

## A bin packing approach to solve the aircraft maintenance task allocation problem

Witteman, Max; Deng, Qichen; Santos, Bruno F.

**DOI**

[10.1016/j.ejor.2021.01.027](https://doi.org/10.1016/j.ejor.2021.01.027)

**Publication date**

2021

**Document Version**

Final published version

**Published in**

European Journal of Operational Research

**Citation (APA)**

Witteman, M., Deng, Q., & Santos, B. F. (2021). A bin packing approach to solve the aircraft maintenance task allocation problem. *European Journal of Operational Research*, 294(1), 365-376.  
<https://doi.org/10.1016/j.ejor.2021.01.027>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# European Journal of Operational Research

journal homepage: [www.elsevier.com/locate/ejor](http://www.elsevier.com/locate/ejor)

Innovative Applications of O.R.

## A bin packing approach to solve the aircraft maintenance task allocation problem

Max Witteman, Qichen Deng\*, Bruno F. Santos

Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands



### ARTICLE INFO

#### Article history:

Received 17 June 2020

Accepted 16 January 2021

Available online 22 January 2021

#### Keywords:

Scheduling

Bin packing problem

Worst-fit decreasing

Task allocation

Aircraft maintenance

### ABSTRACT

This paper addresses the scheduling of aircraft maintenance tasks that must be carried out in multiple maintenance checks to keep a fleet of aircraft airworthy. The allocation of maintenance tasks to maintenance opportunities, also known as the task allocation problem (TAP), is a complex combinatorial problem that needs to be solved daily by maintenance operators. We propose a novel approach capable of efficiently solving the multi-year task allocation problem for a fleet of aircraft in a few minutes. We formulate this problem as a time-constrained variable-sized bin packing problem (TC-VS-BPP), extending the well-known variable-sized bin packing problem (VS-BPP) by adding deadlines, intervals, and arrivals for the repetition of tasks. In particular, we divide the planning horizon into variable size bins to which multidimensional tasks are allocated, subject to available labor power and task deadlines. To solve this problem, we propose a constructive heuristic based on the worst-fit decreasing (WFD) algorithm for TC-VS-BPP. The heuristic is tested and validated using the maintenance data of 45 aircraft from a European airline. Compared with the solution obtained with an approach using an exact method, the proposed heuristic is more than 30% faster for all the test cases discussed with the airline. Most of the cases have optimality gaps below 3%. Even for the extreme case, the optimality gap is still smaller than 5%.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

### 1. Introduction

Modern airliners have thousands of parts, systems, and components that need to be recurrently maintained after undergoing certain flight hours (FH), flight cycles (FC), calendar days (DY), or months (MO). The FH, FC, DY, and MO are known as usage parameters, and their maximums allowed in operation are defined as inspection intervals. The optimal allocation of the maintenance tasks to the best maintenance opportunities is a challenging problem solved daily by maintenance planners. The common approach followed by these planners is to group tasks into maintenance checks (e.g., A-, B-<sup>1</sup>, C- and D-check) to ensure a consistent maintenance program in which all tasks are performed before their associated due dates. A typical A-check includes inspection of the interior or exterior of the airplane with selected areas opened, e.g., checking and servicing the oil, filter replacement, and lubrication (Ackert, 2010). C-check requires thorough inspections of individual systems

and components for serviceability and function. D-check<sup>2</sup> uncovers the airframe, supporting structure and wings for inspection of most structurally significant items.

To determine the optimal start date of the tasks, it is common in practice to adopt a sequential process: first, schedule the A-, C- and D-checks and then allocate maintenance tasks to each check. Although some tasks can quickly be packaged into these letter checks, a large number of other tasks (more than 70% for an Airbus A320 aircraft) are dephased from the intervals of these checks. It means that they either have to be allocated to a more frequent letter check or manually allocated by maintenance operators to different maintenance events based on the suitability of the task to that check and the urgency of performing the task in due time. In practice, both approaches are conducted according to the experience of maintenance planners, leading to inefficiencies.

The task allocation problem (TAP) in aircraft maintenance refers to the process of optimally allocating tasks in predefined maintenance checks. It determines the optimal start dates of aircraft maintenance tasks so that all preventive tasks are performed as close to their due dates as possible. TAP is complicated because of

\* Corresponding author.

E-mail address: [q.deng@tudelft.nl](mailto:q.deng@tudelft.nl) (Q. Deng).<sup>1</sup> B-checks are rarely mentioned in practice. The tasks that could be included in B-checks are commonly incorporated into successive A-checks.<sup>2</sup> Many airlines merge D-check into C-check and label it as a heavy C-check or structural check.

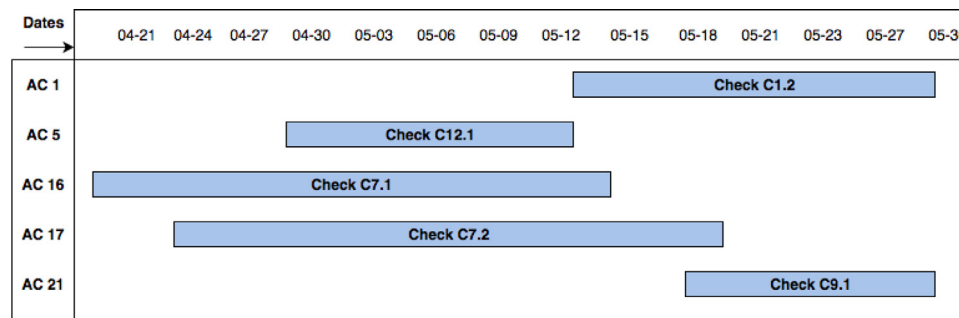


Fig. 1. Snapshot of maintenance overlap situation between aircraft.

its combinatorial nature, and it has to be solved for the entire fleet at the same time. In real-life applications, multiple aircraft checks can be scheduled in parallel, and tasks allocated to these checks will share the maintenance resources. For example, Fig. 1 illustrates a case for five C-checks overlapping in time. Maintenance resources include material, equipment, and a set of labor hours from different skills. Furthermore, the allocation process is intricate also because the maintenance tasks involved in these checks are usually associated with different intervals and elapsed time.

In this paper, we propose a novel approach to efficiently address the TAP, which can solve the problem very fast without compromising the quality of the solution. Maintenance plans are frequently being affected by flight schedules disruptions or the need for unscheduled maintenance tasks, and they constantly need to be revised or even re-planned (Steiner, 2006). Inspired by the bin packing problem (BPP), we consider pre-scheduled maintenance checks to be bins of different (time) dimensions and sharing a multi-dimensional capacity, referring to the multiple types of labor skills involved in the execution of the tasks. The items are the tasks that need to be packed in the bins, and they also subject to time constraints that limit the bin options. We formulate the problem as an extension of the variable size bin packing problem (VS-BPP) (Friesen & Langston, 1986) in which items are repeated within time intervals, and bins have a variable time dimension. This extension of the VS-BPP is named time-constrained VS-BPP (TC-VS-BPP). We present a constructive algorithm to solve this problem efficiently. We test this heuristic in a case study using data from a major European airline and compare the results with the ones obtained using an exact method. The main contribution of this research can be summarized in the following:

- This work is the first to formulate the TAP as a bin packing problem and solve it with an efficient constructive algorithm.
- For the first time, to the best knowledge of authors, the classic VS-BPP formulation is extended to consider time intervals for the allocation of repeated items and variable time dimensions for the bins.
- We adapt the *worst-fit decreasing* algorithm for the classic BPP to efficiently solve the TC-VS-BPP. The resulting constructive algorithm is validated with a real case study and benchmarked against the solution obtained using a commercial linear programming solver.

The outline of this paper is as follows: Section 2 gives an overview of the relevant literature on maintenance related TAPs and bin packing problems (BPPs). The formulation of the TC-VS-BPP for aircraft maintenance is described in Section 3. Section 4 presents a task allocation framework and an associated heuristic algorithm. Section 5 shows a case study from a European airline and the algorithm performance analysis. The last section summarizes the research with concluding remarks and gives an outlook on future work.

## 2. Related work

In this section, we briefly discuss previous works. We divide this literature overview into two subsections. The first subsection reviews the research works dealing with the TAP for aircraft maintenance, with different perspectives and methodologies. The second subsection discusses the literature on the bin packing problem.

