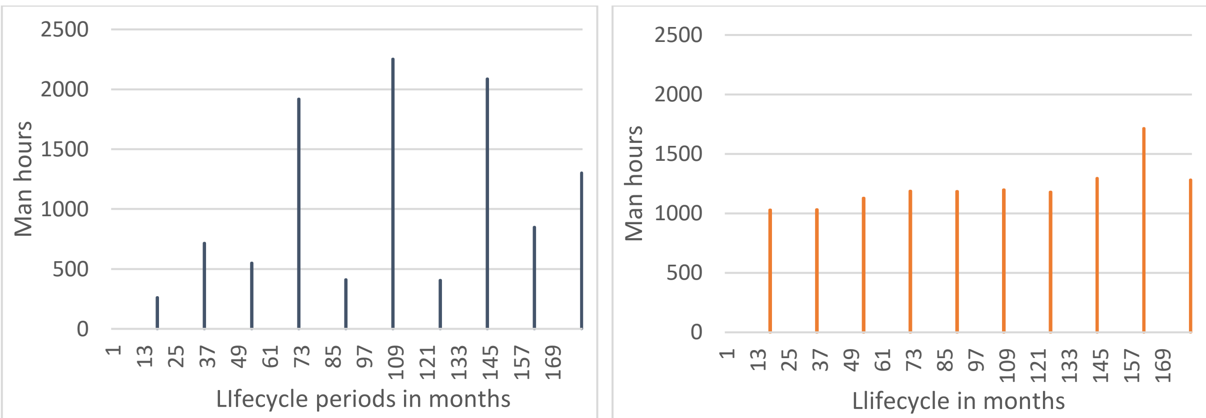


Figure 49: Initial vs. optimised C-check workload distribution with HSC=1200 mh & IRL=100%

An HSC of 1200 mh can apparently not be achieved for the C-check distribution. The optimisation could not continue beyond the 9th C-check. Even if there seems to be enough available capacity at the first 3 periods, the interval of the tasks selected for reallocation was not large enough to reach the first periods, even with an IRL of 100%.

	Initial	**IRL = 100%**
Mean absolute deviation (mh)	651.09	124.19*
Number of reallocations	-	355*
Total labour (mh)	10743.7	12232.59*
Absolute additional labour over 180 months (%)		13.86*
Relative additional labour (%)	-	11.49*

Table 11: Optimisation data for the C-check workload distribution (HSC=1200 mh & IRL=100%) (: optimisation not completed)*

<u>HSC = 1321 mh & IRL = 100%</u>

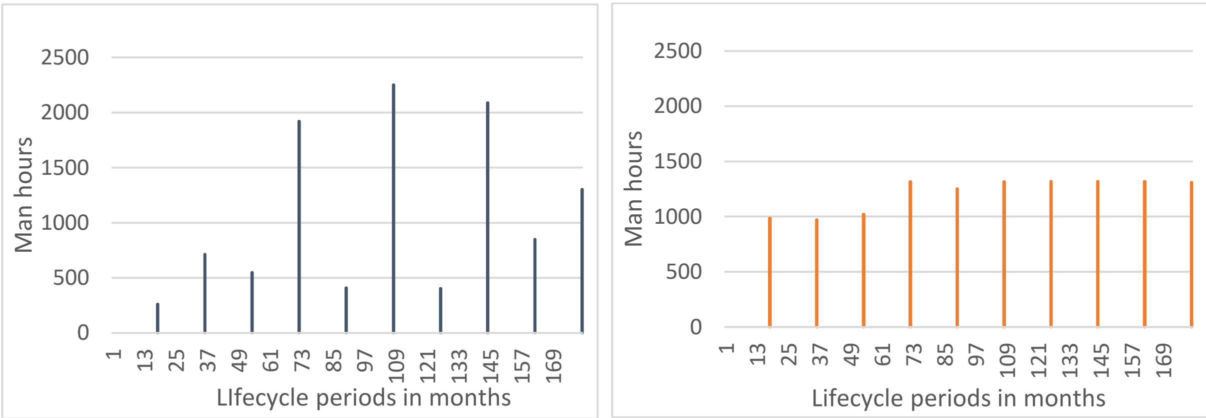


Figure 50: Initial vs. optimised C-check workload distribution with HSC = 1321 mh & IRL = 100%

The lowest achievable value of the HSC turns out to be 1321 mh. This value was achieved for 13% additional labour, which is considerable. But on the other hand, the maximum size of the workforce has been reduced from 2253 mh to 1321 mh.

	Initial	**IRL = 100%**
Mean absolute deviation (mh)	651.09	131.72
Number of reallocations executed	-	357
Total labour (mh)	10743.7	12140
Absolute additional labour over 180 months (%)	-	13
Relative additional labour (%)		12.04

Table 12: Optimisation data for the C-check workload distribution (HSC=1321 mh & IRL=100%)

8.3 EQUALISED WORKLOAD DISTRIBUTION OPTIMISATION

To demonstrate the potential of the reallocation model, the complete workload distribution (A & C-checks included) has been optimised. To achieve this, the tasks that would normally be allocated to C-checks have now been spread out over the preceding A-check by reallocation.

<u>HSC = 215 mh & IRL = 100%</u>

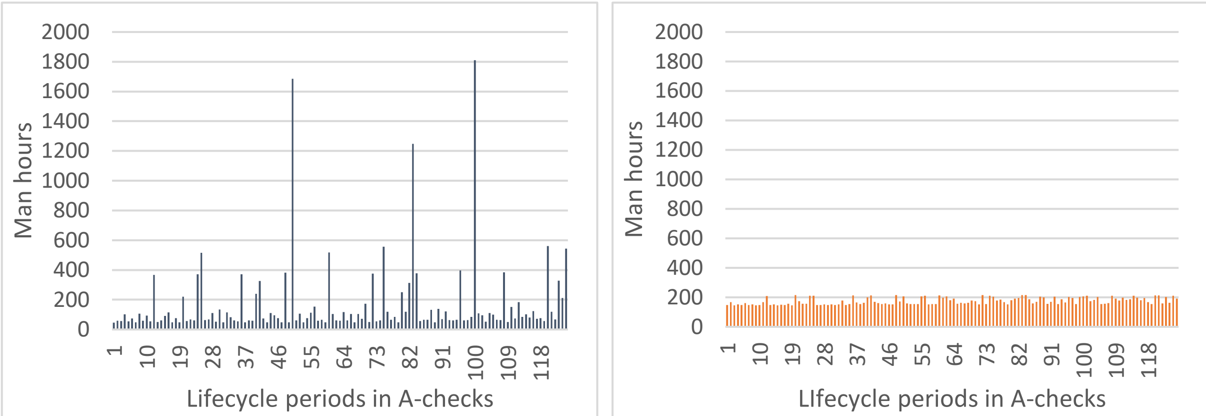


Figure 51: Initial vs. optimised total workload distribution with HSC=215 mh & IRL=100%

Just as for the optimisation of the C-check workload optimisation, some tasks are very large in this distribution, which makes an almost perfectly constant workload distribution difficult to achieve. However, the elimination of the fluctuation is very efficient; only 10% additional labour was generated and the mean absolute deviation has been reduced from 137.47 to just 18.47 mh.

	Initial	**IRL = 100%**
Mean absolute deviation (mh)	137.16	21.42
Number of reallocations executed	-	986
Total labour (mh)	20483.1	22104.64
Absolute additional labour over 126 A-checks (%)	-	7.92
Relative additional labour (%)	-	8.84

Table 13: Optimisation data for the total workload distribution (HSC=215 mh & IRL = 100%)

Figure 52 shows a more detailed view of the optimised distribution of Figure 51. The remaining workload fluctuation is clearly visible. This is caused by the large size of the tasks attributed to the C-checks, which are more difficult to reallocate.
For the current optimisation, the required size of the constant workforce would equal 215 mh for each A-check, which more than doubles the minimum value of 94 mh for the HSC obtained in the A-check (see Figure 46). But on the other hand, there is no need for large operational interruptions anymore since all the large workload peaks have been successfully eliminated.

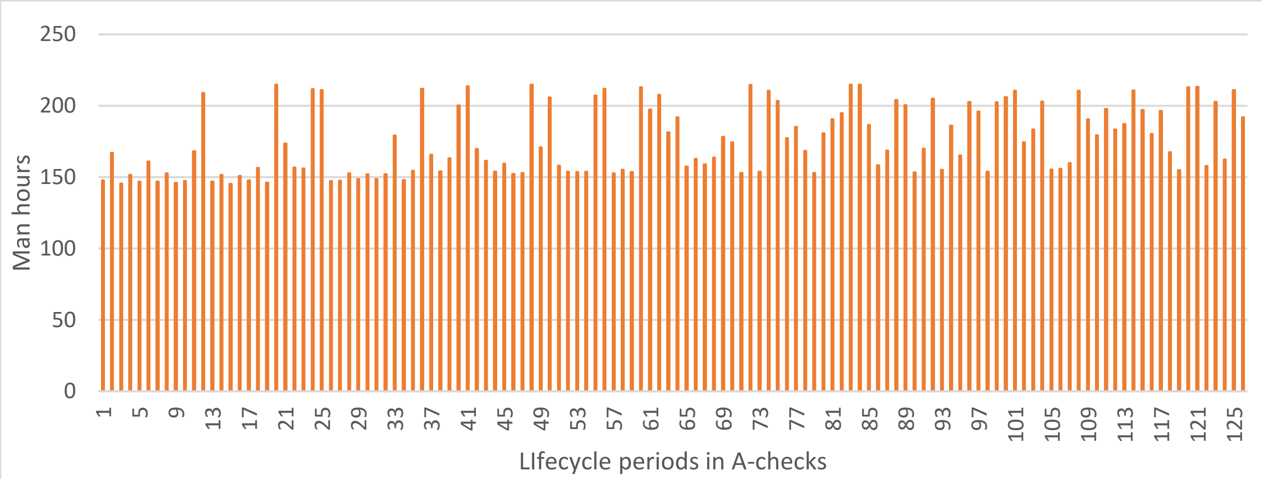


Figure 52: Detailed view of the optimised total workload distribution (HSC=215 mh & IRL=100%)

8.4 SUMMARY

The optimisations carried out for the various workload distributions in this chapter, have shown to be very promising. For the A-check workload distribution, it was possible to reduce the mean deviation from approximately 36 to just 4 mh, which resulted in an almost constant workload distribution that stays below 94 mh for the entire lifecycle (15 years). The additional labour generated (relative and absolute) was kept under 2% which is surprisingly low. For the C-check distribution, the mean deviation was reduced from 651 to 131 mh, keeping the workload fluctuation under 215 mh. Due to the larger tasks in the C-check workload distribution, the elimination of the fluctuation was slightly less effective but still very significant. Only 13% additional labour was generated and the entire workload distribution was kept under 1321 mh.

