Computers & Industrial Engineering 98 (2016) 30–39

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Overhaul planning and exchange scheduling for maintenance services with rotable inventory and limited processing capacity

Murat Erkoc [a,*], Kadir Ertogral [b]

[a] Department of Industrial Engineering, University of Miami, 1251 Memorial Drive, Coral Gables, FL 33146, USA
[b] Department of Industrial Engineering, TOBB University of Economics & Technology, Söğütözü Cad. No: 43, Ankara 06560, Turkey

ARTICLE INFO

Article history:
Received 27 November 2015
Received in revised form 10 May 2016
Accepted 11 May 2016
Available online 13 May 2016

Keywords:
Maintenance planning
Rotable inventory
Full-delay scheduling
Earliness
Aviation

ABSTRACT

Maintenance, repair and overhauling (MRO) of high cost equipment used in many industries are typically subject to regulations set by local governments or international agencies. For example in the aviation industry, critical equipment must be overhauled at certain intervals for continuing permission of use. As such, the overhaul must be completed by strict deadlines. Since the overhaul is typically a long process, MRO companies may implement exchange programs where they carry so called rotable inventory for exchanging expensive modules that require overhaul so that the equipment can continue its services with minimal interruption. The extracted module is overhauled in a capacitated facility and rotated back to the inventory for a future exchange. Since both the rotable inventory and the overhaul process capacity are limited, it may be necessary to carry out some of the exchanges earlier than their deadlines. Early exchanges results in a decrease in the maintenance cycle time of the equipment, which is not desirable for the equipment user. In this paper, we propose an integer programming model so as to minimize total earliness by generating optimal overhaul start times for rotables on parallel processing lines and exchange timetables for orders. We show that the LP relaxation of the proposed model has the integrality property. We develop a practical exact solution algorithm for the model based on a full-delay scheduling approach with backward allocation. The proposed procedure is demonstrated through both a numerical study and a case study from the airline MRO service industry.

1. Introduction

Industrial maintenance and repair includes all the technical and managerial activities carried out to keep a production or service resource available, functional, and safe. Industrial maintenance is especially important in capacity intensive industries with sizeable investments, such as airlines, wafer fabs, and railways. In airlines industry, in particular, maintenance repair and overhaul (MRO) operations are crucial for both ensuring safety and reducing operational disruptions. The maintenance activities are usually subject to regulations and scrutiny enforced by governments or international organizations. Both military and commercial aircrafts must go through MRO at certain intervals defined by either time or flight volume.

The effective management of MRO is not only important in regards to quality and safety, but also from the economic sustainability perspective of the airliners. MRO operations constitute a significant cost item especially in the aviation industry with a substantial business volume (Guajardo, Cohen, Kim, & Netessine, 2012). Therefore, airline companies focus on reducing MRO costs while ensuring that they do not compromise the safety of the airplanes that they operate. Airline companies either carry out MRO in their in-house facilities or outsource it to independent MRO service companies. An important MRO cost factor for airline companies is the disruption of the use of equipment during the MRO process. For example, in the commercial airline industry, the opportunity costs due to downtimes of the planes can amount from a few ten-thousands to hundred thousand dollars per day, depending on the type of the plane and its commercial use. Therefore, reducing turnaround time (TAT) is a key objective in MRO planning.

Inventory management is a crucial factor in MRO for reducing TAT. An important difference between a typical manufacturing system and a MRO system is the fact that MRO systems use so called rotable component inventories or modules, in addition to the regular inventory items (expendables). The rotable inventories consist of expensive components or modules that can be used as loan-outs or exchanges with the customers. The exchange minimizes the TAT

* Corresponding author.
E-mail addresses: merkoc@miami.edu (M. Erkoc), kertogral@etu.edu.tr (K. Ertogral).

http://dx.doi.org/10.1016/j.cie.2016.05.021
0360-8352/

for the airplanes as it only consists of the time spent for uninstalling the used module and installing the new module, without any delays due to other MRO processes. Here we do not consider uninstallation or installation activities of the components, assuming that these activities are carried out by a separate team or production unit, and we only tackle both scheduling of the overhauls and planning the exchanges of rotables.

The exchanged module that requires overhaul joins the MRO firm's rotable inventory and, directly or after some waiting period, enters the overhaul process. Once the overhaul process is completed for the component, it is ready to be used for a future exchange. Thus, the total number of rotable components in MRO system is always constant and a rotable component is in one of the three states; awaiting overhaul, undergoing overhaul process, and ready-to-exchange. The exchange and overhaul schedules determine the formation of these states for all rotables. On one hand, MRO companies aim to operate with low levels of the high-value rotable component inventories to avoid high costs. On the other hand, they must ensure adequate level of customer service.

As pointed out above, the interval times between consecutive MRO's for airliners are strictly regulated. These intervals often times are translated into hard deadlines for MRO services that cannot be violated. For example, the requirement to remove and overhaul a landing gear is every 8–15 years depending on aircraft's use and model. From the cost efficiency perspective, airline companies prefer using this interval times fully and do not want to stop the cycle shorter than the enforced duration. However, the MRO facilities may be compelled to ask some of their customers to bring their airplanes for MRO earlier than their end of cycles due to limited inventory and process capacity. Although feasible, this is not preferable for the airline companies. As such, MRO firms need to efficiently schedule their overhaul operations under their given rotable inventory and process capacity limitations, with the objective of minimizing the early exchanges. In this work, we precisely address this problem.

For a given set of required overhauls and their due dates by the airliners, we propose an integer programming model that minimizes the total earliness under rotable inventory and process capacity constraints. In this context, earliness is defined as the difference between the required exchange deadline (end of cycle) for a rotable in an airplane and the scheduled exchange date for that equipment. Ideally, there should be no gap between the deadlines and the actual exchange dates. However, due to limited number of rotables and processing lines, a gap may be inevitable to obtain a feasible solution where no exchange deadlines are violated. We show that the LP relaxation of the proposed mathematical model has the integrality property. As such, the optimal solution can be directly obtained from the LP relaxation of the model.

We also propose a practical exact solution algorithm based on a so called *full-delay* scheduling for the problem, which can be easily implemented without the need of using any mathematical solver or special computer software. We illustrate the algorithm's implementation using a real-life problem and present a sensitivity analysis that investigates the joint impact of capacity and rotable inventory on the optimal exchange and overhaul processing schedule.

In the next section, we discuss the relevant literature. In Section 3, we present the proposed mathematical programming model. We discuss the properties of the optimal solution and present our exact solution algorithm with an illustrative example in Section 4. In Section 5, we apply the solution algorithm on a case based on real life application from an MRO service company, and investigate the impact of rotable inventory and processing capacity on the optimal solution. We present our conclusions in Section 6.

