Aircraft Line Maintenance Planning Based on PHM Data and Resources Availability Using Large Neighborhood Search

Wlamir Olivares Loesch Vianna[1], Leonardo Ramos Rodrigues[1] and Takashi Yoneyama[2]

[1]*EMBRAER S.A., São José dos Campos, São Paulo, 12227-901, Brazil*

wlamir.vianna@embraer.com.br
leonardo.ramos@ embraer.com.br

[2]*ITA – Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 12228-900, Brazil*

takashi@ita.br

Abstract

Maintenance planning has become a topic of great interest among researchers and industry practitioners in recent years, since it directly impacts the availability and the lifecycle cost of systems. In the aviation industry, maintenance planning becomes even more relevant due to the high availability expectations from aircraft operators and the high costs incurred when an aircraft becomes out of service. For this reason, some minor maintenance activities are carried out near the gate, between two consecutive flight legs. These activities are referred to as aircraft line maintenance. Planning line maintenance activities is critical because a problem in the execution of line maintenance may lead to flight delays and even flight cancellations. This paper presents a methodology for aircraft line maintenance planning including both the troubleshooting tasks and the repair activities. The proposed methodology uses a Large Neighborhood Search (LNS) algorithm in order to find the most appropriated time and location to perform line maintenance activities. The algorithm considers the precedence relation between a troubleshooting task and its respective repair activity, as well as the dispachability constraints included in the MEL (Minimum Equipment List). Resources availability such as spare parts, equipments and personnel are taken into account, as well as the risk of occurrence of an AOG (Aircraft on Ground) event, estimated from PHM (Prognostics and Health Monitoring) data. An AOG event is an event that leads to a flight cancelation. The optimization goal is to minimize the Expected Cost of Repair (ECR) considering both delay and AOG expenses. A numerical example is presented to illustrate the application of the proposed methodology.

1. Introduction

An airline flight operations department needs an efficient planning system in order to successfully manage its expensive assets. Considering that, operational research plays a major role in the airline industry's tactical planning (Barnhart, Belobaba & Odoni, 2003). Most applications in the literature deal with the following areas (Sarac, Batta & Rump, 2006):

- The schedule preparation, where airlines identify a list of flight legs along with departure and arrival times;
- The fleet assignment problem;
- The aircraft routing area; and
- The disruption recovery problem, whose objective is to react to all operational disruptions.

Also, several maintenance planning applications have been proposed in these areas. According to Papakostas, Papachatzakis, Xanthakis, Mourtzis and Chryssolouris (2010), although the increasing progress, most of these approaches have limitations. GO/NOGO decisions are not directly supported and the academic demonstrators so far supporting this kind of functionality have limited intelligence without concurrently taking into consideration parameters such as possible flight delay, cost consequences and actual Remaining Useful Life (RUL) of aircraft systems and components.

Many applications deal with the troubleshooting optimization problem (Langseth & Jensen, 2003; Lín, 2012; Pernesta, Nyberg & Warnquist, 2012; Kalagnanam & Henrion, 1990), which consists of defining the best test sequence in order to optimize the balance between the test cost and the probability that the test will be helpful to isolate the fault. Basically, all non-trivial troubleshooting domains are NP-hard (Vomlelová, 2003), specially those considering operation limitations such as turnaround time (TAT) and resources availability.

This paper proposes a methodology for defining a good line maintenance strategy using a LNS (Large Neighborhood Search) algorithm that takes into account the troubleshooting tasks, the flight plan, resources availability and the health condition of components.

The remaining of this paper is organized as follows. Section 2 describes the formulation of the problem. The proposed methodology is discussed in detail in Section 3. A numerical example to illustrate the application of the proposed methodology is presented in Section 4. Concluding remarks and future research opportunities are given in Section 5.

2. PROBLEM FORMULATION

Airline maintenance activities can be divided into 3 groups: shop maintenance, hangar maintenance and line maintenance. This paper is focused on line maintenance, which is defined by the FAA (2013) as "any unscheduled maintenance resulting from unforeseen events, or scheduled checks where certain servicing and/or inspections do not require specialized training, equipment or facilities".