The equalised concept in which both A- and C-check tasks are combined into a single workload distribution, was easily optimised as well. For less than 8% additional labour, the mean deviation was reduced from 137 to 21.41 mh and the entire workload stayed under 215 mh.

	A-check workload distr.	**C-checks workload distr.**	**Equalised workload distr.**
Min. HSC (mh)	94	1315	215
Min. absolute deviation (mh)	3.89	131.72	21.42
Additional labour	< 2%	< 13 %	< 8%

Table 14 : Summary of the final optimisation results

The use of the IRL during the optimisation had the objective to reduce the additional labour generated by the interval reductions. However, this input is most useful for HSC values above the minimum achievable values, since it limits the capacity of the reallocation process to 'squeeze' the workload distributions. In the end, the maximal reduction of HSC is thought to outweigh the reduction of the additional labour by the IRL. Keeping the IRL equal to 100% seems therefore more appropriate.

9 EVALUATION

The optimised workload distributions obtained in chapter 8 have shown that it is possible to efficiently adapt the maintenance schedule of an aircraft for a much more constant workload distribution, while respecting the intervals prescribed by the MPD and producing limited additional costs (relative additional labour).
This evaluation chapter starts with an approach overview that summarises the essential steps taken in the development of this maintenance schedule optimisation method. This is followed by a more detailed discussion of the advantages/disadvantages related to this method. As a final section, potential improvements for the method and for the software application will be suggested as well.

9.1 APPROACH OVERVIEW

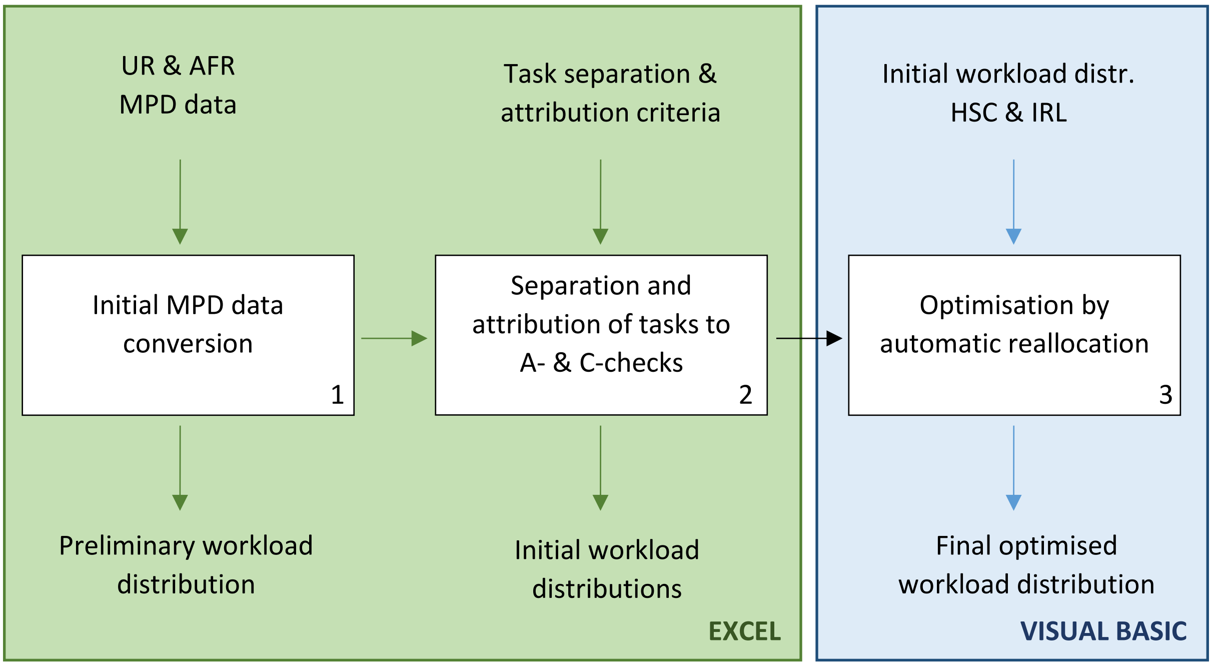


Figure 53: Overview of the elements contributing to the optimisation approach

Figure 53 shows an overview of the elements that have contributed to the final maintenance schedules and workload distributions. During the process, Excel and Visual Basic have been used separately for the different stages of the process. At each stage, inputs and parameters are used to generate both initial and final workload distributions.
At stage 1, the AU, the AFR, the task and flight envelope data from the MPD was used to generate a first impression of the workload distributions that would result from the MPD data.
At stage 2, the tasks are separated and attributed to the different base maintenance opportunities according to specific criteria. Through this generic model, different initial workload distributions can be obtained for a better adaptation of the schedule to the maintenance constraints on the ground. At the same time, initial task lists in the form of .csv

files are generated. At stage 3, these .csv files together with the HSC and IRL are used as inputs for the software assisted reallocation process that is carried out in Visual Basic. The final optimised workload distributions are then obtained.

9.2 ADVANTAGES & DISADVANTAGES

9.2.1 Benefits of a constant workload

In the problem statement of this research, the inconvenience of a workload fluctuation in the fleet context was illustrated. It was concluded that it would not be realistic to assume that an operator can adapt the size of its maintenance workforce to large workload fluctuations, especially if the unpredictability of delays is considered.

However, even if it seems the better option, with large workload fluctuations, a constant workforce will either result in maintenance delays (when workload > workforces size) or in an inefficient usage of the workforce (when workload < workforce size) or other resources such as the hangar, tools, man power etc.

The results of chapter 8 have shown that generating a nearly constant workload distribution is achievable at relatively low additional costs. Moreover, the additional labour generated as a result of interval reductions, is believed to be largely compensated by the much smaller size of the workforce required for constant workloads.

The capacity of the model to adjust the base maintenance schedule of an aircraft such that a minimum and nearly constant workforce size is required along the lifecycle of the aircraft, is believed to be very advantageous. This applies also to the maintenance slot planning organisation in which a constant workforce may considerably reduce complexities related to large fleets, delays, changes in slot allocations etc.

9.2.2 Potential of a generic optimisation model

The optimisation method developed in this chapter only reached a limited generic level. Nevertheless, the advantages of easily generating different initial workload distributions (and maintenance schedules) based on the MPD data and according to specific task criteria and operational parameters, are thought to be very promising.

With the MPD as a basis, there is a possibility to distinguish tasks based on their interval, their size, their type (see Figure 11), the ATA chapter or maintenance program to which they belong and even the aircraft zone in which they are carried out. The generic model automatically allocates the tasks to specific maintenance opportunities and the initial workload distributions (prior to optimisation) are generated. Next to generating separate A- and C-check distributions, the model can also generate a single workload distribution that combines A- and C-check tasks.

These generic capacities of the model lead to maintenance schedules that are more adaptable to the maintenance requirements or constraints of the operator and may result in the more efficient organisation of maintenance activities.

9.2.3 Increased costs due to interval reductions

As explained in 9.2.1, the additional labour generated by the interval reductions in the reallocation process is largely compensated by the much smaller workforce required with a constant workload distribution. But the interval reduction may also produce other additional costs. This applies to the tasks that involve the replacement of components. Through the relatively random reallocation process, the initial lifetime of these components can be significantly reduced, resulting in additional costs.

According to the MPD, most of the components that need to be replaced after some time are negligible in terms of costs; water/air/sensor filters, ULB/ELT batteries, O and D rings on fuel feed pipes, cartridges for fire extinguishers etc. But some tasks also involve the replacement of very expensive parts. Most of them are found in the maintenance program of the power plant (life limited parts in the HP/LP/IP compressor, the hub and LPC drum etc.). These replacements will have a larger impact on additional costs if their intervals are reduced.

9.2.4 Disadvantages of an indiscriminate reallocation process

In the reallocation process, tasks are quite randomly reallocated to new periods. By doing so, each A- or C-check will consist of a task package that is partially different from one maintenance check to the other.

This varying task package affects the level of efficiency that would be achievable through routine and work standardization if each A- or C-check would consist of the same set of tasks. With identical task packages, the workforce can anticipate the work that needs to be carried out and can benefit from routine advantages. After the reallocation process, this may have become impossible for a large of the A- or C-check. Note that if the reallocation process would equally reduce all the intervals between the task sequences (see Figure 34), more homogeneous task sets would take place at the maintenance checks. Unfortunately, this type of interval reduction would lead to a much larger increase of additional labour, as explained in 7.1.

Another potential disadvantage of the reallocation process, is that it may separate or displace tasks that are better carried out together. This may result for example in the excessive opening and closing of access panels. If two different tasks with an identical interval require the opening/closing of the same access panels, carrying out these tasks together at the same maintenance opportunity will save costs and time. Unfortunately, the reallocation process does not take these aspects into consideration and these specific tasks can in principle be reallocated to completely different maintenance opportunities. The advantage of carrying out these tasks together would therefore be lost.

9.3 IMPROVEMENTS

9.3.1 Additional task separation/clustering criteria

Some of the inconveniences mentioned in 9.2.4 can be avoided by adding task criteria to the generic model developed in chapter 6, with which the initial workload distributions were generated. Similarly to the criteria with which the application separates tasks from each other (A- of C-check tasks), a criterion could be implemented to cluster tasks together. These tasks would be reallocated together, avoiding the excessive opening/closing of access panels for example. Clustering symmetrical tasks that require identical activities, may also be a useful option. Since the MPD contains detailed task related information including codes that specify the type of task, the zone, the panels to be removed etc. (see Figure 10), the implementation of this pre-clustering criterion should be relatively simple.

Similarly, the option to prevent tasks from being reallocated, could also be a useful feature for the tasks that require expansive components to be replaced which, as explained in 9.2.3, increase component costs.

9.3.2 Reallocation process improvements

The optimisation process in 7.2.3 is carried out by locally reallocating individual tasks to the maintenance opportunity where the workload is lowest. If the average (or total) excess is reduced after this reallocation, the process continues by selecting the next task. If not, the same task is reallocated to the next maintenance opportunity where the workload is lowest (see 7.3.2).

Instead of reallocating tasks to the period where the workload is lowest, the efficiency of the reallocation process can be improved by reallocating tasks directly to the period for which the average excess is reduced the most. This slightly different approach is thought to reduce the additional labour generated after optimisation, but it may also reduce the number of reallocations required for the entire optimisation.

This process can even be further improved by using algorithms such as Dijkstra's algorithm, which seems adequate for the reallocation process. The aim of this algorithm in to find shortest path between two points in a network. For the reallocation process, the goal would be to find the right order of reallocations (the path) leading to the lowest value of the total excess (shortest length) at the end of the process.

