Computers and Operations Research 95 (2018) 151–162

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An extensive operations and maintenance planning problem with an efficient solution method

Javad Seif, Andrew J. Yu*

Department of Industrial and Systems Engineering, The University of Tennessee, Knoxville, TN 37996-2315, USA

ARTICLE INFO

Article history:
Received 21 July 2017
Revised 14 February 2018
Accepted 22 March 2018
Available online 27 March 2018

Keywords:
Mixed integer programming
Operations and maintenance planning
Availability
Exact solution algorithm
Cutting planes

ABSTRACT

In this paper, we extend the formulation and solution algorithm of a Flight and Maintenance Planning (FMP) problem to cover a wider range of applications. We consider a general Operations and Maintenance Planning (OMP) case where machines are shared between multiple operation schedules as well as have multiple preventive maintenance activities. In this case, machines have different usage-based intervals and the maintenance duration varies for each machine. We formulate and optimize the OMP problem to accommodate the maintenance requirements and physical characteristics of multiple stations. We evaluate the performance of the generalized solution methodology and show its effectiveness and efficiency through computational experiments.

1. Introduction

Operations and maintenance have conflicting objectives with respect to machines; pure operations planning focuses on maximizing the output of the machines while and pure maintenance planning focuses on minimizing machine downtimes. The operational perspective risks machine health and the maintenance perspective ignores the importance of maximum operational throughput. Consequently, simultaneous optimization of operations and maintenance planning is challenging. Although the literature for operations and maintenance planning is largely disconnected (Yao et al., 2005), integration of operations and maintenance problems attracted many researchers in the recent years (Rezg et al., 2014). In this paper, we introduce an Operations and Maintenance Planning (OMP) problem that generalizes the Flight and Maintenance Planning (FMP) problem formulated and solved by Gavranis and Kozanidis (2015). We reconstruct the mixed-integer program and the exact solution algorithm of the FMP for use in OMP. The generalized OMP considers potential scenarios covering a wide range of operations and maintenance planning problems in a variety of industries. Unlike the FMP problem, the generalized OMP problem considers:

- multiple maintenance activities with different time intervals;
- machine-specific maintenance durations;
- multiple types of operations, and each machine is capable of performing one or more than one type of operation;
- multiple maintenance stations with different capacities that accommodate only certain types of machines; and
- overlapping maintenance activities.

Like the FMP, the objective function of the OMP maximizes machine availability over a finite planning horizon. This benefits both operations and maintenance. Maximizing machine availability supports operations by minimizing interruptions caused by unplanned failures. Similarly, scheduled preventive maintenance prolongs machine service life and supports the planned operational tempo. The goal of the OMP is to find a schedule that satisfies both operations and maintenance requirements and maximizes the availability of the machines.

The maintenance planning and scheduling literature typically considers a single generic preventive maintenance (PM) for machines (Yu and Seif, 2016). In practice, machines can have multiple PMs, each with a different time interval. In addition, PM duration differs based on unique characteristics of each machine (Seif et al., 2017). For example, aircraft have multiple PMs (e.g., Type A, Type B, Type C and Type D check). The frequency of each PM is a combination of flight hours and landing-take-off cycles (Sriram and Haghani, 2003) and the duration varies based on the age of the aircraft. Consequently, identical aircraft may have different PM frequency and duration based on an individual aircraft's operational tempo. Maher et al. (2014) suggests considering other maintenance check types in aircraft maintenance routing problems. Similarly, construction and mining machinery PMs are based on hours of operations, e.g., 100 h, 250 h, and 500 h of operation. See (Caterpillar, 2010a, 2010b, 2010c). In this paper, we consider

* Corresponding author.
E-mail addresses: jseif@utk.edu (J. Seif), ajyu@utk.edu (A.J. Yu).

https://doi.org/10.1016/j.cor.2018.03.010
0305-0548/© 2018 Elsevier Ltd. All rights reserved.

multiple PMs with different frequencies and durations for each machine.

The intent of this research is to closely mimic a typical OMP environment where machines are shared between multiple operations. For example, the aircraft industry classifies operations based on flight length; e.g. domestic (short-distance) flights or international (long-distance) flights, or flight capacity and mission; e.g., cargo, commercial, etc. Individual aircraft can be assigned to an operation based on its capability. Similarly, constructions operations are classified based on the nature of the operations; e.g. earthmoving, dirt transportation, etc., or based on the physical location of the operations. Construction machinery is assigned to an operation based on its capability and proximity to the work site. The FMP literature typically considers only one type of operation and one set of identical machines to fulfill operational requirements. For instance, Cho (2011) and Gavranis and Kozanidis (2015) consider a single flight schedule for a finite planning horizon and a set of identical aircraft to fulfill the flight-hour requirement for each planning period. However, in other areas multiple types of assets are considered. In fleet assignment problem, the goal is to optimally assign scheduled flights to aircraft types based on their capacity and performance (Abdelghany et al., 2017). Brown et al. (1990) consider different types of ships with different capabilities (in terms of weapon systems) in the scheduling of naval combatants. Boere (1977) considers multiple aircraft types in the maintenance scheduling of aircraft.

When machines/aircraft are shared among different types of operations, finding an optimal operations/flight schedule can be solved in two different ways. (1) Define a problem for each type of operation and seek the optimal solution for each problem separately, or, (2) define a single integrated problem and seek a solution that optimizes all combined operations. The first approach is often not feasible because the individual schedules may interfere with each other if a machine is assigned to multiple operations within the planning period. The second approach achieves global optimization and feasibility at the expense of sub-optimizing individual problems. In this paper we are trying to follow the second approach in the interest of practicality.

Having multiple operations in OMP has been addressed in the literature. Aramon Bajestani and Beck (2013) consider different flights and different types of aircraft for scheduling repairs in a shop. Brucker and Schlie (1990) generalize job shop scheduling by introducing multi-purpose machines. Our research incorporates a variety of operations into the OMP; multiple equipment operational schedules for a specific planning horizon and consider whether individual machines can perform certain operations are both included.

This research also considers multiple stations in a maintenance shop. This is important since most shops that service large companies, such as an airline, require redundant capability to ensure prompt turnaround and return to service. Regardless, there may be constraints such as the size or number of machines undergoing maintenance simultaneously; the available tools, or the distance between operating sites. In the airline industry, multiple maintenance stations (bases, hangars) may exist at the same airport or at different airports (Papakostas et al., 2010; Talluri, 1998). Feo and Bard (1989) also point out heavy construction or mining machinery might be serviced in a maintenance station separate from smaller machinery based on tool and space requirements.

Hahn and Newman (2008) propose a mixed integer linear programming model for scheduling the deployment and maintenance of United States Coast Guard (USCG) helicopters. They consider multiple maintenance activities with different time intervals, and multiple operational locations. There are many civilian and military applications for our research. The OMP of USCG missions are good examples where multiple types of operations are involved and multiple types of assets are shared between the operations. In the search and rescue mission of USCG, multiple operations, multiple assets, multiple stations and multiple types of heavy maintenance activities are involved. Hahn and Newman (2008) considered inspection combinations in which the heavy maintenance with the lowest time interval is performed whenever a heavy maintenance with a greater time interval is due. We also consider this type of overlapping PMs in our modeling and solution method as will be discussed in the next section.

The rest of the paper is organized as follows. Section 2 defines the generalized OMP problem by extending the mixed integer program (MIP) of the FMP problem. Section 3 describes the redesigned FMP method to solve the OMP problem. Section 4 presents the results of computational experiments designed to evaluate the efficiency and effectiveness of the solution. Section 5 presents the conclusion along with recommendations for future research.

2. Mathematical formulation of OMP

This research considers a set of multi-purpose machines that can perform more than one type of operation. Each machine operates in a finite planning horizon with multiple time periods. The requirement for each type of operation and the operation-hours for each machine is defined at the start of each planning period. The objective is to maximize machine availability to increase readiness.

The term “residual operation time” for each machine is defined as the total time that a machine can perform operations before it should stop for a usage-based PM. Residual operation time is positive if and only if the machine has no PM that is due and is available for operations. Similarly, “residual maintenance time” indicates the total remaining time for performing PMs on a machine before it can be released for either operations or another PM. Assume that machines have more than one type of PM, each with a different usage-based time interval.

Assume maintenance stations have limited capacity to perform PMs and machines are assigned to specific maintenance stations. Constraints on maintenance crew availability also exist. For each period in the planning horizon, maintenance crew availability is known. The maintenance crew is shared between maintenance stations, and traveling and routing between the stations is assumed to be insignificant. Sharing the maintenance crew between multiple stations might not be applicable in some industries, yet it is a useful assumption in this research because it reduces the number of variables and constraints of the model, which enhances the computational experiments. However, we will show that considering separate maintenance crew for different stations is not difficult. Assignment of the crew to maintenance tasks within each station can be studied separately without affecting the solution obtained from the OMP.

