



Scheduling the repair and replacement of individual components in operating systems: a bi-objective mathematical optimization model

Downloaded from: <https://research.chalmers.se>, 2024-05-01 18:41 UTC

Citation for the original published paper (version of record):

Obradovic, G., Strömberg, A., Lundberg, K. (2024). Scheduling the repair and replacement of individual components in operating systems: a bi-objective mathematical optimization model. *Journal of Scheduling*, 27(1): 87-101.
<http://dx.doi.org/10.1007/s10951-023-00800-x>

N.B. When citing this work, cite the original published paper.



Scheduling the repair and replacement of individual components in operating systems: a bi-objective mathematical optimization model

Gabrijela Obradović¹ · Ann-Brith Strömberg¹ · Kristian Lundberg²

Accepted: 7 November 2023
© The Author(s) 2023

Abstract

Preventive maintenance (PM) is performed so that failure is avoided while corrective maintenance is performed after a failure has occurred in order to restore the system back to an operational state. This research aims at scheduling PM activities for a multi-component system within a finite time horizon. We consider a setting with two stakeholders, being the system operator and the maintenance workshop, and two different contract types governing their joint activities, namely an availability contract and a turn-around time contract. Components in the systems that are to be maintained are sent to the maintenance workshop, which needs to schedule and perform all maintenance activities while at the same time satisfying the contract and not exceeding the workshop capacity. Our modelling is based on a mixed-binary linear optimization model of a PM scheduling problem with so-called interval costs over a finite and discretized time horizon. We enhance this scheduling model with the flow of individual components through the maintenance workshop, including stocks of spare components, both those components that need repair and the repaired ones. The resulting scheduling model is then utilized in the optimization of two main contracts, namely maximizing the availability of repaired (or new) components and minimizing the deviation from the contracted turn-around times for the components in the maintenance loop. Each of these objectives is combined with the objective to minimize the costs for maintenance of the operating system, leading to two bi-objective optimization problems. We analyse the two contracting forms between the stakeholders by studying and comparing the Pareto fronts resulting from different parameter settings, regarding minimum allowed stock levels and investments in repair capacity of the workshop. Our bi-objective mixed-binary linear optimization model is able to capture important properties of the results from the contracting forms as well as to show that, in our setting, an availability contract performs better than a turn-around time contract in terms of tractability.

Keywords System maintenance · Workshop scheduling · Mixed-binary linear optimization model · Optimization of contracting forms · Simultaneous scheduling · Bi-objective optimization

1 Introduction

When planning maintenance for a system, the decisions to be made concern when each of its components should

be maintained (i.e. repaired or serviced) and what kind of maintenance should then be performed, with respect to the operational schedule of the system (i.e. maintenance can be done only when the system is not operational). To keep the system functioning, the maintenance of its components has to be planned and performed in good time. Preventive maintenance (PM) can often be planned well in advance, while corrective maintenance (CM) is done after a failure has occurred, to restore a system into operational state, which may come on very short notice and usually higher cost. However, a CM action may provide an opportunity for PM at which the maintenance actions can be rescheduled, starting from the system's current state. We model PM scheduling, while CM is implicitly included by an additional cost which increases with the time between PM occasions. That means

✉ Gabrijela Obradović
gabobradovic@gmail.com; gabobr@chalmers.se

Ann-Brith Strömberg
anstr@chalmers.se

Kristian Lundberg
kristian.lundberg@saabgroup.com

¹ Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden

² Saab AB, Linköping, Sweden

that the longer time in between two PM occasions, the more costly it becomes to do maintenance. This cost structure leads to decreasing risk of unexpected failures (and need for CM). Yu and Strömberg (2021) present a model that uses failure time distributions to model such additional costs.

We consider a setting with one *system operator* and one *maintenance workshop*, which are typically two separate stakeholders, and a *contract* regulating their collaboration. Components that are to be maintained are sent to a maintenance workshop, which needs to schedule and perform all maintenance activities while satisfying the contract, which may define conditions on delivery dates for and/or requirements on the availability of components for the system operator. The workshop's ability to fulfil the contract is dependent on its capacity, in terms of the number of parallel repair lines.

For the case of a contracted delivery date, there is normally a fee to be paid by the maintenance workshop at a late or early delivery. For the case of an availability contract, the workshop normally has to pay a fine when(ever) the number of available (i.e. repaired) components goes below the contracted limit.

We formulate a bi-objective optimization model of the following system of systems: (i) scheduling the PM occasions for the components of the system(s) and (ii) scheduling the repair activities in the maintenance workshop. The main contract types to be analysed are (iii) a component repair turn-around time based contract and (iv) a contract aimed at regulating the availability of components. The objectives considered are (v) minimizing the preventive maintenance costs for the system operator (i.e. set-up costs for the maintenance occasions as well as component replacement costs related to the maintenance intervals), (vi) maximizing the availability (to the system operator) of components repaired by the maintenance workshop and (vii) minimizing the penalty costs (paid by the maintenance workshop) for late or early deliveries of repaired components.

The main contributions of this work are a mathematical model of the simultaneous scheduling of replacement and repair of individual components used in multiple systems, mathematical models of two contracting forms between stakeholders and a comparison of the contracting forms by means of two bi-objective optimization problems.

It is important to understand the operational difference in performance (e.g. in terms of minimal cost) of the different contracts, as defined in (iii) and (iv), respectively. Most supply chain research (see, for example, Akyuz & Erkan 2010) and its industrial understanding focus on how to set up the supply chain in terms of initial stock levels, machine capacities and personnel rostering based on a predicted flow of material, goods or services. There is, however, little understanding of how contracting forms affect the dynamical and/or combinatorial aspects in a supply chain with two or more stakeholders.

Our model can be applied to any system that performs some sort of operations and whose components/parts have to be maintained. Some of many examples are railway and air traffic, and manufacturing machines in industry (see, for example, Robert et al., 2018; Verhoeff et al., 2015; Boliang et al., 2019; Papakostas et al., 2010). Therefore, the motivation behind this research lies in real-world applications.

The result of our model and computations—for a specific application and instance—is a maintenance schedule for individual components, which takes into account the operational requirements on and schedules for the systems, the maintenance requirements for the components of the systems and the capacity of the maintenance workshop.

The assumption we make is that the maintenance workshop and the preventive maintenance scheduling are tightly integrated (i.e. the information is transparent in between the two stakeholders), and we do so because of three reasons. Firstly, it provides a useful planning tool for the case when the workshop is actually integrated with the operating systems (i.e. when there is only one stakeholder). Secondly, when the workshop is controlled by another stakeholder, our tightly integrated model will provide optimistic estimates of achievable results which can be used as benchmarks for comparing with the current results. Lastly, our integration enables an investigation of schedules resulting from different types of contracts between the stakeholders as well as to different capacity levels of the workshop. In Sect. 4, we compare the two contract types, which are based on the availability of components and the turn-around times for the repaired components, respectively. We also study the load level of the workshop for different capacity levels. Results can be used as decision support for the stakeholders/operations management when setting up a contract as well as when making investment decisions (i.e. capacity in the maintenance workshop).

The model of the maintenance scheduling problem presented in this article is partly based on the *preventive maintenance scheduling problem with interval costs* (PMSPIC) model presented in Gustavsson et al. (2014). The PMSPIC considers one system with multiple component types and for which the costs for replacement of components take into account the interval between any two consecutive replacements/maintenance occasions; we generalize this model in the sense that also the individual components are considered and can be placed in any of the systems, as well as allowing for multiple systems. To reduce the probability of unexpected failures, which will reduce the need for CM, we enforce the PM activities to be scheduled before the end of the component's expected life. We also take into account the operational schedules for the systems which lead to time windows in which the different maintenance activities may or must be performed.

One way of generating the operational schedules (e.g. timetables) for the systems considered is presented in Gavranis and Kozanidis (2015), in which the availability of a fleet of aircraft is maximized subject to requirements on the transport missions and maintenance of the aircraft and their components. The results obtained include a tool for deciding which aircraft to fly when and for how long, and at what times the aircraft may and/or must undergo maintenance. The goal is to maximize the fleet availability over the planning horizon while ensuring that the operational and maintenance requirements are met. Methods suggested in that article are used to generate timetables that are input to our model.

The remainder of this article is organized as follows. In Sect. 2, we define the generalized PMSPIC, the maintenance workshop, the stock dynamics modelling and their integration with the operational demand on the systems. We define several objectives in relation to the two stakeholders. In Sect. 3, we present our bi-objective modelling. Tests and results are presented in Sect. 4, and in Sect. 5, we give conclusions and present ideas for future research.

2 Definition of the maintenance scheduling problem

The problem studied in this article is described as follows. A number of systems are operating to fulfil a common production demand; their operating schedules are assumed to be predefined, resulting in certain time windows during which maintenance of the systems' components may be performed. The systems operate their components degrade, which leads to need for maintenance (i.e. service, replacement or repair). At a maintenance occasion, one or several components are taken out of the system, sent to the maintenance workshop for repair and returned back to the stock of repaired components, ready to be used again. The components that are sent for repair are instantly replaced by components that are currently on the stock of repaired components. Therefore, replacement can only be done if there is at least one component (of the same type as components to be replaced) available. Hence, there is a circulating flow of individual components, being used and degraded, replaced, repaired or serviced, and then put back in a system to be used again. This structure of the system of systems is illustrated in Fig. 1. We model this system of systems such that (i) the individual components are tracked, (ii) the operating systems are preserved operational (if possible) and (iii) the capacity of the maintenance workshop is respected. The time is discretized, and we employ a so-called time-indexed modelling. Depending on the length of the planning horizon, the individual components will undergo repair a different number of times.