Figure 54, illustrates Dijkstra's algorithm applied to the reallocation process. Each step corresponds to an excess in the initial workload distribution that needs to be eliminated; each option corresponds to the combination of a task being attributed to a maintenance opportunity. Each option generates additional labour. For this simple example, there are 27 (3^3) possible paths.

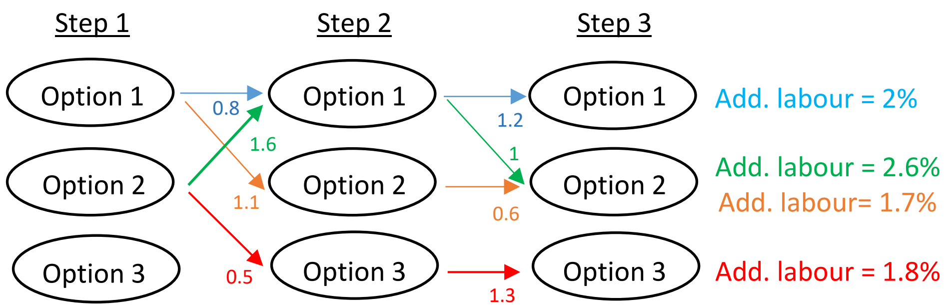


Figure 54: Illustration of Dijkstra's algorithm applied to the reallocation process

Integrating this sort of algorithms in the software application may lead to optimal solutions with minimum workload variations and a minimum number of reallocations.

Another method that may also improve the reallocation process is by starting the reallocation process at the highest workload peak instead of at period 1, and by selecting the largest tasks in terms of size. Since it is easier to allocate large tasks to new periods at the beginning of the process (when available capacities are the largest), it may be easier to obtain flatter workload distributions, especially for the C-check workload distribution that contains tasks of up to 97 mh.

These different process improvements would require the development of a completely different software code. Unfortunately, it was impossible to evaluate these possibilities within the scope of this research.

9.3.3 Additional features and implementations for the basis of a complete and operational software assisted optimisation model

Although the main objective of this research focused on the development an efficient and generic method rather than a fully operational software application, the software model built for the reallocation process is thought to be a very promising part of this research. Further developing it could be very interesting. Therefore, a few suggestions on possible additional features and implementations that may form the basis of a complete and efficient software optimisation package, are briefly discussed in this chapter.

<u>A detailed task list interface</u>

The interface used in the reallocation application and illustrated in 7.3.3 was very simple. It had just two inputs (HSC & IRL) that could be altered. To incorporate the improvements suggested in this chapter, a more detailed version of this interface, focused on the tasks, may be required. For example, by displaying the entire task list in this interface, the tasks that need to be clustered, separated, assigned C-checks exclusively etc. could marked and stored as such, prior to the optimisation process.

Also, modifying task intervals because of regulation changes, could also be an option as well.

Dynamic operational inputs

For the scope of this research, constant average operational inputs (AFR & AU) have been used to calculate the due dates of both tasks and maintenance opportunities, over a period of 15 years. However, these constant values consider the unexpected delays or changes in the operational pattern of the aircraft. For a very precise estimation of due dates, a module keeping track of the aircrafts flight hours and flight cycles, would need to be integrated in this model. This would improve the process of allocating aircraft to slots and it would increase the capacity to anticipate potential delays.

Overview of scheduled tasks at each maintenance opportunity

The primary objective of this research was to 'flatten' the workload distribution by reallocating tasks to different maintenance opportunities. This was done with the use of interval matrices that only displayed the sizes and intervals of each task. To be useful for the operator and the maintenance workforce, the application should clearly indicate the tasks that are due at each maintenance opportunity (A- or C-check). A feature displaying/printing an overview of the MPD task codes for each maintenance opportunity, can be thought of.

Complete integration in Visual Basic (or any other appropriate software program)

As shown in Figure 53, the first steps required in this optimisation approach were carried out in Excel. To facilitate the eventual use of the model, all the inputs used in the current model, including those proposed in this chapter, could be fully integrated in Visual Basic. Ideally, to optimise the maintenance schedule of an aircraft, the operator would only have to upload an MPD and adjust the various required inputs and constraints. A summary of all the features that could be integrated into a single Visual Basic application, is illustrated in Figure 55.

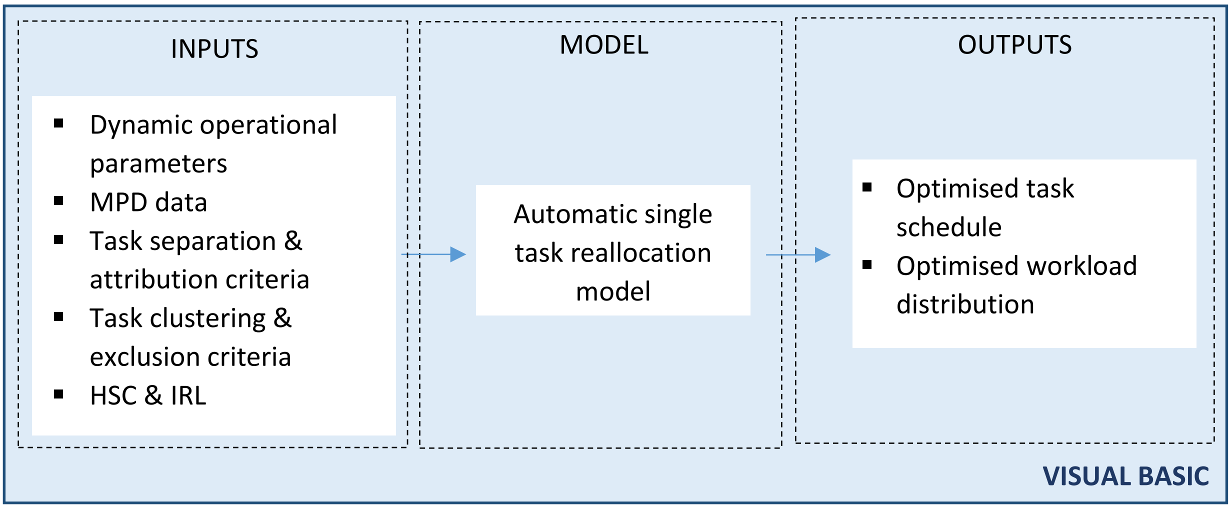


Figure 55: Overview of the integrated maintenance schedule optimisation application

10 CONCLUSION

The optimisation method developed in this research used two essential steps to optimise the maintenance schedule of an aircraft; the generation of the initial MPD workload distributions, and the single task reallocation process through which the final optimised workload distributions were obtained.

Although this method focused on the reallocation process and its software application, the generic model through which the initial workload distributions were obtained, is believed to be a very useful part of it. This model has the potential capacity to separate or cluster tasks based on all the task specifications in the MPD. It then automatically attributes these tasks to specific maintenance opportunities. This step that precedes the task reallocation process, makes the final maintenance schedules more adaptable to the maintenance requirements and/or preferences of the operator and may lead to a more efficient organisation of maintenance activities.

The single task reallocation process applied in this research has shown to be very efficient in minimising the initial workload fluctuation that results from the MPD. Using a software application, the maintenance workload fluctuations have been minimised in a matter of seconds, while respecting the conditions prescribed by the MPD. As such, for the A-checks, the mean absolute deviation of the workload distribution over a period of 15 years, was reduced by 89%, while generating less than 2% additional labour as a result of interval reductions. For the C-checks, the mean absolute deviation of the workload distribution was reduced by 79% with less than 13% additional labour.

The application of an algorithm or a different process order could improve the process even more to reduce the additional labour generated and the number of reallocations needed to eliminate the workload fluctuations.

The additional costs and inconveniences caused by the interval reductions and the relatively random reallocation of tasks, are believed to be largely compensated by the advantages of a constant workload; a much smaller workforce, a considerable simplification of the workforce scheduling activities and a more efficient usage of the workforce and other maintenance resources. This is particularly true when large fleets of aircrafts are considered.

Even if this research aimed to develop a methodology rather than a fully developed software application, it could form the basis for the development of a complete and automated maintenance schedule optimisation model in which ideally, an operator would only have to upload the MPD of any aircraft and set the criteria and inputs it requires (HSC, task clustering/separation criteria etc.). Optimised maintenance schedules with constant workload distributions would be obtained almost instantly, regardless of the type of aircraft, its operational parameters etc.

The ability to quickly generate, adapt and optimise any initial maintenance schedule that results from a MPD, could even facilitate the decision of purchasing new aircraft models with totally different/unknown maintenance schedules.

Finally, the development of such a model that generates optimised maintenance schedules through a ‘free’ rearrangement of maintenance tasks and opportunities, seems to be supported by the current trends in the aircraft maintenance sector. Operators and manufacturers tend to move away from the rigid letter check systems with the objective to obtain more flexible maintenance schedules that are more adaptable to the preferences or requirements of the operator.

REFERENCE LIST

Airbus, Scheduled maintenance seminar, Airbus (2013), *International Civil Aviation Organisation(http://www.coscap-gs.org/Scheduled-Maintenance-Seminar_Part5_May2013.pdf)*

A. Ahmadi, P. Söderholm & U. Kumar (2010), On aircraft scheduled maintenance program development", *Journal of Quality in Maintenance Engineering, Vol. 16 Issue 3 pp. 229 - 255*

A. Ahmadi, P. Söderholm, & U. Kumar (2007), An overview of trends in aircraft maintenance program development: Past, present, and future, *Luleå University of Technology, Luleå, Sweden*

A.K. Muchiri (2002), Maintenance Planning Optimisation for the Boeing 737 Next Generation, *Master Thesis Delft University of Technology*

A. Sharma, G.S. Yadava, S.G. Deshmukh (2011), A literature review and future perspectives on maintenance optimisation, *Journal of Quality in Maintenance Engineering*

C. Şentürk, Şerif Kavsaoğlu and Nikbay (2010), Optimisation of Aircraft Utilization by Reducing Scheduled Maintenance Downtime*, Conference Paper Turkish Technical University, Istanbul*

C. Sriram 1, A. Haghani (2001), An optimisation model for aircraft maintenance scheduling and re-assignment, Department of Civil and Environmental Engineering, *University of Maryland*

Federal Aviation Administration, AFS-330, Maintenance review board procedures (1997), US Department of Transportation, *, Advisory Circular*

H. Yamashina S. Otani (2001), Optimal preventive maintenance planning for multiple elevators, *Journal of Quality in Maintenance Engineering*

J. Beliën, B. Cardoen, E. Demeulemeester (2012), Improving Workforce Scheduling of Aircraft Line Maintenance at Sabena Technics, Interfaces

J. Beliën, E. Demeulemeester, B. Cardoen (2010), Integrated staffing and scheduling for an aircraft line maintenance problem, *University Brussels/Leuven*