For the events that are not related to an AOG situation (i.e., events that do not compromise the aircraft minimal safety requirements and do not cause a flight cancelation), a decision must be made in order to define the sequence and most appropriate time to execute the troubleshoot tasks and the repair activities in order to fix the failure. To accomplish that, the variables described below are considered.

A problem D, that consists in defining the time and location for a group of troubleshooting tasks and repair activities to be performed, needs to be solved. There is a set of faults $F = \{F_1, \dots, F_m\}$ that describes all possible causes for the problem. Each fault $F_i \in F$ has an associated probability p_i that represents the probability of problem D to be caused by fault F_i. Also, a troubleshooting task T_i with duration $d(T_i)$ and a repair activity R_i with duration $d(R_i)$ are associated to each fault F_i.

There is a set of flight legs $L = \{L_1, \dots, L_n\}$ that describes the next flights of the aircraft under consideration. For each flight leg $L_j \in L$, a turnaround time $TAT(L_j)$ describes the time interval available to perform maintenance activities between the *j-th* flight leg and the next one.

A binary matrix $CT(T_i, L_j)$ indicates if the base where the aircraft will be after flight leg L_j has all the resources (materials, maintenance personnel, etc.) required in order to execute the troubleshooting task T_i. Similarly, a binary matrix $CR(R_i, L_j)$ indicates if the base where the aircraft will be after flight leg L_j has all the resources required in order to execute the repair activity R_i.

A matrix $P_{AOG}(F_i, L_j)$ defines the probability of an AOG event to be caused by each fault F_i at each flight leg L_j. Each element of this matrix is estimated based on PHM data. Details on how these estimates are made are presented in Section 4.

A cancellation cost C_{AOG} defines the cost incurred if an AOG event occurs. A delay cost per minute C_{delay} defines the cost incurred for each minute of delay.

Furthermore, the following assumptions are considered:

- Problem D is always caused by a single fault $F_i \in F$.
- New faults or changes in the parameters of matrix $P_{AOG}(F_i, L_j)$ are never introduced during the execution of troubleshooting or repair tasks.
- Every AOG event results in a flight cancellation while every task execution that exceeds the available time $TAT(L_j)$ results in a flight delay.
- The troubleshooting task T_i can isolate only fault $F_i \in F$.
- The repair activity R_i is effective only for fixing fault $F_i \in F$.
- Maintenance activities can not be executed in parallel.
- Maintenance activities can not be split to be carried out in two or more different flight legs.
- The flight plan is fixed independent of task allocation.

The optimization problem consists of defining the following variables:

- The troubleshooting task plan $TP = \{TP_1, \dots, TP_m\}$, where TP_i indicates the most appropriate time to execute the troubleshooting task T_i.

- The repair task plan $RP = \{RP_1, \ldots, RP_m\}$, where RP_i indicates the most appropriate time to execute the repair task R_i.

TP and RP are related to the flight legs L. It implies that TP_i and RP_i have integer values. For example, if $TP_2 = 3$, it means that the troubleshooting task associated to fault F_2 will be carried out after the third flight leg.

A time availability matrix A must be built. Each element A_{ij} of matrix A contains the amount of time in $TAT(L_j)$ that is not used by any troubleshooting task or repair activity, considering that problem D was caused by fault F_i. The expressions to calculate the elements of matrix A are shown in Eq. (1).

$$A_{ij} = \begin{cases} TAT(L_j) - \sum d(T_i) \cdot \delta_{ij} & j < RP_i \\ TAT(L_j) - d(R_i) - \sum d(T_i) \cdot \delta_{ij} & j = RP_i \\ TAT(L_j) & j > RP_i \end{cases} \tag{1}$$

where δ_{ij} is a binary variable that assumes the value "1" if $TP_i = j$ and the value "0" otherwise.