### 2.1. Maintenance task allocation

In one of the initial studies on TAP of aircraft maintenance, Van Buskirk et al. (2002) combined the maintenance task allocation with aircraft operation to one single problem. The authors presented a two-stage system that supports maintenance chiefs in planning both aircraft operations and maintenance activities. The first stage assigns the planes to flight operations using a custom-built, multi-level greedy search algorithm. The second stage schedules all maintenance activities according to a constraint satisfaction problem. The authors tested the system with 17 jets, and results indicate that the system can schedule 3750 maintenance activities for a 3-month planning horizon within 20 minutes. The authors also state that the goal was to plan the activities given various constraints: calendar-based actions have to be done within a specific time window; usage-based actions have to be done when the usage clock on a part or subsystem reaches a particular value; personnel has to be available to do the job (mechanics can only do jobs that they are qualified for), and maintenance jobs have to be inspected by a quality/safety inspector and so forth. However, this initial work does not optimize the maintenance schedule given that support for the flight operation was the top priority.

In contrast to Van Buskirk et al. (2002), Steiner (2006) presented a heuristic for aircraft maintenance planning, aiming at minimizing the overall number of maintenance actions and uniformly distributing the capacity and flying hours over a given time horizon. The main idea was to split the whole process into subprocesses that could be handled computationally fast at the same time. Determining the optimal position of the maintenance actions was the least difficult one, whereas the balancing step was the most challenging one. Even under various settings and constraints, the proposed algorithms have shown to work reliably, fast, and with good optimization results. According to the case study for a 5-year time horizon, the number of tasks scheduled per fleet was around 50–500. The time to compute a new maintenance plan was about 15 minutes.

In practice, many large airlines adopt the top-down approach by appropriately grouping maintenance tasks into large packages and fitting them into letter checks. Muchiri and Smit (2009) followed this approach and developed a maintenance item allocation model (MIAM) to cluster aircraft maintenance tasks into packages. The MIAM first simulates the aircraft utilization, calculates when

a maintenance item turns due, and then fits each maintenance item into a package. The authors use the concept of *de-escalation* to assess the quality of their MIAM, which can be interpreted as the loss associated with maintenance items being performed more frequently than necessary. The authors proposed a translation of the de-escalation into additional labor costs essential in the long-term to perform extra maintenance activities. According to a case study of a Boeing 737-NG aircraft, the authors claimed that introducing an initial de-escalation, i.e., performing the first base maintenance before its due date, leads to a lower de-escalation labor cost over time. The authors obtained the best result for an initial de-escalation of 30 days, leading to a savings of 248 labor hours (or €13,902) for a single aircraft. The importance of [Muchiri and Smit \(2009\)](#) is that it provides an alternative of assessing maintenance costs using the causal relationship between expense and labor hours.

Maintenance operation costs, in more detail, include the costs of maintenance tools, labor hours, and aircraft spare parts. Each maintenance task associates a cost. Since there are 1000–3000 tasks involved in aircraft maintenance, and many tasks can be performed in parallel, one of the biggest challenges is to execute the right maintenance task at the right time. Assigning priorities to maintenance tasks, such as the rule of “the most urgent task first”, can significantly reduce problem complexity. [Hölzel, Schröder, Schilling, and Gollnick \(2012\)](#) considered this aspect and presented an optimization method for aircraft maintenance task allocation integrating simulations of aircraft life-cycles. In a real-life application, the authors obtained the best results when sorting the tasks by cost (labor hours) in descending order. In this way, the optimizer allocated the most expensive tasks to maintenance opportunities closer to the end of the lives of the components.

From an efficiency perspective, finding the best maintenance opportunities and allocating maintenance tasks one after another is exceptionally time-consuming. Since each task has some basic properties to indicate similarities, such as ATA code, maintenance interval, zone, and check type, it is more convenient to combine several similar tasks into a work package and reduce the total number of tasks. [Li, Zuo, Lei, Liang, and Lu \(2015\)](#) followed this idea and gave different weights on properties to indicate task similarities. Based on engineering experience, weighting factors 0.05, 0.8, 0.05, and 0.1 are assigned to ATA code, maintenance interval, zone, and check, respectively. The authors solved the TAP of an airline using a fuzzy C-means clustering algorithm. Although convergence and improvements were both achieved, the authors stated there are still some pitfalls that need to be investigated, such as the influence of model parameters on solution quality and convergence rate.

In general, the literature on TAP, especially for a long term planning horizon, is very limited. Some of them address TAP on aircraft level ([Li et al., 2015](#); [Muchiri & Smit, 2009](#)), while others on fleet level ([Hölzel et al., 2012](#); [Steiner, 2006](#); [Van Buskirk et al., 2002](#)). Even in the research work of TAP in fleet level, the authors tackled task allocation of each aircraft independently, and eventually looped over the entire fleet. Furthermore, none of those related works has assessed the optimality of the proposed models or heuristics. There is no comparison of how close the solution from proposed models or heuristics to local/global optimum.

## 2.2. The bin packing problem

Despite the various task allocation models and methods discussed before, the TAP of aircraft maintenance is very analogous with bin packing problem (BPP), where for TAP, the maintenance opportunities are equivalent to bins, and maintenance tasks are considered as items. The keys to solving BPP are the bin selection and item allocation. For bin selection strategies, [Johnson](#)

(1974) lists four fundamental and widely used algorithms, *next-fit* (NF), *first-fit* (FF), *best-fit* (BF), and *worst-fit* (WF):

- *Next-Fit* (NF): If the item fits in the same bin as the previous item, put it there. Otherwise, open a new bin and put it in there.
- *First-Fit* (FF): Put each item as you come to it into the oldest (earliest opened) bin into which it fits. Only open a new bin if an item does not fit into any previous bin.
- *Best-Fit* (BF): Put items in bins in a way that it maximizes the utilization of the bins that already have been opened.
- *Worst-Fit* (WF): Put each item into the emptiest bin among those with something in them. Only start a new bin if the item does not fit into any bin that has already been started. If there are two or more bins already started which are tied for emptiest, use the bin opened earliest from among those tied.

If all items are the same size, there is no difference in the four algorithms. Since items are very likely to have different sizes, the allocation of items to bins becomes intricate and time-consuming. And this may involve shifting bin contents continuously until the item list is empty. Thus, some researchers proposed prioritizing the items before putting them in bins. [Johnson \(1972\)](#) has suggested some alternatives to the FF and BF. The author states that if the items are sorted in descending order (i.e., the largest item goes first), the worst-case behavior of bin packing problems can be significantly improved. Therefore, it is now a common step to prioritize items before allocation when solving the BPP. The resulting algorithms are the equivalent *first-fit decreasing* (FFD) and *best-fit decreasing* (BFD) algorithms. Similarly, there are also *next-fit decreasing* (NFD) and *worst fit decreasing* (WFD) algorithms.

In practice, not only items can have various sizes, but also bins can have different capacities, and this leads to variable-sized BPP (VS-BPP). VS-BPP is an extension of the classic BPP, in which bins no longer have the same size, and the cost of a bin is proportional to its size ([Friesen & Langston, 1986](#)). VS-BPP is more challenging since putting items in bins affects the selections of opening new bins later on and item allocations and vice versa. VS-BPP is NP-hard ([Correia, Gouveia, & da Gama, 2008](#)). Researchers tend to solve it using approximation algorithms instead of finding the exact global optimum. [Friesen and Langston \(1986\)](#) listed some algorithms for VS-BPP, such as *next-fit using largest bins only* (NFL), and *first fit decreasing using largest bins and at the end repack to smallest possible bins* (FFDLR). The authors also showed that allowing repacking small bins and shifting bin contents improves algorithm efficiency. And the FFDLR has better worst-case performance than NFL because there is no repacking in the NFL. [Friesen and Langston \(1986\)](#) further developed a new algorithm *first fit decreasing using the largest bins, but shifting as necessary* (FFDLS) to dynamically shifting bin contents during the construction of packing. Case studies prove that with dynamically shifting bin contents, FFDLS outperforms both NFL and FFDLR in the worst cases.

While [Friesen and Langston \(1986\)](#) is one of the first works in VS-BPP, research in this topic continues and flourishes in many other studies ([Correia et al., 2008](#); [Csirik, 1989](#); [Haouari & Serairi, 2009](#); [Kang & Park, 2003](#)). The main focus of these studies is on the development of algorithms. There is no deadline for putting each item in bins in all of those studies. VS-BPP in scheduling, especially maintenance planning, is very distinct from other fields due to time constraints. For example, each maintenance task associates a due date. In VS-BPP, it is equivalent to imposing a deadline for each item (each item has to be put in a bin before a specific time). Besides, and some tasks have to be performed repeatedly. Once the task is executed, we can anticipate the next arrival time of the same task.