2. Literature review

Over the past couple of decades a significant volume of research tackled problems regarding MRO management from a variety of viewpoints. Early work in MRO area is generally about planning activities for internal MRO needs in a company. Dekker (1996) summarizes this line of research. The author underlines that most of the models and solution approaches were introduced in 80s and 90s and estimates a growing role of MRO in the future due to increased use of high capital equipment both in manufacturing and service sectors. In a following work, Dekker and Scarf (1998) discuss applications of several models in MRO area and shows the importance of the optimization models in MRO in comparison to the use of more qualitative approaches. They demonstrate the power of analytical models needed to tackle complex problems prevalent in MRO management. A more recent review is reported in Nicolai and Dekker (2008).

Many studies regarding MRO in aviation in recent years focus on the MRO supply chains specifically pertaining to the spare parts and component inventory management. Related research tackles forecasting spare parts consumption (Ghobbar & Friend, 2003; Regattieri, Gamberi, Gamberini, & Manzini, 2005), inventory control with cost and downtime minimization (Safaei, Banjevic, & Jardine, 2011; MacDonnell & Clegg, 2011; van Jaarsveld, Dollevoet, & Dekker, 2012; Gu, Zhang, & Li, 2015), integrated rotable inventory control and overhaul capacity management (Buyukkaramikli, van Ooijen, & Bertrand, 2013), and strategies for inventory pooling and integration across aviation supply chains (Kilpi & Vepsäläinen, 2004; MacDonnell & Clegg, 2007; Kilpi, Töyli, & Vepsäläinen, 2009).

Our research in this study primarily focuses on MRO planning, where we explicitly incorporate capacity and inventory constraints into the optimal scheduling of the overhaul processes. We specifically consider rotables that can be exchanged with the components that require overhaul so as to eliminate downtimes. The most relevant works to the current study are due to Luh, Yu, Soorapanth, Khibnik, and Rajamani (2005), Joo (2009), Joo and Min (2011), and Arts and Flapper (2015). Luh et al. (2005) suggest a mathematical programming approach that integrates scheduling exchanges and inventory control problems. The solution approach is a Lagrangian relaxation scheme. Unlike our model, in this study the due dates of the overhaul requests are taken as the arrival times, and they did not consider early arrivals of requests. The scheduling part of their work concentrates on planning service activities in order to meet the requests. Again unlike our model, their approach allows delays in satisfying a request with an associated penalty. However, the general tendency in aviation MRO practice is to minimize the delays in meeting requests through exchange policies. Joo (2009) tackles the problem of scheduling exchange of a single type rotable with the objective of minimization of the earliness. Similar to our setting, the rotables are scheduled to be exchanged with rotable components for a given a set of aircraft fleet of similar type with overhaul deadlines. The deadlines are not allowed to be violated to avoid downtimes. As such, the exchanges are to be scheduled no later than their respective deadlines. For a given initial inventory of rotables, the author proposes a polynomial time algorithm that optimally schedules the exchange times and overhaul processes. The generation of the schedule is constrained by the availability of rotable inventory level only. However, his model implicitly assumes unlimited number of processing lines and thus, unlimited maintenance capacity. In another paper, Joo and Min (2011) extend this model by incorporating rotable inventory decision by means of a budget constraint. They employ a goal programming approach, where the due date satisfaction and budget constraints are taken as soft constraints. The resulting

multi objective model is solved directly with a commercial solver. They use weighted objectives aiming at minimization of the violation of these soft constraints. In our study, for practical relevance, we explicitly incorporate the capacity constraint into the scheduling problem and propose an exact solution procedure for the exchanges and the overhaul processes.

In a more recent study, Arts and Flapper (2015) introduce an aggregated production planning model which considers both the rotable inventory and workforce planning decisions in a long term planning context. The model uses two time buckets; month and year. The workforce capacity levels can only be changed at the beginning of the years with an associated cost. The model does not consider exact timing of the rotable requests but rather it focuses on satisfying monthly total demand by varying the workforce levels. The objective of the model that the authors employ minimizes the sum of several cost factors, such as labor, material, and inventory costs, throughout the life cycle of the fleet of vehicles, whereas we focus on earliness. The model tries to determine the workforce levels for each period in order to provide the total required capacity for maintenance operations, and not the exact scheduling of the overhauls and the exchanges for each individual request. Compared to this setting, our model provides a lower resolution operational scheduling framework that tries to meet each individual request on time with the total earliness objective and takes teams of workers as a single unit process capacity.

3. Problem formulation

We consider a MRO company that adopts an "exchange" program to deliver the overhaul services to its customers. There has been a growing trend in recent years for airlines and MRO companies establishing exchange programs where the equipment owner exchanges its equipment that needs overhaul with a recently overhauled equipment from the MRO company's rotable inventory, at a specific date which is no later than the customer's deadline (Mwanalushi, 2012). Determining the exchange date depends on the company's available rotable inventory stocks as well as the customer's deadline. Following an exchange, each received equipment set enters the overhaul process and becomes ready for the next exchange on completion of its overhaul. The turnover time depends on the overhaul processing time and the available capacity. Ideally, an exchange takes places exactly at the company's deadline. However, limited levels of rotable stocks and capacity may compel the MRO company request that some of the orders to arrive earlier for the exchange.

We propose a mathematical optimization model that minimizes the total earliness of exchange dates. We assume that there are known set of customer orders with due dates, overhaul process times, capacity, and rotable inventory. Table 1 lists the nomenclature employed in the proposed model. All orders are for the same rotable type, and each rotable module requires an overhaul processing time denoted by p. In this setting, we consider a line corresponds to a team of special workers handling explicitly an overhaul for a particular high-value rotable component, which constitutes a large portion of the overall MRO operations, such as overhauling a special type of landing gear. The number of rotable inventories carried by the company is represented by s. As mentioned above, each time an exchange takes place, the received equipment needs to be overhauled. The MRO firm's capacity dedicated for the product type is limited by the number of parallel lines, each of which can process one module at a time and no preemption is allowed during the process. As such, the overhaul process of a particular rotable inventory item can only start when at least one of the parallel lines is available.

At any given period, inventory is composed of three groups depending on the state of the module at hand; 1. *Exchange-ready inventory* is the group of equipment whose overhaul is completed and ready-to-go. 2. Equipment modules that are currently undergoing the overhaul process make up the *in-process inventory*. 3. The inventory received from the customer following an exchange awaiting overhaul. We refer to this group as the *on-hold inventory*. A given rotable inventory is in this state either because there is no available processing line for its overhaul or its scheduled start time is delayed in the production schedule.

As we mention above, at a given period t the rotables are composed of exchange-ready inventory (denoted by H_t), on-hold

Table 1
Nomenclature.