The Expected Cost of Repair considering that the problem D was caused by fault F_i, ECR_i, is calculated as shown in Eq.(2).

$$\begin{aligned} ECR_i &= C_{AOG} \cdot P_{AOG}(i, RP_i) + \\ &+ C_{delay} \cdot \sum_{j=1}^{RP_i} máx\left[0; -A_{i,j}\right] \cdot \left[1 - P_{AOG}(i, j)\right] \end{aligned} \tag{2}$$

Finally, the Total Expected Cost of Repair, $TECR$, is obtained according to Eq. (3).

$$TECR = \sum_{i=1}^{m} ECR_i \cdot p_i \tag{3}$$

The $TECR$ is the objective function of the optimization problem. The problem constraints are listed below.

- $CT(T_i, TP_i) = 1$, for $i = 1, \ldots, m$
- $CR(R_i, RP_i) = 1$, for $i = 1, \ldots, m$
- $RP_i \geq TP_i$, for $i = 1, \ldots, m$

The two first constraints are related to the resources availability and base limitations to execute each troubleshooting task and repair activity, while the last constraint is related to the precedence relation between a troubleshooting task and its respective repair activity.

Once TP and RP are defined, maintenance activities are carried out according to Algorithm 1.

Algorithm 1: Plan Execution

```
fault ← 0
for j ← 1, n do
    for i ← 1, m do
        if TP_i = j then
            Execute T_i
            if F_i = true then
                fault ← i
            end if
        end if
        if RP_i = j and fault = i then
            Execute R_i
            return
        end if
    end for
end for
```

3. OPTIMIZATION METHODOLOGY

Considering the complexity of the current problem (NP-Complete), a heuristic method will be used and therefore an optimal solution is not guaranteed. In this paper, constraint programming (CP) and local search (LS) will be used to solve the optimization problem described in the previous Section. Although this method has a higher change of being stuck in a local minima solution compared to other more robust optimization methods such as Simulated Annealing and Genetic Algorithms, the decision to choose this method was made due to its low computational cost and relative easy implementation. The use of several methods to solve similar problems can be found in literature. Papakostas et al. (2010) used a multi-criteria mechanism for deferring maintenance actions. Langseth and Jensen (2003) presented a greedy heuristic algorithm for fault diagnosis. Optimal partitioning (Ottosen & Jensen, 2011) and Bayesian networks (Pernesta et al., 2012) have also been used to solve maintenance planning problems.

The efficiency of a troubleshooting task is used by many authors in maintenance action solutions (Ottosen & Jensen, 2011). The efficiency of a troubleshooting task is defined in Eq. (4).

$$eff(T_i) = \frac{p(T_i)}{d(T_i)} \tag{4}$$

The efficiency ranks the troubleshooting tasks in order of the most probable action with less cost, or duration in this particular case. Tasks with higher efficiency values are carried out previous to others with lower efficiency values. One good strategy to begin with is to perform the tasks in the efficiency order as soon as possible with no delay. In other words, $A_{ij} \geq 0$ for every F_i and L_j. The same strategy is applicable to the repair tasks plan. Although this strategy does not consider the AOG costs, it is a good starting point for the optimization problem.

Considering the starting point strategy described above, the combination of LS and CP and the search space of the problem, a Large Neighborhood Search (LNS) optimization algorithm was implemented.

LNS is a heuristic algorithm that generally starts with a feasible solution and iteratively tries to obtain a better solution by searching the "neighborhood" of the current solution. A critical issue in the design of a neighborhood search algorithm is the choice of the neighborhood structure, i. e., the manner in which the neighborhood is defined. At the same time, the larger the neighborhood, the longer it takes to search the neighborhood at each iteration (Ahujaa, Ergunb, Orlinc & Punnend, 2002). In this paper, the neighborhood was defined based on two operations:

- Swapping: Exchange the execution time of two tasks.
- Shifting: Anticipate or postpone a task.

The first operation evaluates if allocating the time reserved for one task to another one with different efficiency reduces the AOG risk with the possible impact of increasing delay costs. The second operation evaluates if executing a task before improves costs by reducing AOG risks with the possible impact of increasing delay costs.