The arrival times of items and item allocation deadlines make the maintenance scheduling related VS-BPP unique and more



complex. Some researchers categorize the VS-BPP, in which each item has an associated arrival time and allocation deadline, as time-constrained VS-BPP (TC-VS-BPP). In one of the very few available references, Fazi, van Woensel, and Fransoo (2012) presents a Markov Chain Monte Carlo (MCMC) heuristic to address the TC-VS-BPP in a working paper. The main difference between TC-VS-BPP and VS-BPP is that in TC-VS-BPP, the arrival times of the items have specific patterns, e.g., a probability distribution in Fazi et al. (2012), and each item has to be allocated before a particular deadline. The MCMC heuristic is a combination of local search and Monte Carlo sampling. It starts with a simple greedy approach to obtain an initial feasible solution. In this step, the authors create two non-ordered lists for bins and items, respectively, and apply the FF algorithm to put items in bins. After that, the authors use MCMC to improve the initial feasible solutions iteratively. One interesting finding from Fazi et al. (2012) is that when time constraints are introduced, smaller and faster bins are preferred to meet the deadlines. But in the classical VS-BPP, items are often concentrated in few high capacitated bins. Two main features in TC-VS-BPP, arrival times of the items and deadlines of the items (Fazi et al., 2012), are also common in maintenance scheduling. Since most of the maintenance tasks have deadlines and follow periodic patterns, once a task is performed, we can already anticipate its next execution.

The review of the literature on TAP, BPP, VS-BPP, TC-VS-BPP, and corresponding solution techniques indicates that an aircraft maintenance TAP is similar to TC-VS-BPP in the model formulation in terms of maintenance capacity constraints, availability of each maintenance hangar, the different costs in task execution, workloads of performing tasks, task execution intervals, and deadlines of the maintenance tasks, meaning that the solution strategies, such as NFD/FFD/BFD/WFD, to BPP/VS-BPP/TC-VS-BPP, can be used to address TAP. Based on the findings from the literature, we propose a constructive heuristic based on the WFD algorithm to solve the long-term aircraft maintenance TAP. The main reason is that more than 55% of the tasks belong to heavy maintenance (C/D-check), and we want to let the available workforce address as many heavy maintenance tasks (large task blocks) as possible in aircraft C-checks. In our problem, we are not trying to reduce the number of bins being used – these were already pre-defined in the maintenance schedule and as a consequence of the overlapping of multiple checks in time. Furthermore, we want to spread the tasks over the multiple bins in such a way that we avoid resource limitations at any point. So the idea is always to allocate the item to the bin with the minimum load (or higher resources available). Since our work focuses on practical application, instead of worst-case performance analysis, we compare the results from the heuristic to a solution from exact methods.

### 3. Problem formulation

In this section we define the TC-VS-BPP for aircraft maintenance task allocation. We start the section with specifying the problem and its scope (Section 3.1), followed by a description of the assumptions followed (Section 3.2). In Section 3.3 we introduce some model considerations, including the concept of *time segment* and the generation of the *task items* in our TC-VS-BPP. Finally, in Section 3.4 we present the optimization model formulation.

#### 3.1. Problem definition and scope

##### 3.1.1. Task Classes

In the aircraft maintenance context, tasks can represent regular maintenance jobs needed for the continuous airworthiness of the aircraft or repairing works that need to be performed to correct

malfunctions or damage. Accordingly, the tasks can be divided into two main classes (Ackert, 2018):

- **Routine Tasks:** these are the regular tasks outlined in a Maintenance Planning Document (MPD) provided by the aircraft manufacture or defined by the airline in their Operator Approved Maintenance Program (OAMP). These tasks have to be scheduled within certain fixed intervals, specified in terms of usage parameters such as FH, FC, and calendar days. A routine task has to be performed before one of the usage parameters reaches the specified interval.
- **Non-Routine Tasks:** these are non-scheduled tasks that can result from defaults or damage identified when executing a routine task, pilot reports, or abnormal events such as hard landings or ground damages. They can also represent abnormal maintenance interventions suggested by, e.g., the aircraft manufacturer (service bulletins) or the regulatory body (airworthiness directives). When generated, these tasks are also associated with a time window for their execution. And this time window can vary from having to perform the task before the next flight to a couple of weeks after they were generated.

##### 3.1.2. Task intervals

The aircraft maintenance tasks, regardless of being routine or non-routine, have to be allocated to a maintenance event. These events include line maintenance inspections (i.e., performed at the ramp or remote stands during the turn-around time of the aircraft) and hangar inspections. In this article, we only consider the latter and ignore the small tasks usually performed during line maintenance inspections.

##### Workforce

The available workforce constrains the task allocation to maintenance check; each maintenance task is associated with the workforce requirements to perform the task. The maintenance workforce is divided per skill types (e.g., engines and flight control systems, avionics, aircraft metallic structure, and painting technicians). It is limited per day or shift, according to the daily workforce schedule. In this study, the availability of the workforce per skill is an input to the model. The number of hours needed per skill type is a characteristic of the task, which can only be allocated to a maintenance opportunity if there is enough workforce for all skill types involved in task execution.

##### 3.1.3. Time horizon

Given that routine tasks have to be scheduled based on intervals and that these intervals are re-started every time the tasks are performed, the TAP should consider a time horizon that is large enough to cover at least two following task executions. The reason being that, otherwise, a possible action could be to delay the first task as much as possible, disregarding the possibility of executing the tasks the next time. And this can result in a poor or unfeasible solution in the long-term. For this reason, given that some tasks having very large intervals (i.e., some are not performed every year), a multi-year planning horizon is adopted.

##### 3.1.4. Sequential approach

To plan hangar inspection tasks, we follow a sequential approach, consistent with the practice of most airlines. That is, we assume that the aircraft maintenance check scheduling (AMCS) was solved beforehand and that an optimal letter check schedule is provided. According to this schedule, each check is considered as a maintenance opportunity to perform a maintenance task. Consequently, the goal of the TAP is to allocate the maintenance tasks to the opportunities that are as close as possible to their due dates.

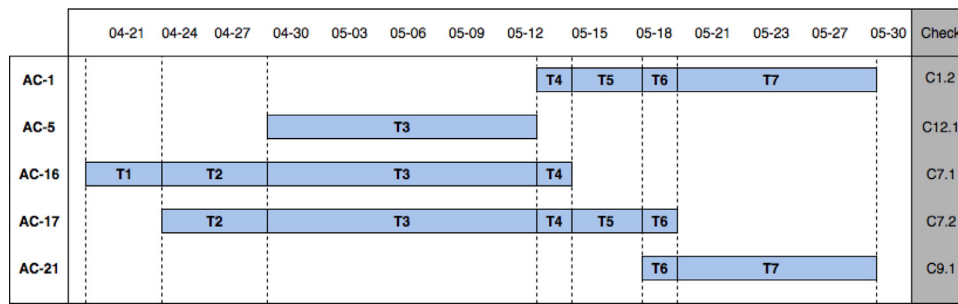


Fig. 2. Overlapped maintenance checks are divided into several time segments.

### 3.2. Assumptions

This research is subject to the following assumptions:

- A. There are sufficient aircraft spare parts and available maintenance tools and equipment, without constraining the optimal allocation of tasks.
- B. The optimal allocation of tasks is constrained by the workforce available. The optimal distribution of tasks per shift or worker is not considered in the TAP.
- C. A-check tasks can be performed in a C-check, but not the other way around.
- D. Non-routine tasks generated while executing other tasks can also be performed during the same check, and this is considered by augmenting the task duration and labor power needed according to “non-routine rates” estimated from historical data.

The first two assumptions are reasonable, considering that the TAP is a long-term problem and spare parts, maintenance tools and equipment, and labor force are planned following the maintenance schedule. Assumptions A3 and A4 are common in practice. The first, because the resources, skills, and time needed to perform most C-check tasks are not compatible with the planning of an A-check. The second, because the differing tasks from a hangar check can result in pressure to perform these tasks another day, eventually causing disruptions in operations. Therefore, airlines usually prefer to pre-allocate a time and workforce buffer in each maintenance check to execute these non-routine tasks.

### 3.3. Model considerations

#### 3.3.1. Time segments

In practice, maintenance operators are typically confronted with situations of overlapped maintenance checks, in which several aircraft undergo the same type of maintenance check at the time and therefore competing for the limited maintenance resources. Fig. 2 depicts an example of such a schedule, and five aircraft are scheduled to perform C-checks maintenance between Apr 21<sup>st</sup> and May 30<sup>th</sup>. During these overlap periods, resources have to be shared, constraining the optimal allocation of tasks.