Sets:	
I	Set of exchange orders for the rotable module
T	Set of time periods in the planning horizon
Parameters:	
s	Initial number of the rotable modules in inventory
k	Number of parallel overhaul lines
p	Number of periods to complete an overhaul process for a rotable module
d_i	Due date for the exchange of the ith order
Decision variables	
X_{it}	1 if the exchange for order i is made at period t and 0 otherwise
Y_t	Number of overhaul starts at the beginning of period t
H_t	Inventory ready for exchange at the end of period t
B_t	Inventory of rotables waiting for overhaul at the end of period t
W_t	Inventory of rotables being overhauled at period t

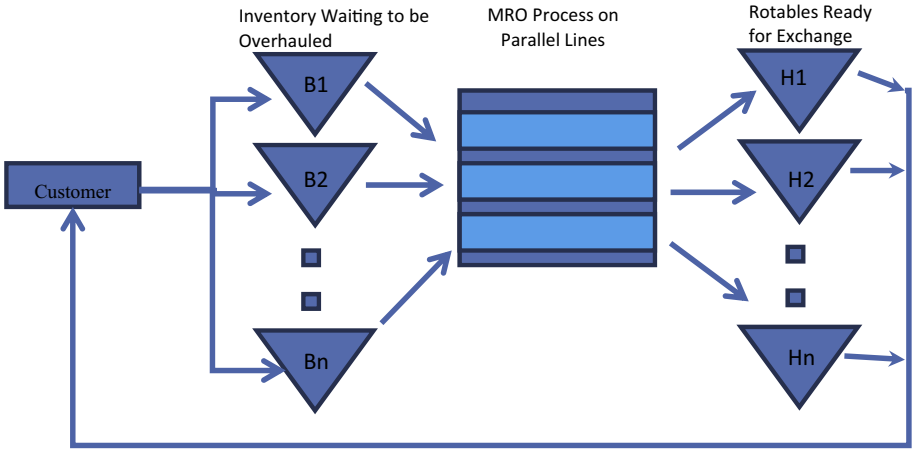

Fig. 1. Flow of rotables.

inventory (denoted by B_t), and in-process inventory (denoted by W_t). Clearly, at any period it holds that $S = H_t + B_t + W_t$. The flow of rotables is illustrated in Fig. 1. In a period, the spread of the inventory can be changed by three types of events. First, an equipment exchange may take place ($\sum_{i\in I} x_{it}$). The number of exchanges is limited by the exchange-ready inventory level. The equipment received from the customer(s) is either added to the on-hold inventory or enter into the overhaul process directly. The module (s) that enter into the overhaul process (y_t) are deducted from the on-hold inventory. Clearly, the size of this group is limited by the number of available process lines (*i.e.*, $W_t \leqslant K$). The third event that leads to the replenishment of the exchange-ready inventory occurs when any overhaul processes started exactly p periods earlier are completed (y_{t-p}). As in the previous case, this number is also limited by the number of available process lines.

The setting explained above indicates that the overall timeliness of equipment exchanges are mainly constrained by the number of equipment sets that the service company rotates its inventory (s) and the number of the processing lines (k) that it operates. As expected, higher inventory leads to improved timeliness for exchanges. Moreover, the throughput of the rotation is determined by the number of processing lines. An efficient schedule of the overhaul operations and the exchange dates should be generated within these operational constraints.

The optimization model presented below aims to minimize the total earliness;

$$(\boldsymbol{P}) \quad \min \sum_{i\in I}\sum_{t=1}^{d_i}(d_i - t)x_{it} \tag{1}$$

$$\text{s.t.} \quad \sum_{t=1}^{d_i} x_{it} = 1, \quad \forall i \in I \tag{2}$$

$$B_t = B_{t-1} + \sum_{i\in I} X_{it} - W_t, \quad \forall t \in T \setminus \{0\} \tag{3}$$

$$H_t = H_{t-1} - \sum_{i\in I} X_{it}, \quad \forall t \in T \setminus \{0\} : t < p \tag{4}$$

$$H_t = H_{t-1} - \sum_{i\in I} X_{it} + W_{t-P}, \quad \forall t \in T : t \geqslant p \tag{5}$$

$$\sum_{l=t}^{t+p-1} W_l \leqslant k, \quad \forall t \in T : t \leqslant |T| - p + 1 \tag{6}$$

$$\sum_{i\in I} X_{it} \leqslant H_{t-1}, \quad \forall t \in T : t < p \tag{7}$$

$$\sum_{i\in I} X_{it} \leqslant H_{t-1} + W_{t-P}, \quad \forall t \in T : t \geqslant p \tag{8}$$

$$H_0 = s \tag{9}$$

$$B_0 = 0 \tag{10}$$

$$X_{it} \in \{0,1\}; \quad W_t, H_t, B_t \geqslant 0 \quad \text{and integer} \tag{11}$$

The objective function given in (1) returns the sum of the difference between the time of exchange and the deadline for each exchange. The summation gives the total earliness. First constraint ensures that all exchanges take place before their deadlines. Constraints (3)–(5) establish the flow balance for the on-hold and finished inventories. While any exchanged equipment joins the on-hold inventory waiting to be overhauled, the overhaul starts are removed from on-hold inventory and join the in-process inventory. At the same time, finished inventory level decreases when an exchange takes place and increases when an overhaul process is completed for any in-process module. Inequality (6) forms the capacity constraint. It requires that the number of overhaul starts during any time interval with length of the unit overhauling time (*i.e.*, p) cannot exceed the number of parallel process lines (k). In other words, the constraint ensures that the maximum in-process inventory level is limited by the number of processing lines at any period during the planning horizon.

Constraints (7) and (8) limit the number of exchanges for any given period by the finished inventory level which contains exchange-ready modules. Constraints (9) and (10) set the initial conditions for the finished and on-hold inventories, respectively. Without loss of generality, it is assumed that all rotable inventory items are ready for exchange at the beginning of the planning horizon.

The model employs a constant processing time for all rotables since a single job type is assumed. One can see that we can have some deviations in overhaul times due to randomness of the nature of the work content. Our model assumes this randomness is not significant. Even if we have high variation in overhaul times, the results of the model are still valuable due to the fact that what determines the completion time of a particular overhaul is really the accumulation of the overhauls times executed in a process lines, not an individual overhaul time, and coefficient of the variation of the total overhaul time will get smaller as the number of overhauls increases.

One major advantage of the above model is that it has the integrality property as shown below:

Proposition 1. *LP relaxation of model* ***P*** *has the integrality property.*

Proof. Let A be the matrix of coefficients for the decision variables in the constraints of **P** with relaxed integrality restrictions. Every entry of A is -1, 0, or +1, and therefore, all square sub matrixes of A have a determinant of 0, +1 or -1 implying that A is totally unimodular. Hence, every basis of A has a determinant of -1 or +1. Since the right hand side vector of the model is composed of integers, we conclude that all extreme points of set A will correspond to an integer basic feasible solution. □

The above result reveals that the optimal solution to the Linear Programming (LP) model obtained by relaxing the integrality constraints in (11) is feasible and optimal for **P** as well. As such, the proposed model is effective in finding optimal solutions with computational efficiency.