M. Lohatepanont, C. Barnhart (2004) Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment, *Transportation Science*

M. Shannon P. Ackert (2010), Basics of Aircraft Maintenance Programs for Financiers, Evaluation & Insights of Commercial Aircraft Maintenance Programs

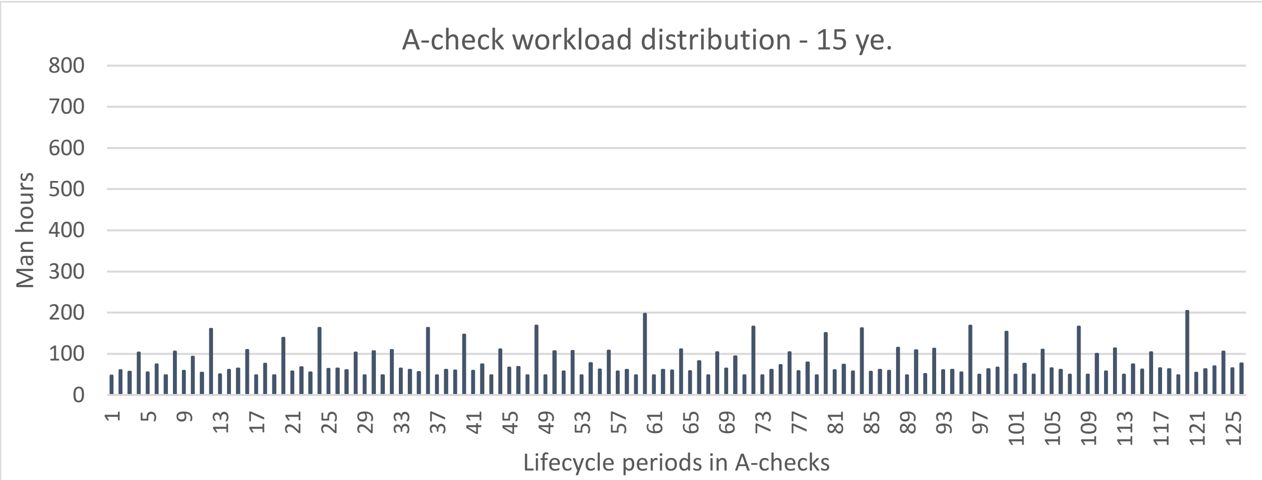
Transport Studies Group University of Westminster London (2008), Innovative Cooperative Actions of R&D in EUROCONTROL Programme CARE INO III, Dynamic Cost Indexing, Technical Discussion Document 9.0

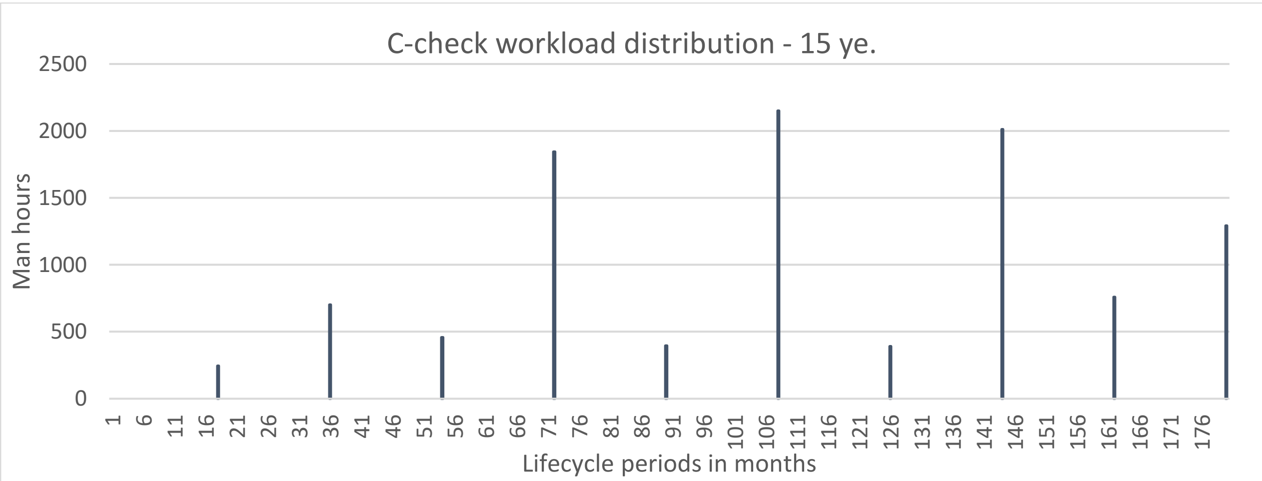
APPENDIX I : VARIOUS INITIAL WORKLOAD DISTRIBUTIONS

In chapter 6, the model that generates the initial workload distributions uses several task separation criteria to attribute tasks to A- or C-checks (interval, size and category, see 6.1).
In this appendix, different initial workload distributions have been generated according to different task selection criteria to see how these criteria affect the initial workload distributions.
The boxes above the workload distributions (screenshots of the actual model in excel), indicate the task selection criteria and aircraft operational parameters that have been set for the corresponding workload distributions (see next page).

Aircraft operational param.	
Average Fly Route in FH (AFR)	10
Average Utilisation in % (AU)	90

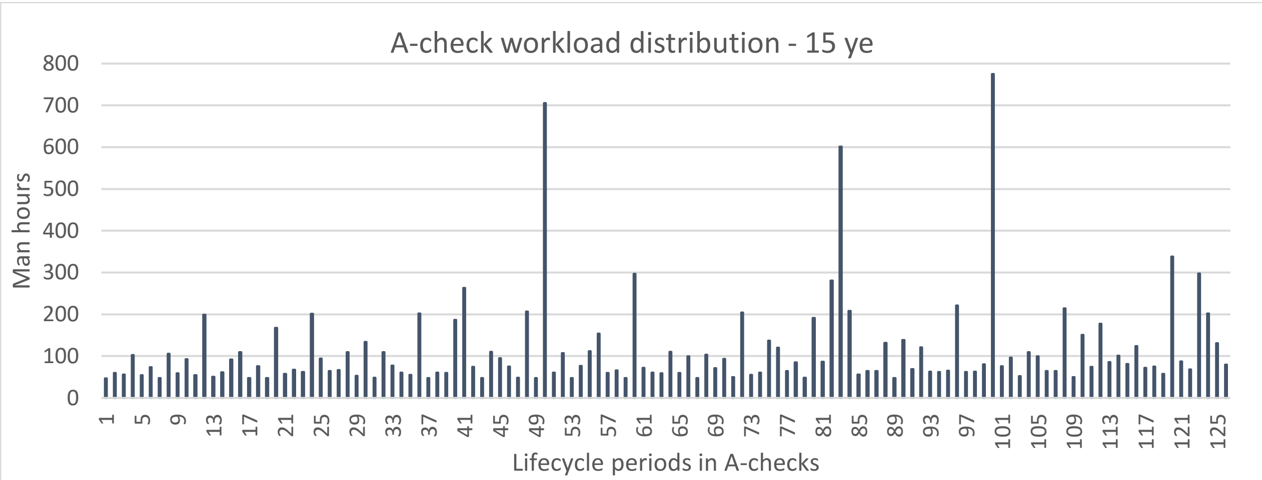
Task selection criteria	A-check	C-check
Systems & Components	☑	☐
Power Plant	☐	☑
APU	☐	☑
Zonal	☐	☑
Structures	☐	☑
Include/Exclude tasks (mh)	< 2	> 2
All C-check tasks included	☐	☐

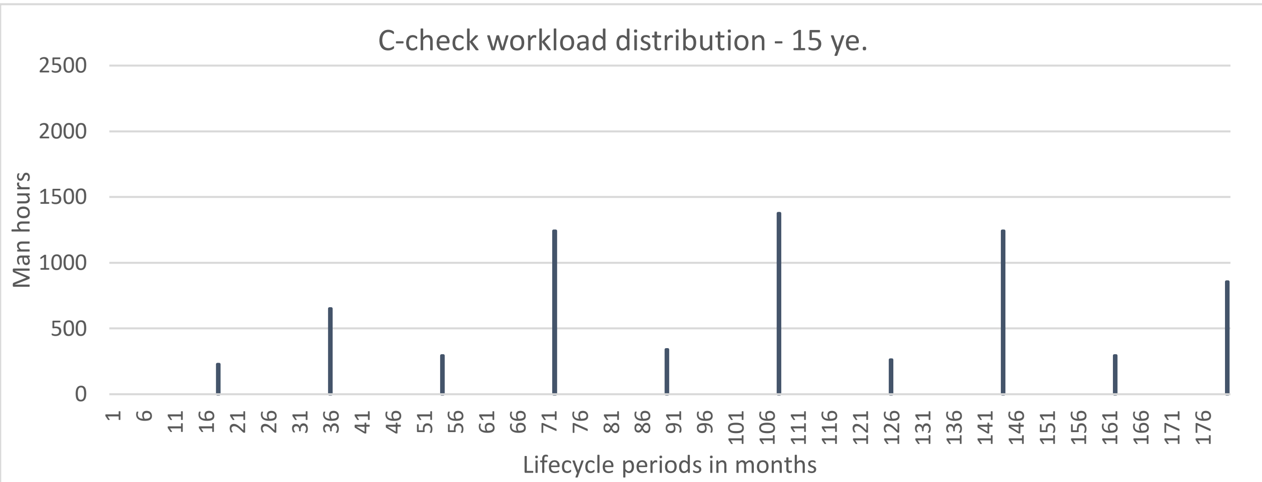




Aircraft operational param.	
Average Fly Route in FH (AFR)	10
Average Utilisation in % (AU)	90