For given operational requirements, physical capacity, and maintenance crew availability, the objective of the OMP problem is to find the optimum plan for operations and maintenance of each machine, which maximizes the overall readiness of the machines in response to unplanned failures, external threats, or unexpected increases in operational requirements. Scheduled PMs are performed according to vendor suggested usage-based maintenance intervals.

The following notations are used in the mixed integer linear program of OMP:

Sets

N	Set of all machines.
P	Set of PM activities.
O	Set of operational activities.
M	Set of maintenance stations.

Parameters

T	Number of time periods in the planning horizon; also, the period at which completion of all operations is due.
$R_{o,t}$	Requirement for operation type $o \in O$ at period $t = 1, \ldots, T$.
H_t	Time capacity of human resources/workforce for performing PMs in period $t = 1, \ldots, T$.
C_m	Capacity of maintenance station $m \in M$; the number of machines it can hold.
Y_p	Usage-based PM interval. Partial residual operation time (PROT) of a machine with respect to PM type $p \in P$ immediately after the PM is performed on the machine.
$G_{n,p}$	Duration of PM type $p \in P$ on machine $n \in N$.
$A^1_{n,p}$	Binary availability state of machine $n \in N$ at the beginning of the first-time period with respect to PM type $p \in P$.
$Y^1_{n,p}$	PROT of machine $n \in N$ at the beginning of the first-time period for PM type $p \in P$.
$G^1_{n,p}$	Residual maintenance time of PM type $p \in P$ of machine $n \in N$ at the beginning of the first-time period.
X^{max}	Maximum operational time of a machine in a single period.
Y^{min}	Lower bound on the PROT of available machines.
G^{min}	Lower bound on the residual maintenance time.
$B_{n,o}$	Binary capability state of machine $n \in N$ performing operations of type $o \in O$.
$E_{n,m}$	Binary state of whether machine $n \in N$ is assigned to the maintenance station $m \in M$, where $\sum_{m \in M} E_{n,m} = 1$.
K	A sufficiently large number.

Decision variables

$a_{n,p,\,t}$	Partial availability state. Binary decision variable taking the value 1 if machine $n \in N$ is available (can still operate), considering *only* PM type $p \in P$, in period $t = 1, \ldots, T$, and 0, otherwise.
$A_{n,t}$	Actual availability state. Binary decision variable taking the value 1 if machine $n \in N$ is available in period $t = 1, \ldots, T$, considering *all* its PMs, and 0, otherwise.
$y_{n,p,\,t}$	Partial residual operating time (PROT) of machine $n \in N$ at the beginning of period $t = 1, \ldots, T$ with respect to PM type $p \in P$.
$Z_{n,t}$	Actual residual operating time (AROT). Minimum of PROTs of machine $n \in N$ at the beginning of the period t, $t = 1, \ldots, T$.
$x_{n,o,\,t}$	Amount of time that machine $n \in N$ spent on operation $o \in O$ in period $t = 1, \ldots, T$.
$g_{n,p,\,t}$	Residual maintenance time of PM type $p \in P$ of machine $n \in N$ at the beginning of period $t = 1, \ldots, T$.
$h_{n,p,\,t}$	Amount of time that the maintenance crew spent on PM type $p \in P$ on machine $n \in N$ in period $t = 1, \ldots, T$.
$d_{n,p,\,t}$	Binary decision variable taking the value 1 if PM type $p \in P$ on machine $n \in N$ completes in period $t = 1, \ldots, T$.
$f_{n,p,\,t}$	Binary decision variable taking the value 1 if machine $n \in N$ must stop because of PM type $p \in P$ in the beginning of period $t = 1, \ldots, T$.
$q_t, u_{n,p,t}, r_{n,p,t}, \tilde{y}_{n,p,t}, b_{n,p,t}$	Auxiliary variables.

The OMP model is formulated as follows.

$$\text{maximize } COA = \sum_{n \in N} \sum_{t=2}^{T+1} Z_{n,t} \tag{1}$$

Subject to:

$$y_{n,p,t+1} + \tilde{y}_{n,p,t+1} = y_{n,p,t} - \sum_{o \in O} x_{n,o,t}{}^{Bn,o} \qquad n \in N, p \in p, p \neq p', t = \ldots, T \tag{2a}$$

$$y_{n,p',t+1} = y_{n,p',t} - \sum_{o \in O} x_{n,o,t} B_{n,o} + Y_{p'} d_{n,p',t+1}, \qquad n \in N, \quad t = 1, \ldots, T \tag{2b}$$

$$d_{n,p,t+1} \geq a_{n,p,t+1} - a_{n,p,t}, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{3}$$

$$a_{n,p,t+1} - a_{n,p,t} + 1.1\left(1 - d_{n,p,t+1}\right) \geq 0.1, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{4}$$

$$g_{n,p,t+1} = g_{n,p,t} - h_{n,p,t} + G_{n,p} f_{n,p,t+1}, \quad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{5}$$

$$f_{n,p,t+1} \geq a_{n,p,t} - a_{n,p,t+1}, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{6}$$

$$a_{n,p,t} - a_{n,p,t+1} + 1.1\left(1 - f_{n,p,t+1}\right) \geq 0.1, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{7}$$

$$\sum_{n \in N} x_{n,o,t} B_{n,o} = R_{o,t}, \qquad o \in O, \ t = 1, \ldots, T \tag{8}$$

$$\sum_{n \in N} E_{n,m}(1 - A_{n,t}) \leq C_m, \qquad m \in M, \ t = 2, \ldots, T+1 \tag{9}$$

$$\sum_{p \in P} \sum_{n \in N} h_{n,p,t} \leq H_t, \qquad t = 1, \ldots, T \tag{10}$$

$$H_t \leq \sum_{p \in P} \sum_{n \in N} h_{n,p,t} + K(1 - q_t), \qquad t = 1, \ldots, T \tag{11}$$

$$\sum_{p \in P} \sum_{n \in N} g_{n,p,t} \leq \sum_{p \in P} \sum_{n \in N} h_{n,p,t} + K q_t, \qquad t = 1, \ldots, T \tag{12}$$

$$y_{n,p,t} + K u_{n,p,t} \leq K, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{13}$$

$$a_{n,p,t+1} \leq \left(y_{n,p,t} - \sum_{o \in O} x_{n,o,t} B_{n,o} \right) K + K u_{n,p,t}, \quad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{14}$$

$$g_{n,p,t} + K r_{n,p,t} \leq K, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{15}$$

$$1 - a_{n,p,t+1} \leq (g_{n,p,t} - h_{n,p,t})K + K r_{n,p,t}, \qquad n \in N, \ p \in P, \ t = 1, \ldots, T \tag{16}$$

$$y_{n,p,t} \le a_{n,p,t} Y_p, \qquad n \in N,\ p \in P,\ t = 2, \ldots, T+1 \tag{17a}$$

$$\tilde{y}_{n,p,t} \le (1 - a_{n,p,t}) Y_p, \qquad n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{17b}$$

$$g_{n,p,t} \le (1 - a_{n,p,t}) G_{n,p}, \qquad n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{18}$$

$$\sum_{o \in O} x_{n,o,t} B_{n,o} \le X^{max} A_{n,t}, \qquad n \in N,\ t = 1, \ldots, T \tag{19}$$

$$y_{n,p,t} \ge Y^{min} a_{n,p,t}, \qquad n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{20}$$

$$g_{n,p,t} \ge G^{min}(1 - a_{n,p,t}),\ \ n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{21}$$

$$\sum_{o \in O} x_{n,o,t} B_{n,o} \le Z_{n,t}, \qquad n \in N,\ t = 1, \ldots, T \tag{22}$$

$$h_{n,p,t} \le g_{n,p,t}, \qquad n \in N,\ \ p \in P,\ \ t = 1, \ldots, T \tag{23}$$

$$a_{n,p,1} = A^1_{n,p}, \qquad n \in N,\ \ p \in P \tag{24}$$

$$y_{n,p,1} = Y^1_{n,p}, \qquad n \in N,\ \ p \in P \tag{25}$$

$$g_{n,p,1} = G^1_{n,p}, \qquad n \in N,\ \ p \in P \tag{26}$$

$$A_{n,t} \le a_{n,p,t},\ \ n \in N,\ \ p \in P,\ \ t = 1, \ldots, T+1 \tag{27}$$

$$Z_{n,t} \le y_{n,p,t},\ \ n \in N,\ \ p \in P,\ \ t = 1, \ldots, T+1 \tag{28}$$

$$y_{n,p,t} \ge Y_{p'} b_{n,p,t}, \qquad n \in N,\ p \ne p',\ p \in P,\ t = 2, \ldots, T+1 \tag{29}$$

$$b_{n,p,t} \le a_{n,p,t} \qquad n \in N,\ \ p \ne p',\ p \in\ P,\ t = 2, \ldots, T+1 \tag{30}$$