We divide the planning horizon depicted in Fig. 2 into time segments. A time segment is created every time the overlap conditions change. In Fig. 2, the overlap of checks change on Apr 24<sup>th</sup> and 30<sup>th</sup>, May 12<sup>th</sup>, 15<sup>th</sup>, 18<sup>th</sup>, 21<sup>st</sup> and 30<sup>th</sup>. Therefore, we create seven time segments: Apr 21<sup>st</sup>–24<sup>th</sup>, Apr 24<sup>th</sup>–30<sup>th</sup>, Apr 30<sup>th</sup>–May 12<sup>th</sup>, May 12<sup>th</sup>–15<sup>th</sup>, May 15<sup>th</sup>–18<sup>th</sup>, May 18<sup>th</sup>–21<sup>st</sup> and May 21<sup>st</sup>–30<sup>th</sup>. Each time segment of an aircraft is considered to be a bin, with a given duration (in days) and constrained by the labor available on these days for each given skill type. For example, AC-1 has four bins, T4–T8; AC-5 has only one bin, T3; AC-16 has four bins, T1–T4; AC-17 has five bins, T2–T6; AC-21 has two bins, T6 and T7. It is worth mentioning that all the bins and their associated sizes are defined based on the maintenance check schedule

and kept open. Unlike the classic BPP, we do not need to open a bin when we allocate items (tasks). For the rest of the paper, when we refer to TC-VS-BPP, we also imply that all the bins are pre-determined.

#### 3.3.2. Task items and maintenance opportunities

Most routine tasks have to be scheduled more than once for the same aircraft over the time horizon considered. For example, a task that has to be performed in every A-check (about every 7–8 weeks), may have to be executed 38 times in a 5-year horizon. In our approach, we consider each occurrence of these tasks to be an item in our TC-VS-BPP. That is, a routine task that has to be executed at most  $N$  times in the planning horizon will be translated into  $N$  tasks items in our optimization model. To do so, we have to estimate the maximum number of repetitions in the planning horizon. Table 1 illustrates our approach for a given task of a specific aircraft. In this example, the maintenance task has to be performed every ten weeks, while the aircraft A-checks are performed every seven weeks. There are five maintenance events during the time horizon for the execution of the task (four aircraft A-checks and one aircraft C-check, presented in chronological order). This task can be executed from two times (only in A2 and A3) to five times (in every maintenance check), which can be translated as five task items in the task allocation. The procedure for creating task items and defining the respective maintenance opportunities can be summarized as follows:

- Step 1: The maintenance opportunities for the first execution of the task are determined, according to the state of the task at the start of the planning horizon and its inspection interval.
- Step 2: If the earliest maintenance opportunity for the previous task item is the last maintenance event in the planning horizon, we stop. Otherwise, we create a new task item (next execution).
- Step 3: For the new task item (new execution),
  - o Step 3.1: the first maintenance opportunity is the maintenance event right after the earliest maintenance opportunity from the previous task item;
  - o Step 3.2: the last maintenance opportunity is the last maintenance event, within the planning horizon, that can be considered before the end of its fixed interval.
  - o Step 3.3: all maintenance events between the first and last maintenance opportunities are considered in the set of maintenance opportunities.
- Step 4: Go back to Step 2.

For the task items which can potentially be allocated to a maintenance check after the end of the planning horizon, we create a *fictitious maintenance opportunity* (bin). The fictitious bin is needed because, eventually, not all task items have to be allocated within the planning horizon to keep the aircraft airworthy. The fictitious bin is added on the day right after the end of the planning horizon, associated with infinite resources and no costs, and it is considered as a bin for all task items that can be scheduled after the end

**Table 1**

Illustration of the maintenance opportunities for repeated items of one maintenance task with an inspection interval of 10 weeks (Task 1<sub>1</sub>–1<sub>5</sub> represent the 1<sup>st</sup>–5<sup>th</sup> execution of the same task). The value of “1” indicates that the associated maintenance check (column) is a possible maintenance opportunity for the execution (row).

	A1 week 1	A2 week 8	C1 week 12	A3 week 15	A4 week 22	Fictitious opportunity
Task 1 <sub>1</sub>	1	1	0	0	0	0
Task 1 <sub>2</sub>	0	1	1	1	0	0
Task 1 <sub>3</sub>	0	0	1	1	1	1
Task 1 <sub>4</sub>	0	0	0	1	1	1
Task 1 <sub>5</sub>	0	0	0	0	1	1

of the planning horizon. This step-wise approach, repeated to all maintenance tasks, will result in a list of task items  $N_k$  per aircraft  $k$  and the respective set of maintenance opportunities  $R_{i,k}$  associated with each task  $i$  in the list.

There are several task execution plans for the example presented in Table 1. Each plan is associated with a de-escalation cost, depending on the letter checks that the task is executed. We can choose a high-cost plan in which the task is executed in every maintenance check (i.e., five times in the planning horizon) or a low-cost plan in which the task is executed only twice, during the A1 and the A3 checks. Even for two plans with the same number of total executions of a task, the de-escalation is different. For instance, performing the task in **A2**, **C1** and **A3** results in a de-escalation cost of:

$$(10 - 8) + [10 - (12 - 8)] + [10 - (15 - 12)] = 15 \text{ weeks} \quad (1)$$

executing the task in **A2**, **A3** and **A4** results in a de-escalation cost of:

$$(10 - 8) + [10 - (15 - 8)] + [10 - (22 - 15)] = 8 \text{ weeks} \quad (2)$$

We can observe from (1) and (2) that the latter execution plan has a lower de-escalation cost, and the goal of task allocation is to find the task execution plan with the lowest cost, given the resources available and the urgency of other tasks “competing” for the same maintenance opportunities.

### 3.4. Problem formulation

#### 3.4.1. Nomenclature

• **Sets**

- $i$ : task indicator
- $K$ : set of aircraft
- $N_k$ : set of task items for aircraft  $k$  ( $k \in K$ )
- $T_k$ : set of time segments for aircraft  $k$  ( $k \in K$ )
- $R_{i,k}$ : set of time segments for task item  $i$  ( $i \in N_k$ ) of aircraft  $k$  ( $k \in K$ )
- $J$ : set of skills
- $O_{i,k}$  unit set with the task item that follows task item  $i$  ( $i \in N_k$ ) of aircraft  $k$  ( $k \in K$ )

• **Parameters**

- $c_{i,k}^t$ : cost of allocating task item  $i$  ( $i \in N_k$ ) from aircraft  $k$  ( $k \in K$ ) to maintenance opportunity belonging to time segment  $t$  ( $t \in T_k$ )
- $\overline{GR}_t^j$ : amount of available labor hours of skill type  $j$  ( $j \in J$ ) at time segment  $t$
- $GR_{i,k}^j$ : amount of labor hours of skill type  $j$  prescribed to perform task item  $i$  of aircraft  $k$
- $\sigma^{j,l}$ : “non-routine rate” indicating the amount of labor hours needed from skill type  $l$  for every labor-hour prescribed from skill type  $j$  (note:  $\sigma^{j,j} \geq 1.0 \quad \forall j \in J$ )
- $\overline{d}_{i,k}$ : maximum number of days between rescheduling task item  $i$  ( $t \in T_k$ ) for aircraft  $k$  ( $k \in K$ )

- $d^t$ : number of days from the start of the planning horizon till maintenance opportunity belonging to time segment  $t$
- $\overline{fh}_{i,k}$ : maximum number of flight-hours between rescheduling task item  $i$  for aircraft  $k$
- $fh^t$ : number of accumulated flight-hours from the start of the planning horizon till maintenance opportunity belonging to time segment  $t$
- $\overline{fc}_{i,k}$ : maximum number of flight-cycles between rescheduling task item  $i$  for aircraft  $k$
- $fc^t$ : number of accumulated flight-cycles from the start of the planning horizon till maintenance opportunity belonging to time segment  $t$
- $O\_day_i$ : total days of aircraft operations from the start of the planning horizon to the due date of performing task item  $i$ , following the task fix interval and if no resource constraints are considered
- $interval_i$ : average fix interval for task item  $i$  measured in days
- $labor\_rate_j$ : labor rate, per hour, of skill type  $j$  ( $j \in J$ )
- $other\_costs_{i,k}$ : non-labor costs associated with task item  $i$  ( $i \in N_k$ ) of aircraft  $k$  ( $k \in K$ ), such as costs of spare parts and tooling

• **Decision variables**

- $x_{i,k}^t$ : 1 if task item  $i$  is assigned to maintenance opportunity belonging to time segment  $t$  for aircraft  $k$ , and 0 otherwise

#### Mixed Integer Linear Programming (MILP) Formulation

Given a long-term aircraft A- and C-check schedule, we formulated the TAP as a 0–1 MILP model.