We start by making a formal definition of the generalized PMSPIC—which models the replacement scheduling

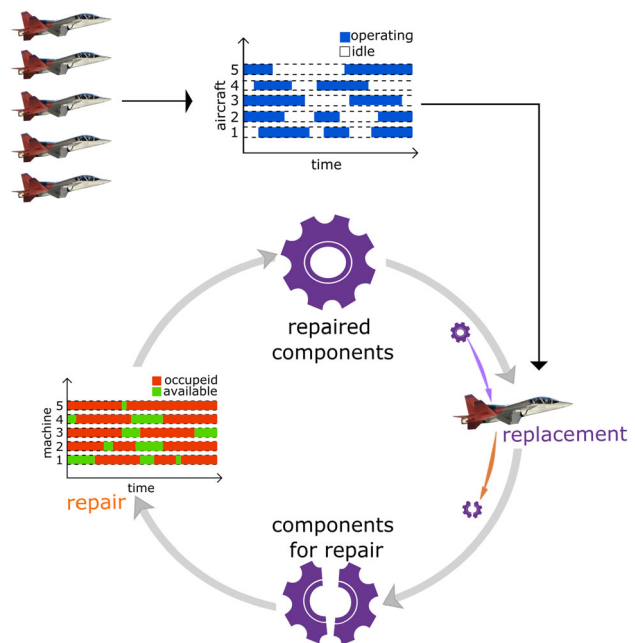


Fig. 1 Operating systems and the maintenance workshop, with the operational demand as input and the scheduling of component replacement and repair as output

for the components of the systems considered—along with a mixed-binary linear optimization formulation. Then, we model the scheduling of the maintenance workshop using mixed-binary linear optimization. These systems are then integrated through the dynamics of the stocks of components waiting to be maintained and those that have finished maintenance and are available to be used again by the systems. The section is concluded with a summary of the combined mixed-integer linear optimization formulation.

2.1 The generalized preventive maintenance scheduling problem with interval costs

The *generalized preventive maintenance scheduling problem with interval costs* (GPMSPIC) is defined as follows; cf. Gustavsson et al. (2014).

Definition 1 (GPMSPIC) Consider K systems $k \in \mathcal{K} := \{1, \dots, K\}$ with component types $i \in \mathcal{I} := \{1, \dots, I\}$, the set of individual components of type i defined as $\mathcal{J}_i := \{1, \dots, J_i\}$, and a set $\mathcal{T} := \{1, \dots, T\}$ of time steps at which maintenance of the systems can be performed, where T represents the planning horizon. A PM schedule consists of a set of scheduled replacement times in \mathcal{T} for each system k and component type i . A maintenance occasion for system k at time t generates the maintenance occasion cost d_t^k . If PM of an individual component j of type i , denoted with (i, j) , in system k is scheduled at the times $s \in \mathcal{T} \cup \{0\}$ and $t \in \{s + 1, \dots, T + 1\}$, but not in the (possibly empty)

time interval $\{s + 1, \dots, t - 1\}$, then the maintenance interval, denoted (s, t) , generates the interval cost c_{st}^i . Find a PM schedule that minimizes the sum of maintenance occasion costs and interval costs. \square

Any special case of the GPMSPIC such that $K = J_i = 1$ coincides with the PMSPIC. According to Gustavsson et al. (2014)—see also Arkin et al. (1989) and Boctor et al. (2004)—the PMSPIC is NP-hard¹ which implies that the GPMSPIC is also NP-hard. The main practical implication of this property is that the optimal scheduling of the PM occasions for the components of the system(s) is a computationally demanding problem.

We next model the GPMSPIC as a binary linear optimization problem. With the decision variables being defined as

$$x_{st}^{ijk} = \begin{cases} 1, & \text{if individual component } j \text{ of type } i \text{ in system } k \\ & \text{receives PM at times } s \text{ and } t, \text{ but not in between,} \\ 0, & \text{otherwise,} \end{cases} \quad j \in \mathcal{J}_i, i \in \mathcal{I}, k \in \mathcal{K}, \\ 0 \leq s < t \leq T + 1, \\ z_t^k = \begin{cases} 1, & \text{if maintenance of system } k \text{ occurs at time } t, \\ 0, & \text{otherwise,} \end{cases} \quad k \in \mathcal{K}, t \in \mathcal{T},$$

the feasible set is modelled by the equality and inequality constraints

$$\sum_{j \in \mathcal{J}_i} \sum_{s=0}^{t-1} x_{st}^{ijk} = \sum_{j \in \mathcal{J}_i} \sum_{r=t+1}^{T+1} x_{tr}^{ijk}, \quad i \in \mathcal{I}, t \in \mathcal{T}, k \in \mathcal{K}, \quad (1a)$$

$$\sum_{j \in \mathcal{J}_i} \sum_{r=1}^{T+1} x_{0r}^{ijk} = 1, \quad i \in \mathcal{I}, k \in \mathcal{K}, \quad (1b)$$

$$\sum_{j \in \mathcal{J}_i} \sum_{s=0}^{t-1} x_{st}^{ijk} \leq z_t^k, \quad i \in \mathcal{I}, t \in \mathcal{T}, k \in \mathcal{K}, \quad (1c)$$

$$\sum_{k \in \mathcal{K}} \sum_{s=0}^{t-1} x_{st}^{ijk} \leq 1, \quad j \in \mathcal{J}_i, i \in \mathcal{I}, t \in \mathcal{T}, \quad (1d)$$

$$x_{st}^{ijk} = 0, \quad \bar{t}_i \leq s + \bar{t}_i < t \leq T + 1, \\ j \in \mathcal{J}_i, i \in \mathcal{I}, k \in \mathcal{K}. \quad (1e)$$

For each system k and component type i , a maintenance interval starts at time 0, which is modelled by (1b), while the constraints (1a) ensure that the same number (i.e. 0 or 1) of maintenance intervals ends and starts at time t . The constraints (1c) ensure that if a maintenance interval of component type i in system k ends at time t , then maintenance of system k must occur at time t . The constraints (1d) ensure that each component (i, j) is in at most one system k at each time t . The constraints (1e) prevent any maintenance interval for component type $i \in \mathcal{I}$ from being longer than $\bar{t}_i \leq T$, which prevents from having to perform corrective maintenance.

¹ A decision problem is in NP if the answer “yes” can be verified in polynomial time. A decision problem is NP-hard if any NP problem can be reduced to it in polynomial time. A decision problem is NP-complete if it is in NP and NP-hard (Conforti et al., 2014, Ch. 1.3)

2.2 The maintenance workshop scheduling problem

Components that should be maintained are sent to the *maintenance workshop*, which contains a number of (identical) parallel machines for component repair, each of which has a repair capacity of one unit while each component repair requires one unit of this capacity per time step during a pre-specified and consecutive (i.e. preemption is not allowed) number of time steps. When a component arrives at the workshop, it is available for repair and assigned a due date, at which the repair should be finished, and the component be returned back to the system operator. This problem is identified as an *identical parallel machines scheduling problem* (IPMSP; commonly denoted $P \parallel \sum C_j$); see Brucker and Knust (2012, Ch. 1.2.2). A solution to the maintenance workshop scheduling problem specifies at which time each component arriving at the workshop should start maintenance. If a component is delivered after its due date, the maintenance workshop has to pay a fee to the system operator.

Definition 2 (IPMSP) Consider a set $\mathcal{L} := \{1, \dots, L\}$ of identical component repair machines and the individual components $j \in \mathcal{J}_i$ of types $i \in \mathcal{I}$ that arrive at the workshop at given time points $t^{ij} \in \mathcal{T}$. Each component (i, j) has a repair time $p^i > 0$ (number of time steps it takes for repair) and a due date $d^{ij} \geq t^{ij} + p^i$ (number of time steps within which a component should be repaired and returned). At most $L \geq 1$ machines can operate simultaneously. A component that finishes repair prior to or after its due date generates a non-negative penalty cost. Find a schedule for the maintenance workshop such that the sum of the penalty costs for late and early deliveries of the repaired components is minimized. \square

The IPMSP with a (weighted) sum objective is polynomially solvable (Lawler & Lenstra, 1993, Ch. 8.0), whereas its version with a minimax, i.e. makespan, objective is NP-hard (Brucker & Knust, 2012, Ch. 2.1).

To model this as a mixed-integer optimization problem, we define for each $j \in \mathcal{J}_i$, $i \in \mathcal{I}$, and $t \in \mathcal{T}$, the decision variables as

$$u_t^{ij} = \begin{cases} 1, & \text{if component } (i, j) \text{ starts repair at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the number ℓ_t of active parallel machines at time t should fulfil the constraints

$$0 \leq \ell_t = \ell_{t-1} + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} (u_t^{ij} - u_{t-p^i}^{ij}) \leq L, \quad t \in \mathcal{T}, \quad (2)$$

where ℓ_0 and u_t^{ij} , $t \leq 0$, are initial (fixed) values that constitute input to the model; see (6). The constraints (2) state that

the number of active parallel machines at time t equals the number of active machines in the previous time step (i.e. $t-1$) plus the difference between the numbers of components starting and finishing repair (i.e. the number of parallel machines being activated and deactivated, respectively) at time step t ; they also state that the number of activated machines at any time step must be in the interval $[0, L]$. In our study, we also vary the number, L , of parallel machines, to enable decision support for capacity investments in the maintenance workshop.