Next, we introduce a special algorithm that can be used to find the optimal solution to the above model. The advantage of the algorithm is that it does not require any mathematical solver or special computer software and it can be easily implemented using simple scripts or cataloging. Moreover, the algorithm does not require much data processing and memory which would be the case with the mathematical model especially for large size instances.

4. The exact solution algorithm

When process capacity is ample, that is, when the number of parallel processing lines exceeds the number of rotables at hand, the above problem can be solved easily (*i.e.* $k \geqslant s$). Joo (2009) proposes a polynomial time exact solution algorithm for this case. The algorithm is based on a backward allocation where the equipment exchange with the latest due date is scheduled first. In this setting, all exchanges are scheduled as close to their due dates as possible, and each exchange is assigned to a certain rotable in the exchange-ready inventory. Some of the exchanges must be scheduled early in order to make the timely (*i.e.*, feasible) delivery of the subsequent exchanges possible.

Having $k \geqslant s$ is equivalent to the uncapacitated problem since we cannot simultaneously process more than s rotables at any given period. However, it typically is more common to have $k < s$. Since establishing each overhaul process line would mean both making high investments for special equipment and machinery and keeping a team of skilled workers with high salaries on payroll, it makes financially much more sense to have some extra inventory

in order to avoid delays in exchanges. In our study, we propose an optimal solution methodology for this more general case where the process capacity is important. For $k < s$, the exchange process is not only constrained by the availability of the rotables but also by the available processing lines. Therefore, the timing of the exchanges should be determined based on the processing capacity in addition to availability of the exchange inventory. In what follows, we discuss the properties of the problem which help us construct an efficient exact solution procedure for scheduling module exchanges and their designated MRO processes optimally.

4.1. Problem properties

We present two essential optimality properties. The proposed algorithm is developed based on these properties. In describing the properties below, we use the term "partial schedule". The partial schedule is the incomplete schedule where only a subset of the jobs is allotted to the processors. It is generated by assigning available rotable inventories to the requests from latest to the earliest in their due dates, while scheduling overhauls of rotables in a backward fashion in time.

Proposition 2. *In a partial schedule, we do not violate the optimality of a solution if we always assign an available rotable to the latest unsatisfied request whose due date is greater than or equal to the period the rotable becomes available.*

Proof. Consider a partial schedule constructed in a backward fashion from right to the left in time. Let f_i be the overhaul finish time for a last overhaul (i) in a partial schedule and f_{i-1} the overhaul finish time for the rotable that comes second to the last overhaul. Also, let d_k and d_{k-1} be the due dates of the latest unsatisfied request (k) and the second latest unsatisfied requests ($k-1$), respectively ($d_k \geqslant d_{k-1}$). There could be two cases regarding the finish time of the last overhaul;

Case I: $d_{k-1} \leqslant f_i \leqslant d_k$

In this case, it is obviously optimal to assign the rotable to the latest request in order to minimize the total earliness.

Case II: $f_i \leqslant d_{k-1} \leqslant d_k$

In this case, we can either assign the rotable to open request k or $k-1$. If we assign it to open request k the marginal contribution of this assignment ($mc1$) to the total earliness objective will be $mc1 = (d_k - f_i) + (d_{k-1} - f_{i-1})$. If, on the other hand, we assign the rotable to request $k-1$, the marginal contribution of the assignment will be $mc2 = (d_{k-1} - f_i) + (d_k - f_{i-1})$. It is easy to see that, since the processing times are identical, $mc1 = mc2$. Therefore, in this case as well, we would not violate optimality if we assign any available rotable to the latest unsatisfied request. □

Based on the proposition we conclude that we must consider the exchange requests from latest to earliest when we assign available rotable inventory as we construct the schedule in a backward fashion.

Proposition 3. *In a partial schedule, delaying an overhaul of a rotable as late as possible without violating the processor constraint so that the finish time of the rotable is closest to the due date of the latest unsatisfied request, does not violate the optimality of the partial schedule.*

Proof. Let f_i be the overhaul finish time for a last overhaul i in a partial schedule and let r_k be the due date of the latest unsatisfied requests k in a partial schedule ($f_i < r$). In a given partial schedule, let h be the first point in time in between fi and r_k such that the number of processors busy drops to $K-1$, i.e. one processor becomes available at h. We can have two cases;

Case I: $r_k - h < p$

In this case we cannot shift the start of overhaul any further to the right in the schedule because of processor availability.

Case II: $r_k - h \geqslant p$

In this case, shifting the overhaul to the right as much as possible and setting $f_i = r_k$ will improve the total earliness. □

Based on Proposition 3, we try to construct so called *full-delay* schedules where no overhaul can be shifted to the right without affecting the schedule of other overhauls. Optimality of the solution algorithm is based on fact that the suggested backward scheduling procedure uses the moves that do not violate the optimality based on the last two propositions In other words, the procedure repeats the moves in line with two propositions. Therefore the moves do not cut out the optimal solution (or solutions) and the final solution found by applying these moves is guaranteed to be optimal, if the solution is feasible. A constructed solution is feasible if the start time of the first overhaul is greater than or equal to the first period in the planning horizon.

The solution algorithm considers the requests one by one from latest to the earliest in a backward fashion. It tries to satisfy the latest open request first by an overhaul which finishes as close as possible to the due date of the latest unsatisfied request. Therefore, if it converges to a feasible solution, the converged solution must be an optimal solution. We give the details of the *full-delay* scheduling algorithm below.

4.2. Full-delay scheduling algorithm

In this section, we introduce a single-pass constructive polynomial-time algorithm that can be used to obtain the exact solutions for real-life size problems. The approach to construct the schedule of overhauls and exchanges is based on two properties in line with the propositions given above;

(1) Schedule the exchanges from latest due date to the earliest and designate each overhaul of a rotable for a particular exchange order.
(2) Schedule the start of the designated overhaul as late as possible without violating the exchange due dates and the capacity constraints.

We refer to this proposed algorithm as *Earliest Due-Date Full-Delay* (*EDF*) scheduling, where beginning with the exchange order with latest due date, overhaul task for the designated rotable inventory to be used in the exchange is always scheduled at the latest possible time slot on one of the parallel processors. Full-delay ensures that all exchanges are carried out as late as possible but before their due dates. Since the overhaul times are identical, the optimal solution is guaranteed.

To help explain the proposed full-delay scheduling algorithm, we use a numerical example. The example includes 13 overhaul exchanges with due dates in days as given in Table 2. The overhaul process for each item (p) is 30 days. Suppose the MRO company carries four rotable items as initial inventory ($S = 4$) and has two parallel lines ($K = 2$) to carry out the overhaul processes.