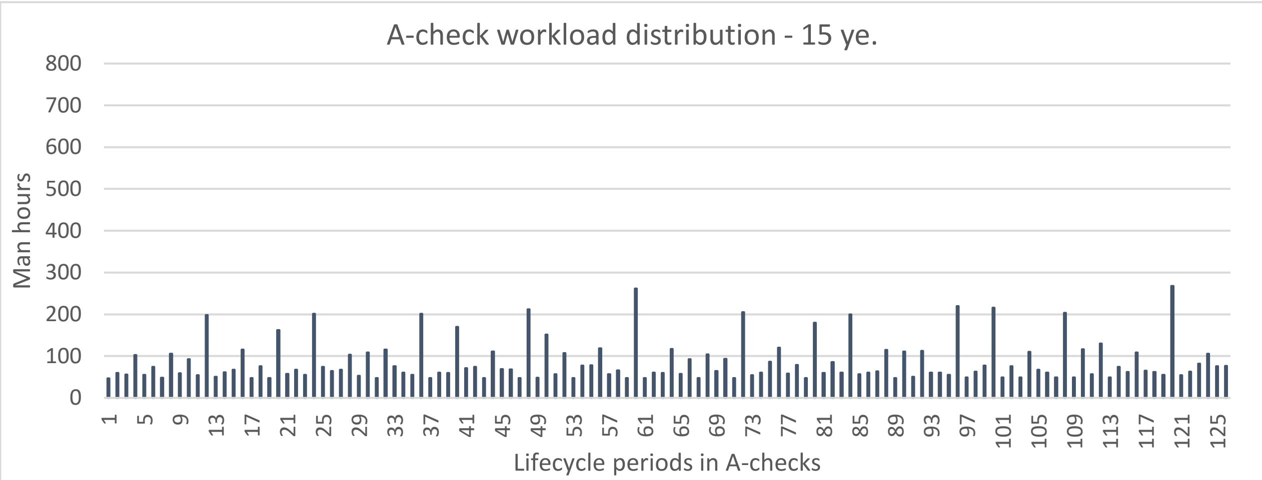
Task selection criteria	A-check	C-check
Systems & Components	☑	☐
Power Plant	☐	☑
APU	☐	☑
Zonal	☐	☑
Structures	☑	☐
Include/Exclude tasks (mh)	< 25	> 25
All A-/C-checks combined	☐	☐

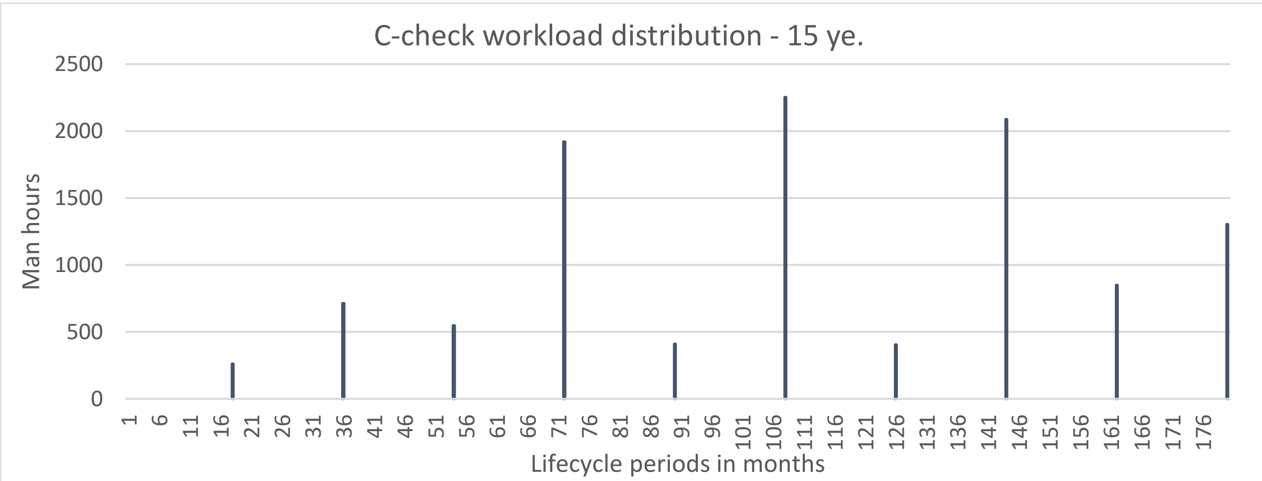




Aircraft operational param.	
Average Fly Route in FH (AFR)	10
Average Utilisation in % (AU)	90

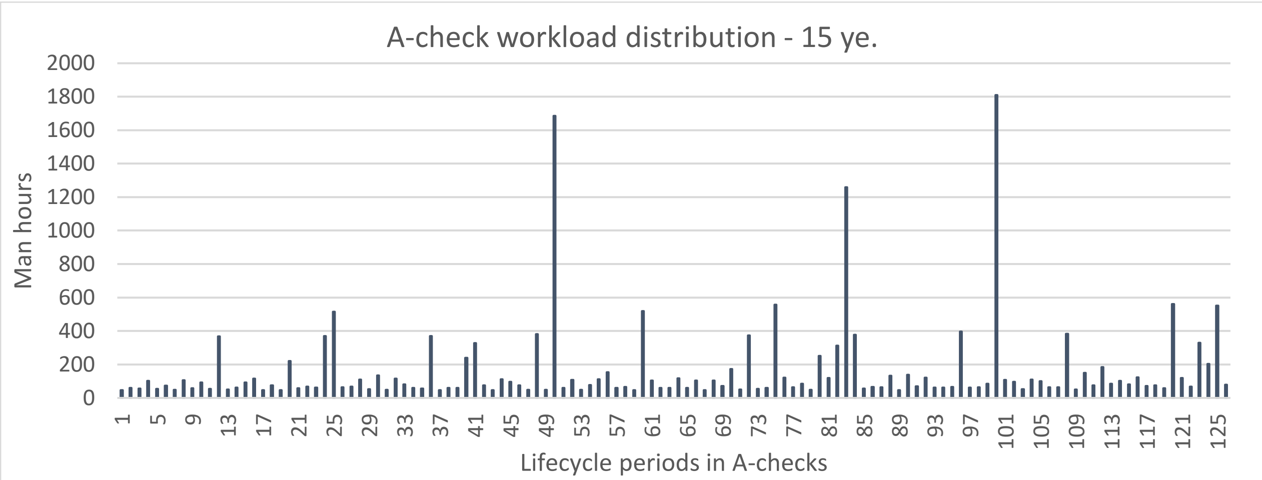
Task selection criteria	A-check	C-check
Systems & Components	☑	☐
Power Plant	☑	☐
APU	☑	☐
Zonal	☑	☐
Structures	☐	☑
Include/Exclude tasks (mh)	< 12	> 12
All C-check tasks included	☐	☐

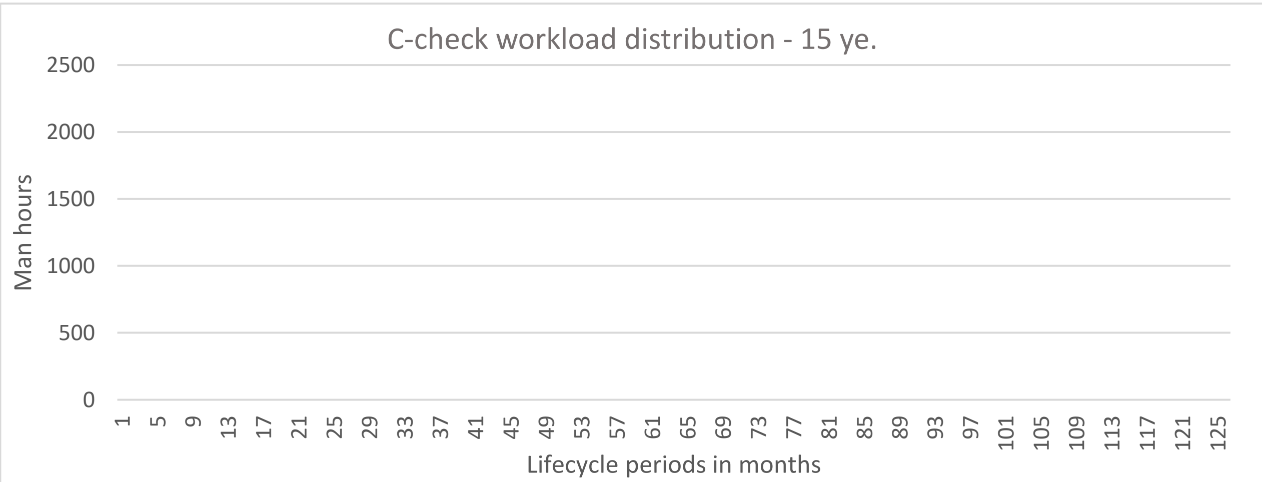




Aircraft operational param.	
Average Fly Route in FH (AFR)	10
Average Utilisation in % (AU)	90

Task selection criteria	A-check		C-check	
Systems & Components	☐		☐	
Power Plant	☐		☐	
APU	☐		☐	
Zonal	☐		☐	
Structures	☐		☐	
Include/Exclude tasks (mh)	<	-	>	-
All C-check tasks included	☑		☑	