$$\min \sum_{k \in K} \sum_{i \in N_k} \sum_{t \in R_{i,k}} c_{i,k}^t \times x_{i,k}^t \quad (3)$$

subject to:

$$\sum_{t \in R_{i,k}} x_{i,k}^t = 1 \quad \forall i \in N_k \quad \forall k \in K \quad (4)$$

$$\sum_{k \in K} \sum_{i \in N_k} \sum_{j \in J} GR_{i,k}^j \times x_{i,k}^t \times \sigma^{j,l} \leq \overline{GR}_t^l \quad \forall t \in T_k \quad \forall l \in J \quad (5)$$

$$\sum_{m \in R_{p,k}} d^m \times x_{p,k}^m - \sum_{t \in R_{i,k}} d^t \times x_{i,k}^t \leq \overline{d}_{i,k} \quad \forall i \in N_k \quad \forall p \in O_{i,k} \quad \forall k \in K \quad (6)$$

$$\sum_{m \in R_{p,k}} fh^m \times x_{p,k}^m - \sum_{t \in R_{i,k}} fh^t \times x_{i,k}^t \leq \overline{fh}_{i,k} \quad \forall i \in N_k \quad \forall p \in O_{i,k} \quad \forall k \in K \quad (7)$$

$$\sum_{m \in R_{p,k}} fc^m \times x_{p,k}^m - \sum_{t \in R_{i,k}} fc^t \times x_{i,k}^t \leq \overline{fc}_{i,k} \quad \forall i \in N_k \quad \forall p \in O_{i,k} \quad \forall k \in K \quad (8)$$

$$x_{i,k}^t \in \{0, 1\} \quad \forall k \in K \quad \forall i \in N_k \quad \forall t \in T_k \quad (9)$$

The objective function (3) aims at minimizing the total maintenance costs, which reflect the de-escalation costs associated with scheduling the task earlier than its due date and, consequently, having to perform the task more frequently in the future. To compute these costs, we estimate the due date to allocate the task item beforehand. For example, if a maintenance task is to replace an aircraft component, based on its previous execution date and the associated maintenance interval, we simulate the utilization of the component using the average aircraft’s daily utilization. In this way, we can estimate the next due date of replacing this component and its ideal maximum utilization  $O\_day_i$ . The de-escalation costs can then be calculated by comparing how earlier the task item is allocated when compared with its desired day (Hölzel et al., 2012):

$$c_{i,k}^t = \frac{O\_day_i - d^t}{interval_i} \times \left[ \sum_{j \in J} \left( \sum_{l \in L} GR_{l,k}^j \times \sigma^{l,j} \right) \times labor\_rate_j + other\_costs_i \right] \quad (10)$$

The de-escalation costs indicated by (10) is a reference cost used as a proxy of the goal of scheduling the tasks as later as possible, or as less frequent as possible. In (10), the cost of allocating task item  $i$  of aircraft  $k$  to maintenance opportunity  $t$  is a function of the wasted interval of the task (first term), the labor hours required to perform the task (second term), the labor hours cost per labor skill (third term) and additional costs associated with maintenance task  $i$  such as the cost for materials or expensive tooling (last term). And this formulation aims at allocating tasks to the maintenance opportunity closer to its due date while giving a higher priority to labor-intensive tasks and tasks involving many labor skills or high additional costs.

Constraints (4) guarantee that each task item is allocated exactly once, either to a maintenance event or to the fictitious maintenance event after the planning horizon. Constraints (5) make sure that the available labor hours for each skill type is not exceeded in each of the maintenance time segments. The left-hand side of these constraints sums the labor hours needed to perform each task item, including the workforce needed to perform the task and, eventually, associated “non-routine” tasks. These two sets of constraints are the ones that define the classic VS-BPP. The other three set of constraints (6)–(8) are the features of TC-VS-BPP and also ones that represent the maintenance time-intervals. They imply the arrivals and deadline of tasks. Constraints (6) guarantee that a subsequent task item is scheduled within the number of days defined in the fix interval for the respective task, while constraints (7) and (8) reflect the fix interval in terms of flight-hours and flight-cycles, respectively.

#### 4. Task allocation framework

The same as BPP, TC-VS-BPP is also NP-hard (Garey & Johnson, 1990). Optimal solutions to small TC-VS-BPPs can be obtained using exact methods. Still, unfortunately, when the size of the problem grows, the running times of these exact methods become prohibitive, especially for practical implementations. For this reason, we propose a constructive heuristic to solve the TAP efficiently. The proposed approach is an iterative process based on the WFD algorithm. To the TAP for aircraft maintenance, we start by sorting the tasks from the multiple aircraft into decreasing order of priority and then allocate those tasks one after another to the suitable bin that has a lower load. In this section, we provide details on the proposed constructive heuristic, explaining the general framework, including the input data (Section 4.1), the necessary pre-computation (Section 4.2) and the algorithm itself (Section 4.3).

#### 4.1. Input data

Four sets of input data are needed to formulate and solve the TAP. The first set consists of maintenance task information present in the OAMP for the considered aircraft fleet. This information is not necessarily limited to maintenance tasks described in the MPD. It could include additional maintenance tasks as required by the airline, service bulletins, airworthiness directives, deferred defects, or modifications (Ackert, 2018). Furthermore, information about the last executed date of the routine tasks is used to calculate the first due-date of the maintenance task. The second set includes the estimated daily aircraft utilization, in DY, FH, and FC, of each aircraft for the entire time horizon. For the short term, these values could be obtained using aircraft routes or flight schedules, while in the long run, the most common approach is to use average aircraft utilization per day of the week, per month, or season. It is convenient, however, to use the same input values used to produce the maintenance check schedule. The third set of input is the available workforce per skill type, per day, for the entire time horizon. Again, detailed daily schedules could be provided for the short term, while the average workforce per day can be used for the longer term. The last set of data used is the A- and C-check schedule, defining the starting dates and duration of all checks in the planning horizon for each aircraft in the fleet.

#### 4.2. Pre-computation

A set of pre-computation steps are necessary before initiating the constructive task allocation algorithm. These steps can be divided into task items and bins related pre-computations. Starting with the task items related steps, maintenance tasks from the same aircraft that have identical intervals, in terms of FH, FC, and DY, are clustered together to reduce the number of tasks to be considered. For the resulting tasks, a set of task items are created, following the procedure explained in Section 3.3. The following step is to compute the due-dates for the first item of the maintenance tasks. And this is done by considering the initial state of each task (i.e., number of FH, FC, and DY since its previous execution), the task intervals as defined by the OEM or airline, and the simulation of the aircraft utilization over time. Some tasks, such as deferred defects or modifications, can be input already with fixed due dates instead of task intervals.

For the bin related steps, the checks schedule is used to divide the maintenance opportunities into bins, as explained in Section 3.3.1. The bins are variable in size and discrete, composed by a set of days. After that, we continue to convert the labor power obtained per day into labor power available per bin.

#### 4.3. Constructive heuristic

A constructive heuristic based on WFD is proposed for task allocation. The pseudo-code of the heuristic is presented in Algorithm 1, while the main procedures of the heuristic are explained next.

##### Sort Task List

After uploading the input data, the first procedure is to sort the task items list according to the priorities of the items included in the list. The prioritization is done according to a prioritization function  $p(i)$  that classifies each task item  $i$ . This prioritization function divides task items into three classes:

- *High Priority* – these are items from maintenance tasks that have an interval equal to the interval of the aircraft checks. The allocation process for these items is trivial since those tasks have to be allocated to all equivalent checks in the schedule.