Considering the neighborhood structure defined for the problem, the proposed methodology finds a solution according to Algorithm 2.

Function "*Initial_Solution*" generates the troubleshooting and repair plans (TP and RP) based on the task efficiency rank as previously discussed. Function "Swap" generates all possible strategies by swapping two tasks at a time. Function "*Shift*" generates all possible strategies by shifting each task to all available times. Function "$TECR$" estimates the Total Expected Cost of Repair for each alternative. This algorithm estimates $TECR$ for all possible neighbors considering all tasks swapping and shifting and, if any neighbor presents a lower $TECR$ compared to the current solution, this neighbor is selected as the new current solution. The process is repeated until there is no neighbor with lower $TECR$ compared to the current solution.

Algorithm 2: LNS Algorithm

```
Current_Solution ← Initial_Solution()
Stop_Criteria = False
while Stop_Criteria = False do
    Swap_Neighbors = Swap (Current_Solution)
    Shif_Neighbors = Shift (Current_Solution)
    Best_Swap = arg(min(TECR(Swap_Neighbors)))
    Best_Shift = arg(min(TECR(Shift_Neighbors)))
    Best_Neighbor = arg(min(TECR([Best_Swap,Best_Shift)))
    if TECR(Best_Neighbor) < TECR(Current_Solution) then
        Current_Solution = Best_Neighbor
    else
        Stop_Criteria = True
    end if
end while
```

In the proposed methodology, only the swap between two tasks and the shift of one task was considered. It can limit the search area. The neighborhood search is executed only for the current best solution and the algorithm always converges for the same strategy independent of how many trials are made for the same inputs.

4. NUMERICAL EXAMPLE

In this Section, an example of application of the proposed methodology is presented. In this example, we suppose that a "Bleed 1 Fail" message associated to the failure of an aircraft bleed system 1 happened and a decision should be made in order to isolate and repair the fault. There are six faults that may cause the problem:

1. Shutoff Bleed Valve.
2. Transitory Condition (Spurious Message).
3. Torque Motor Controller.
4. Motor Drive Module.
5. Cross Bleed Valve.
6. Pressure Sensor.

The probability vector p and tasks duration used in this example are:

$$p = [0.45 \quad 0.30 \quad 0.10 \quad 0.05 \quad 0.05 \quad 0.05]$$

$$d[T_1 ... T_6] = [90 \quad 10 \quad 15 \quad 60 \quad 80 \quad 70]$$

$$d[R_1 ... R_6] = [0 \quad 0 \quad 80 \quad 0 \quad 0 \quad 0]$$

The faults whose troubleshooting task results in repairing the fault has a repair duration equals to 0. To illustrate this situation consider the following example: The troubleshooting tasks to isolate the Shutoff Bleed Valve includes replacing the valve to verify whether the fault is removed and, if that is the case, no more activities are required.

The delay cost C_{delay} and the AOG cost C_{AOG} were arbitrarily chosen and defined as 300 and 120,000, respectively.

The variables associated to the flight plan and bases resources used in this example are:

$TAT = [40 \quad 37 \quad 33 \quad 55 \quad 814 \quad 62 \quad 66 \quad 34 \quad 730 \quad 25 \quad 839 \quad 33 \quad 45]$

$$CT = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad CR = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The AOG probability is associated to the probability of failure of bleed system 2, considering that Bleed System 1 is unavailable and the aircraft must not be dispatched with both bleed systems failed. In order to estimate the probability distribution of a failure in the bleed system 2 to occur, the PHM algorithm proposed by Gomes, Ferreira, Cabral, Galvão & Yoneyama (2010) was used. Figure 1 shows the Bleed Valve from System 2 degradation indexes for this method over the last 30 days previous to the fault message.

The degradation indexes were used in order to estimate the coefficients of a linear curve using a least square regression method with a confidence level of 95%. The coefficients and the covariance matrix obtained from the data shown below.