Our model assumes deterministic processing times, p^i , in the maintenance workshop. In practical maintenance applications, however, the real processing times are occasionally revealed only after the component has arrived at the maintenance workshop, which we denote as *unexpected events*. In Sect. 4.4, we suggest a rescheduling procedure that takes such unexpected events into account.

To connect the mathematical models of the IPMSP and the GPMSPIC, we next introduce the stock dynamics modelling.

2.3 The stock dynamics

When an individual component (i, j) is taken out of system k it is sent—with no time delay—to the stock of damaged components, where it stays until it is scheduled for repair. The transport time between the stock of damaged components and the maintenance workshop is denoted δ_a^i . Upon being repaired, it goes to the stock of repaired (i.e. as good as new) components—with a transport time denoted δ_b^i —where it is kept until its scheduled time for placement into a(nother) system $k \in \mathcal{K}$. We assume that all transport times are represented by non-negative integers.

The integration of the models of the GPMSPIC and the IPMSP requires the modelling of the two stocks of damaged and repaired components, respectively. We introduce the following variables for all $j \in \mathcal{J}_i$ and $i \in \mathcal{I}$:

$$a_t^{ij} (b_t^{ij}) = \begin{cases} 1, & \text{if individual component } j \text{ of type } i \\ & \text{is on the stock of damaged (repaired)} \\ & \text{components at time } t \in \mathcal{T} \cup \{0\}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\alpha_t^{ij} (\beta_t^{ij}) = \begin{cases} 1, & \text{if individual component } j \text{ of type } i \text{ is taken} \\ & \text{out of (placed in) one of the systems } k \in \mathcal{K} \\ & \text{at time } t \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases}$$

The stock of damaged components is then modelled by the constraints

$$\alpha_t^{ij} = \sum_{k \in \mathcal{K}} \sum_{s=0}^{t-1} x_{st}^{ijk}, \quad j \in \mathcal{J}_i, i \in \mathcal{I}, t \in \mathcal{T}, \quad (3a)$$

$$a_t^{ij} = a_{t-1}^{ij} + \alpha_t^{ij} - u_{t+\delta_a^i}^{ij} \in \{0, 1\},$$

$$t \in \{1 - \delta_a^i, \dots, T + 1\}, j \in \mathcal{J}_i, i \in \mathcal{I}. \quad (3b)$$

The constraints (3a) connect the variables from the GPMSPIC with the stocks: if a component (i, j) is taken out of any of the systems $k \in \mathcal{K}$ at time t , α_t^{ij} will take the value 1; otherwise, α_t^{ij} takes the value 0.² The constraints (3b) provide the state of component (i, j) at time t : whether it is on the stock of damaged components (i.e. $a_t^{ij} = 1$) or not (i.e. $a_t^{ij} = 0$). The state of a component at time t depends on its state in the previous time step $t-1$, whether it is taken out of any system k and placed on the stock at time step t , and whether it is starting maintenance at time step $t + \delta_a^i$. We let $\alpha_t^{ij} := 0, t \in \{1 - \delta_a^i, \dots, 0\}$, while the variable a_0^{ij} constitutes (fixed) input data and must fulfil (6).

The stock of repaired components is modelled analogously, as

$$\beta_t^{ij} = \sum_{k \in \mathcal{K}} \sum_{r=t+1}^{T+1} x_{tr}^{ijk}, \quad j \in \mathcal{J}_i, i \in \mathcal{I}, t \in \mathcal{T} \quad (4a)$$

$$b_t^{ij} = b_{t-1}^{ij} - \beta_t^{ij} + u_{t-\delta_b^i-p^i}^{ij} \in \{0, 1\},$$

$$j \in \mathcal{J}_i, i \in \mathcal{I}, t \in \mathcal{T} \quad (4b)$$

$$\sum_{j \in \mathcal{J}_i} b_t^{ij} \geq \underline{b}^i, \quad i \in \mathcal{I}, t \in \mathcal{T}. \quad (4c)$$

The constraints (4a) represent the connection between the stocks of repaired components and the GPMSPIC. If component (i, j) is placed into any system k at time t , β_t^{ij} will take the value 1; otherwise, β_t^{ij} takes the value 0. In (4b), the individual states of the components at time t are updated: a component is either on the stock (i.e. $b_t^{ij} = 1$) or it is not (i.e. $b_t^{ij} = 0$). A component's state on the stock of repaired components at time t is affected by its state in the previous time step $t-1$, whether it is placed in some system k at time t , and whether it will arrive to the stock at time t after being repaired (i.e. if $u_{t-\delta_b^i-p^i}^{ij} = 1$, meaning that component (i, j) started maintenance at time $t - \delta_b^i - p^i$ and thus arrives to the stock of repaired components at time t).³ Then, in (4c) it is expressed that the sum of the variables b_t^{ij} over the individual components, i.e. the stock level of repaired components per

² Note that the constraints (3a) combined with the binary requirements on the variables α_t^{ij} make the constraints (1d) redundant.

³ The variables b_0^{ij} , β_0^{ij} and $u_t^{ij}, t \in \{1 - \delta_b^i - p^i, \dots, 0\}$, comprise (fixed) input data, which must fulfil the constraints (6).

component type i at time t , may not fall short of the lower stock limit \underline{b}^i .

The constraints (3) and (4) control the stocks/inventory of damaged and repaired components, respectively, and enable the control of the levels of these stocks subject to relevant constraints.

2.4 Integration with the operational demand of the systems

What drives the need for maintenance of components, and constitutes the input to our modelling, is the *operational demand*: we assume that operational schedules are given for the systems $k \in \mathcal{K}$ which are such that the demand for operations can be fulfilled. For our maintenance planning problem, these schedules are represented in terms of time intervals when the system is either operating—at which times maintenance cannot be performed—or accessible for maintenance. In other words, PM may not be scheduled while a system is operating. In the case of railway systems (Lidén, 2020), each train would be assigned time slots when it should operate (i.e. perform transports of goods or passengers); hence, PM may be scheduled only in between these time slots. In the case of offshore wind turbine maintenance (Shafiee et al., 2013), the operational demand is fulfilled by wind energy production, while maintenance work can be done only during time periods of not too harsh weather conditions. When planning any PM occasion the (predicted or planned) operational schedules for the systems provide time windows during which maintenance may be performed. As input to the integrated GPMSPIC and IPMSP model, for all $t \in \mathcal{T}$ and all $k \in \mathcal{K}$ we thus let the parameters

$$\bar{z}_t^k = \begin{cases} 1, & \text{if PM is allowed to be scheduled for system } k \text{ at time } t, \\ 0, & \text{otherwise,} \end{cases}$$

define upper limits on the variables representing the maintenance occasions, as

$$z_t^k \leq \bar{z}_t^k, \quad t \in \mathcal{T}, k \in \mathcal{K}, \quad (5)$$

the fulfilment of which implies that the time windows for PM are respected.

2.5 Initialization

In our modelling, the planning horizon is assumed to start at time step 0. The events and states of the systems, in terms of variables x_{0r}^{ijk} , a_0^{ij} and b_0^{ij} , and u_t^{ij} for a number of preceding time steps, thus need to be initialized such that the constraints (1d), (3b) and (4b) are fulfilled for the initial indices. The nec-

essary and sufficient initializations of variables are expressed by the constraints

$$\sum_{k \in \mathcal{K}} \sum_{r=1}^{T+1} x_{0r}^{ijk} + a_0^{ij} + b_0^{ij} + \sum_{t=1-p^i-\delta_b^i}^0 u_t^{ij} = 1, \quad j \in \mathcal{J}_i, i \in \mathcal{I}, \quad (6)$$

implying that each component individual is at exactly one place at time 0, i.e. either in one of the systems $k \in \mathcal{K}$, in one of the two stocks or in the maintenance workshop.

2.6 The complete model of the system of systems

In summary, the set of feasible solutions to our maintenance scheduling problem is modelled by the constraints (1), (2), (3), (4), (5) and (6), with binary requirements⁴ on the variables x_{st}^{ijk} , z_t^k , u_t^{ij} , a_t^{ij} , b_t^{ij} , α_t^{ij} and β_t^{ij} , for all relevant values of the indices, while $\ell_t \in \mathbb{Z}_+$ should hold⁵ for $t \in \mathcal{T}$.

3 Contracts and optimization objectives

3.1 Contracts between the stakeholders

In order to compare two different types of contracts between the stakeholders—*availability* and *turn-around time* contracts—and their dependence on the capacity level in the maintenance workshop, we define three objectives: (i) minimization of the maintenance cost for the system operator; (ii) maximization of the availability of components on the stock of repaired components (which can be modelled as minimizing the risk for lack of repaired components); and (iii) minimization of the penalty for late and early deliveries of repaired components, which is paid by the maintenance workshop to the system operator.

We study the two contracts by defining two bi-objective optimization problems from the objectives (i)–(iii). The first problem is composed by the minimization of the maintenance cost and the maximization of the availability of components on the stock of repaired components, i.e., objectives (i) and (ii). The second problem is composed by the minimization of the maintenance costs and the minimization of the penalty for lateness and earliness, i.e. objectives (i) and (iii). In both problems, the capacity level of the workshop is varied, while the set of feasible schedules is defined by all the constraints

⁴ If all variables z_t^k , x_{st}^{ijk} and u_t^{ij} have binary values, the binary requirements on the variables a_t^{ij} , b_t^{ij} , α_t^{ij} and β_t^{ij} can be relaxed to values in the interval $[0, 1]$, since any corresponding mixed-integer linear optimization problem will possess binary optimal solutions.