**Algorithm 1** Task Allocation Algorithm.

```

1:  $N \leftarrow$  set of task items from all aircraft,  $N = \cup N_k$ 
2:  $\overline{GR}_t^j \leftarrow$  available labor hours from skill  $j$  in bin  $t$ 
3:  $GR_{i,k}^j \leftarrow$  amount of labor hours of skill  $j$  prescribed to perform
   task item  $i$  of aircraft  $k$ 
4:  $\sigma_{j,m} \leftarrow$  “non-routine rate” from skill  $m$  from every hour of skill  $j$ 

5: procedure SORT TASK ITEMS LIST
6:   Sort and reindex  $N$  so that  $p(i_1) \geq p(i_2) \geq \dots \geq p(i_n)$   $\triangleright$ 
   Prioritization of task items
7: end procedure

8: procedure TASK ITEMS LOOP
9:   while  $N \neq \emptyset$  do
10:     Select  $i$  from  $N$   $\triangleright$  Select the first task in the list
11:      $R_i \leftarrow R_{i,k} \cup t_0$   $\triangleright$  Add  $t_0$  as a fictitious opportunity
12:     Sort and reindex  $R_i$  so that  $\sum_{j \in J} \overline{GR}_{t_{i,1}}^j \geq \sum_{j \in J} \overline{GR}_{t_{i,2}}^j \geq \dots \geq$ 
        $\sum_{j \in J} \overline{GR}_{t_{i,n}}^j$ 
13:     procedure ALLOCATE TO BIN
14:        $n \leftarrow 0$ 
15:       while  $n < |R_i|$  do
16:          $n \leftarrow n + 1$ 
17:         if  $\overline{GR}_t^j \geq \sum_{j \in J} GR_{i,k}^j \times \sigma_{j,m} \quad \forall m \in J$  then
18:           Allocate  $i$  to  $t_{i,n}$ 
19:           Set  $\overline{GR}_{t_{i,n}}^j = \overline{GR}_{t_{i,n}}^j - \sum_{j \in J} GR_{i,k}^j \times \sigma_{j,m} \quad \forall m \in J$ 
20:           Compute next due-date for task item  $i$ 
21:           if Next due-date not within time horizon then
22:              $N \leftarrow N \setminus \{i\}$   $\triangleright$  Remove the maintenance task
23:           else
24:             Sort Task Items List
25:           end if
26:           break
27:         end if
28:         if  $n = |R_i|$  then  $\triangleright$  In case of no allocation possible
29:           Allocate  $i$  to  $t_0$ 
30:           Report Alert
31:         end if
32:       end while
33:     end procedure
34:   end while
35: end procedure

```

This strategy of starting the allocation process with these tasks follows the scheduling practice observed in practice, assigning the workforce necessary to these tasks before starting the allocation of maintenance tasks with more flexibility.

- *Medium Priority* – these are the maintenance tasks dephased from the aircraft checks intervals. Each of these tasks has an interval length larger than the A-check interval (e.g., the task in Table 1) and hence they will not necessarily be allocated to every maintenance check.
- *Low Priority* – these are the maintenance tasks with a low frequency of occurrence. They are dephased from the aircraft checks by, at least, being able to skip at least one A-check from any day within the planning horizon. These tasks have some flexibility, and they can be allocated at last.

The tasks within each of these classes are sorted by the maintenance costs, as expressed in the second and third terms in (10).

4.3.1. Task items loop

Task item loop (TIL) is the main procedure of the algorithm. The goal is to choose the best maintenance opportunity that minimizes the maintenance costs, as defined in (10), and to select from the bins the one that less compromises the best allocation of subsequent task items. After sorting all the task items according to their costs, the first task item has the highest priority; the second task item has the second-highest priority, and so forth. We define a list of bins that would allow a feasible allocation of task item  $i$  before the associate task interval is expired according to the maintenance check schedule. A *fictitious* bin ( $t_0$ ) is added to this list of bins in case none of the available bins has enough resources to allocate the task item. Other than that, we will not create any new bin during the task allocation.

After that, we sort the available bins for task item  $i$  according to the maintenance resources within bins in descending order. Namely, the bin with the most resources is always the first to assign the task in it. After that, the allocation of each task item following a “worst bin” selection process in the fourth step. Therefore, the TIL procedure gives a higher preference to the bins closer to the due date of the task item and, among these bins, to the ones that have more available maintenance resources.

4.3.2. Allocation of tasks to bins

The next procedure is to allocate the task items to a bin, following the sorted list of bins. If the bin under consideration has enough available labor hours for the necessary skills to perform the respective maintenance task, we allocate an item to the bin. In this case, we subtract the labor hours consumed to execute the task from the total available labor hours from that bin. The next step is to check the need to remove the task that has been allocated. For a routine task, we simulate the evolution of usage parameters after allocation and estimate its new date according to aircraft daily utilization. If the next due date is beyond the end of the time horizon, we just remove the task from the task item list. For a non-routine task, since they are not recurrently performed, we generate a new due date after the end of the planning horizon. For the case of running out of bins to which the task can be allocated, we generate an alert and put the task into fictitious bin  $t_0$ . This fictitious bin includes all the tasks that have not been allocated to any available bin, and we will inform the maintenance controller and let them address those tasks.

5. Case study

In this section, we present a case study on a major European airline and illustrate the applicability of the TAP approach. The input data includes aircraft utilization, a 4-year maintenance schedule generated by the dynamic programming based methodology described in the paper of Deng, Santos, and Curran (2020), task information from a heterogeneous fleet of 45 aircraft, and an associated estimation of available workforce per day. Our airline partner currently follows a manual process to allocate the aircraft maintenance tasks to checks, supported by a digital solution that keeps track of the open tasks and suggests a prioritization of maintenance activities. There are two maintenance planners in the airline doing this job for the entire fleet.

We consider eight skill types and that the productivity factor of each worker is equivalent to 4.8 productive labor hours per day, following the airline practice. The remaining hours of the labor shift are dedicated to transitioning meetings between work shifts, collection of materials or equipment, obtaining information about the maintenance task, reporting, and ancillary activities.

The results from this case study are discussed in Section 5.1, followed by an analysis of the current airline practice of not performing any aircraft C-check tasks in an A-check (Section 5.2). In

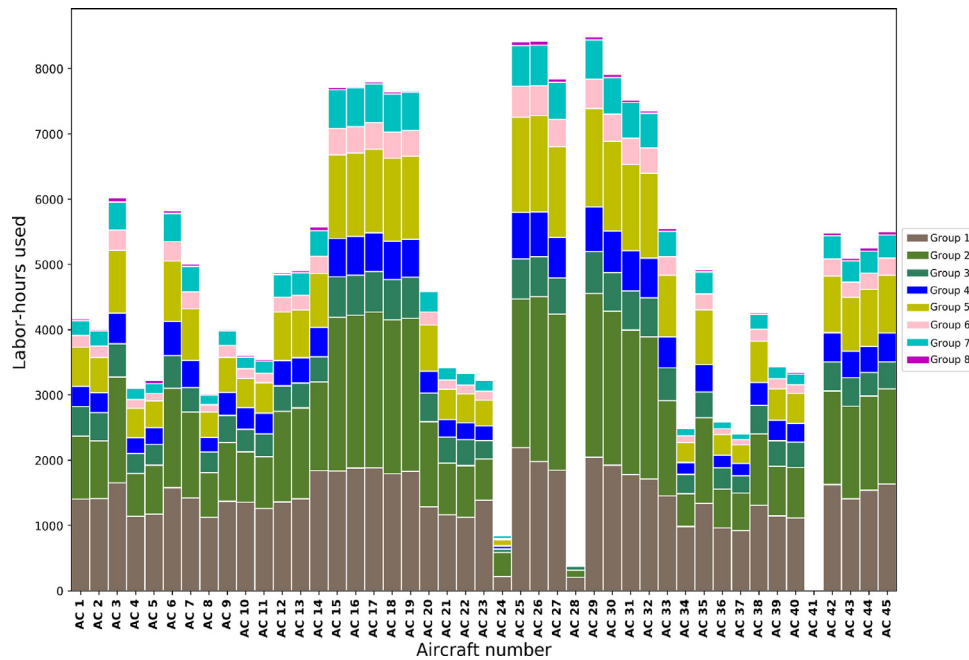


Fig. 3. Labor hours distribution per aircraft and skill type.

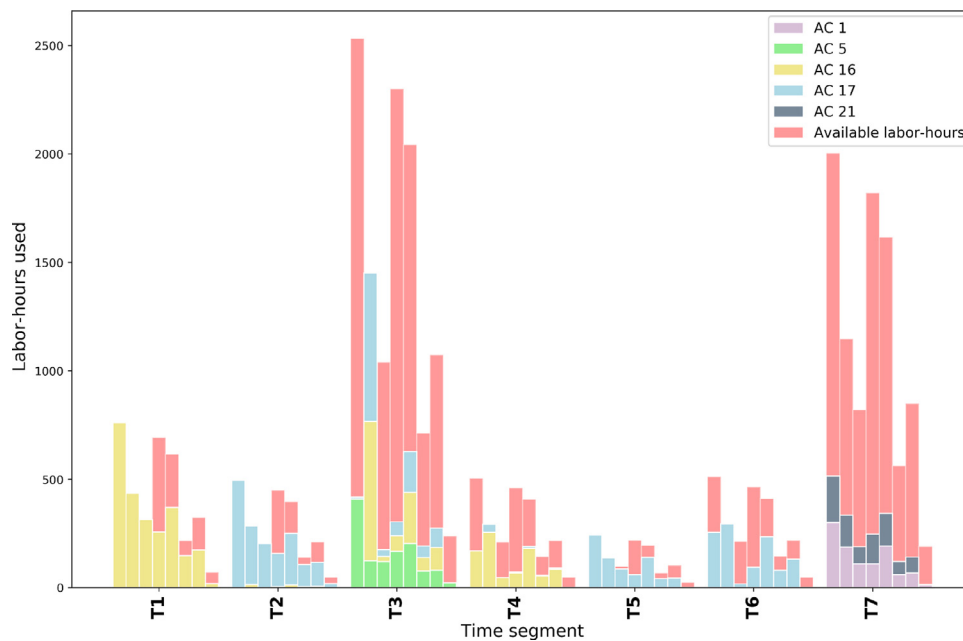


Fig. 4. The amount of labor hours used for each skill type during the time segments within the overlap situation. Each time segment has eight different bars and each bar represent a particular skill type (Group 1, Group 2, ..., Group 8).