$b_0 = 72.07$

$b_1 = 1.66$

$$\mathrm{cov}_{b0,b1} = \begin{bmatrix} 2.90 & 0.15 \\ 0.15 & 0.01 \end{bmatrix}$$

where b_0 and b_1 are the coefficients of the linear model $y = b_0 + b_1 \cdot x$. In this model, the dependent variable y is the degradation index and the independent variable x is the time.

Monte Carlo simulation was used to generate a set of 5,000 realizations of the bleed system 2 degradation index evolution and the associated time of failure. Here, the time of failure is associated to the time when the degradation index reaches 100%.

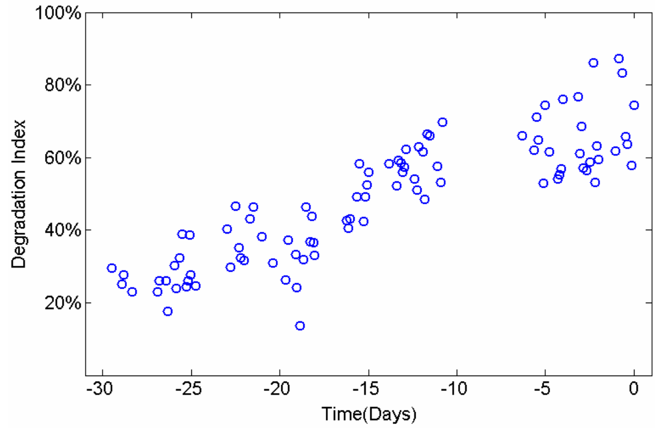


Figure 1. Bleed System 2 past degradation indexes.

The set of time of failures obtained from the Monte Carlo simulation was then used to fit a Weibull distribution for the Remaining Useful Life (RUL) of bleed system 2.

The probability of an AOG event to occur due to a bleed system 2 failure associated to each leg was estimated by inserting the date of each flight at the Cumulative Distribution Function (CDF) of the Weibull distribution. Figure 2 shows the CDF obtained for this example.

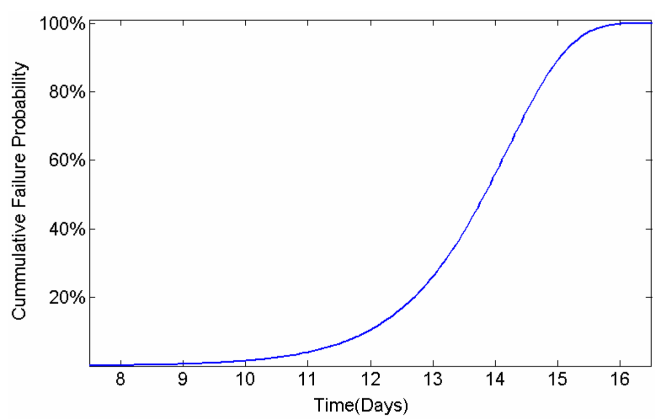


Figure 2. CDF obtained from the Weibull curve fitting.

These probabilities are associated to every fault except fault F_2 (Transitory Condition), which does not turns the Bleed System unavailable. Considering the CDF presented in Figure 2 and the TAT values, the probability matrix associated to each failure, P_L, is:

$$P_L = [.028 \quad .029 \quad .030 \quad .035 \quad .043 \quad .046 \\ .051 \quad .053 \quad .063 \quad .065 \quad .080 \quad .084 \quad .087]$$

The matrix P_{AOG} for the example is:

$$P_{AOG} = [P_L' \quad zeros(13,1) \quad P_L' \quad P_L' \quad P_L' \quad P_L']$$

After all variables are defined, the proposed methodology was used and the Total Expected Cost of Repair was estimated. The troubleshooting plan, TP, and repair plan, RP, obtained for this example were:

$$TP = [6 \quad 1 \quad 1 \quad 4 \quad 8 \quad 13]$$

$$RP = [6 \quad 1 \quad 6 \quad 4 \quad 8 \quad 13]$$

$$TECR = 11{,}376$$

Comparisons were made to two other possible strategies. The first one, which will be called "Method A", is a conservative strategy that recommends executing all tasks at the first possible base, even if it causes a flight delay. The second one, which will be called "Method B", recommends not executing any preventive action. Instead, it recommends executing a corrective repair action once the failure has happened. It makes the Total Expected Cost of Repair to be equal to the AOG cost $(TECR = C_{AOG})$. The Total Expected Costs of Repair estimated from the application of the methods are summarized in Table 1.