$$b_{n,p,t} \le 1 - a_{n,p',t}, \qquad n \in N,\ p \ne p',\ \ p \in\ P,\ t\ =\ 2, \ldots, T+1 \tag{31}$$

$$b_{n,p,t} \ge a_{n,p,t} - a_{n,p',t}, \qquad n \in\ N,\ p \ne p',\ \ p \in P,\ t = 2, \ldots, T+1 \tag{32}$$

$$x_{n,o,t} \ge 0,\ \ n \in N,\ \ o \in O,\ \ t = 1, \ldots, T \tag{33}$$

$$h_{n,p,t} \ge 0, \qquad n \in N,\ \ p \in P,\ \ t = 1, \ldots, T \tag{34}$$

$$y_{n,p,t}, \tilde{y}_{n,p,t},\ \ g_{n,p,t} \ge 0 n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{35}$$

$$u_{n,p,t},\ \ r_{n,p,t}\ \textit{binary}, \qquad n \in N,\ \ p \in P,\ \ t = 1, \ldots, T \tag{36}$$

$$a_{n,p,t},\ \ b_{n,p,t}\ \textit{binary}, \qquad n \in N,\ \ p \in P,\ \ t = 2, \ldots, T+1 \tag{37}$$

$$A_{n,t},\ \ Z_t,\ \ q_t\ \textit{binary}, \qquad n \in N,\ \ t = 1, \ldots, T \tag{38}$$

$$Z_{n,t} \ge 0, \qquad n \in N,\ \ t = 1, \ldots, T \tag{39}$$

The objective function maximizes the Cumulative Operations Availability (COA), which is equal to the sum of the AROTs of the machines over the planning horizon, as shown in Eq. (1). Because the residual times of the machines are given inputs (constraints) for the first-time period, the first-time period is eliminated from the equation. One extra period ($T+1$) is included to maximize the availability at the beginning of the next planning horizon. An alternative objective function is introduced in Section 6.

In Constraint set (2), PROTs for a machine at the beginning of the next period ($\forall p \in P$) are calculated in one of the following ways, depending on the status of the machine with respect to p. If the machine is *available* with respect to p, $y_{n,p,\ t}$ is equal to its current value minus the sum of its operating time in the current period. Note that all types of operations contribute in the same way to depleting the residual operating time. As was addressed in Introduction, if the model did not consider all types of operations and separate solutions were to be generated for various operations, the aggregated solutions would be prone to being infeasible from a practical standpoint (assigning a machine to two or more operations within the same time window, or assigning the machine to an operation it is not capabale of performing). If the machine is *unavailable* in the current period ($\exists p^* \in P$ where $a_{n,p^*,t} = 0$), $y_{n,p,\ t}$ does not change. If the machine is unavailable with respect to p, becomes available in the next period ($d_{n,p,\ t+1} = 1$), $y_{n,p,\ t}$ equals its nominal value (Y_p). The auxiliary variable $\tilde{y}_{n,p,t}$ takes the value of $y_{n,p,\ t}$ when it is forced to be equal to zero (refer to Constraint sets 29–32). Constraint set (5) has a similar logic. Each residual maintenance time of a machine in the beginning of the next period ($\forall p \in P$) is calculated by subtracting PM duration for the current period from its current value. If the PM starts in the next period, the binary variable $f_{n,p,\ t+1}$ takes the value 1, and the residual maintenance time is reset to the nominal PM duration of the machine, namely $G_{n,p}$.

Constraint sets (3), (4), (6) and (7) determine the start and completion of a PM and ensure that the binary variables $d_{n,p,\ t}$ and $f_{n,p,\ t}$ take the correct value. The pair ($a_{n,p,\ t}, a_{n,p,\ t+1}$) takes the value (0, 0), (0, 1), (1, 0) and (1, 1). For example, the pair (0, 1) implies that PM type $p \in P$ of machine $n \in N$ completes in the current period and the machine becomes *partially* available in the next period. In this case, $d_{n,p,\ t+1}$ takes the value 1 as expressed in Constraint set (3). For other pairs, where the PM is incomplete, Constraint set (4) forces $d_{n,p,\ t+1}$ to take the value 0. A similar logic applies to $f_{n,p,\ t}$ in Constraint sets (6) and (7). A machine becomes available when all the partial availability states are equal to 1, as shown in Constraint set (27).

Constraint set (8) guarantees that all of the operating requirements ($\forall o \in O$) are satisfied in all time periods of the planning horizon, while considering the capabilities of the machines. Constraint sets (9) and (10) ensure that the physical and time capacities of the stations are not violated in any time period. When the maintenance crew does not perform a PM, maintenance stations are considered idle. The stations should not be idle if there is at least one machine waiting for maintenance. Because the objective function is a maximization objective, all of the capacity will be used. Constraint sets (10–12) are resource constraints and ensure that the total time spent on all machines is not greater than the available time of the maintenance crew (time capacity of the stations), and ensure that the total time spent maintaining machines in each period t is equal to the minimum of the capacity of the maintenance crew (H_t) and the total maintenance time required in that period ($\sum_{n \in N} \sum_{p \in P} g_{n,p,t}$). Constraint sets (11) and (12) express this in a linear form. The auxiliary binary variable q_t turns on either (11) or (12) and turns off the other one; $q_t = 1$ if $\sum_{p \in P} \sum_{n \in N} h_{n,p,t} \le \sum_{p \in P} \sum_{n \in N} g_{n,p,t}$ and $q_t = 0$ otherwise.

Constraint sets (13) and (14) work together to ensure that a specific partial availability variable in the next time period ($\mathrm{a}_{\mathrm{n,p,\ t+1}}$) takes the value 0, if the respective PROT in the current period ($\mathrm{y}_{\mathrm{n,p,\ t}}$) is used up (dropped to zero). The auxiliary binary variable $u_{n,p,\,t}$ takes 0 if $y_{n,p,\,t} > 0$. This is expressed in Constraint (13). If $y_{n,p,\,t} > 0$ and is used up ($y_{n,p,t} = \sum_{o\in O} x_{n,o,t}B_{n,o}$), the right-hand side of (14) becomes 0, which enforces $a_{n,p,\,t+1} = 0$. Constraints (15) and (16) have a similar mechanism. They ensure that $a_{n,p,\,t+1}$ becomes equal to 1 if the respective maintenance requirement is met in the current period ($g_{n,p,\,t} > 0$ and $g_{n,p,\,t} = h_{n,p,\,t}$). These four sets of constraints enforce the right value for $a_{n,p,\,t}$ only when the respective PROT is used up or the respective residual maintenance time is fulfilled. Constraints 17a) and ((18) ensure the following relationships always exist between $a_{n,p,\,t}$, $y_{n,p,\,t}$, and $g_{n,p,\,t}$, $\forall n,\, p,\, t$: $a_{n,p,t} = 0 \Leftrightarrow g_{n,p,t} > 0$ & $y_{n,p,t} = 0$, and $a_{n,p,t} = 1 \Leftrightarrow g_{n,p,t} = 0$ & $y_{n,p,t} > 0$. Constraint set (17b) ensures the auxiliary variable $\tilde{y}_{n,p,t}$ becomes positive and takes the remainder value of $y_{n,p,\,t}$ when the machine becomes unavailable and $y_{n,p,\,t}$ is equal to zero. Refer to Constraint sets (29–32).

Constraint set (19) ensures the sum of periodic operating times of a machine is not greater than the maximum operating time allowed in a period, given that the machine is *available*. Constraint sets (20) and (21) impose a lower bound on PROTs and residual maintenance times to prevent accuracy errors when implementing the model. Without these constraints, there is a chance that a machine is considered available/unavailable when PROT/residual maintenance time is negligible, yet positive. Constraint set (22) ensures the sum of operating times scheduled for a machine in each period is not greater than its AROT. Constraint set (23) forces the amount of time spent by the maintenance crew on a machine in each period ($h_{n,p,\,t}$) to be less than or equal to the time needed ($g_{n,p,\,t}$).

Constraint sets (24)–(26) fix the values of partial availabilities, PROTs, and residual maintenance time of machines at the beginning of the first period according to the initial state of the system. Constraint set (27) sets the actual availability state of a machine equal to zero if any of its partial availability states become zero. This means that if any of the PROTs of a machine becomes less than or equal to Y^{min}, the respective binary variable (partial availability state) also becomes zero and the machine should be stopped for maintenance. Constraint set (28) states that the AROT of a machine, which is to be maximized, is less than or equal to all its PROTs. Constraint sets (29–32) linearize the constraint set $y_{n,p,\,t} \geq a_{n,p,\,t}(1 - a_{n,p',\,t})Y_{p'}, n \in N,\; p \neq p',\; p \in P,\; t = 2, \ldots, T+1$ where p' is the index of the PM with the smallest time interval. This constraint has practical significance; when a machine is already in the maintenance station, all the PMs associated with the PROTs whose current values are smaller than the smallest PM interval ($Y_{p'}$) should also be performed. Note that the value $Y_{p'}$ can be replaced by any desired value, yet this value seems relevant.