⁵ Due to the constraints (2) and the binary requirements on the variables u_t^{ij} , the variables ℓ_t can be modelled as continuous, non-negative variables.

defined in Sect. 2. In Sect. 4, these two problems will be studied from a bi-objective point of view (Ehrgott, 2005) and the results are compared and analysed.

3.2 Optimization objectives

Below follows our detailed modelling of the four objectives defined in Sect. 3.1.

Minimizing costs for maintenance set-up and intervals Each maintenance occasion yields a set-up/maintenance cost for the system operator. Besides this, there is an interval cost for every component which is determined based on the length of the interval between two consecutive maintenance occasions. We assume that the interval cost is non-decreasing with the length of the interval. The rationale behind this assumption that (i) the longer time the component has been used for operations, the more costly will the maintenance be, and (ii) it enables the enforcement of scheduling the maintenance at the latest at the end of each individual component's life (cf. (1e)). From the system operator's point of view, the objective is to minimize the total costs for maintenance during a prespecified time period. We mathematically formulate this objective as to

$$\text{minimize } \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} d_t z_t^k + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} \sum_{t=1}^{T+1} \sum_{s=0}^{t-1} c_{st}^{ijk}, \quad (7)$$

where the first sum represents the maintenance set-up costs and the second the interval costs. Every maintenance occasion for the system k (when $z_t^k = 1$) generates a cost $d_t > 0$ while every maintenance interval (s, t) for a component (i, j) in system k (when $x_{st}^{ijk} = 1$) yields an interval cost $c_{st}^{ijk} > 0$, which is such that $c_{st}^{ijk} \geq c_{sr}^{ijk}$ for all $r \leq t$.

Minimizing the risk for lack of spare parts To ensure that the operational schedule is undisturbed, or at least that the disturbance is minimal, it is crucial to have enough spare components available. Then, whenever an unexpected failure occurs, the damaged component can be replaced by a new one without the planned operations having to be stopped. We maximize a weighted average of the number of repaired (or new) components available, which is modelled as to

$$\text{maximize } \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} w^i \sum_{j \in \mathcal{J}_i} b_t^{ij}, \quad (8)$$

where $w^i > 0$ is an objective weight assigned to component type $i \in \mathcal{I}$. The value of this objective function corresponds to the (weighted) average number of repaired components available at each single time step, to be compared with the total number, $\sum_{i \in \mathcal{I}} J_i$, of components in the system.

Minimize risk of exceeding the contracted turn-around times for component repair The “turn-around time” of an

individual component (i, j) is defined as the time when it is taken out of one of the systems in \mathcal{K} (i.e. a time t such that $\alpha_t^{ij} = 1$) until it has become repaired and is available for usage again in one of the systems (i.e. a t such that $u_{t-p^i-\delta_b^i}^{ij} = 1$). The total turn-around time, v_{tat}^{ij} , for component individual (i, j) , $j \in \mathcal{J}_i$, $i \in \mathcal{I}$, over the planning period \mathcal{T} , is thus computed as

$$v_{\text{tat}}^{ij} = \sum_{t=0}^{T+1} \left((t + p^i + \delta_b^i) u_t^{ij} - t \alpha_t^{ij} \right), \quad (9a)$$

where the term $(p^i + \delta_b^i) u_0^{ij}$ is positive if component (i, j) is initially on the stock of damaged components, and the equalities $u_{T+1}^{ij} = a_0^{ij} - u_0^{ij} + \sum_{t \in \mathcal{T}} (\alpha_t^{ij} - u_t^{ij})$ and $\alpha_{T+1}^{ij} = 0$ hold.⁶ The shortest possible turn-around time for a component of type i equals $\delta_a^i + p^i + \delta_b^i$, i.e. the sum of the repair time in the maintenance workshop and the time required for the transportation to and from the workshop.

Letting $c_{\text{delay}}^{ij} > 0$ and $c_{\text{early}}^{ij} \in (0, c_{\text{delay}}^{ij}]$ denote the penalties for late and early, respectively, delivery of a repaired component, this objective is expressed as to

$$\text{minimize } \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} \left(c_{\text{delay}}^{ij} v_{\text{delay}}^{ij} - c_{\text{early}}^{ij} v_{\text{early}}^{ij} \right), \quad (9b)$$

where v_{delay}^{ij} (v_{early}^{ij}) denotes the total delay (earliness) for component (i, j) over the planning period. These variables are due to the constraints

$$v_{\text{early}}^{ij} \leq v_{\text{tat}}^{ij} - d^{ij} \left(a_0^{ij} + \sum_{t=1}^{T+1} \alpha_t^{ij} \right) \leq v_{\text{delay}}^{ij}, \quad (9c)$$

$$v_{\text{early}}^{ij} \leq 0 \leq v_{\text{delay}}^{ij}. \quad (9d)$$

Due to the construction of (9) at least one of v_{early}^{ij} and v_{delay}^{ij} will attain the value 0 when the objective (9b) is at optimum (a component will be either early, late, or on time; in the latter case $v_{\text{early}}^{ij} = v_{\text{delay}}^{ij} = 0$ holds). If the turn-around time v_{tat}^{ij} is longer (shorter) than the due date d^{ij} , the component is late (early); thus, $v_{\text{delay}} > 0$ ($v_{\text{delay}} = 0$) and $v_{\text{early}} = 0$ ($v_{\text{early}} < 0$). By construction of (9), we aggregate these variables over the whole planning period. Hence, the objective (9b) minimizes the total penalty for late and early component delivery.

⁶ Note that the use of the variables a_0^{ij} and u_{T+1}^{ij} lead to a possible underestimate of v_{tat}^{ij} , as we possibly shorten the v_{tat}^{ij} for components which were initialized on the stock of components to be repaired at $t = 0$ and components which did not finish repair until $t = T + 1$.

It follows that v_{early}^{ij} (v_{delay}^{ij}), as defined in (9c)–(9d), will possibly be under(over)estimated.

In Sect. 4, we present our performed tests of the two bi-objective optimization problems along with the results obtained.

4 Application: implementation, tests and results

We present an application from the aerospace industry, from a collaboration with the Swedish aerospace and defence company Saab AB. The instance sizes are considered reasonable from a practical application point of view and the data sets used are based on knowledge mediated from the industrial partner; all numbers are normalized. Our implementation is made using Julia (2012) and JuMP (see Dunning et al., 2017), and the computations are performed by Gurobi (2020) on a laptop computer with a 2.4 GHz Intel Core i5 processor and 8 GB of RAM memory. The computer used has eight available processors. Gurobi usually uses all cores available, but can choose to use less. We investigated the thread count: for all the results reported Gurobi used all eight threads. Comparing with results obtained using single thread operations, we observed that computations take longer than allowing for multi-thread operations.

4.1 The main test instance and bi-objective settings

As a test case, we consider $K = 5$ systems, each having $I = 3$ component types and $J_i = 10$ individual components of each type $i = 1, 2, 3$. The operational and maintenance related differences of the component types are reflected by their respective processing times in the maintenance workshop, as well as their respective due dates, which are chosen randomly within the same order of magnitude. The different component types are also assigned different structures of their interval costs, all modelled with higher costs for longer time in between any two maintenance occasions, which reflect the increased risk of having to perform CM. For the turn-around time objective (9), the penalties for lateness, c_{delay}^{ij} , vary over the component types, and the penalties for earliness are set to $c_{\text{early}}^{ij} = c_{\text{delay}}^{ij}/2$. The planning horizon is $T = 20$ time steps and we alter the workshop capacity between $L = 7$ and $L = 10$ parallel machines. We have tested two main cases of our planning problem: (i) with no lower limits on the stocks of repaired components, i.e. with $\underline{b}^i = 0$, $i \in \mathcal{I}$, and (ii) with the lower limits $\underline{b}^i = 1$, $i \in \mathcal{I}$. The timetable for the systems' operations is generated by application of the model in Gavranis and Kozanidis (2015) to the set of systems \mathcal{K} over the whole planning period \mathcal{T} .

Our tests consider an availability contract and a turn-around time contract, each of which is modelled as a bi-objective optimization problem. For both contract types, the system operator's objective is modelled as to minimize

the total costs for maintenance, i.e. the objective (7). The maximization of the availability is achieved by the objective (8), while the minimization of the penalty for late and early deliveries is achieved by the objective defined in (9).

When solving a multi-objective optimization problem, one is usually interested in finding Pareto optimal, or efficient solutions; see, for example, Ehrgott et al. (2005, Ch. 2.1). A solution is called Pareto optimal if none of the objective functions can be improved in value without degrading at least one of the other objectives' values. To find points on the Pareto front—the set of all Pareto optimal points—we employ the ϵ -constraint method (see Mavrotas, 2009), which—in the bi-objective case—optimizes iteratively one objective function while the other is being constrained.