Section 5.3, we validate the results obtained using the proposed task allocation heuristic. We suggest assessing the algorithm performance by comparing it with the solution obtained when using an exact method for solving the MILP presented in Section 3.4. Furthermore, all results obtained by the proposed heuristic were validated by the maintenance planners of the airline partner.

5.1. Optimization results

In this subsection, we apply the proposed task allocation algorithm to the case study, following the airline current policy of not allowing to allocate C-check tasks to A-check maintenance opportunities. The problem was solved in less than 14 minutes by

the algorithm. The outcome is a 4-year, fleet-wide task allocation plan that satisfies labor-hour constraints and tasks fix intervals. The plan includes around 85 thousand task items, from which 24% of them are C-check tasks, and 76% are A-check tasks. Despite this, the C-check tasks consume about 65.5% of the labor hours allocated to perform the tasks. The algorithm achieves an average de-escalation of 205 days for C-check tasks and 19.3 days for A-check tasks.

Fig. 3 shows the distribution of labor hours per skill for the maintenance of all aircraft in the fleet for the full planning horizon. There is significant diversity in the required labor hours among the aircraft. And the difference in aircraft age, the number of C-check events in the maintenance schedule, and the differences in terms

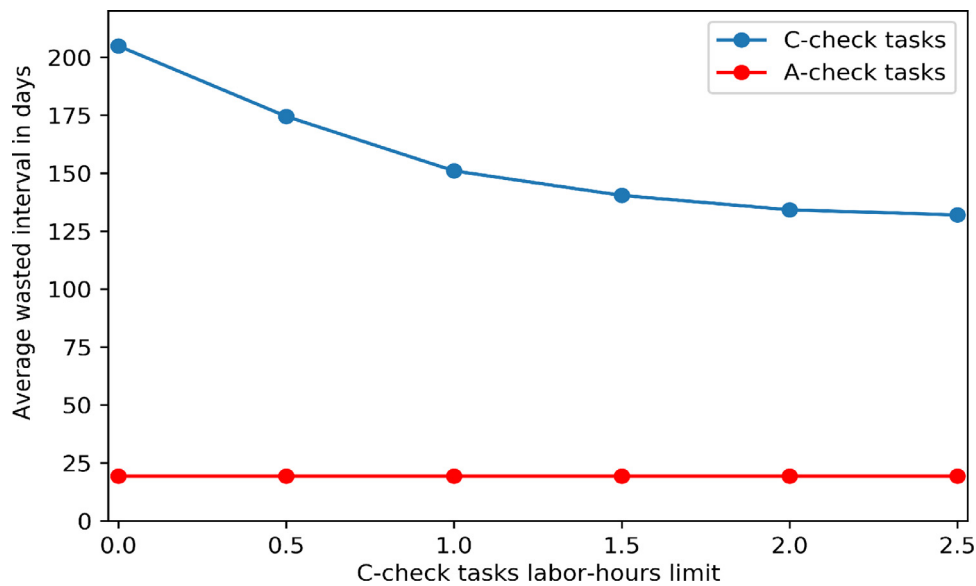


Fig. 5. Average wasted interval in days for increased C-check task labor hours thresholds.

of aircraft utilization cause this diversity. For instance, aircraft AC-41 is phased-out a few days after the start of the planning horizon, while AC-24 is phased out one and half years after the beginning of the planning horizon, following a minor C-check and 10 A-checks. In the same way, it is possible to identify the aircraft that perform a C-check early in the planning horizon and hence have to undergo three C-check before the end of the planning horizon. And this applies to aircraft AC-25, AC-26, and AC-29.

To analyze the maintenance plan in more detail, we decided to focus on the overlap situation presented in Fig. 2. The allocation of labor hours per time segment is depicted in Fig. 4. In this figure, there are eight bars per time-segment, representing the eight different skill types. We observe that the first six time-segments consume all the available labor hours of the Group 2 skill type. And this restricts the allocation of tasks requiring labor hours from Group 2 skill for AC-5, AC-16, and AC-17 since these aircraft will have a fully constrained overlap situation. And this forces some of the A-check tasks from these aircraft to be allocated to a previous A-check. Similarly, there are also C-check tasks being anticipated at an earlier C-check. In the latter case, it means that some components are inspected or replaced about two years earlier than intended.

### 5.2. Flexible task allocation policy

In this subsection, we question the current airline policy of not allocating any C-check task to A-check maintenance opportunities, even though we observe that there is a surplus of labor hours in the A-checks scheduled. Several small C-check tasks would fit in an A-check, in terms of time and resources needed. For this reason, we performed a simulation in which these C-check tasks are allowed to be allocated to A-check opportunities. We carry out the analysis considering different thresholds for the size of these tasks. After discussing with maintenance planners from the airline, we agree on using the labor hours needed for the task as the reference metric for task size, and to consider a threshold varying from zero to 2.5 labor hours.

The simulation results (presented in Fig. 5) indicate that the de-escalation of C-check tasks can be reduced from more 205 days to 132 days when allowing C-check tasks within 2.5 labor hours to be executed on A-check opportunities. From the results, it can also be concluded that the marginal gain of extending the threshold re-

duces as the threshold increases in value. In fact, it can be inferred from Fig. 5 that, for this airline, after a labor hours threshold of 2.0 or 2.5, there are barely any benefits of extending this threshold. The reason being that very few C-check tasks consume more than 2.5 labor hours and can still be allocated in an A-check without compromising the allocation of the A-check tasks to their best A-check opportunities.

### 5.3. Algorithm performance analysis

To analyze the performance of the task allocation algorithm, we compare it with the performance of an approach using an exact method to solve the TAP formulated in Section 3.4. To provide a more detailed comparison, we decided to vary the productivity factor of the workforce, from the initial considered 4.8 labor hours per day to a restricted case of 3.2 labor hours per day.

To compute solutions with the exact method in a reasonable time for the more restricted cases, we follow an iterative process for the creation of task items and maintenance opportunities for each maintenance task (Section 3.3.1). That is, we initially run the MILP, then add new items and maintenance opportunities for those task items that had the constraints violated and rerun the MILP until the problem becomes feasible. For the 4.8 labor hours case, the MILP formulation resulted in 1.15 million decision variables and 373 thousand constraints. The task allocation algorithm is coded in Python 3.7, while the exact method is addressed using the commercial solver Gurobi. The results from both approaches are computed on an Intel Core i7 2.6GHz laptop with 8GB ram.

We summarize the results in Table 2. Each line of Table 2 compares the computation times and presents the optimality gap for a given productivity factor between two different approaches, where the results obtained from the solver is used as a reference. While the computation time of the exact method (MILP solver) explodes with the decrease of the productivity factor, the same does not happen to the proposed heuristic algorithm. The proposed heuristic is more than 30% faster than the exact method for the default productivity factor of 4.8 labor hours, and the optimality gap is only 0.03%. Even though the productivity labor hours decrease to 3.2, the optimality gap is still within 5%.

It is worth mentioning that for the most constrained test case, the exact method requires about 4.9 hours to compute the optimal solution, while the proposed heuristic needs less than 15 minutes.

**Table 2**  
Simulation results of performance analysis.

Productivity Labor hours	Computational time (s)		Solution Gap
	MILP solver	Heuristic	Heuristic vs. Solver
4.80	1,135	773	0.03%
4.72	1,148	775	0.03%
4.64	1,154	776	0.03%
4.56	1,159	778	0.04%
4.48	1,168	779	0.05%
4.40	1,172	787	0.11%
4.32	1,181	792	0.23%
4.24	1,186	798	0.36%
4.16	1,187	803	0.54%
4.08	1,195	811	0.81%
4.00	1,639	818	1.17%
3.92	1,821	822	1.44%
3.84	1,903	828	1.61%
3.76	2,570	835	1.43%
3.68	3,097	839	1.90%
3.60	3,857	846	2.45%
3.52	4,702	851	2.88%
3.44	5,679	861	3.38%
3.36	8,243	866	3.89%
3.28	13,828	871	4.47%
3.20	17,636	879	4.95%

In summary, from a perspective of solution quality, the solution gap between the heuristic algorithm and the optimal solution is within 5% for all test cases. For cases where the productivity factor was higher than 4.0 labor hours, the solution gap is below 1%, and this confirms that the heuristic algorithm is capable of producing good solutions in minutes for a realistic TAP with a fleet of 45 aircraft.