Before the execution of the scheduling algorithm, all exchange orders are ordered from the latest due date to the earliest. At each iteration, the overhaul task for an exchange order with the latest deadline is scheduled. In this respect, the scheduling follows a backward process. In the proposed optimization algorithm, orders are indexed in descending order of their due dates (*i.e.*, $d_i \geqslant d_{i+1}$). Let L_v denote the scheduled overhaul start time for the rotable inventory item v ($v = 1, \ldots, S$) in a partial schedule at any iteration. Initially, we set the overhaul start times for all inventory equal to the latest due date in the set of all orders (or any value greater than

Table 2
Exchange due dates for the numerical example (in days).

Order #	1	2	3	4	5	6	7	8	9	10	11	12	13
Due date	215	192	176	164	152	150	150	137	124	104	81	81	77

the due date of the last order). We let EARLY represent the running total for the earliness. We let U_i and E_i denote the start time of overhaul process and the time of exchange for the rotable to be used to satisfy exchange order i, respectively. Moreover, we define M_j as the last scheduled overhaul start time on processor j in the partial schedule already constructed in any iteration. The initial step of the algorithm is given below:

Step 0: ***Initialize***
if $k > s$ *then* $k \leftarrow s$
$d_{max} \leftarrow d_1$
for $v = 1$ *to* s $\{L_v \leftarrow d_1\}$
EARLY = 0

In the initialization step, we make sure that the inventory is overhauled on at most min(S,K) parallel lines. Clearly, when we have more lines than inventory, the excess lines cannot be used since there will not be enough rotable modules to process. We recall that this case corresponds to the uncapacitated setting discussed in Joo (2009), where constraint (6) is redundant.

Following the full-delay approach, the schedule is constructed in a way that the exchange periods for the first s orders (s orders with the latest due dates) are allocated to their respective due dates in a backward fashion. Therefore, once the initialization is completed, we construct the schedule in two main phases. We schedule the exchange of the first S orders just on time. For this, we first schedule the overhaul processes for all k process lines for the first k orders ($i = 1, \ldots, k$) and then schedule the overhaul processes for the following $s - k$ orders ($i = k + 1, \ldots, s$). After scheduling the overhauls for the first S exchange orders, we schedule the overhauls for the remaining ($N - S$) orders as described below.

Step 1: ***Schedule the overhaul starts and exchanges for orders 1 to k***
for $i = 1$ *to* k $\{E_i \leftarrow d_i;\ U_i \leftarrow E_i - p;\ M_i \leftarrow U_i;\ L_i \leftarrow U_i\}$

In this step all orders are exchanged on time with no earliness, and the algorithm schedules the corresponding overhaul processes in a way that they are completed just on time for the exchanges. To illustrate this, consider the example given in Table 2. At the end of the first step, exchange for the first two orders are scheduled (since $k = 2$), where $E_1 = 215$ and $E_2 = 192$. Hence, the corresponding overhaul starts are set at $U_1 = 215 - 30 = 185$ and $U_2 = 192 - 30 = 162$ on processing lines 1 and 2, respectively. Consequently, in the partial schedule, the processing lines 1 and 2 become busy at $M_1 = 185$ and $M_2 = 162$, respectively. Likewise, the latest overhaul start times for rotables 1 and 2 are $L_1 = 185$ and $L_2 = 162$, respectively.

At this point since we still have $s - k$ available rotables, next $s - k$ orders can also be scheduled for on time exchanges. Let $j_{\max}$ represent the processing line with the highest scheduled processing start time in the current partial schedule, that is, $j_{\max} = \text{-argmax}_{j\in[1,K]}\ (M_j)$. The next overhaul start is always allocated to the line $j_{\max}$ under the adopted full-delay regime. For example, in the partial schedule of our numerical example $j_{\max} = 1$ because the last scheduled overhaul process on this line has larger start time than that of the other processing line.

Step 2: ***Complete the scheduling for orders K + 1 to S***
for $i = k + 1$ *to* s $\{E_i \leftarrow d_i;\ U_i \leftarrow min(E_i, M_{jmax}) - p;\ M_{jmax} \leftarrow U_i;$
$L_i \leftarrow U_i\}$

We note that at any time no more than K modules can be in the overhaul process. As such, the overhaul process of a rotable inventory designated for an order can start just in time (*i.e.* $U_i = E_i - p$) if and only if there is an available processing line at that time. Otherwise, the start time is shifted back to the latest possible period which is $M_{jmax} - p$. M_{jmax} is updated each time an overhaul process is scheduled.

Following Step 2, in our example, the exchange dates for orders 3 and 4 are scheduled on time using rotables 3 and 4, respectively. The overhaul corresponding to order 3 is scheduled on processing line 1 since this line has the largest start time in the partial schedule. The overhaul is scheduled to start at $U_3 = \min(176, 185) - 30 = 146$ and thus, $L_3 = 146$. At this point, the latest scheduled overhaul start time on this processing line becomes 146 as well (*i.e.*, $M_1 = 146$). Consequently, now $j_{\max} = 2$ since $M_2 > M_1$. Next, overhaul process associated with order 4 is scheduled with $U_4 = \min(164, 162) - 30 = 132$, $L_4 = 132$, and $M_2 = 132$. At this point, $j_{\max}$ becomes 1 since now $M_1 > M_2$. While the overhaul process is completed just in time for the exchange of order 3, it needs to finish before the date of the exchange for order 4 because the processing line will be busy with another overhaul process at the time of the exchange.

After overhaul start and exchange periods for the first S rotables are determined, the remaining $N - S$ orders are scheduled continuing with the backward process. For the remaining iterations, let $v_{\max}$ represent the rotable inventory item with the highest process start time in the current partial schedule, that is, $v_{\max} = \text{-argmax}_{v\in[1,S]}\ (L_v)$.

Step 3: ***Complete the scheduling for the remaining N − s orders***
for $i = s + 1$ *to* $N - s$ $\{E_i \leftarrow min(L_{vmax}, d_i);\ U_i \leftarrow min(E_i, M_{jmax}) -$
$p;\ M_{jmax} \leftarrow U_i;\ L_{vmax} \leftarrow U_i;$
if $L_{vmax} < 0$ then break: solution is infeasible
EARLY = EARLY + $(d_i - E_i)\}$
for $i = N - s$ *to* N $\{E_i \leftarrow min(L_{vmax}, d_i);\ U_i \leftarrow min(E_i, M_{jmax}) - p;$
$M_{jmax} \leftarrow U_i;\ L_{vmax} \leftarrow U_i;$
EARLY = EARLY + $(d_i - E_i)\}$

We note that there is no need for overhaul scheduling for the last S orders. Since these orders have the earliest s due dates, each corresponds to the initial exchange for each of the rotable items, which are assumed to be ready for exchange at the beginning of the planning period. However, a virtual overhaul schedule is nevertheless generated for these orders so as to update $v_{\max}$ values and hence, their exchange dates. Therefore, feasibility check is needed for only orders $s + 1$ thru $N - s$, not for the last s orders.