4.2 Results from the computational tests

For the availability contract, i.e. the objectives (7) and (8), we employed two levels of the lower limits on the stocks of repaired components: $\underline{b}_i \in \{0, 1\}$, $i \in \mathcal{I}$, and the workshop capacity $L = 10$. When the lower limit \underline{b}_i on the stocks of repaired components was set to 0 (1), the maximum average availability was in the interval $[2.9, 14.5]$ ($[5.25, 14.75]$), and the lower limit ϵ on the average availability was varied in the interval $[2.9, 14.5]$ ($[5.25, 14.75]$) with an increment of 0.5. The resulting Pareto fronts are plotted in Fig. 2a. For each value of ϵ , the minimal maintenance costs increase slightly when \underline{b}_i increase from 0 to 1. However, for large average numbers of repaired components on the stock (in our case, when the average over the planning period exceeds twelve components) the two Pareto fronts approach each other; this is due to the lower stock limit being 0 or 1 losing impact as the average availability increases.

For the turn-around time contract, i.e. the objectives defined by (7) and (9), we employed two levels of the lower limits on the stocks of repaired components: $\underline{b}_i \in \{0, 1\}$, $i \in \mathcal{I}$, and the workshop capacity $L = 10$. For $\underline{b}_i = 0$ (1), ϵ was varied in the interval $[8690, 11250]$ ($[8690, 11055]$) with an increment of 200. The resulting Pareto fronts are plotted in Fig. 2b. The differences between the respective minimal maintenance costs—for the lower limit on the stock of repaired components being 0 and 1, respectively—are larger than those resulting from the availability contract; these differences seem approximately constant when the value ϵ of the upper limit on the delay/earliness penalty is varied.

The minimal and maximal maintenance costs corresponding to the availability and turn-around time contracts, as illustrated in Fig. 2, are listed in Table 1.

The numbers illustrated in Figs. 3, 4, 5, 6, 7, 8 and 9 correspond to the Pareto optimal solutions resulting from the sum of the objectives in (8) and (7) for the availability contract, and (9b) and (7) for the turn-around time contract, respectively.

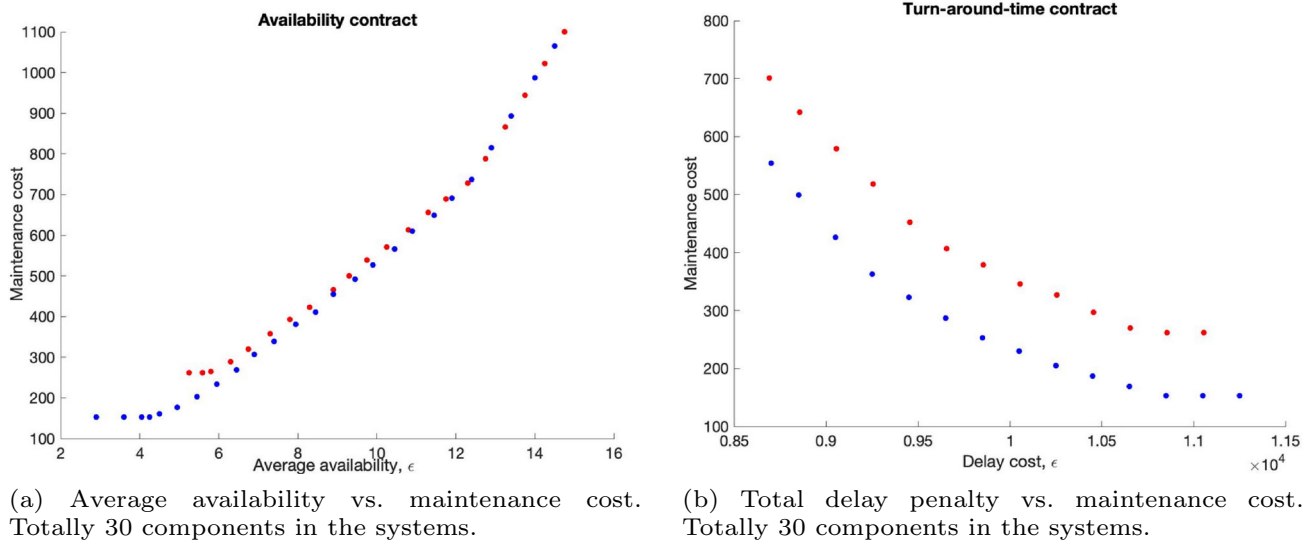


Fig. 2 Computed points on the respective Pareto fronts, for $L = 10$. The blue and red dots correspond to $\underline{b}^i = 0$ and $\underline{b}^i = 1$, respectively, $i \in \mathcal{I}$

Table 1 Minimal and maximal maintenance costs resulting from the availability and turn-around time contracts, for $\underline{b}^i \in \{0, 1\}$, $i \in \mathcal{I}$

| Stock limit \underline{b}^i | Minimum maintenance cost Both contracts | Maximum maintenance cost | |
|----------------------------------|--|--------------------------|---------------------------|
| | | Availability contract | Turn-around time contract |
| 0 | 153 | 1065 | 554 |
| 1 | 262 | 1100 | 701 |

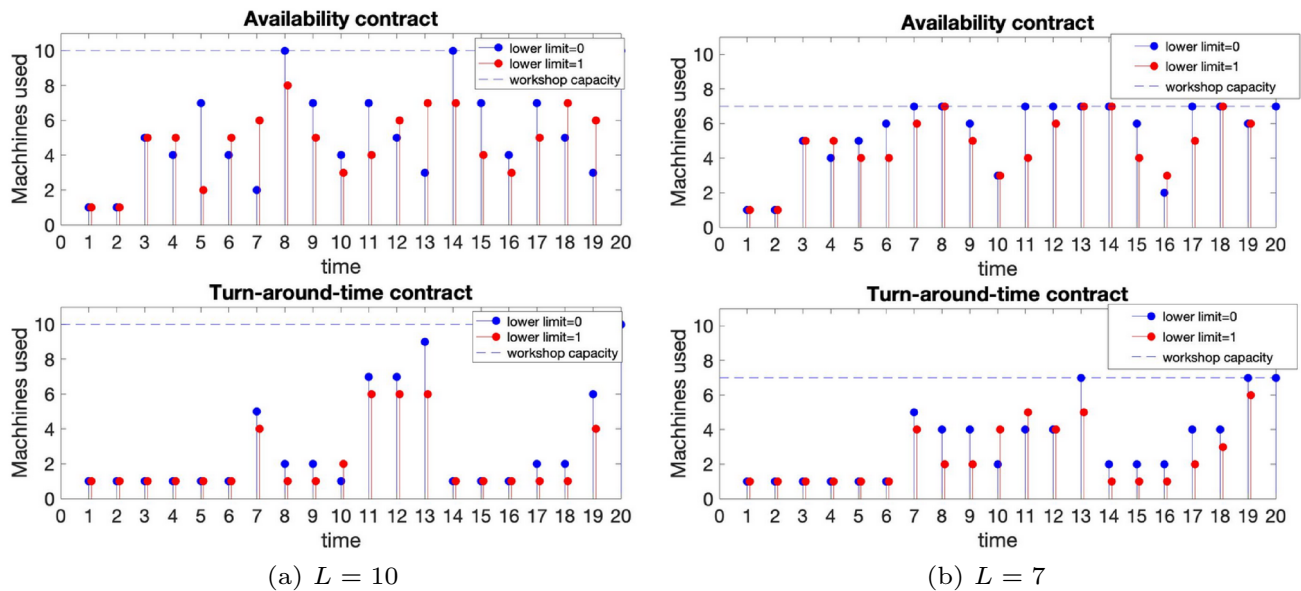
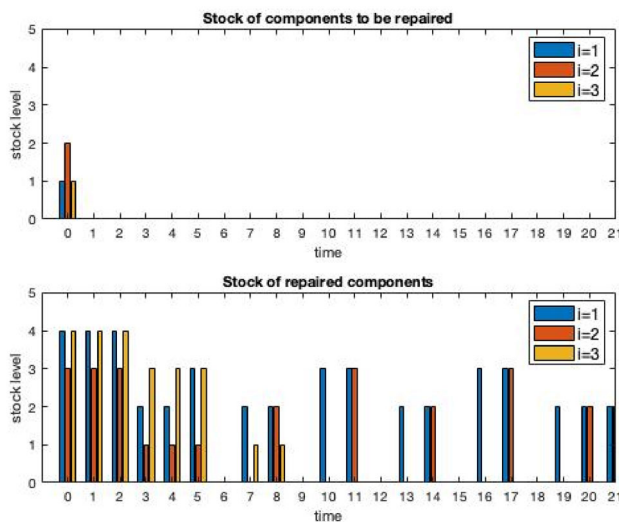
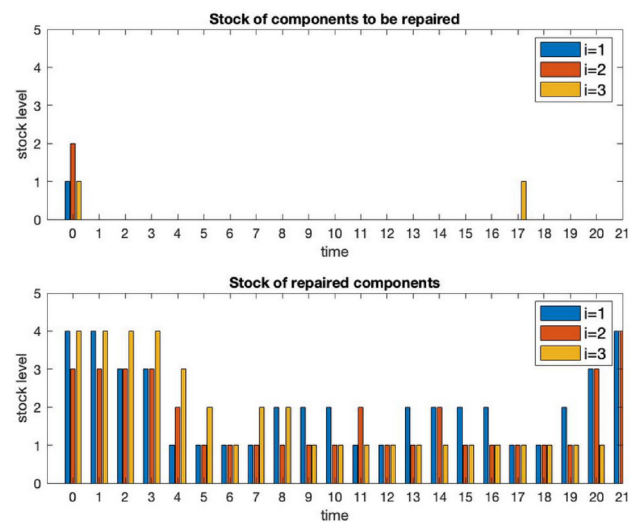