With a few changes, the model can also handle those applications where the maintenance crew is not shared between multiple stations. This can be controlled by adding the index m to the variable $h_{n,p,\,t}$ and q_t, and parameter H_t. Constraint sets (10–12) also need to be modified as follows. We consider the original constraints in the rest of the paper. However, depending on the applications, the users can replace Constraints (10–12) with the following constraints.

$$\sum_{p\in P}\sum_{n\in N} h^m_{n,p,t} \leq H^m_t, \qquad t = 1, \ldots, T, \quad m \in M \tag{10b}$$

$$H^m_t \leq \sum_{p\in P}\sum_{n\in N} h^m_{n,p,t} + K(1 - q_{m,t}), \qquad t = 1, \ldots, T, \quad m \in M \tag{11b}$$

$$\sum_{p\in P}\sum_{n\in N} E_{n,m} g_{n,p,t} \leq \sum_{p\in P}\sum_{n\in N} h^m_{n,p,t} + K q_{m,t}, \qquad t = 1, \ldots, T, \quad m \in M \tag{12b}$$

The rest of the constraints (33–39) ensure that all variables are within their boundaries. In the next Section, the solution methodology is presented.

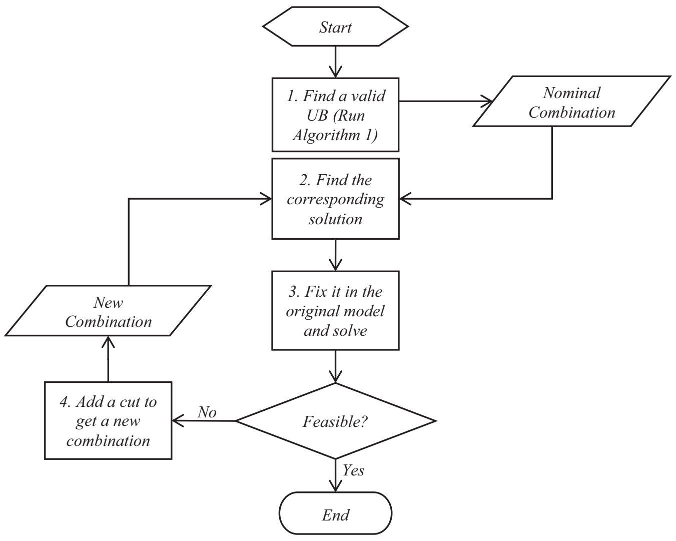
Start
1. Find a valid UB (Run Algorithm 1)
Nominal Combination
2. Find the corresponding solution
New Combination
3. Fix it in the original model and solve
4. Add a cut to get a new combination
No
Feasible?
Yes
End

Fig. 1. The outline of the solution methodology.

3. Solution methodology

Fig. 1 shows the solution methodology to find the optimal solution for the OMP problem. The steps of the solution methodology are as follows.

Step 1 – Find a valid upper bound (UB) by running Algorithm 1. Algorithm 1 has two outputs: the UB value for the objective function (1) and a combination of the machines that enter and exit the maintenance stations over the planning horizon (the nominal combination). This combination does not specify which machines enter/exit the maintenance stations in each time period. It only gives the number of machines that enter/exist the stations in each time period.

Step 2 – Find a solution that is consistent with the combination. The values of the decision variables, when considered together, can represent a specific combination. In this step we try to find these values such that they represent the combination that is found in the previous step.

Step 3 – Fix the solution obtained from Step 2 in the OMP problem defined by Constraint sets (2–32) to see if the solution is feasible. If the solution is feasible, it is also the optimal solution because it maximizes the objective function. If not, we need to find another combination that yields the same UB.

Step 4 – Cut the current combination and find a new one. In this step we solve a separate mixed integer program (MIP). The input of this MIP is the current combination and the output is a new combination. We generate new cuts until a new combination for which there exists a feasible solution is found.

These steps are discussed in details in the following subsections.

3.1. Finding a valid upper bound

Gavranis and Kozanidis (2015) discovered that availability of a unit of aircraft depends solely on the combination of the aircraft that enter and exit the maintenance station over the planning horizon, and the number of combinations is finite. They proved that the optimal policy for the maintenance station for FMP problems is first-in-first-out (FIFO) because a policy that maximizes the flow of aircraft entering and existing a maintenance station also maximizes the objective function value.

Algorithm 1 Finding an upper bound for the cumulative operations availability (COA) problem.

```
1 set $COA = T\sum_{n\in N}\min(Y^1_{n,p},\ \forall p\in P)$
2 set $C^{res}_m = C_m - unavilable\ machines\ of\ class\ m,\ \forall m\in M$
3 set $TR^{pln}_{o,0} = 0,\ TR^{act}_{o,0} = 0, \forall o\in O$
4 set $tlast_n = 1,\ \forall n\in N$
5
6 //ITERATIONS
7 for $t=1$ to $T$ do
8     $TR^{pln}_{o,t} = TR^{pln}_{o,(t-1)} + R_{o,t},\ \forall o\in O$
9     $TR^{act}_{o,t} = TR^{act}_{o,(t-1)},\ \forall o\in O$
10    $H^{res} = H_t$
11    set maintenanceLine as the list of unavailable machines across all maintenance stations
12    order maintenanceLine by whether all maintenance needs can be fulfilled in the current period (so they can exit the maintenance station); break the ties by reordering the machines in non-decreasing order of the machines' total residual maintenance time
13    //MAINTENANCE BLOCK
14    while there is at least one stopped machine in any of the stations (maintenanceLine has at least one member) & $H^{res} > 0$, do
15        select the first machine from maintenanceLine; let $q$ be the index of the selected machine, $q\in N$; let $m$ be the index of the machine's maintenance station, $E_{n,m}=1$; let $\bar{P}\subseteq P$ be the set of all PM types of machine $q$ if $g_{q,p,\ t} > G^{min},\ p\in\bar{P}$;
16        if $H^{res} - g_{q,p,\ t} \geq 0,\ \forall p\in\bar{P}$ then
17            set $H^{res} -= g_{q,p,\ t},\ \forall p\in\bar{P}$
18            set $g_{q,p,\ t} = 0,\ \forall p\in\bar{P}$
19            set $y_{q,p,\bar{t}} = Y_p,\ \ \forall p\in\bar{P},\ \ \bar{t} = t+1,\ldots,T+1$
20            set $ex_{m,(t+1)} += 1$
21            set $C^{res}_m += 1$
22            set $tlast_q = t+1$
23        else, then
24            for all $p\in\bar{P}$ do
25                $service_p = \min(H^{res}, g_{q,p,\ t})$
26                set $H^{res} -= service_p$
27                set $g_{q,p,\ t} -= service_p$
28        remove q from maintenanceLine
29        order maintenanceLine by whether all maintenance needs can be fulfilled in the current period (so they can exit the maintenance station); break the tie by reordering the machines in non-decreasing order of machines' total residual maintenance times
30    for all $n\in N$ do
31        if $tlast_n = t+1$ then
32            set $COA += (T-t+1)Y_{p'}$
33
34    //OPERATIONS BLOCK
35    set operationLine as the list of all available machines (that can be used for different operations)
36    order operationLine in non-decreasing order of AROTs (min. of machines' residual operation time)
37    while operationLine has at least one machine whose respective maintenance station has at least one empty spot do
38        select the first machine in operationLine; let $q$ be the index of the selected machine; let $m$ be the index of $q$'s respective maintenance station
39        if $tlast_q > 1$ then
40            set $ylast_q = \min\{y_{q,p,\ t} | \forall p\in P\}$
41        else if $tlast_q = 1$ then
42            set $ylast_q = Y_{p'}$
43        define $\bar{O}\subseteq O$ as the set of all the operations that machines of class $m$ can perform
44        set $\alpha = \sum_{o\in\bar{O}} TR^{act}_{o,t}$
45        set $\beta = \sum_{o\in\bar{O}} TR^{pln}_{o,t}$
46        if $ylast_q \leq X^{max}(t - tlast_q + 1)$ & $ylast_q + \alpha \leq \beta$ then
47            for all $o\in\bar{O}$ do
48                if $ylast_q > 0$ then
49                    set $operation = \min(ylast_q, TR^{pln}_{o,t} - TR^{act}_{o,t})$
50                    set $TR^{act}_{o,t} += operation$
51                    set $ylast_q -= operation$
52                    set $y_{q,p,\bar{t}} -= operation,\ \forall p\in P\ ,\bar{t} = t+1,\ldots,T+1$
53                else, then
54                    break
55        for all $p\in P$ do
56            if $y_{q,p,\ t+1} < Y^{min}$ or $(y_{q,p',\ t+1} < Y^{min}$ and $y_{q,p,\ t+1} < Y_{p'})$ then
57                set $g_{q,p,\bar{t}} = G_{q,p},\ \bar{t} = t+1,\ldots,T+1$
58                set $y_{q,p,\bar{t}} = 0,\ \bar{t} = t+1,\ldots,T+1$
60        remove machine q from operationLine
61        set $en_{m,t} += 1$
62        set $C^{res}_m -= 1$
63        if $C^{res}_m = 0$ then
64            for all $n\in$ operationLine do
65                if $E_{n,m} = 1$ then
66                    remove $n$ from operationLine
67    set $COA -= \sum_{o\in O}(T-t+1)R_{o,t}$
```

For OMP problems where machines require multiple PMs, FIFO is not the optimal policy. A policy that maximizes shop flow is optimal except when multiple operations are required. The upper bound FMP algorithm was redesigned for OMP. Considerations in redesigning the UB algorithm for FMP are explained, followed by a description of the redesigned algorithm.