In order to ensure full-delay, in Step 3, the algorithm selects the rotable with the latest scheduled overhaul start in the partial

schedule to determine the next overhaul start and exchange dates. At this stage, exchange dates are constrained by the availability of both the rotable items and the processing lines. The exchange for order i must be carried out at an earlier date if on the day the exchange is due the rotable item is already assigned to an overhaul process corresponding to another order, that is, $L_{vmax} < d_i$. This can be illustrated in Table 3, which recaps the overall scheduling process for our numerical example. Orders 5 and 6 can be exchanged on time since the overhaul processes for rotable items 1 and 2 can be started early enough. On the other hand, the exchange of order 7 must be scheduled earlier than its due date – day 150 – since the rotable item 3, which is used for this exchange, is scheduled for overhaul start on day 146. The overhaul start scheduled for day 146 is associated with the exchange of order 3. As such, the exchange of order 7 cannot take place after day 146. Consequently, the exchange is scheduled for day 146, 4 days preceding the due date of the order.

Fig. 2 provides a visual depiction of the exact solution generated by the proposed algorithm for the numerical example. Note that, in Fig. 2, each row corresponds to a rotable inventory item and no more than two schedule blocks can overlap on a column since there are two processing lines. The minimum total earliness for this problem results with 49 days. We note that the left-most schedule blocks (overhauls) for the rotable items are not actual overhaul processes since the rotables are ready for exchange and the overhauls are not necessary prior to the first four exchanges.

Table 3
Completed schedule for the numerical example.

Order	Due date (d_i)	Exchange date (E_i)	Overhaul start (U_i)	Process line (j)	Rotable item (v)	Earliness
1	215	215	185	1	1	0
2	192	192	162	2	2	0
3	176	176	146	1	3	0
4	164	164	132	2	4	0
5	152	152	116	1	1	0
6	150	150	102	2	2	0
7	150	146	86	1	3	4
8	137	132	72	2	4	5
9	124	116	56	1	1	8
10	104	102	–	–	2	2
11	81	81	–	–	3	0
12	81	72	–	–	4	9
13	77	56	–	–	1	21

5. A case study: MRO exchange schedule for landing gear

In this section, we apply the proposed algorithm to a real life case in order to illustrate our solution methodology. We carry out a sensitivity analysis to investigate the impact of rotable inventory and processing capacity on the performance of the optimal schedule. The case is built based on our research collaboration with an industry partner who is among the top ten MRO companies in the World operating in the airline industry. The name of the company as well as the specifics of the module type is not disclosed in this study for privacy concerns. We focus on an exchange program of the MRO program for landing gear of a specific regional jet type. A regional jet (RJ) is a term used for a class of short to medium-

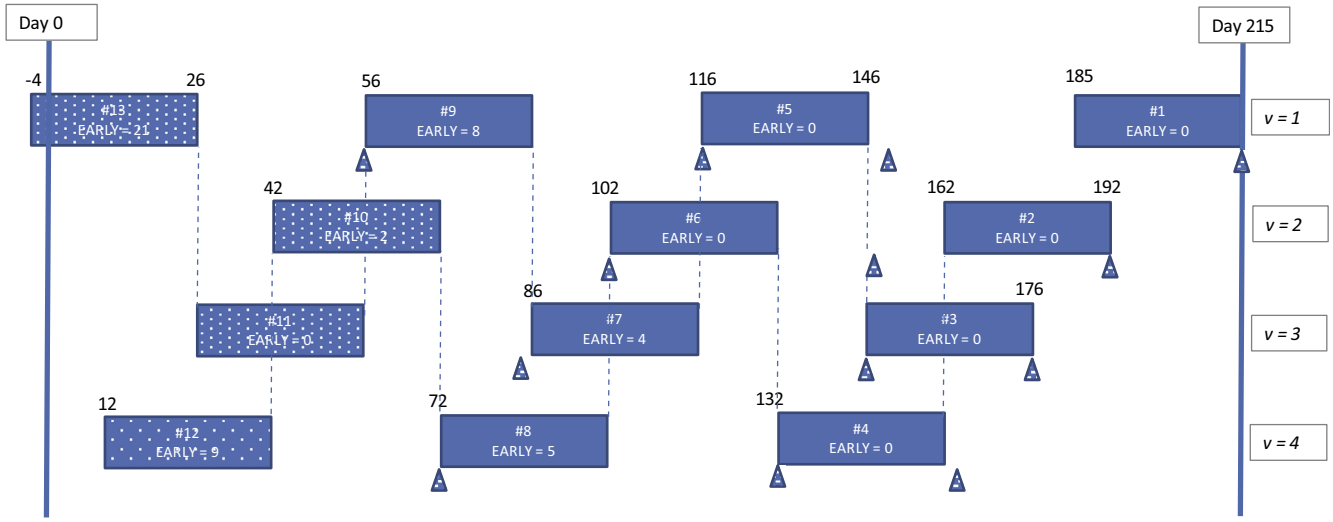

Fig. 2. Final schedule of overhaul processes and exchanges with two lines and four rotable inventory items (exchange days are indicated with Δ).

Table 4
Exchange due dates for the case study (in days).

Order	Due Date	Order	Due Date	Order	Due Date	Order	Due Date	Order	Due Date	Order	Due Date	Order	Due Date	Order	Due Date
1	1829	11	1707	21	1485	31	1402	41	1331	51	1252	61	623	71	198
2	1806	12	1705	22	1481	32	1400	42	1307	52	1232	62	543	72	198
3	1790	13	1699	23	1481	33	1400	43	1303	53	1230	63	540	73	198
4	1768	14	1594	24	1478	34	1380	44	1302	54	1218	64	530	74	170
5	1766	15	1549	25	1455	35	1379	45	1295	55	847	65	513	75	78
6	1744	16	1542	26	1441	36	1377	46	1285	56	798	66	482	76	78
7	1710	17	1532	27	1428	37	1363	47	1282	57	758	67	467	77	75
8	1709	18	1523	28	1414	38	1345	48	1273	58	756	68	460	78	75
9	1708	19	1521	29	1405	39	1343	49	1271	59	711	69	439	79	48
10	1708	20	1518	30	1403	40	1337	50	1256	60	643	70	426	80	48

range turbofan powered airliners. The company has MRO contracts with several customers who operate this type of jets. The contracts stipulate a total of 80 exchanges with customer specified due-dates over a 5-year time horizon. As mentioned earlier, since these due dates are typically governed by the Federal Aviation Administration (FAA) regulations, all due-dates are considered as firm deadlines. Overhaul time is given as 30 days.