Fig. 3 Resulting numbers of active parallel machines over time, i.e. ℓ_t , $t \in \mathcal{T}$, for the availability and turn-around time contract, respectively. The blue (red) points correspond to a lower limit of $\underline{b}^i = 0$ (1) on the stock of repaired components

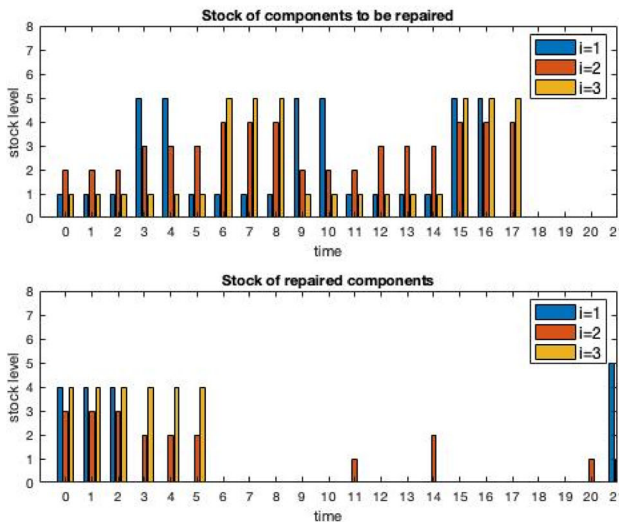


(a) No requirements on the stock of repaired components: $\underline{b}^i = 0$, $i \in \mathcal{I}$. Here, maintenance cost = 153 and average availability = 4.25.

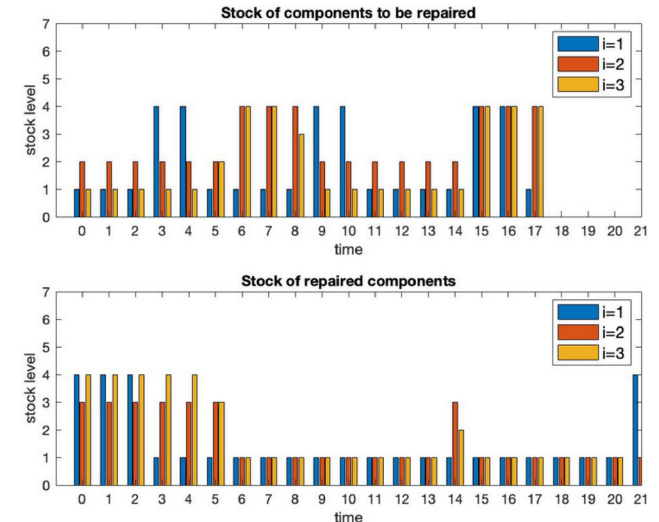


(b) Lower limit on the stock of repaired components, $\underline{b}^i = 1$, $i \in \mathcal{I}$. Here, maintenance cost = 262 and average availability = 5.59.

Fig. 4 Resulting stock levels for the availability contract, $L = 10$



(a) No requirements on the stock of repaired components: $\underline{b}^i = 0$, $i \in \mathcal{I}$. Here, maintenance cost = 554 and total delay penalty = 8700.

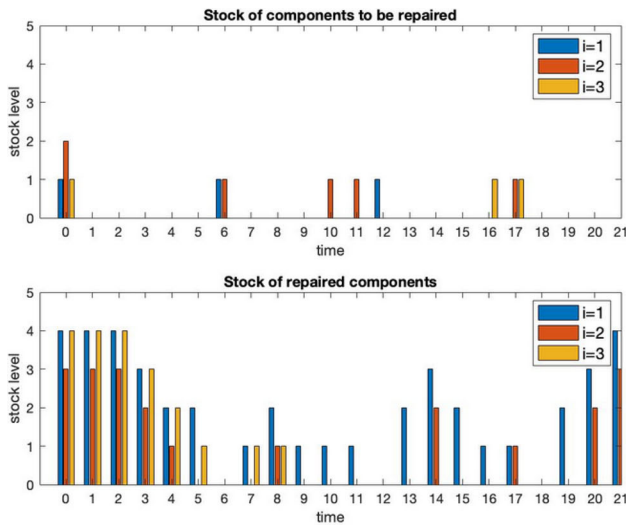


(b) Lower limit on the stock of repaired components, $\underline{b}^i = 1$, $i \in \mathcal{I}$. Here, maintenance cost = 701 and total delay penalty = 8690.

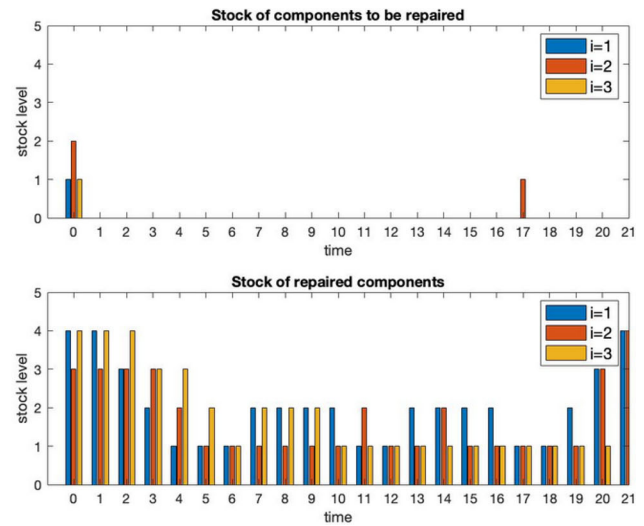
Fig. 5 Resulting stock levels for the turn-around time contract, with penalties for delayed and early deliveries, $L = 10$

Figure 3 illustrates the number of active parallel repair lines over time, as resulting from the availability and turn-around time contracts, for the workshop capacity being $L = 10$ and $L = 7$, and with the lower limit on the stock of repaired components being 0 and 1, respectively. The Pareto points corresponding to Fig. 3a, i.e. for $L = 10$ and $\underline{b}^i = 0(1)$, $i \in \mathcal{I}$, are determined as follows: For the availability contract, maintenance cost = 153 (262) and average availability = 4.249 (5.59); for the turn-around time contract, maintenance cost = 554 (701) and delay cost = 8700 (8690).

It is noticeable that an availability contract—as compared to a turn-around time contract—puts more demand on the workshop in order to fulfil the availability requirements on repaired components. When reducing the workshop capacity below $L = 7$, for the instance considered, finding optimal schedules becomes computationally too expensive for the case of a turn-around time contract. For the availability contract, the corresponding effect occurs when the workshop capacity goes below $L = 6$.

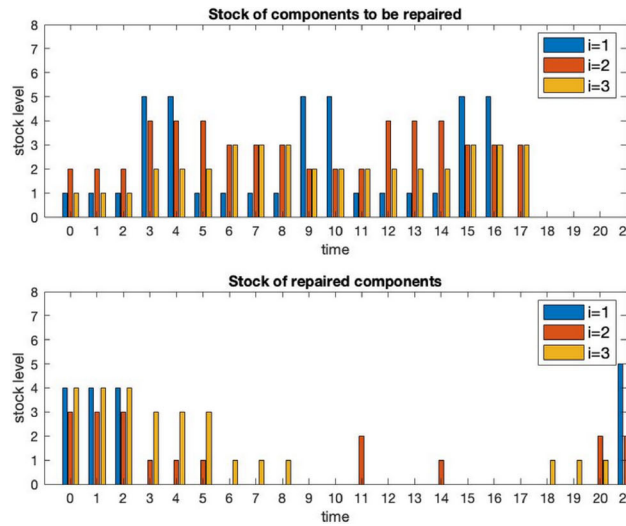


(a) No requirements on the stock of repaired components: $\underline{b}^i = 0, i \in \mathcal{I}$.

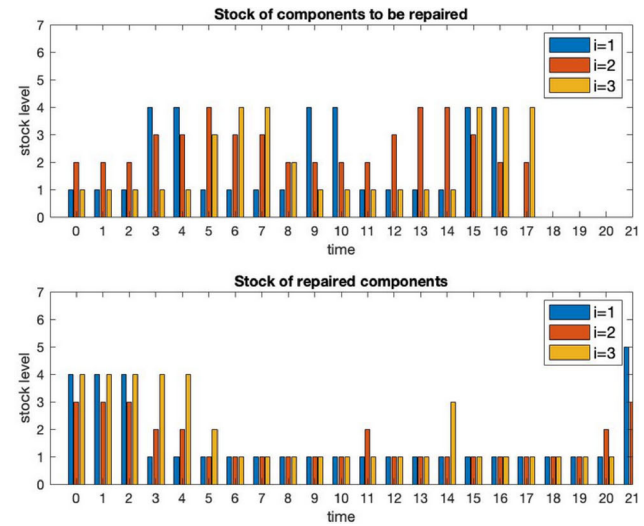


(b) Lower limit on the stock of repaired components, $\underline{b}^i = 1, i \in \mathcal{I}$.

Fig. 6 Resulting stock levels for the availability contract, $L = 7$



(a) No requirements on the stock of repaired components: $\underline{b}^i = 0, i \in \mathcal{I}$.



(b) Lower limit on the stock of repaired components, $\underline{b}^i = 1, i \in \mathcal{I}$.

Fig. 7 Resulting stock levels for the turn-around time contract with penalties for early and delayed deliveries, $L = 7$

The stock dynamics resulting from the Pareto points corresponding to the availability and turn-around time contract are illustrated in Figs. 4 and 5, respectively. The main difference in stock levels for the respective two contract types can be observed in the stock of components to be repaired. Namely, with an availability contract, damaged components are being repaired as soon as they arrive to the stock of components to be repaired while with a turn-around time contract, the number of damaged components on the stock is significantly larger over time. The same conclusion applies for both $\underline{b}^i = 0$ and $\underline{b}^i = 1$.