APPENDIX II : VISUAL BASIC CODE

```
// Áircraft maintenance.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include <math.h>
#include <Strsafe.h>

#include "Áircraft maintenance.h"

#define MAX_LOADSTRING 100
#define BUFFERSIZE     18000
#define MAXTASKS       720
#define MAXPERIODS      1400


// Prototypes
intFindTaskNo(int);
intSearchPossibleFit(int,int);

// Global Variables:
HINSTANCEhInst;// current instance
WCHARszTitle[MAX_LOADSTRING];// The title bar text
WCHARszWindowClass[MAX_LOADSTRING];// the main window class name

// Global varialbles;
intvooraflyne=3;
intscheduleHorizon=10;
doubleIntervalReductionFactor=0.7;
intnTask=720;
inti;
TCHAR*WorkDirectory;
inttaskno;
inttask;
//period number]
intperno;
intno_of_tasks=nTask;
floatTotalHours[MAXPERIODS];
doubleManHours[MAXPERIODS][MAXTASKS];
intInterval[MAXPERIODS][MAXTASKS];
intThreshold[MAXTASKS];
intCurrentTask;
intCurrentPeriod;
intFoundPerno;
intHangarCap=150;
intno_of_periods=MAXPERIODS;
intReductionLimit=100;/* interval reduction limit*/
floatAverage;
floatTotal;
floatAverageExcess;
intrest;
char*szBuffer;
char*lpBuffer;
intno_acts;
charszBuf[BUFFERSIZE];
intDistance;

// Forward declarations of functions included in this code module:
ATOMMyRegisterClass(HINSTANCEhInstance);
BOOLInitInstance(HINSTANCE,int);
intSaveOptimisedDistribution(HWND);
LRESULTCALLBACKWndProc(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKAbout(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKOpen(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKPrintEverything(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKInput(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKTaskList(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKInitialDistribution(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKPrintInitDistr(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKPrintOptDist(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKOptimisedDistribution(HWND,UINT,WPARAM,LPARAM);
INT_PTRCALLBACKSetting(HWND,UINT,WPARAM,LPARAM);
```

```
VOIDScroll(HWND,int,WORD);
doubleCalculateTotalManHours(void);



structtask
{
unsignedcharNPD_chapter[6];
intInterval;
floatManHours;
intThreshold;
}Task[MAXTASKS];



struct{
intntasks;
}data;



struct{
intSceduledHorizon;
intHangarSlotCapacity;
floatIntervalReductionLimit;
charWorkDirectory[40];
}MySystem;


intAPIENTRYwWinMain(_In_HINSTANCEhInstance,
_In_opt_HINSTANCEhPrevInstance,
_In_LPWSTRlpCmdLine,
_In_intnCmdShow)
{
UNREFERENCED_PARAMETER(hPrevInstance);
UNREFERENCED_PARAMETER(lpCmdLine);

// TODO: Place code here.

// Initialize global strings
LoadStringW(hInstance,IDS_APP_TITLE,szTitle,MAX_LOADSTRING);
LoadStringW(hInstance,IDC_IRCRAFTMAINTENANCE,szWindowClass,MAX_LOADSTRING);
MyRegisterClass(hInstance);

// Perform application initialization:
if(!InitInstance(hInstance,nCmdShow))
{
returnFALSE;
}

HACCELhAccelTable=LoadAccelerators(hInstance,MAKEINTRESOURCE(IDC_IRCRAFTMAINTENANCE));

MSGmsg;

// Main message loop:
while(GetMessage(&msg,nullptr,0,0))
{
if(!TranslateAccelerator(msg.hwnd,hAccelTable,&msg))
{
TranslateMessage(&msg);
DispatchMessage(&msg);
}
}

return(int)msg.wParam;
}


intPrintEverything(HWNDhWnd)
{

HDChDC;
//    SIZE size;
intntasks;
```

```
intitask;

LPCWSTRlpText[100];


hDC=GetDC(hWnd);
wsprintf((LPWSTR)lpText,L" 1            2               3            4               5");
TextOut(hDC,200,20,(LPCWSTR)lpText,lstrlen((LPCWSTR)lpText));
wsprintf((LPWSTR)lpText,L" Interval values");
TextOut(hDC,15,20,(LPCWSTR)lpText,lstrlen((LPCWSTR)lpText));
wsprintf((LPWSTR)lpText,L" Man hours");
TextOut(hDC,100,20,(LPCWSTR)lpText,lstrlen((LPCWSTR)lpText));
ntasks=data.ntasks;
for(itask=1;itask<=ntasks;itask++)

{
wsprintf((LPWSTR)lpText,L"Task %d \n",itask);
TextOut(hDC,20,50+itask*20,(LPCWSTR)lpText,lstrlen((LPCWSTR)lpText));

}
return0;
}

//
// FUNCTION: MyRegisterClass()
//
// PURPOSE: Registers the window class.
//
ATOMMyRegisterClass(HINSTANCEhInstance)
{
WNDCLASSEXWwcex;

wcex.cbSize=sizeof(WNDCLASSEX);

wcex.style=CS_HREDRAW|CS_VREDRAW;
wcex.lpfnWndProc=WndProc;
wcex.cbClsExtra=0;
wcex.cbWndExtra=0;
wcex.hInstance=hInstance;
wcex.hIcon=LoadIcon(hInstance,MAKEINTRESOURCE(IDI_IRCRAFTMAINTENANCE));
wcex.hCursor=LoadCursor(nullptr,IDC_ARROW);
wcex.hbrBackground=(HBRUSH)(COLOR_WINDOW+1);
wcex.lpszMenuName=MAKEINTRESOURCEW(IDC_IRCRAFTMAINTENANCE);
wcex.lpszClassName=szWindowClass;
wcex.hIconSm=LoadIcon(wcex.hInstance,MAKEINTRESOURCE(IDI_SMALL));

returnRegisterClassExW(&wcex);
}

////  FUNCTION: InitInstance(HINSTANCE, int)
////  PURPOSE: Saves instance handle and creates main window
////  COMMENTS:
////      In this function, we save the instance handle in a global variable and
//       create and display the main program window.
//
BOOLInitInstance(HINSTANCEhInstance,intnCmdShow)
{
hInst=hInstance;// Store instance handle in our global variable

HWNDhWnd=CreateWindowW(szWindowClass,szTitle,WS_OVERLAPPEDWINDOW,
CW_USEDEFAULT,0,CW_USEDEFAULT,0,nullptr,nullptr,hInstance,nullptr);

if(!hWnd)
{
returnFALSE;
}

ShowWindow(hWnd,nCmdShow);
UpdateWindow(hWnd);

returnTRUE;
}

//
```

```
// FUNCTION: WndProc(HWND, UINT, WPARAM, LPARAM)
//// PURPOSE: Processes messages for the main window.
//// WM_COMMAND - process the application menu
// WM_PAINT - Paint the main window
// WM_DESTROY - post a quit message and return
////
LRESULTCALLBACKWndProc(HWNDhWnd,UINTmessage,WPARAMwParam,LPARAMlParam)
{
// TCHAR szBuffer[40];
// DWORD dwArgs[3];
LPCTSTRlpText=L"String to fit";
LPCTSTRTextString=L"MyText";
RECTWindowRect;
HDChDC;
intx;

Task[1].Interval=1;


switch(message)
{
caseWM_COMMAND:
{
intwmId=LOWORD(wParam);
// Parse the menu selections:
switch(wmId){

caseIDM_ABOUT:
DialogBox(hInst,MAKEINTRESOURCE(IDD_ABOUTBOX),hWnd,About);
break;

caseID_PRINT_INITDISTR:
{
hDC=GetDC(hWnd);
SetRect(&WindowRect,10,10,400,50);
x=sizeof(TextString);
DrawTextW(hDC,TextString,sizeof(TextString),&WindowRect,DT_CENTER);
ReleaseDC(hWnd,hDC);
}
break;

caseID_FILE_OPEN:
DialogBox(hInst,MAKEINTRESOURCE(IDD_OPEN),hWnd,Open);
break;

caseID_PRINTEVERYTHING:
PrintEverything(hWnd);
//DialogBox(hInst, MAKEINTRESOURCE(IDD_PRINTEVERYTHING), hWnd, PrintEverything);
break;

caseID_INPUT:
DialogBox(hInst,MAKEINTRESOURCE(IDD_INPUT_TASK_TABLE),hWnd,Input);
break;

caseID_TASK_LIST:
DialogBox(hInst,MAKEINTRESOURCE(IDD_TASK_LIST),hWnd,TaskList);
break;

caseID_MAIN_SCHEDULE:
DialogBox(hInst,MAKEINTRESOURCE(IDD_INITIAL_DISTRIBUTION),hWnd,InitialDistribution);
break;

caseID_OPTIMISED_SCHEDULE:
DialogBox(hInst,MAKEINTRESOURCE(IDD_OPTIMISED_DISTRIBUTION),hWnd,OptimisedDistribution);
break;


caseID_SETTING:
DialogBox(hInst,MAKEINTRESOURCE(IDD_SETTING),hWnd,Setting);
break;

caseIDM_EXIT:
DestroyWindow(hWnd);
break;
```

```
default:
returnDefWindowProc(hWnd,message,wParam,lParam);
}
}

caseWM_PAINT:
{
PAINTSTRUCTps;
HDChdc=BeginPaint(hWnd,&ps);
// TODO: Add any drawing code that uses hdc here...
EndPaint(hWnd,&ps);
}
break;

caseWM_DESTROY:
PostQuitMessage(0);
break;

default:
returnDefWindowProc(hWnd,message,wParam,lParam);
}
return0;
}

// Message handler for about box.
INT_PTRCALLBACKAbout(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
UNREFERENCED_PARAMETER(lParam);
switch(message)
{
caseWM_INITDIALOG:
return(INT_PTR)TRUE;

caseWM_COMMAND:
{
if(LOWORD(wParam)==IDOK||LOWORD(wParam)==IDCANCEL)
{
EndDialog(hDlg,LOWORD(wParam));
return(INT_PTR)TRUE;
}
break;
}
return(INT_PTR)FALSE;
}
}

INT_PTRCALLBACKOpen(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
UNREFERENCED_PARAMETER(lParam);
staticHANDLEhFile;
LPTSTRlpszDirName=NULL;

charszBuf[BUFFERSIZE];
//    char lpBuffer[BUFFERSIZE];

charFileName[100]="RealTaskList.csv";
intdwRead;
//    char MijnString[10];
char*header1;
char*next_token;
char*next_string;
inttask_no;
interror;

charmystring[5];
chars;

char*param1;
char*param2;
char*param3;
char*param4;
char*param5;
//char *param6;
```

```
switch(message)
{
caseWM_INITDIALOG:
{
no_acts=0;
SetDlgItemText(hDlg,IDC_EDIT1,L"C:\\Aircraft maintenance\\");
SetDlgItemText(hDlg,IDC_EDIT3,L"RealTaskList");
//     ErrorCode = GetLastError();                 ,IDC_EDIT
}
break;

caseWM_COMMAND:
{
switch(wParam)
{
caseIDC_BUTTON1:
{
//                     strcpy_s(FileToOpen, WorkDirectory);
//                     strcat_s(FileToOpen, FileName);

hFile=CreateFileW(L"C:\\Aircraft
maintenance\\RealTaskList.csv",GENERIC_READ,0,NULL,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,NULL);

if(hFile==INVALID_HANDLE_VALUE)
{
if(MessageBox(hDlg,L"Kan file niet openen ",NULL,MB_YESNO)==IDYES)
{
}
}
else
{
if(MessageBox(hDlg,L"Reading Task List \n",L"Task list",MB_YESNO)==IDYES)
{
}
}

ReadFile(hFile,szBuf,BUFFERSIZE,(LPDWORD)&dwRead,NULL);