We normalize the due-date data with respect to time 0 (now). The due date data are listed in Table 4 in descending order and also depicted in Fig. 3. We apply our solution algorithm to this data set under varying numbers of initial rotable inventory and processing lines. Specifically, our analysis varies number of rotables, s, from 2 to 9 and employs the same range for the number of processing lines, k. We note that with only one rotable, there is no feasible solution for any k in $\{2 \ldots 9\}$. Likewise, with a single processing line, no feasible solution exists for any s in $\{2 \ldots 9\}$. As such, we take the minimum number of rotables and processing lines as 2.

For the case problem, each instance involves 95,651 decision variables and 7369 constraints. The resulting optimal total earliness values are presented in Table 5.

As expected, the total earliness is nonincreasing in both the number of rotables and the processing lines. The data reveals that the total earliness does not improve beyond a certain number of processing lines for a given number of rotables. This is obvious for the case $k \geqslant s$, where the extra processing lines cannot be used since the simultaneous overhauling processes are limited by the number of rotables. As such, the optimal solution for these cases will not change and will be identical to the case with $k = s$. However, we observe that capacity saturation may occur even when $k < s$, that is when there are more rotables than the processing lines. For example, with 6 rotables, the total earliness converges to 29 days for 4 processing lines and does not improve with any additional line. Such is also the case with 7 or more rotables.

As exemplified in Fig. 4, the return from each additional rotable diminishes. That is, the level of drop in earliness gets smaller as we increase the number of rotables. Same is also the case for processing lines. As expected, when both the number of rotables and the number of processing lines are sufficiently high, total earliness converges to zero. With four or more processing lines, the total earliness goes down to zero with 8 rotables. It would take 19 and 10 rotables with 2 and 3 processing lines, respectively, to reach at zero earliness. From these results, we observe that while total earliness may not always converge to zero with increasing number of processing lines, it always does so when the number of rotables gets sufficiently high for a fixed number of processing lines, which is in line with the intuition.

The marginal impact of rotables and processing lines on the total earliness differ and are context specific. The gain from additional processing line does not always surpass the gain from an additional rotable inventory. Comparison of the marginal gains is illustrated in Table 6 with three types of entries. A cell with "L" indicates a higher gain to be realized by increasing the number of processing lines, whereas "R" signifies that it is preferable to increase the rotable inventory. The entry "N" represents the cases where the total earliness is zero and as such, no gain is realized by increasing either the number of rotables or the number of processing lines. For example, when the company has five rotables that can be overhauled on three lines, the decrease in total earliness will be higher when an additional processing line is used (*i.e.,* from 659 days to 118 days) compared to acquiring an additional rotable (*i.e.,* from 659 days to 411 days). Then with four processing lines, the company is better off with an additional rotable (a decrease from 118 days to 29 days). It is intuitive that as the company has more rotables in its inventory, the marginal gain from additional processing lines increases since the processing lines becomes the bottleneck. On the other hand when the number of processing lines is high enough, rotable inventory becomes the bottleneck.

The comparison of the impacts of the rotables and the processing lines is the typical comparison between inventory and capacity. Clearly, both have impact on the performance in any production or service operations and either one may become a bottleneck. To gain further insights about the tradeoff between inventory and capacity, we solve the same problem set with longer and shorter

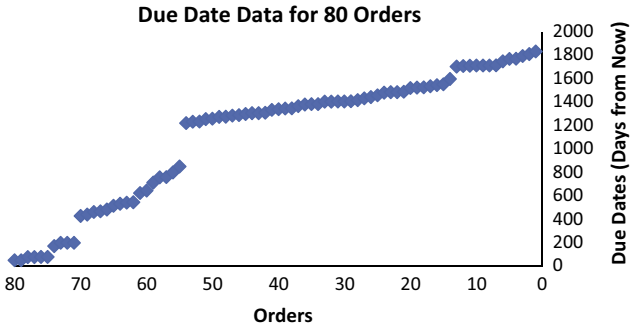

Fig. 3. Spread of due dates for the future exchanges.

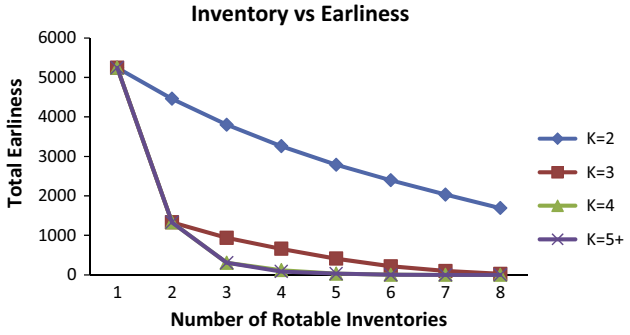

Fig. 4. Impact of initial rotable inventory level on total earliness under varying process capacities.

Table 5
Optimal total earliness values.

Rotables\Lines	2	3	4	5	6	7	8	9
2	5243							
3	4461	1331						
4	3803	939	306					
5	3259	659	118	84				
6	2787	411	29	29	29			
7	2395	216	2	2	2	2		
8	2031	98	0	0	0	0	0	
9	1691	23	0	0	0	0	0	0

Table 6
Preferences for additional processing lines versus additional rotables ($p = 30$ days).

Rotables\Lines	2	3	4	5	6	7	8
2	R						
3	L	R					
4	L	L	R				
5	L	L	L	R			
6	L	L	L	R	R		
7	L	L	L	R	R	R	
8	L	L	L	R	R	R	R

Table 7
Preferences for additional processing lines versus additional rotables ($p = 20$ days).

Rotables\Lines	2	3	4	5	6	7	8
2	R						
3	L	R					
4	L	L	R				
5	L	L	R	R			
6	L	L	R	R	R		
7	L	L	N	N	N	N	
8	L	L	N	N	N	N	N

Table 8
Preferences for additional processing lines versus additional rotables ($p = 40$ days).

Rotables\Lines	2	3	4	5	6	7	8
2	R						
3	L	R					
4	L	R	R				
5	L	R	R	R			
6	L	R	R	R	R		
7	L	N	N	N	N	N	
8	N	N	N	N	N	N	N

processing times, namely with $p = 20$ days and $p = 40$ days. The marginal impact comparison tables for these cases are given in Tables 7 and 8. Both tables reveal that when the overhaul processing times are short, rotable inventory becomes the bottleneck more often than the processing lines. On the other hand, if the processing times are longer, additional processing lines contribute to the earliness performance relatively more in higher number of cases. The intuition is that longer processing times lead to lower inventory turnovers for each processing line relatively reducing the marginal benefit of additional inventory. Such comparison can be used to help MRO firms decide on the direction of capacity expansion especially under budget constraints.