Reducing the workshop capacity from $L = 10$ to $L = 7$ parallel machines leads to different stock levels, as presented in Figs. 6 and 7, for the availability and turn-around time contract, respectively.

4.3 A larger test instance

In order to investigate the computational complexity of our model, we consider a larger instance of $K = 10$ systems, each having $I = 5$ component types and $J_i = 15$ individual components of each type $i = 1, \dots, 5$. We extend the time

Table 2 Instance sizes and computing times for the turn-around time contract

| Instance size | | | | | # Rows | # Variables | # Binary variables | | Gap (%) | Solution Time | Presolve Time (s) |
|---------------|-------|-----|-----|-----|-----------|-------------|--------------------|----------------|---------|---------------|-------------------|
| I | J_i | K | T | L | | | Original | After presolve | | | |
| 3 | 10 | 5 | 20 | 10 | 41,193 | 75,238 | 73,510 | 22,237 | 0.45 | 20 min | 0.48 |
| | | | | | | | | | 0 | 6.9 h | 0.43 |
| 5 | 15 | 10 | 50 | 20 | 1,031,340 | 2,020,318 | 2,011,340 | 459,206 | 1.0 | 9.6 h | 12.26 |
| | | | | | | | | | 0.45 | 124.5 h | 11.9 |

horizon from 20 to 50 time steps and the workshop capacity from 10 to 20 parallel lines. The size of the resulting mixed-binary linear optimization model for the turn-around time contract is shown in Table 2, which also reveals that obtaining a feasible solution with a verified duality gap of 1% requires around 10 hours of computing time, while reducing the gap below 0.5% takes around 124 hours. While the smaller instances are solved to optimality in a reasonable computing time, the larger ones require a significantly longer computing time to reach a duality gap of 0.45%; details are listed in Table 2. Presolve times are approximately 0.45 seconds for the first and 12 seconds for the second instance reported in Table 2; hence, the solver quickly eliminates redundant variables and/or constraints.⁷

In terms of the specific application to maintenance of military aircraft, a real instance size would differ from the ones shown in Table 2. The number, I , of component types would be larger (typically in the range of 20–30), while the number, J_i , of individual components could be smaller than the considered value of 15 (since the components considered are usually quite expensive). Capacity in the maintenance workshop, L , could vary as well but most of the time, L is not very high. The fleet size (i.e. number of systems) would be in the range of $K = 10$ aircraft. The length of the planning horizon depends heavily on the length of each time step (e.g. one hour, one half day or one day). The use of our model in different applications, will result in different instance sizes. For example, rail traffic or commercial airlines instances would have a larger number of systems (i.e. train sets/aircraft). Our preliminary tests with the larger instance reveal that our mixed-binary linear optimization model is computationally demanding, especially for reasonably large instance sizes. One way to tackle this problem is to not keep track of the individual components $j \in \mathcal{J}_i$ in the systems, but consider the component types $i \in \mathcal{I}$ only. The advantage of such an approach will be a significant reduction of the problem size as well as of its computational complexity, and thus also a considerable reduction of computing times for corresponding instance sizes. On the other hand, since in the

turn-around time contract as modelled in (9), the due dates and turn-around times are defined per individual component, the disregard of individual components will call for a different modelling. For an availability contract, the approach of not keeping track of individual components is, however, tractable, since the availability of components is defined per component type. In Table 3, we present the size of the resulting mixed-integer linear optimization problem resulting from an aggregation over individual components (regarding all relevant variables in the model) and an availability contract. It is noticeable that the problem size as well as computing times have reduced significantly as compared to the mixed-binary linear optimization problem for the turn-around time contract (shown in Table 2).

4.4 Simulation of unexpected events in the workshop

Our model assumes deterministic processing times in the maintenance workshop. Uncertainty, such as unexpected events, may, however, affect the processing times. A reasonable situation is the following: when a component arrives at the workshop for repair it is first examined, at which occasion further damages may be detected which lengthen the processing time for the component repair. This scenario is incorporated in our model by a re-solution of the scheduling problem for the new processing time(s), from the point in time at which the damages were detected (i.e. the time step when the component at hand started repair in the workshop). The re-solution is performed whenever an unexpected event is revealed. The “complete” schedule employed is then composed by the computed part schedules, as defined by each consecutive pair of time steps at which a(ny) longer processing time(s) were detected. This consideration of uncertainty in the processing times is summarized in the rescheduling Algorithm 1.

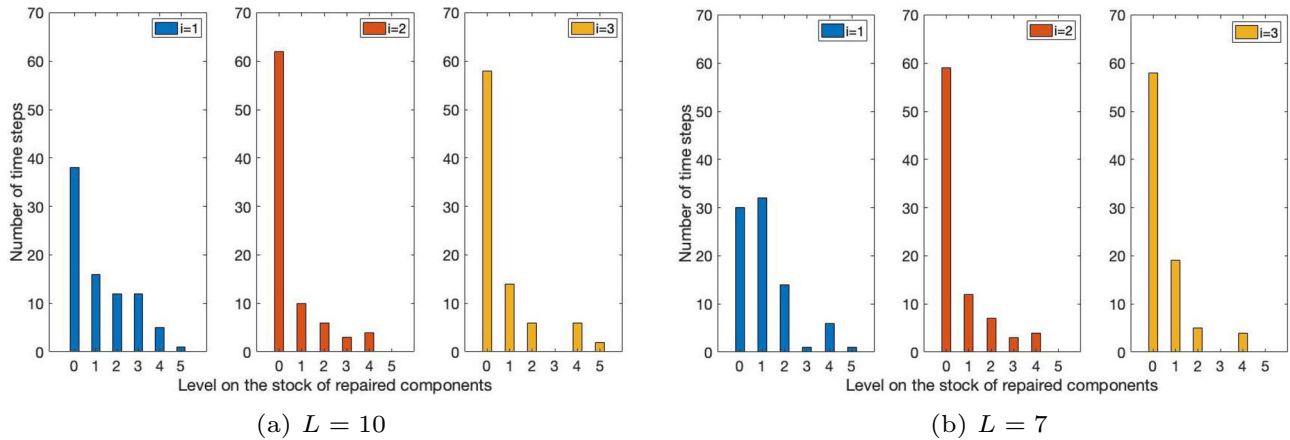
We define an event by the 4-tuple $\{\bar{t}, \bar{i}, \bar{j}, \bar{p}_{\bar{i}}^{\bar{i}\bar{j}}\}$, where $\bar{t} \in \mathcal{T}$ denotes the time step of the event, $\bar{i} \in \mathcal{I}$ denotes the component type, $\bar{j} \in \mathcal{J}_{\bar{i}}$ denotes the individual component, and $\bar{p}_{\bar{i}}^{\bar{i}\bar{j}}$ denotes the estimated processing time for the component individual (\bar{i}, \bar{j}) that started maintenance at time \bar{t} ; it is assumed that $\bar{p}_{\bar{i}}^{\bar{i}\bar{j}} \geq p^{\bar{i}}$ holds. An event at time \bar{t} is then con-

⁷ Note that the small differences in presolve times when solving one instance more than once come from slightly different solve times when solving the instance each time.

Table 3 Instance sizes and computing times for the availability contract with the aggregation over individual components

| Instance size | | | | | # Rows | # Variables | # Integer variables | | Gap (%) | Solution Time (s) | Presolve Time (s) |
|---------------|-------|-----|-----|-----|--------|-------------|---------------------|----------------|---------|-------------------|-------------------|
| I | J_i | K | T | L | | | Original | After presolve | | | |
| 3 | 15 | 10 | 20 | 10 | 2618 | 14,410 | 13,430 | 2048 | 0 | 0.1 | 0.05 |
| 5 | 15 | 10 | 40 | 10 | 8032 | 87,750 | 84,450 | 12,739 | 0 | 36.16 | 0.24 |
| 5 | 15 | 11 | 40 | 10 | 8637 | 96,405 | 92,895 | 14,129 | 0 | 19.89 | 0.09 |

The lower limit on the stock of available components is $\underline{b}^i = 1$, $i \in \mathcal{I}$, for all instances considered

**Fig. 8** Distribution of stock levels per component type $i \in \mathcal{I}$ on the stock of repaired components for capacities $L \in \{7, 10\}$ in the maintenance workshop

sidered as an unexpected event if it holds that $\bar{p}_i^{\bar{i}\bar{j}} \geq \bar{p}^i + \pi^i$, for some prespecified value of $\pi^i > 0$.

We next show some results obtained from the rescheduling opportunity in Algorithm 1. For demonstration and simplicity, we choose the availability contract with $\underline{b}^i = 0$, for each $i \in \mathcal{I}$. An analogous rescheduling can be done for a turn-around time contract. Four unexpected events in the maintenance workshop were sampled (i.e. four processing times were longer than expected), leading to five iterations of Algorithm 1. The resulting distributions of stock levels over time for repaired components are shown in Fig. 8, for the capacity in the maintenance workshop being $L = 10$ and $L = 7$, respectively. It is noticeable that for many time steps there are no repaired components on the stock (i.e. the stock being on its lower limit). In comparison with the stock of repaired components reported in Fig. 6a for $L = 7$, the stock levels are on average lower, which comes as a consequence of (some of) the processing times being longer. Moreover, we observe that a higher number of components on the stock occurs at only a few time steps.