## 6. Conclusion

The task allocation problem (TAP) of aircraft maintenance is defined as assigning tasks to their optimal maintenance opportunities. In this research, we formulate TAP as a time-constrained variable-sized bin packing problem (TC-VS-BPP), in which we treat maintenance opportunities as bins and the tasks as items, and there are time constraints on both bins and items. TC-VS-BPP is NP-hard and, therefore, challenging to solve for large case instances. For this reason, we proposed a constructive heuristic to solve the TAP (TC-VS-BPP). The proposed approach is an efficient iterative process based on the worst-fit decreasing (WFD) algorithm. According to a real-life case study on a heterogeneous fleet of 45 aircraft, the heuristic is more than 30% faster than an exact method, while the solution gap is smaller than 0.1%. For the most restricted test case, the solution from the heuristic is only 5% worse than the solution obtained from the exact method, while being much faster. The computation time of TAP is essential in the aircraft maintenance domain since changes to the priority/urgency of existing tasks or new (non-routine) tasks can require running the proposed constructive heuristic many times per day. Therefore, an algorithm that runs in a reasonable and stable computational time, regardless of how restrictive is the problem, is something very useful.

During the case study, we are told that some airline technicians work just part-time at the hangar, and we overestimated the maintenance capacity if we set the productivity labor hours to 8 (all technicians are working full time, 8 hours a day). The maintenance capacity constraint (5) is not the main restriction during the task allocation process. Since there is no other data to support sensitivity analysis, we change the productivity labor hours to test the proposed heuristic in a more constrained context.

The research presented in this paper is also one of the requirements from the airline, continuing the work of aircraft main-

tenance check scheduling optimization described in Deng et al. (2020). The dynamic programming based methodology in Deng et al. (2020) first determines the optimal start dates of all maintenance check for the entire fleet, and the optimal maintenance check schedule indicates in which checks a maintenance task can be allocated. Otherwise, without a maintenance check schedule, it is very time-consuming to know when, which aircraft, and what maintenance tasks should be performed. The results of the maintenance task allocation are the task execution plans for all maintenance checks, which help the technicians to execute the right task, on the right aircraft, at the right time.

We structure the task allocation problem of aircraft maintenance as a bin packing problem (BPP) so that it can be solved quickly using the worst-fit decreasing algorithm. Whenever unscheduled maintenance tasks occur, we can use the dynamic programming based methodology presented in Deng et al. (2020) to obtain a new maintenance check schedule, and then apply the task allocation framework to update the tasks accordingly. The task allocation framework is suitable for real-life applications. It can provide near-optimal solutions to the TAP, significantly reducing the workload currently required in practice for the creation of maintenance plans. Besides, given that it runs in minutes, it can potentially be used to dynamically adjust the task allocation plans given flight schedule disruptions during operations or emergency of unscheduled tasks during the execution of maintenance inspections. Furthermore, the task allocation framework can be used to test or analyze different maintenance concepts or policies, as demonstrated in Section 5.2.

Future research on this work may consider the stochasticity associated with the TAP problem, or explore the uncertainty related to, e.g., the emerge of “non-routine tasks” or the aircraft utilization over the planning horizon. And this could enhance the robustness of the outgoing task execution plan. Furthermore, a stochastic approach could extend the current work to consider health prognostics and diagnostics, investigating the possibility of incorporating condition-based maintenance in the proposed framework. An alternative interesting future research direction is to integrate the maintenance check schedule optimizer with the task allocation framework proposed. And this could improve the overall quality of the maintenance plan, including checks schedule and task allocation per check.

## Acknowledgments

This research work is part of AIRMES project, which received funding from the Clean Sky 2 Joint Undertaking under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 681858. We would like to show gratitude to the associated AIRMES partners for providing their aircraft maintenance data and validation of the proposed methodology and its practical relevance. For more information please visit [www.airmes-project.eu](http://www.airmes-project.eu).

## References

- Ackert, S. P. (2010). Basics of aircraft maintenance programs for financiers. [http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics\\_of\\_aircraft\\_maintenance\\_programs\\_for\\_financiers\\_v1.pdf](http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers_v1.pdf). (Accessed on June 6, 2020) [http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics\\_of\\_aircraft\\_maintenance\\_programs\\_for\\_financiers\\_v1.pdf](http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers_v1.pdf).
- Ackert, S. P. (2018). Aircraft Maintenance Handbook for Financiers. [http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/aircraft\\_mx\\_handbook\\_for\\_financiers\\_v1.pdf](http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/aircraft_mx_handbook_for_financiers_v1.pdf). (Accessed on June 6, 2020) [http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/aircraft\\_mx\\_handbook\\_for\\_financiers\\_v1.pdf](http://www.aircraftmonitor.com/uploads/1/5/9/9/15993320/aircraft_mx_handbook_for_financiers_v1.pdf).
- Correia, I., Gouveia, L., & da Gama, F. S. (2008). Solving the variable size bin packing problem with discretized formulations. *Computers & Operations Research*, 35, 2103–2113. <https://doi.org/10.1016/j.cor.2006.10.014>.
- Csirik, J. (1989). An on-line algorithm for variable-sized bin packing. *Acta Informatica*, 26, 697–709. <https://doi.org/10.1007/BF00289157>.



- Deng, Q., Santos, B. F., & Curran, R. (2020). A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281, 256–273. <https://doi.org/10.1016/j.ejor.2019.08.025>.
- Fazi, S., van Woensel, T., & Fransoo, J. C. (2012). A Stochastic Variable Size Bin Packing Problem with Time Constraints. <https://pure.tue.nl/ws/files/3678899/565767590859814.pdf>. (Accessed on June 6, 2020) <https://pure.tue.nl/ws/files/3678899/565767590859814.pdf>.
- Friesen, D. K., & Langston, M. A. (1986). Variable sized bin packing. *SIAM Journal on Computing*, 15(1), 222–230. <https://doi.org/10.1137/0215016>.
- Garey, M. R., & Johnson, D. S. (1990). *Computers and intractability: a guide to the theory of NP-Completeness*. USA: W. H. Freeman & Co..
- Haouari, M., & Serairi, M. (2009). Heuristics for the variable sized bin-packing problem. *Computers & Operations Research*, 36(10), 2877–2884. <https://doi.org/10.1016/j.cor.2008.12.016>.
- Hölzel, N., Schröder, C., Schilling, T., & Gollnick, V. (2012). A Maintenance Packaging and Scheduling Optimization Method for Future Aircraft. In *Air transport and operations symposium*. <https://doi.org/10.3233/978-1-61499-119-9-343>.
- Johnson, D. S. (1972). Fast allocation algorithms. In *Proceedings of the 13th annual symposium on switching and automata theory*. <https://doi.org/10.1109/SWAT.1972.4>.
- Johnson, D. S. (1974). Fast algorithms for bin packing. *Journal of Computer and System Science*, 8, 272–314. [https://doi.org/10.1016/S0022-0000\(74\)80026-7](https://doi.org/10.1016/S0022-0000(74)80026-7).
- Kang, J., & Park, S. (2003). Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147, 365–372. [https://doi.org/10.1016/S0377-2217\(02\)00247-3](https://doi.org/10.1016/S0377-2217(02)00247-3).
- Li, H., Zuo, H., Lei, D., Liang, K., & Lu, T. (2015). Optimal combination of aircraft maintenance tasks by a novel simplex optimization method. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2015/169310>.
- Muchiri, A. K., & Smit, K. (2009). Application of maintenance interval de-escalation in base maintenance planning optimization. *Enterprise Risk Management*, 1(2). <https://doi.org/10.5296/erm.v1i2.179>.
- Steiner, A. (2006). A heuristic method for aircraft maintenance scheduling under various constraints. In *Proceedings of the 6th swiss transport research conference*.
- Van Buskirk, C., Dawant, B., Karsai, G., Sprinkle, J., Szokoli, G., Suwanmongkol, K., & Currer, R. (2002). Computer-Aided Aircraft Maintenance Scheduling. *Technical Report*. Institute for Software-Integrated Systems.