// read first header
header1=strtok_s(szBuf,"\n",&next_token);
task_no=1;
do
{
param1=strtok_s(NULL,";",&next_token);
param2=strtok_s(NULL,";",&next_token);
param3=strtok_s(NULL,";",&next_token);
param4=strtok_s(NULL,"\n",&next_token);
if(strcmp(param1,"EOF")==0)break;
//                            strcpy_s(Task[task_no].NPD_chapter,param1.c_str());
//                            s = mystring[*param1];
Task[task_no].Threshold=atoi(param2);
Task[task_no].Interval=atoi(param3);
strtok_s(param4,"\r",&next_string);
Task[task_no].ManHours=atof(param4);
task_no++;
}while(strcmp(param1,"EOF")!=0);
no_of_tasks=task_no;
CloseHandle(hFile);
EndDialog(hDlg,IDOK);
returnTRUE;
}
break;

caseIDCANCEL:
{
EndDialog(hDlg,IDCANCEL);
}


caseIDOK:
{
EndDialog(hDlg,IDOK);
```

```
}
}
}
}
return(INT_PTR)FALSE;
}


/*
INT_PTR CALLBACK PrintInitDistr(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    LPCTSTR lpText = L"Mijn Data";
int array;

    switch (message)
    {
        case WM_COMMAND:
        {
            switch (LOWORD(wParam))
            {
                case IDC_BUTTON1:
                {
                    HDC hDC;
                    SIZE size;
                    hDC = GetDC(hDlg);
                    Rectangle(hDC, 10, 20, 220, 50);
                    TextOut(hDC, 15, 20, lpText, sizeof(lpText));

                }
                break;

                case IDOK:
                {
                    EndDialog(hDlg, IDOK);
                    return (INT_PTR)TRUE;
                }

                case IDCANCEL:
                {
                    EndDialog(hDlg, IDCANCEL);
                    return (INT_PTR)TRUE;
                }
            } // end of
        } // end of WM_COMMAND
    }  // end of switch(message)
}
*/


INT_PTRCALLBACKPrintEverything(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
TCHARszBuffer[40];
DWORDdwArgs[3];


switch(message)
{
caseWM_INITDIALOG:
{
HDChDC=GetDC(hDlg);
LPCWSTRszEdit1=L"Interval Values";
SetDlgItemText(hDlg,IDC_EDIT1,szEdit1);
wsprintf(szBuffer,(LPCWSTR)"De text regel 1 is  %s \n",L"text output\n",dwArgs);
TextOut(hDC,10,20,szBuffer,lstrlen(szBuffer));

}

break;

caseWM_COMMAND:
{
if(LOWORD(wParam)==IDOK||LOWORD(wParam)==IDCANCEL)
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}
```

```
}
break;
}
return(INT_PTR)FALSE;
}

INT_PTRCALLBACKInput(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
TCHARszBuffer[40];
DWORDdwArgs[3];


switch(message)
{
caseWM_INITDIALOG:
{
HDChDC=GetDC(hDlg);
LPCWSTRszEdit1=L"Interval Values";
SetDlgItemText(hDlg,IDC_EDIT1,szEdit1);
wsprintf(szBuffer,(LPCWSTR)"De text regel 1 is  %s \n",L"text output\n",dwArgs);
TextOut(hDC,10,20,szBuffer,lstrlen(szBuffer));
}
break;

caseWM_COMMAND:
{
if(LOWORD(wParam)==IDOK||LOWORD(wParam)==IDCANCEL)
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}
}
break;
}
return(INT_PTR)FALSE;
}


INT_PTRCALLBACKTaskList(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
staticHWNDhControl=NULL;
staticintnDspLines;
staticintnNumPos;
staticintnNumItems;
staticintnCurPos=0;
staticintinterval;
intconstarraysize=30;

LPCTSTRpszFormat=TEXT("Task %d  ");

size_tcbDest=arraysize*sizeof(TCHAR);
TCHAR*pszText=TEXT(" |");

inti;



switch(message)
{
caseWM_INITDIALOG:
{
}
break;


caseWM_COMMAND:
{
switch(LOWORD(wParam))
{
caseIDC_BUTTON2:// input task list
{
intconstarraysize=30;
TCHARpszDest[arraysize];
```

```
// Add items to list.
HWNDhwndList=GetDlgItem(hDlg,IDC_LIST1);
for(i=1;i<no_of_tasks;i++)
{
LPCTSTRpszFormat=TEXT(" %s  %3d  %s  %3d  %3d  %.2f ");
TCHAR*pszTxt=TEXT("Task ");
HRESULThr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszTxt,i,Task[i].NPD_chapter,Task[i
].Threshold,Task[i].Interval,Task[i].ManHours);
intpos=(int)SendMessage(hwndList,LB_ADDSTRING,0,(LPARAM)pszDest);
}
// Set input focus to the list box.
SetFocus(hwndList);

}
break;



caseIDC_BUTTON4:// View tasklist
{

}
break;


caseIDC_LIST1:
{
switch(HIWORD(wParam))
{
caseLBN_SELCHANGE:
{
HWNDhwndList=GetDlgItem(hDlg,IDC_LIST1);
// Get selected index.
intlbItem=(int)SendMessage(hwndList,LB_GETCURSEL,0,0);
// Get item data.
inti=(int)SendMessage(hwndList,LB_GETITEMDATA,lbItem,0);
LPCTSTRpszFormat=TEXT(" %s \t %3d \t %d  \t %.2f ");
TCHAR*pszTxt=TEXT("Task ");
}
}
}
returnTRUE;

caseIDOK:
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}

caseIDCANCEL:
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}
}
}

}
return(INT_PTR)FALSE;
}

intCreateOptimisedTaskList(HWNDhDlg)
{
HANDLEhFile;
charszBuf[BUFFERSIZE];
DWORDdwWritten;

DeleteFile(L"C:\\Aircraft maintenance\\OptimisedTaskList.csv");
hFile=CreateFileW(L"C:\\Aircraft
maintenance\\OptimisedTaskList.csv",GENERIC_WRITE,0,NULL,OPEN_ALWAYS,FILE_ATTRIBUTE_NORMAL,NULL
);
// print header
sprintf_s(szBuf,"\r\n %s \r\n","Optimised Task List \r\n");
SetFilePointer(hFile,0,NULL,FILE_BEGIN);
```

```
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);

for(taskno=1;taskno<=no_of_tasks;taskno++)
{
sprintf_s(szBuf,"%3d %3d",Threshold[taskno],Interval[taskno]);

}

CloseHandle(hFile);
returnTRUE;
}



INT_PTRCALLBACKInitialDistribution(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
TCHARAverageText[20];

// TCHAR pszDest[arraysize];
TCHAR*pszText=TEXT(" ");
intconstarraysize=40;
TCHARpszDest[arraysize];
size_tcbDest;
LPCTSTRpszFormat=TEXT("%s %.2f");
TCHAR*pskSrc=TEXT("n");
staticHANDLEhFile;
HRESULThr;
DWORDdwWritten;
DWORDhighest_perno;
intdwRead;
charnewBuf[20];
charbuf[10];
char*header1;
char*next_token;
char*header2;
char*header3;
char*header4;
interror;
//     char mystring[10];
BOOL*result=FALSE;
intfirstexcess=FALSE;
floatmanhours;
intinterval=1;
intstored_interval;
intdelay;
boolexcess=TRUE;

switch(message)
{
caseWM_INITDIALOG:
{
ReductionLimit=100;
//                  WriteText(100,100,L"PROCESS DÄTA")
SetDlgItemInt(hDlg,IDC_EDIT2,HangarCap,NULL);
SetDlgItemInt(hDlg,IDC_EDIT13,ReductionLimit,NULL);
}
break;

caseWM_COMMAND:
{
switch(LOWORD(wParam))
{
caseIDC_BUTTON1://setup initial distribution
{
no_acts=0;
for(taskno=1;taskno<no_of_tasks-1;taskno++)
{
if(Task[taskno].Threshold!=0)
{
perno=Task[taskno].Threshold;
}
else
{
perno=1;
```

```
}

for(perno;perno<no_of_periods;perno++)
{
rest=((perno-Task[taskno].Threshold)%Task[taskno].Interval);
if(rest==0)
{
ManHours[perno][taskno]=Task[taskno].ManHours;
}
else
ManHours[perno][taskno]=0;
}
}

// Geneate total hours per period
for(perno=1;perno<no_of_periods;perno++)
{
TotalHours[perno]=0;

for(taskno=1;taskno<(no_of_tasks-1);taskno++)
{
TotalHours[perno]+=ManHours[perno][taskno];
}

}

// calculate Average level
Total=0;
for(perno=1;perno<no_of_periods;perno++)
{
Total+=TotalHours[perno];
Average=(Total)/no_of_periods;
}

// print average level
cbDest=arraysize*sizeof(TCHAR);
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,Average);
SetDlgItemText(hDlg,IDC_EDIT1,(LPCTSTR)pszDest);

// search highest distribution level
highest_perno=1;
for(perno=1;perno<no_of_periods;perno++)
{
if(TotalHours[perno]>TotalHours[highest_perno])
highest_perno=perno;
}
cbDest=arraysize*sizeof(TCHAR);
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,TotalHours[highest_perno]);
SetDlgItemText(hDlg,IDC_EDIT7,(LPCTSTR)pszDest);
SetDlgItemInt(hDlg,IDC_EDIT8,highest_perno,FALSE);
SetDlgItemInt(hDlg,IDC_EDIT10,perno,FALSE);


// calculate first excess
perno=0;
while(firstexcess==FALSE)
{
perno++;
if(TotalHours[perno]>HangarCap)
{
firstexcess=TRUE;
}
if(perno==no_of_periods)break;
}
if(perno==no_of_periods)
{
EndDialog(hDlg,10);
}

cbDest=arraysize*sizeof(TCHAR);
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,TotalHours[perno]);
SetDlgItemText(hDlg,IDC_EDIT9,(LPCTSTR)pszDest);
SetDlgItemInt(hDlg,IDC_EDIT8,perno,FALSE);
// search highest interval
```

```
//SelectHighestInterval(perno);
taskno=FindTaskNo(perno);
SetDlgItemInt(hDlg,IDC_EDIT5,taskno,NULL);
SetDlgItemInt(hDlg,IDC_EDIT6,Task[taskno].Interval,NULL);
CurrentPeriod=perno;
CurrentTask=taskno;


// calculate average excess
AverageExcess=0;
for(perno=1;perno<no_of_periods;perno++)
{
if(TotalHours[perno]>HangarCap)AverageExcess+=(TotalHours[perno]-HangarCap);
}
AverageExcess=AverageExcess/no_of_periods;
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,AverageExcess);
SetDlgItemText(hDlg,IDC_EDIT4,(LPCTSTR)pszDest);
}
break;

caseIDC_BUTTON2:
{
ReductionLimit=GetDlgItemInt(hDlg,IDC_EDIT13,result,FALSE);
}
break;

caseIDC_BUTTON3:
{
HangarCap=GetDlgItemInt(hDlg,IDC_EDIT2,result,FALSE);
}
break;


caseIDC_BUTTON4://Save initial distribution
{
DeleteFile(L"C:\\Aircraft maintenance\\InitialDistribution.csv");

hFile=CreateFileW(L"C:\\Aircraft
maintenance\\InitialDistribution.csv",GENERIC_WRITE,0,NULL,OPEN_ALWAYS,FILE_ATTRIBUTE_NORMAL,NULL);
// print header