3.1.1. *Considerations in designing the OMP algorithm*

In FMP, there is only one residual flight time for every aircraft (only one single PM), while in OMP, there are multiple partial residual operation times (PROTs) with an actual residual operation time (AROT, equivalent to the residual flight time) equal to their minimum. When there are multiple PMs, a machine is stopped when *any* of its PROTs is equal to zero.

Since machines can have more than one PM, a new machine entering one of the maintenance stations may have a total maintenance duration less than machines already in the maintenance station. This is because the new machine might have one short PM due while a machine already in the maintenance station might have multiple long PMs due. Consequently, the idea that the first machine that enters the maintenance station has the least residual maintenance time is not true. If the machine with the lowest “sum of maintenance residual times across all maintenance stations” exits first, the maximum flow will be retained and the result will be the same as the FIFO policy for FMP.

In contrast to FMO, the residual operating time (AROT) is not necessarily equal to a single minimal value in OMP; rather, it is equal to the minimum of its PROTs. Some PROTs are reset after exiting the maintenance station, others may be smaller than the ones reset if no maintenance work was performed in the station; e.g., were not equal to zero when the machine was stopped. This will affect the value of the objective function. Constraint sets (29–32) deal with this problem; the AROT is always $Y_{p'}$ for a machine after exiting the maintenance station. Therefore, the best policy is to first work on the machine with the least sum of maintenance work without worrying about the availability of the aircraft when it exits the station. Then, the objective becomes maximizing the total flow of machines into and out of the maintenance stations.

Another consideration is that different maintenance stations have different capacities. The maintenance crew available time is not dependent on, or related to, any maintenance station; e.g., they can work at any station, so crew resources do not affect the policy. However, machines must frequent only the proper maintenance station, i.e., the maintenance station to which the machines belong. Similarly, the available capacity of each maintenance station should be affected by only the machines it can accommodate.

In OMP, where multiple operations exist, it is not necessarily optimal to work on a machine with the minimum maintenance work. Consider two machines A and B; the maintenance work needed for machine A is less than machine B. Machines A and B have different capabilities, so their operational load is also different. The operations machine A can resume upon release from a maintenance station is less than its actual residual time; however, machine B operations are greater than its AROT. As a result, working on machine A first is suboptimal because if machine B is worked first it will return to the station and hence, the flow will be higher. This *can* lead to slight sub-optimality in cases where different operations requirements are not similarly distributed across the planning horizon. The degree to which this sub-optimality can occur is assessed in Section 4.

3.1.2. *The algorithm*

Constraint set (8) postulates distribution of operating periods among operational requirements is unimportant. If operational requirements are met, it does not matter which machines were used to complete an operation. If the maintenance workforce completes the PMs for the machine with the lowest *total* residual maintenance time first, maximum COA (value of the objective function) will be attained. Any other policy leads to suboptimal solutions since interrupting work during a PM delays its return to service. The object is to return a machine to service as soon as possible; consequently, maintenance work should commence on a machine with the lowest total residual maintenance time.

Although the FIFO policy was shown to be optimal by Gavranis and Kozanidis (2015), it does not work here. However, the flow of machines into and out of the maintenance stations can be maximized by adapting an algorithm that uses a similar approach. Let $en_{m,t}$ and $ex_{m,t}$ be the number of machines of class m that enter and exit maintenance station m at the beginning of time period t, and $cx_t = \sum_{m\in M}\sum_{k=2}^{t} ex_{m,k}$ is the cumulative number of all machines exiting the maintenance stations up to the period t where $t = 1, \ldots, T+1$. The algorithm will prove that a policy that continuously serves a machine with the lowest total residual maintenance time until all its PM services are complete leads to maximum availability of machines regardless of the machine class.

Proposition 1. *Maximization of COA is equivalent to maximizing* $\sum_{t=2}^{T+1} cx_t$.

Proof. For time period t in the planning horizon, total AROT is equal to the initial residual time per operation minus the time to fulfill the operational requirement plus the residual time of machines that have exited maintenance stations, namely $T\sum_{n\in N} Z_{n,1} - \sum_{o\in O}\sum_{k=1}^{t-1} R_{o,k} + \sum_{m\in M}\sum_{k=2}^{t} Y_{p'} ex_{m,k}$ where $Y_{p'}$ is the minimum of PM intervals, $p' \in P$. As a result, the objective function (COA), after considering all the time periods, is equal to $\sum_{t=2}^{T+1}(\sum_{o\in O}\sum_{n\in N} Z_{n,1} - \sum_{k=1}^{t-1} R_{o,t} + \sum_{m\in M}\sum_{k=2}^{t} Y_{p'} ex_{m,k}) = T\sum_{n\in N} Z_{n,1} - \sum_{o\in O}\sum_{t=1}^{T}(T-t+1)R_{o,t} + \sum_{t=1}^{T}\sum_{m\in M}(T-t+1)Y_{p'} ex_{m,t+1}$, where the only variable is $ex_{m,t}$. The last term can be rewritten as $\sum_{m\in M}(TY_{p'} ex_{m,2} + (T-1)Y_{p'} ex_{m,3} + (T-2)Y_{p'} ex_{m,4} + \ldots + Y_{p'} ex_{m,T+1}) = Y_{p'}\sum_{m\in M}(Tex_{m,2} + (T-1)ex_{m,3} + (T-2)ex_{m,4} + \ldots + ex_{m,T+1})$. Therefore, maximizing the objective function of the OMP problem is equivalent to maximizing the expression $\sum_{m\in M}(Tex_{m,2} + (T-1)ex_{m,3} + (T-2)ex_{m,4} + \ldots + ex_{m,T+1}) = \sum_{t=1}^{T}\sum_{m\in M}(T-t+1)ex_{m,t+1}$. Because $\sum_{t=2}^{T+1} cx_t = \sum_{t=2}^{T+1}(\sum_{m\in M}\sum_{k=2}^{t} ex_{m,k}) = \sum_{t=1}^{T}\sum_{m\in M}(T-t+1)ex_{m,t+1}$, the validity of the Proposition 1 is established.

Proposition 2. *The optimal policy for maximizing COA is one that not only maximizes the number of machines exiting the maintenance stations* ($\sum_{t=2}^{T+1} cx_t$)*, but also maximizes the number of machines entering the maintenance stations* ($\sum_{t=2}^{T+1}\sum_{m\in M} en_{m,t}$).

Proof. The number of machines that exit the maintenance stations depends on the number of machines that enter the stations. Therefore, maximizing the number of machines that enter the stations maximizes the value of COA obtained from a policy that maximizes the number of machines that exit the stations.

From Propositions 1 and 2, the following conclusions can be drawn.

Corollary 1. *For a special case where* $|O| = 1$*, the policy in assignment of the machines to operations is to always use first the machine whose AROT is the minimum because this maximizes the flow of machines into the maintenance stations.*

Corollary 2. *When* $|O| > 1$*, the policy in assignment of the machines to operations is to always use first the machine whose AROT can be depleted first, considering that each machine can take only certain types of operations; the demand that one machine can satisfy might be different from another machine.*

Corollary 3. For a special case where $|O| = 1$, $|P| = 1$, $|M| = 1$, and $G_{n,p} = G'$, $\forall n \in N$, the optimal maintenance policy is a first-in-first-out (FIFO) policy in which the machine that comes first to the maintenance stations is always serviced first continuously until the service is completed and the machine is ready to be ready for operations. In this case all machines have only a single maintenance activity ($|P| = 1$) and the duration of that activity is the same for all machines (G'). Therefore, if the machine that enters the station is served first and is worked on continuously until the service is completed, it is guaranteed that a machine with the minimum maintenance time is worked on first which is the optimal approach for increasing the number of machines that exit the station. Note that this case is the only case that Gavranis and Kozanidis (2015) consider.

Corollary 4. For a special case where $|O| = 1$, and $|M| = 1$, the optimal maintenance policy is not a first-in-first-out (FIFO) policy because the machine that enters the maintenance station first, does not necessarily have the minimum maintenance time. This is due to having multiple maintenance activities ($|P| \geq 1$) and having different maintenance durations for different machines ($G_{n,p}$, $\forall p \in P$). In this case the optimal policy is one that continuously works on the machine that can be released sooner.