With the proposed exact solution algorithm, each instance in the above analysis can be solved almost instantaneously using a PC with Intel Core i5 processor and 16 GB of 1.6 GHz memory. In order to demonstrate the computational efficiency of the proposed model and the full-delay algorithm, we used AMPL/CPLEX (version 12.6) and a simple AMPL code (without calling any solver) respectively. We used the case study problem discussed above as the base problem and generated additional instances with larger problem sizes by adding jobs. Specifically, we ran the mathematical model for instances with 80, 160, 240, 320, 400, 480, 720, and 880 jobs. For the larger instances, the due date of each additional job was generated by adding a random interval to the previous job's due date, where these random intervals range from 0 to 40 days, uniformly distributed. We observed that while the computation times varied from a couple of seconds to 10 min with the mathematical model and CPLEX, the proposed full-delay algorithm was able to solve all instances *altogether* under one fourth of a second on the same workstation. Our analysis indicates that both approaches are computationally efficient. However, the full-delay algorithm does not require any professional solver tool and allows for practical implementation on widely available tools (such as MS Excel/VBA).

6. Conclusions

MRO of critical and high-value equipment is an inescapable support activity for continuation of uninterrupted operations for many manufacturers and service providers. In industries, where safety is the crucial part of the operations, the maintenance and overhaul of the equipment used is subject to stern regulations and requirements. In the airline industry, the MRO of the airplane parts and components must be carried out in specific intervals defined with time windows or flying hours. As such, the MRO service for aircraft components has to be completed before hard deadlines. The MRO service providers are expected not only to satisfy these enforced deadlines but also deliver with short turnaround times for their customers' efficient utilization of the equipment. To minimize the customers' downtime, some service providers adopt exchange programs where customer's equipment is exchanged with a ready-to-go module and the uninstalled component enters into the overhaul process as a rotable item for a future exchange. Because of the constrained overhaul capacity and limited number of rotables, the MRO companies may be compelled to ask for an exchange before the customer's deadline. However, this practice ends the equipment's operational cycle early and as such, is not preferable for the customer.

In this paper, we introduce an integer programming model, as the first model of the problem to the best of our knowledge, for optimal overhaul and exchange scheduling that minimizes the total earliness of exchange times under capacity and inventory constraints. The capacity is defined by the number of parallel processing lines, and the inventory is composed of finite number of rotables. We also introduce an exact polynomial time algorithm for the problem and illustrate its implementation using a case from a real-life practice. We present a sensitivity analysis that investigates the joint impact of capacity and inventory on the optimal schedule. The analysis shows that the marginal benefit of an additional rotable inventory or an additional process line is context specific and depends on which one is the bottleneck for the MRO firm.

The problem can be extended in different directions. For a future work direction one can generalize our model to include multiple module types with varying overhaul process requirements. Another direction is to incorporate rotable inventory decisions into the problem with budget constraints or using multi-objective models. Uncertainty in demand arrivals and processing times are also potential extensions to our work.

Acknowledgments

This research was partially supported by the Science Fellowships & Grant Programs Department of The Scientific & Technological Research Council of Turkey (TUBITAK), BIDEB #2221. We are grateful to two anonymous referees whose comments have significantly contributed to improvement of our paper.

References

Arts, J., & Flapper, S. D. (2015). Aggregate overhaul and supply chain planning for rotables. *Annals of Operations Research, 224*, 77–100.
Buyukkaramikli, N. C., van Ooijen, H. P. G., & Bertrand, J. W. M. (2013). Integrating inventory control and capacity management at a maintenance service provider. *Annals of Operations Research*. forthcoming.
Dekker, R. (1996). Applications of maintenance optimization models: A review and analysis. *Reliability Engineering and System Safety, 51*, 229–240.
Dekker, R., & Scarf, P. A. (1998). On the impact of optimisation models in maintenance decision making: The state of the art. *Reliability Engineering and System Safety, 60*, 111–119.
Ghobbar, A. A., & Friend, C. H. (2003). Evaluation of forecasting methods for intermittent parts demand in the field of aviation: A predictive model. *Computers & Operations Research, 30*, 2097–2114.
Gu, J., Zhang, G., & Li, K. W. (2015). Efficient aircraft space parts inventory management under demand uncertainty. *Journal of Air Transport Management, 42*, 101–109.
Guajardo, J. A., Cohen, M. A., Kim, S.-H., & Netessine, S. (2012). Impact of performance-based contracting on product reliability: An empirical analysis. *Management Science, 58*(5), 961–979.
Joo, S. J. (2009). Scheduling preventive maintenance for modular designed components: A dynamic approach. *European Journal of Operational Research, 192*, 512–520.

Joo, S. H., & Min, H. (2011). A multiple objective approach to scheduling the preventive maintenance of modular aircraft components. *International Journal of Services and Operations, 9*(1), 18–31.

Kilpi, J., Töyli, J., & Vepsäläinen, A. P. J. (2009). Cooperative strategies for the availability services of repairable aircraft components. *International Journal of Production Economics, 117*, 360–370.

Kilpi, J., & Vepsäläinen, A. P. J. (2004). Pooling of spare components between airlines. *Journal of Air Transport Management, 10*, 137–146.

Luh, P. B., Yu, D., Soorapanth, S., Khibnik, A. I., & Rajamani, R. (2005). A Lagrangian relaxation based approach to schedule asset overhaul and repair services. *IEEE Transactions on Automation Science and Engineering, 2*(2), 145–167.

MacDonnell, M., & Clegg, B. (2007). Designing a support system for aerospace maintenance supply chains. *Journal of Manufacturing Technology Management, 18*, 139–151.

MacDonnell, M., & Clegg, B. (2011). A new inventory model for aircraft spares. In N. Altay & L. A. Litteral (Eds.), *Service parts management*. London: Springer-Verlag.

Mwanalushi, K. (2012). *Managing rotable inventories*. AviTrader MRO. November issue.

Nicolai, R., & Dekker, R. (2008). Optimal maintenance of multi-component systems: A review. In *Complex System Maintenance Handbook* (pp. 263–286). London: Springer.

Regattieri, A., Gamberi, M., Gamberini, R., & Manzini, R. (2005). Managing lumpy demand for aircraft spare parts. *Journal of Air Transport Management, 11*, 426–431.

Safaei, N., Banjevic, D., & Jardine, A. K. S. (2011). Workforce-constrained maintenance scheduling for military aircraft fleet: A case study. *Annals of Operations Research, 186*, 295–316.

van Jaarsveld, W., Dollevoet, T., & Dekker, R. (2012). *Spare parts inventory control for an aircraft component repair shop* Working Paper.Rotterdam, Netherlands: Erasmus University.