sprintf_s(szBuf,"\r\n %s  \r\n","Initial Distribution\r\n");
SetFilePointer(hFile,0,NULL,FILE_BEGIN);
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// 1 kolom inspringen
sprintf_s(szBuf,",");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// getal 1 t/m 140 afdrukken
for(perno=1;perno<=no_of_periods;perno++)
{
sprintf_s(szBuf,"   %4d,",perno);
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
}
// nieuwe regel
sprintf_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// printf lp buffer
for(taskno=1;taskno<no_of_tasks;taskno++)
{
sprintf_s(szBuf,"%s %4d, ","Task",taskno);
for(perno=1;perno<no_of_periods;perno++)
{
sprintf_s(newBuf," %.2f,",ManHours[perno][taskno]);
strcat_s(szBuf,newBuf);
}
strcat_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
}
// extra regel toevoegen
sprintf_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// print totalen
sprintf_s(szBuf,"Totalen,");
for(perno=1;perno<=no_of_periods;perno++)
{
```

```
sprintf_s(newBuf,"%.2f,",TotalHours[perno]);
strcat_s(szBuf,newBuf);
}
strcat_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
CloseHandle(hFile);
}
break;


caseIDC_BUTTON6://generate optimised distribution
{
no_acts+=1;
SetDlgItemInt(hDlg,IDC_EDIT12,no_acts,FALSE);
// search the colom with the smalest value
if(SearchPossibleFit(CurrentPeriod,CurrentTask))
{
Task[CurrentTask].Threshold=FoundPerno;
}
else
{
MessageBox(hDlg,L"End of searching",L"No possibel fit",IDCANCEL);
}

for(taskno=1;taskno<no_of_tasks-1;taskno++)
{
if(Task[taskno].Threshold!=0)
{
perno=Task[taskno].Threshold;
}
else
{
perno=1;
}

//for (perno = 1; perno < no_of_periods; perno++)
for(perno=Task[taskno].Threshold;perno<no_of_periods;perno++)
{
if(perno==Task[taskno].Threshold)
{
ManHours[perno][taskno]=Task[taskno].ManHours;
}
else
{
rest=((perno-Task[taskno].Threshold)%Task[taskno].Interval);
if(rest==0)
{
ManHours[perno][taskno]=Task[taskno].ManHours;
}
elseManHours[perno][taskno]=0;
}
}
}

// Geneate total hours per period
for(perno=1;perno<no_of_periods;perno++)
{
TotalHours[perno]=0;
for(taskno=1;taskno<(no_of_tasks-1);taskno++)
{
TotalHours[perno]+=ManHours[perno][taskno];
}
}

// calculate Average level
Total=0;
for(perno=1;perno<no_of_periods;perno++)
{
Total+=TotalHours[perno];
Average=(Total)/no_of_periods;
}

// print average level
cbDest=arraysize*sizeof(TCHAR);
```

```
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,Average);
SetDlgItemText(hDlg,IDC_EDIT1,(LPCTSTR)pszDest);

// search highest distribution level
highest_perno=1;
for(perno=1;perno<no_of_periods;perno++)
{
if(TotalHours[perno]>TotalHours[highest_perno])
highest_perno=perno;
}
cbDest=arraysize*sizeof(TCHAR);
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,TotalHours[highest_perno]);
SetDlgItemText(hDlg,IDC_EDIT7,(LPCTSTR)pszDest);
SetDlgItemInt(hDlg,IDC_EDIT8,highest_perno,FALSE);

// calculate excess
perno=0;
for(perno=0;perno<no_of_periods;perno++)
{
if(TotalHours[perno]>HangarCap)
{
excess=TRUE;
break;
}
}
if(excess==TRUE)
{
cbDest=arraysize*sizeof(TCHAR);
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,TotalHours[perno]);
SetDlgItemText(hDlg,IDC_EDIT9,(LPCTSTR)pszDest);
SetDlgItemInt(hDlg,IDC_EDIT8,perno,FALSE);
// search highest interval
//SelectHighestInterval(perno);
taskno=FindTaskNo(perno);
SetDlgItemInt(hDlg,IDC_EDIT5,taskno,NULL);
SetDlgItemInt(hDlg,IDC_EDIT6,Task[taskno].Interval,NULL);
CurrentPeriod=perno;
CurrentTask=taskno;
// calculate average excess
AverageExcess=0;
for(perno=1;perno<no_of_periods;perno++)
{
if(TotalHours[perno]>HangarCap)AverageExcess+=(TotalHours[perno]-HangarCap);
}
AverageExcess=AverageExcess/no_of_periods;
hr=StringCbPrintf((STRSAFE_LPWSTR)pszDest,cbDest,pszFormat,pszText,AverageExcess);
SetDlgItemText(hDlg,IDC_EDIT4,(LPCTSTR)pszDest);
}
else
{
SaveOptimisedDistribution(hDlg);
MessageBox(hDlg,L"End of optimisation ",L"Generation ended",IDOK);
excess=FALSE;
}
CalculateTotalManHours();
}
break;

caseIDC_BUTTON5://Save optimised distribution
{
SaveOptimisedDistribution(hDlg);
}
break;

caseIDOK:
{
EndDialog(hDlg,IDOK);
return(INT_PTR)TRUE;
}


caseIDCANCEL:
{
EndDialog(hDlg,IDCANCEL);
```

```
return(INT_PTR)TRUE;
}
}
}
}
return(INT_PTR)FALSE;
}


INT_PTRCALLBACKOptimisedDistribution(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
TCHARszBuffer[40];
DWORDdwArgs[3];


switch(message)
{
caseWM_INITDIALOG:
{
HDChDC=GetDC(hDlg);
LPCWSTRszEdit1=L"Interval Values";
SetDlgItemText(hDlg,IDC_EDIT1,szEdit1);
wsprintf(szBuffer,(LPCWSTR)"De text regel 1 is  %s \n",L"text output\n",dwArgs);
TextOut(hDC,10,20,szBuffer,lstrlen(szBuffer));

}
break;

caseWM_COMMAND:
{
if(LOWORD(wParam)==IDOK||LOWORD(wParam)==IDCANCEL)
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}
}
break;
}
return(INT_PTR)FALSE;

}


intSaveOptimisedDistribution(HWNDhDlg)
{
staticHANDLEhFile;
DWORDdwWritten;
DWORDhighest_perno;
intdwRead;
charnewBuf[20];

DeleteFile(L"C:\\Aircraft maintenance\\OptimisedDistribution.csv");

hFile=CreateFileW(L"C:\\Aircraft
maintenance\\OptimisedDistribution.csv",GENERIC_WRITE,0,NULL,OPEN_ALWAYS,FILE_ATTRIBUTE_NORMAL,N
ULL);
// print header
sprintf_s(szBuf,"\r\n %s  \r\n","Optimised Distribution\r\n");

SetFilePointer(hFile,0,NULL,FILE_BEGIN);

WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// 1 kolom inspringen
sprintf_s(szBuf,",");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// getal 1 t/m 140 afdrukken
for(perno=1;perno<=no_of_periods;perno++)
{
sprintf_s(szBuf,"%4d ,",perno);
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
}
// nieuwe regel
sprintf_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
```

```
// printf lp buffer
for(taskno=1;taskno<no_of_tasks;taskno++)
{
sprintf_s(szBuf,"%s %4d, ","Task",taskno);
for(perno=1;perno<no_of_periods;perno++)
{
sprintf_s(newBuf," %.2f,",ManHours[perno][taskno]);
strcat_s(szBuf,newBuf);
}
strcat_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
}
// extra regel toevoegen
sprintf_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
// print totalen
sprintf_s(szBuf,"Totalen,");
for(perno=1;perno<=no_of_periods;perno++)
{
sprintf_s(newBuf,"%.2f ,",TotalHours[perno]);
strcat_s(szBuf,newBuf);
}
strcat_s(szBuf,"\r\n");
WriteFile(hFile,szBuf,strlen(szBuf),&dwWritten,NULL);
CloseHandle(hFile);


return0;
}

INT_PTRCALLBACKSetting(HWNDhDlg,UINTmessage,WPARAMwParam,LPARAMlParam)
{
TCHARszBuffer[40];
DWORDdwArgs[3];
intNo_tasks=1500;
intSceduledHorizon=140;
intHangarSlotCapacity=3;

switch(message)
{


caseWM_INITDIALOG:
{
// Work directory
SetDlgItemText(hDlg,IDC_EDIT2,L"C:\\TEMP");
// HangarSlotCapacity
SetDlgItemInt(hDlg,IDC_EDIT3,HangarSlotCapacity,NULL);
// Interval reduction limit
SetDlgItemText(hDlg,IDC_EDIT4,L"0,7");
// Sceduled Horizon
SetDlgItemInt(hDlg,IDC_EDIT5,SceduledHorizon,NULL);
// Number of tasks
SetDlgItemInt(hDlg,IDC_EDIT6,No_tasks,NULL);
}

break;

caseWM_COMMAND:
{

if(LOWORD(wParam)==IDOK||LOWORD(wParam)==IDCANCEL)
{
EndDialog(hDlg,IDCANCEL);
return(INT_PTR)TRUE;
}
}
break;
}
return(INT_PTR)FALSE;

}
```

```
intFindTaskNo(intcurperno)
{

intresult=0;
intnewresult;
intinterval;

for(taskno=1;taskno<=no_of_tasks;taskno++)
{
// valid task
if(ManHours[curperno][taskno]>0)
{
newresult=Task[taskno].Interval;
if(newresult>result)
{
result=newresult;
task=taskno;

}
}
}
returntask;
}


intSearchPossibleFit(intCurrentPeriod,inttaskno)
{
// search lowest level of ManHours in range

ints1=0;
ints2;
ints3;

intCurrentInterval;
intLowPerno;
intHighPerno;

// calculate search window
HighPerno=CurrentPeriod-1;
LowPerno=CurrentPeriod-Task[taskno].Interval+1;
Distance=HighPerno-LowPerno;
rest=(Distance*(100-ReductionLimit))%100;
Distance=((Distance*(100-ReductionLimit))/100);
if(rest>0.5)Distance+=1;

LowPerno=HighPerno-Distance;
if(LowPerno<0)LowPerno=1;
// TotalHours[FoundPerno] = 0;
FoundPerno=LowPerno;
for(perno=LowPerno;perno<=HighPerno;perno++)-->hiermoetvolgensmijn"=<"staantoch?
{
if(TotalHours[perno]<TotalHours[FoundPerno])-->dezemoetalleen"<"zijnvolgensmij(?)
{
FoundPerno=perno;
}
}
if(TotalHours[FoundPerno]+ManHours[CurrentPeriod][taskno]<HangarCap)returnTRUE;
else
{
returnFALSE;
}

}

doubleCalculateTotalManHours()
{
doubletotal=0;
for(i=1;i<=no_of_periods;i++)
{
total+=TotalHours[perno];
}
returntotal;
}
```

```
/*
LRESULT CALLBACK ReadDlgProc(HWND hDlg, UINT Message, WPARAM wParam, LPARAM lParam)
{
	static char  szTemp[255];
	switch (Message)
	{
		case WM_INITDIALOG: {
			DlgDirList(hDlg, L"*.*", IDC_LIST, IDC_DIRECTORY, DDL_DIRECTORY | DDL_DRIVES);
		}
		break;

		case WM_COMMAND:
		{
			switch (LOWORD(wParam))
			{
				case IDC_LIST: {
					if (HIWORD(wParam) == LBN_DBLCLK)
					{
						if ( DlgDirSelectEx( hDlg, szTemp, sizeof( szTemp ), idc-)
					}

				}
			}

		}

	}

}*/
```