Corollary 5. For a special case where $|O| = 1$, the optimal maintenance policy is the one that continuously works on the machine that can be released sooner across all maintenance stations ($|M| \geq 1$).

Corollary 6. When $|O| > 1$, the optimal maintenance policy is *not necessarily* the one that continuously works on the machine that can be released sooner than all the other machines across all maintenance stations, and the reason for this is as follows. The requirements for different operations are not necessarily distributed identically over the planning horizon and the machines are not necessarily identical in terms of the types of operations they can take. Therefore, because increasing the flow into the maintenance stations is also important in optimality (Proposition 2), servicing the machine that can exit sooner does not necessarily increase the overall flow. There might exist a machine that can exist later or at the same time, but has more capabilities in satisfying the operations requirements and can be stopped sooner after being released from a maintenance station.

The policy in Corollary 5 guarantees the maximum value for $\sum_{t=2}^{T+1} cx_t$ when $|O| = 1$. While there may be another policy or solution that yields the maximum value, this policy ensures the maximum value is never overlooked.

Corollary 6 and Proposition 2 state that in order to maximize the number of machines exiting the maintenance stations, the number of machines entering the stations should be also maximized. The goal is to maximize the flow of machines into and out of the maintenance stations. In each period, depending on the value of the unsatisfied operation requirements, a machine that can be stopped for maintenance as soon as possible will be used to satisfy the operational requirements so that flow is maximized.

Algorithm 1 shows a procedure that finds a valid upper bound on the optimal COA when $|O| = 1$. As will be shown in the next section, when $|O| > 1$, the algorithm yields either the optimal value or a near optimal value for COA. To increase the speed of the algorithm, not all the constraints are considered. The combination of the machines that enter and exit the maintenance stations over the planning horizon and yield the upper bound is not necessarily feasible. However, there will be a feasible combination that yields the same bound. Expressions $a += b$ and $a -= b$ used in the pseudo code indicate adding b to the current value of a and subtracting b from it, respectively. The following notations are used.

C_m^{res}	Available number of empty spots in maintenance station m (residual capacity),
H^{res}	Available time of maintenance crew (residual time capacity),
$TR_{o,t}^{pln}$	Cumulative operational load (planned) required for operation type $o \in O$ up to time period t (by convention, $TR_{o,0}^{pln} = 0,\ \forall o \in O$),
$TR_{o,t}^{act}$	Cumulative operational load (actual) fulfilled for operation type $o \in O$ up to time period t (by convention, $TR_{o,0}^{pln} = 0,\ \forall o \in O$),
$tlast_n$	Last period that machine $n \in N$ exited its maintenance station (by convention, $tlast_n = 1$ if machine n has never exited its maintenance station),
$ylast_n$	Auxiliary variable that is set equal to $Y_{p'}$ if $tlast_n > 1$, and $\min Y_{n,p}^1,\ \forall p \in P$ otherwise.

Algorithm 1 has two main blocks in each period; 1) Lines 14–33, performing maintenance on the unavailable machines in such a way that more machines exit the maintenance stations and, 2) Lines 34–66, assigning operational requirements to the available machines with the least residual operation time to maximize flow of machines into and out of the maintenance stations.

The value of objective function (COA) is calculated by, 1) considering the residual operating time for all machines in the first period for each period in the planning horizon (Line 1); and, 2) adding the residual operating time of each machine exiting the maintenance stations in the next time period (Line 31) to the value of COA per time period in the remaining time periods in the planning horizon (Line 32); and, 3) deducting all cumulative operating requirements from COA (Line 67).

The maintenance block (Lines 14–33) is formed by a set of all machines in the maintenance. The set is sorted by checking whether a machine can be fully served in the current period so it can exit. Ties are broken by resorting the machines based on the total time they need for maintenance. The algorithm checks each machine to see what can be done to it (Lines 15–27), it is removed from the set and the set is resorted (Line 35) before selecting the next machine. This is based on the assumption that the previous machine impacted maintenance resources.

Line 16 checks whether all PMs of the selected machine can be fully performed in the current period; if yes, the machine exits the maintenance station and the value of variables such as residual time and station capacity are updated. If the machine cannot be checked out from the station at the current period (Lines 24–27), as much maintenance work as possible will be performed on the machine. Note that when the first machine in the list cannot be fully served, there is no other machine in the list that can be fully served. Therefore, using resources for partially serving a machine while another machine waiting in the line can be fully served is not a concern.

The operations block (Lines 34–66) considers all available machines as a set or list (Line 35). After the machines are sorted by their AROT, the algorithm checks each machine in the sorted list starting from the first one. Lines 39–46 check whether the machine can be sent to the appropriate maintenance station in the next period. If both conditions in Line 46 are true, there are enough maintenance tasks for this machine so the minimum of its PROTs is fully used. Whether or not the machine passes these conditions, it will be removed from the set and the algorithm checks the next machine. This procedure continues until there are no machines left in the list. Constraint sets (29–32) are implemented in Lines 57–58.

Before checking the first machine in the list (*operationLine*), the algorithm checks for all machines where there is an empty spot in the required maintenance station. If one is not available, the machine is removed from the list since they cannot be stopped in the next period. This justifies why the maintenance blocks comes before the operations block.

Proposition 3. Algorithm 1 provides a valid upper bound for the objective function of the OMP problem when $|O|=1$.

Proof. Corollaries 1–5 proved that the optimal policy for the OMP problem is that machine that can be released soonest across all the maintenance stations is served first. The premise of that policy is to maximize the flow of machines into and out of the maintenance stations. Algorithm 1, in each period $t=2, \ldots, T+1$, computes an upper bound for the number of machines that can enter maintenance stations from the first period up to the current period to maximize flow into the maintenance stations. It also insures the number of machines sent to the maintenance stations is maximized. As a result, the actual number of machines entering the maintenance stations in the first t periods cannot be greater than what the algorithm returns. Therefore, the *COA* computed in Algorithm 1 is a valid upper bound for the OMP problem.

Algorithm 1 returns the upper bound and the number of machines that enter or exit each maintenance station in each period, called a machine *combination*. Several or no solutions for the OMP problem may realize a combination. If there is no solution realizes a combination, the combination is infeasible and needs to be excluded from further consideration. Therefore, a check for feasible combinations must be performed, and constraints added to eliminate infeasible combinations. The combination found first by the algorithm is called *the nominal combination.*

3.2. Feasibility check

Feasibility check of a combination is performed by fixing the value of decision variables that represent a combination in the original problem and trying to solve it to see if a solution that realizes that combination exists. When integer variables such as $a_{n,p,\,t}$, $d_{n,p,\,t}$, $f_{n,p,\,t}$, etc. are fixed, the original problem is very easy to solve and the feasibility check is quick.

3.3. Adding cuts

If an infeasible combination is found, another combination that yields the current bound (COA) for the objective function must be sought. Constraints that exclude the infeasible combination and find a new combination are added. The variables $en_{m,t}$ and $ex_{m,t}$ are augmented with a new superscript to show whether they represent the nominal combination (*nom*), current combination (*cur*), or the new combination (*new*) by generalizing the method that Gavranis and Kozanidis (2015) used. *nom* and *cur* are associated with known values for the variables (parameters of the following formulation).

Only $en^{new}_{m,t}$ and $ex^{new}_{m,t}$ represent variables. Unlike FMP, there can be more than one maintenance station ($|M| \geq 1$) depending on the number of classes of the machines. This does not affect the method used in FMP in a negative way, as feasibility of any combination is ultimately checked through the original problem. The following is a mixed-integer formulation that excludes the current combination and finds a new one. The objective function of this formulation can be arbitrarily chosen.

$$\sum_{m\in M}\sum_{t=1}^{T} Y_{p'}(T-t+1)ex^{new}_{m,t+1} = COA^* - T\sum_{n\in N}\min\left(Y^1_{n,p},\ \forall p\in P\right) + \sum_{t=1}^{T}\sum_{o\in O}(T-t+1)R_{o,t} \quad (40)$$

$$\sum_{k=2}^{t} ex^{new}_{m,k} \leq \sum_{k=2}^{t} ex^{nom}_{m,k} \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (41)$$

$$\sum_{k=2}^{t} en^{new}_{m,k} \leq \sum_{k=2}^{t} en^{nom}_{m,k}, \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (42)$$

$$st^{new}_{m,t} = st^{new}_{m,t-1} + en^{new}_{m,t} - ex^{new}_{m,t} \quad \forall m\in M,\ t=2,\ldots,T+1 \quad (43)$$

$$st^{new}_{m,t} \leq C_m \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (44)$$

$$ex^{new}_{m,t} \leq st^{new}_{m,t-1} \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (45)$$

$$\sum_{m\in M}\left(\sum_{t=2}^{T+1}\left|en^{cur}_{m,t}-en^{new}_{m,t}\right| + \sum_{t=2}^{T+1}\left|ex^{cur}_{m,t}-ex^{new}_{m,t}\right|\right) \geq 1, \quad (46)$$

$$en^{new}_{m,t},\ ex^{new}_{m,t},\ st^{new}_{m,t} \in \mathbb{Z} \geq 0 \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (47)$$