The corresponding distributions for the active repair lines in the maintenance workshop are presented in Fig. 9. Setting the workshop capacity to $L = 7$ repair lines leads to the workshop working at its full capacity most of the time; see Fig. 9b. Increasing the number of repair lines to $L = 10$ yields more freedom in the workshop to distribute the workload, and the number of time steps at which the workshop operates at full capacity is reduced by more than a half; see Fig. 9a.

Algorithm 1 Rescheduling algorithm

- 1: Denote the time horizon $\mathcal{T} = \{t_{\text{start}}, \dots, t_{\text{end}}\}$ and initialize $t_{\text{start}} := 1$ and $t_{\text{end}} := T$
- 2: **repeat**
- 3: Solve the full model (Sect. 2.6) over \mathcal{T} and a specified objective (Sect. 3.2)
- 4: Sample an event in the maintenance workshop: $\{\bar{i}, \bar{i}, \bar{j}, \bar{p}_i^{\bar{i}\bar{j}}\}$ (such that $\bar{i} \geq t_{\text{start}}$)
- 5: Fix all variable values over time steps $\{t_{\text{start}}, \dots, \bar{i}\} \cap \mathcal{T}$ to the solution in step 3
- 6: Assign new values: $\bar{p}^i := \bar{p}_i^{\bar{i}\bar{j}}; \hat{t} := \min\{\bar{i}, t_{\text{end}}\}; t_{\text{start}} := \hat{t} + 1; t_{\text{end}} := \hat{t} + T$
 \Rightarrow updated processing time and time horizon \mathcal{T}
- 7: **until** [# iterations or $t_{\text{end}} \geq T_{\text{max}}$]

5 Conclusions and future research

The solutions resulting from our modelling can be used to find a lower limit for an optimal performance of a collaboration between stakeholders who govern a common system of systems. Moreover, our modelling enables an investigation of contracting forms between stakeholders. It also provides a planning tool for the case when the maintenance workshop and the system operator are integrated. We conclude that an availability contract is more computationally tractable than a turn-around time contract as it allows removal of individual components in the model, thereby availability contract type is more suitable for larger instances and real-world problems.

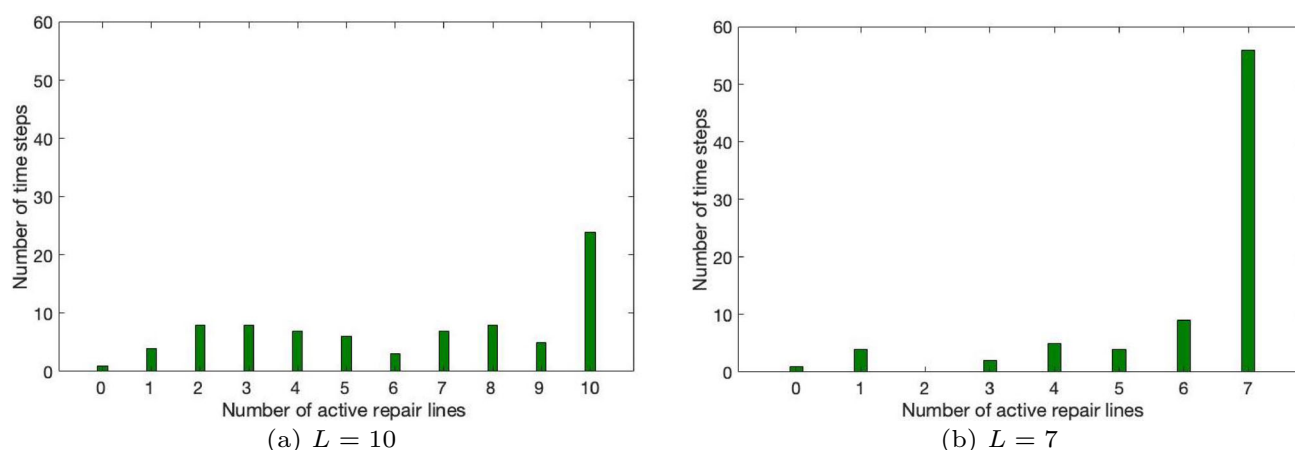


Fig. 9 Distribution of number of active repair lines in the maintenance workshop for total number of repair lines $L \in \{7, 10\}$

In our intended application, there are typically several optional maintenance workshops and/or maintenance companies, who may enter into the cooperation by means of different contracting forms. Taking these generalizations into account is a topic for further research.

We start from an NP-hard problem (PMSPIC), generalize it to consider individual components and incorporate the maintenance workshop, the stock dynamics, and the delay and availability objectives with this problem. Therefore, our problem has a high complexity and is computationally very demanding, as shown in Table 2. One important future research question is therefore how to reduce the computing times. For larger instances, our current model and solution approach are computationally intractable and are subject to further investigation and development, especially in the case of the turn-around time contract as defined in (9), which introduces non-binary coefficients in the constraint matrix. As mentioned in Sect. 4, considering component types only, and not individual components, will simplify the problem significantly, but in that case we will need to develop a new model to express the turn-around time objective and constraints.

Other means for reducing computing times include investigating the polyhedral properties of, as well as mathematical decomposition approaches for, the mixed-binary linear optimization problem formulated in this article.

Another extension, which is important for the intended application of this work, is to include corrective maintenance (CM), in terms of the risk of having to perform CM. At the current stage, the means to handle unexpected failures are to (i) reduce the risk for such failures by not allowing too large maintenance intervals (cf. the constraints (1e)) and (ii) reschedule the maintenance plan whenever an unexpected event occurs (see Sect. 4.4). Since short-term changes in the operational schedules for the systems, as well as in the schedules for the maintenance workshop, are often inconvenient and sometimes not even feasible, the rescheduling should (if

possible) be such that the solution remains fixed for a certain number of time steps.

Acknowledgements The research leading to the results presented in this paper was financially supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA; Project Number 2017-04879), Chalmers University of Technology, and Saab AB.

Funding Open access funding provided by Chalmers University of Technology.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Akyuz, G. A., & Erkan, T. E. (2010). Supply chain performance measurement: A literature review. *International Journal of Production Research*, 48(17), 5137–5155. <https://doi.org/10.1080/00207540903089536>
- Arkin, E., Jones, D., & Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2), 61–66.
- Boctor, F., Laporte, G., & Renaud, J. (2004). Models and algorithms for the dynamic demand joint replenishment problem. *International Journal of Production Research*, 42(13), 2667–2678.
- Boliang, L., Wu, J., Lin, R., Wang, J., Wang, H., & Zhang, X. (2019). Optimization of high-level preventive maintenance scheduling for high-speed trains. *Reliability Engineering & System Safety*, 183, 261–275.
- Brucker, P., & Knust, S. (2012). *Complex scheduling* (2nd ed.). Berlin: GOR-Publications. Springer.

- Conforti, M., Cornejo, G., & Zambelli, G. (2014). Integer programming. Graduate texts in mathematics, vol. 271. Springer, Switzerland.
- Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2), 295–320. <https://doi.org/10.1137/15M1020575>
- Ehrgott, M. (2005). *Multicriteria optimization* (2nd ed.). Berlin: Springer. <https://doi.org/10.1007/3-540-27659-9>
- 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. *European Journal of Operational Research*, 242, 631–643.
- Gurobi. (2020). Gurobi optimizer reference manual. <http://www.gurobi.com>.
- Gustavsson, E., Patriksson, M., Strömberg, A. Q. B., Wojciechowski, A., & Önnheim, M. (2014). The preventive maintenance scheduling problem with interval costs. *Computers & Industrial Engineering*, 76, 390–400.
- Julia. (2012). Version 1.5. Julia: A fast dynamic language for technical computing. <https://docs.julialang.org/en/v1/>.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1993). Chapter 9: Sequencing and scheduling: Algorithms and complexity. In: *Logistics of Production and Inventory. Handbooks in Operations Research and Management Science* (Vol. 4, pp. 445–522). Amsterdam: Elsevier.
- Lidén, T. (2020). Coordinating maintenance windows and train traffic: A case study. *Public Transport*, 12, 261–298. <https://doi.org/10.1007/s12469-020-00232-2>
- Mavrotas, G. (2009). Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213, 455–465.
- Papakostas, N., Papachatzakis, P., Kanthakis, V., Mourtzis, D., & Chrysosolouris, G. (2010). An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48, 604–612.
- Robert, E., Berenguer, C., Bouvard, K., Tedie, H., & Lesobre, H. (2018). Joint dynamic scheduling of missions and maintenance for a commercial heavy vehicle: Value of on-line information. *IFAC PapersOnLine*, 51(24), 837–842. 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFE-PROCESS 2018.
- Shafiee, M., Patriksson, M., & Strömberg, A. Q. B. (2013). An optimal number-dependent preventive maintenance strategy for offshore wind turbine blades considering logistics. *Advances in Operations Research*, 2013, 205847. <https://doi.org/10.1155/2013/205847>
- Verhoeff, M., Verhagen, W. J. C., & Curran, R. (2015). Maximizing operational readiness in military aviation by optimizing flight and maintenance planning. *Transportation Research Procedia*, 10, 941–950.
- Yu, Q., & Strömberg, A. Q. B. (2021). Mathematical optimization models for long-term maintenance scheduling of wind power systems. [arxiv:2105.06666](https://arxiv.org/abs/2105.06666).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.