Constraint set (47) infers the only decision variables of this mixed-integer linear program are $en^{new}_{m,t}$, $ex^{new}_{m,t}$, and $st^{new}_{m,t}$, all of which determine the new combination to be found. The rest are known parameters from previous combinations. $st^{new}_{m,t}$ is the number of stopped machines in station m in time period t. Constraint (40) ensures the current level for the objective function (COA^*), associated with the current combination, is obtained by the new combination. Constraint sets (41) and (42) set an upper bound for the cumulative number of machines that can exit and enter each maintenance station in each period, and is equal to the value associated with the nominal combination. The number of stopped machines held in each maintenance station in each period is equal to the same number in the previous period, plus the number of machines entering the station in the current period, minus the number of machines that exited, as shown in Constraint set (43). Constraint sets (44) and (45) ensure the number of machines held in each station do not exceed the capacity of the station, and the number of machines exiting a station is not greater than the number of machines held in it. Constraint (46) cuts the current combination by ensuring that at least one of the variables $en^{new}_{m,t}$ or $ex^{new}_{m,t}$, $\forall m\in M, t=2,\ldots,T+1$ takes a value different from the corresponding parameter in the current combination ($en^{cur}_{m,t}$, $ex^{cur}_{m,t}$). This constraint is nonlinear and can be replaced by the following linearized constraints:

$$en^{new}_{m,t} - en^{cur}_{m,t} \leq z^{en}_{m,t} \leq en^{new}_{m,t} - en^{cur}_{m,t} + Ku^{en}_{m,t}, \qquad \forall m\in M,\ t=2,\ldots,T+1 \quad (48)$$

$$en^{cur}_{m,t} - en^{new}_{m,t} \leq z^{en}_{m,t} \leq en^{cur}_{m,t} - en^{new}_{m,t} + K\left(1-u^{en}_{m,t}\right), \quad \forall m\in M, t=2,\ldots,T+1 \quad (49)$$

$$ex^{new}_{m,t} - ex^{cur}_{m,t} \leq z^{ex}_{m,t} \leq ex^{new}_{m,t} - ex^{cur}_{m,t} + Ku^{ex}_{m,t}, \qquad \forall m\in M, t=2,\ldots,T+1 \quad (50)$$

$$ex^{cur}_{m,t} - ex^{new}_{m,t} \leq z^{ex}_{m,t} \leq ex^{cur}_{m,t} - ex^{new}_{m,t} + K\left(1-u^{ex}_{m,t}\right), \quad \forall m\in M, t=2,\ldots,T+1 \quad (51)$$

$$\sum_{m\in M}\sum_{t=2}^{T} z^{en}_{m,t} + z^{ex}_{m,t} \geq 1, \quad (52)$$

$$u^{en}_{m,t}, u^{ex}_{m,t} \in \{0,1\}, \quad (53)$$

$$z^{en}_{m,t}, z^{ex}_{m,t} \geq 0, \tag{54}$$

With this formulation, $z^{en}_{m,t}$ and $z^{en}_{m,t}$ will be equal to $|en^{cur}_{m,t} - en^{new}_{m,t}|$ and $|ex^{cur}_{m,t} - ex^{new}_{m,t}|$, respectively, and $u^{en}_{m,t}$ and $u^{ex}_{m,t}$ are auxiliary variables that make one of the pair constraints (48,49) and (50,51) redundant, while preserving the linearity of the constraints.

4. Computational experiments

In this section, the computational experiments for the problem variations that evaluate the performance of the solution methodology are presented. The solution methodology was implemented in Python and is compared against Gurobi (2017) as a popular commercial solver. In military applications such as the Air Force, 60–80 aircraft are typically scheduled with as many as 100 in special cases (Gavranis and Kozanidis, 2015). Fleet size of commercial airlines is significantly larger; i.e., 300–1600 for the largest airlines (Falcus, 2016). The number of machines in construction projects has similar sized equipment fleets, based on the authors' observation. As was also shown by Gavranis and Kozanidis (2015), commercial solvers cannot handle problems with more than about 50 aircraft, in a reasonable time. Since this problem is a generalization of theirs, this limitation is even more rigid. The experiments were performed on a computer cluster made up of 11 Dell DSS 1500 servers, each containing 2 Intel Xeon E5-2630 8-core processors running at 2.4 GHz. Each server has 32GB of RAM. There are a total of 176 processor cores and 352GB of RAM. By utilizing high-performance CPU and memory for the computations, it is possible to test larger numbers. Nevertheless, a 30 min timeout for Gurobi and the algorithm is considered a *reasonable time*.

The experiments are designed for $|N| = 6, 12, 25, 50,$ and 100 machines (aircraft). The number of machines, $|N|$, is the only parameter that determines the problem size. In the computational experiments, we only increased the number of machines because in practice, other parameters such as the length of the planning horizon, number of PMs, or the number of operations do not vary significantly. For example, the planning horizon is usually 6 months and the number of PMs is around 3 (A, B, and C checks) in flight scheduling. We used three different PM intervals in order to cover a wide range of scenarios. Since the solver could not accommodate problems of a larger size, the performance of the algorithm for challenging problems with up to 1000 machines (aircraft) was also tested. In all of the following computational experiments, 30 random test problems were solved for each problem size.

For generating the parameters of the random test problems, the method used by Gavranis and Kozanidis (2015) was used so the test problems were as realistic as possible. C_m, the capacity of maintenance station m, was set equal to $0.15|N_m|$ and rounded up to the nearest integer. The number of inoperable machines in the station m is a discrete random variable uniformly distributed in the interval $[0, C_m]$.

Three PMs were considered for all problem instances with time intervals of $Y_1 = 125,\ Y_2 = 300,\ Y_3 = 500$ h. The initial residual operating time of the available machines for to each PM ($Y^1_{n,p}$) are random numbers uniformly distributed in the interval $[Y^{min}, Y_p]$. For the unavailable machines in the stations, the initial residual maintenance time for to each PM was generated in the interval $[G^{min}, G_{n,p}]$, given that the residual operation time is not completely depleted. A uniform random integer in the fixed interval [20, 40] was generated as the demand for operation o in period t ($R_{o,t}$). We used a fixed interval because in practice, having more facilities does not lead to having more demand. Each machines was randomly assigned to one (and only one) of the maintenance stations. The capability state of each machine in each operation type

Table 1
Comparison of the computational times (in seconds) when $|O| = 1$.

$\|N\|$	Gurobi			Proposed algorithm		
	Min.	Max.	Avg.	Min.	Max.	Avg.
6	0.17	1.33	0.56	0.02	0.09	0.03
12	0.67	4.53	1.66	0.05	0.15	0.08
25	4.53	102.56	23.09	0.17	1.71	0.46
50	5.97	1675.86	234.05	0.24	0.54	0.34
100				0.80	46.50	16.10

($B_{n,o}$) was randomly determined. The rest of the parameters were set as follows. $T = 6$, $X^{max} = 60$, $G^{min} = 0.1$, $Y^{min} = 0.1$, $H_t = [329, 348, 354, 344, 326, 335]$.

Because the total capacity of the fleet is influenced by their residual operating times and these values are taken randomly from the range $[Y^{min}, Y_p]$, infinite number of possible scenarios (in terms of how far or close the capacity is to the demand) can happen when generating a test problem. Indeed this is the very reason we generated 30 random test problems. If the total capacity is less than the total demand, the problem is going to be infeasible because it has been assumed that the capacity is enough to satisfy the demand: there exists at least one feasible solution. When generating the test problems for each problem size, infeasible test problems were discarded and more test problems were generated until 30 feasible test problems were found. Having infeasible test problems generated for each problem size ensures that computationally challenging scenarios are being covered: tight problem instances in which capacity and demand are very close.

Recall that the FMP problem was generalized for the OMP by introducing multiple PMs, multiple maintenance stations, and multiple operations with multi-purpose machines. As was stated in Corollary 6, when there are multiple operations involved, a slightly sub-optimal solution may result. Table 1 contrasts the performance of the solution methodology with Gurobi, for the case of multiple PMs, multiple stations, but only one operation. Because the algorithm found the optimal solution for the test problems, the gap (0.00%) is not reported. Average, minimum, and maximum computational time for Gurobi and the proposed solution algorithm are presented in Table 1. When $|N| = 100$ the solver could not solve the test problem within the reasonable time. As the results show, proposed algorithm finds the optimal solution very efficiently.

Table 2 contrasts the performance of the proposed algorithm with that of Gurobi for fully generalized problem instances where three types of operations for each machine exists in the planning horizon. Minimum, maximum, and average computational times of Gurobi and the proposed algorithm are shown. Since the algorithm can give suboptimal solutions, the gap between the exact optimal value of the objective function obtained from Gurobi and the best value found by the algorithm is also shown.

When multiple operations are involved, there is a possibility that the machine that satisfies the condition of Line 46 in the upper-bound algorithm can be assigned to operations in different ways. The goal of this assignment is to deplete the residual operating time of the machine so the machine can be stopped as soon as possible. This can lead to suboptimal solutions because all the possible assignments cannot be exhausted in a continuous domain via the proposed algorithm. In cases of sub-optimal solutions, another assignment may lead to selecting a machine that can be stopped for maintenance sooner than a machine originally selected. Therefore, the algorithm can be improved when multiple operations are involved. However, results show the proposed algorithm performs satisfactorily even when multiple operations are involved. Note that since the results of Gurobi could not be obtained within the time limit for the last two rows of Table 2, it

Table 2
Comparison of the computational times (in seconds) and quality of solutions when $|O| > 1$.

$|N|$	Time						Gap			
	Gurobi			Proposed algorithm			Distance to optimality			Times optimal solution was found
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	
6	0.18	1.35	0.63	0.03	1.94	0.20	0.00%	5.27%	3.01%	30.77%
12	1.12	7.28	2.47	0.04	1.83	0.18	0.00%	6.66%	1.57%	46.94%
25	3.21	152.93	26.12	0.10	4.42	0.38	0.00%	3.86%	0.81%	53.33%
50				0.14	0.52	0.27				
100				1.05	2.21	1.44				

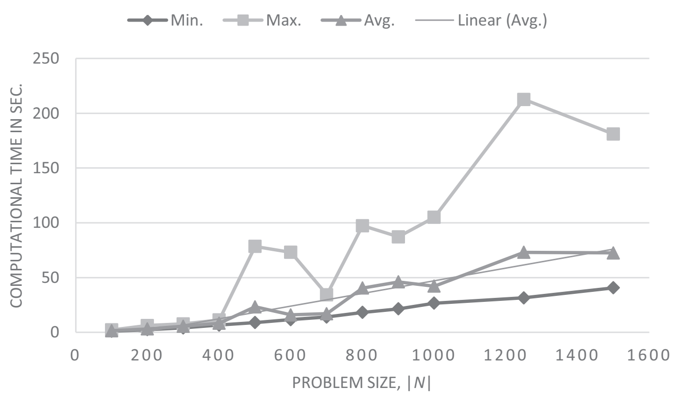

Fig. 2. Computational times of the proposed algorithm for challenging problem sizes.

was not possible to compare the results of the proposed algorithm with Gurobi. The values reported in those two lines are without considering the cuts. One observation is that both the distance to optimality and the number of times that the optimal solution is found (distance to optimality = 0.00%) improves as the number of machines (fleet size) increases.

Since Gurobi fails to solve the problems within the reasonable time when the number of machines $|N| \geq 50$ a series of experiments where the number of machines is more challenging and close to real-world scenarios, i.e. $|N| \in [100, 1500]$, were performed to examine the computational time of the algorithm. Fig. 2 shows the results of this experiment. The results show the algorithm is fast and the average computational time increases linearly with the number of machines.

5. Conclusion

In this paper, formulation of an existing flight and maintenance planning problem (FMP) was extended to an operations and maintenance planning (OMP) problem where multiple PMs, multiple maintenance stations, and multiple operations exist. The exact solution algorithm previously introduced for FMP was also extended to OMP. The results show that when there are multiple operations with dissimilar requirements distribution across the planning horizon, the extended algorithm can lead to sub-optimal solutions. However, the results proved that the quality of the solutions were still satisfactory. Performance of the extended algorithm was also tested for large sizes of the problem where up to 1500 machines are involved in planning. The results illustrated the efficiency of the algorithm. The presented model and its solution method do not consider unexpected changes in demand or unexpected failures. Because the solution method is relatively fast, it is possible to handle unexpected changes in the original operations and maintenance requirements by updating the input data and solving the problem again. This can be implemented as a recurring process where the model and the solution method are applied. For some cases where random failures and emergency maintenance are frequent, the presented model can be improved by turning it to a stochastic program.

This work can be extended in several other directions as follows.

- The model and the solution algorithm can be further generalized by considering maintenance workforce planning;
- unplanned maintenance and random failures can be incorporated in a stochastic version of the model;
- the solution algorithm can be improved for the special case where multiple operations are involved; and;
- since different operations have different priorities, an alternative objective function, weighted cumulative operation availability (WCOA), can assign weights to each operation. For WCOA, unless the operations are equally critical, the solution methodology presented in this paper will be sub-optimal.

Each of these extensions can make the model and/or the solution methodology very challenging. Implementation of this work in different applications such as in the OMP of USCG missions along with numerical analysis can also be of high value.

Acknowledgment

The authors are grateful to the referees for their constructive inputs and many useful suggestions. We would also like to thank Dr. Janice Tolk for improving the manuscript.

References

Abdelghany, A., Abdelghany, K., Azadian, F., 2017. Airline flight schedule planning under competition. Comput. Oper. Res. 87, 20–39. https://doi.org/10.1016/j.cor.2017.05.013.
Aramon Bajestani, M., Beck, J.C., 2013. Scheduling a dynamic aircraft repair shop with limited repair resources. J. Artif. Intell. Res. 47, 35–70.
Boere, N., 1977. Air Canada saves with aircraft maintenance scheduling. Interfaces 7 (3), 1–13.
Brown, G.G., Goodman, C.E., Wood, R.K., 1990. Annual scheduling of Atlantic Fleet naval combatants. Oper. Res. 38 (2), 249–259.
Brucker, P., Schlie, R., 1990. Job-shop scheduling with multi-purpose machines. Computing 45 (4), 369–375.
Caterpillar. (2010a). 725 and 730 OEM articulated trucks - maintenance intervals operation & maintenance manual excerpt Retrieved from https://safety.cat.com.
Caterpillar. (2010b). 988 wheel loader - maintenance intervals operation & maintenance manual excerpt Retrieved from https://safety.cat.com.
Caterpillar. (2010c). M312 and M315 excavators - maintenance intervals operation & maintenance manual excerpt Retrieved from https://safety.cat.com.
Cho, P.Y., 2011. Optimal Scheduling of Fighter Aircraft Maintenance. Massachusetts Institute of Technology.
Falcus, M. (2016). Top 10 world's largest airlines. Retrieved from http://www.airportspotting.com/top-10-worlds-largest-airlines/.
Feo, T.A., Bard, J.F., 1989. Flight scheduling and maintenance base planning. Manage. Sci. 35 (12), 1415–1432.
Gavranis, A., Kozanidis, G., 2015. An exact solution algorithm for maximizing the fleet availability of a unit of aircraft subject to flight and maintenance requirements. Eur. J. Oper. Res. 242 (2), 631–643. http://dx.doi.org/10.1016/j.ejor.2014.10.016.
Gurobi. (2017). Gurobi 7.0.2. Retrieved from http://www.gurobi.com/downloads/gurobi-optimizer.
Hahn, R.A., Newman, A.M., 2008. Scheduling United States coast guard helicopter deployment and maintenance at clearwater air station, Florida. Comput. Oper. Res. 35 (6), 1829–1843. http://dx.doi.org/10.1016/j.cor.2006.09.015.

Maher, S.J., Desaulniers, G., Soumis, F., 2014. Recoverable robust single day aircraft maintenance routing problem. Comput. Oper. Res. 51, 130–145. http://dx.doi.org/10.1016/j.cor.2014.03.007.

Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., Chryssolouris, G., 2010. An approach to operational aircraft maintenance planning. Decis. Support Syst. 48 (4), 604–612.

Rezg, N., Dellagi, S., Khatad, A., 2014. Joint Optimization of Maintenance and Production Policies. John Wiley & Sons.

Seif, J., Yu, A.J., Rahmanniyay, F., 2017. Modelling and optimization of a bi-objective flow shop scheduling with diverse maintenance requirements. Int. J. Prod. Res. 1–22.

Sriram, C., Haghani, A., 2003. An optimization model for aircraft maintenance scheduling and re-assignment. Transp. Res. Part A 37 (1), 29–48.

Talluri, K.T., 1998. The four-day aircraft maintenance routing problem. Transp. Sci. 32 (1), 43–53.

Yao, X., Xie, X., Fu, M.C., Marcus, S.I., 2005. Optimal joint preventive maintenance and production policies. Naval Res. Logistics (NRL) 52 (7), 668–681.

Yu, A.J., Seif, J., 2016. Minimizing tardiness and maintenance costs in flow shop scheduling by a lower-bound-based GA. Comput. Indus. Eng. 97, 26–40.