Table 1. Result comparison

Optimization Method	TP	RP	TECR
LNS	[6 1 1 4 8 13]	[6 1 6 4 8 13]	11,376
Method A	[1 1 1 1 1 1]	[1 1 1 1 1 1]	48,150
Method B	N/A	N/A	120,000

From these results it is possible to see how effective the proposed methodology was in this example. It found a much cheaper solution compared to the other strategies.

5. CONCLUSIONS

This paper presented an aircraft line maintenance planning methodology including both the troubleshooting tasks and the repair activities to be carried out during the turn-around time. Resources availability and flight plan were considered. The probability of an AOG event to occur was also taken into account, based on RUL estimates obtained from a PHM system. A Large Neighborhood Search (LNS) algorithm was used in order to optimize the expected cost of repair (ECR), considering delay and AOG costs. A numerical example was presented to illustrate the application of the proposed methodology. The results showed that the proposed methodology is promising, although it does not guarantee that the optimal TECR will be found.

Improvements in the proposed methodology could be made by implementing more robust neighborhood definitions. Some operations that could be used in the neighborhood definition include compounded swaps, cyclical shifts, assignments/matching (Ahujaa et al, 2002), optimal partitions (Ottosen & Jensen, 2011) and multi-start search algorithms.

Future work opportunities include evaluating the proposed methodology with real data, proposing new operations for the neighborhood definition. Another direction for future research is to evaluate the performance of different local search algorithms such as Simulated Annealing, Tabu Search, Ant Colony Optimization and Genetic Algorithms to solve the troubleshooting optimization problem. Also, adding new constraints to the optimization problem in order to consider other aspects such as safety and risk requirements is an interesting direction for future research. The assumptions adopted in this paper may not consider some real operational situations such as the occurrence of multiple failures, the execution of troubleshooting tasks in parallel and the flexibilization of flight plans. Proposing a more robust optimization troubleshooting model that considers all these operational situations is also a topic for future research in this area.

ACKNOWLEDGEMENT

The authors acknowledge the support of FINEP (grant 1498/07).

REFERENCES

Ahujaa, R. K., Ergunb, O., Orlinc, J. B., & Punnend, A. P. (2002). A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, vol. 123, pp. 75-102.

Barnhart, C., Belobaba, P., & Odoni, A. R. (2003). Applications of operations research in the air transport industry. *Transportation Science*, vol. 37, pp. 368-391.

Federal Aviation Administration (FAA) (2013). Air operator and air agency certification and application process. chapter 14.

Gomes, J. P. P., Ferreira, B. C., Cabral, D., Galvão, R. K. H., & Yoneyama, T. (2010). Health monitoring of a pneumatic valve using a PIT based technique. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*. October 10-16, Portland.

Kalagnanam J, & Henrion M. (1990). A comparison of decision analysis and expert rules for sequential analysis. *Uncertainty in Artificial Intelligence*, vol. 4, pp. 271-281.

Langseth, H., & Jensen, F. V. (2003). Decision theoretic troubleshooting of coherent systems. *Reliability Engineering and System Safety*, vol. 80, pp. 49-62.

Lín, V. (2012). Decision-theoretic troubleshooting: Hardness of approximation. *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models*. September 19-21, Granada.

Ottosen, T. J., & Jensen, F. V. (2011). When to test? Troubleshooting with postponed system test. *Expert Systems with Applications*, vol. 38, pp. 12142-12150.

Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., & Chryssolouris, G. (2010). An approach to operational aircraft maintenance planning. *Decision Support Systems*, vol. 48, issue 4, pp. 604-612.

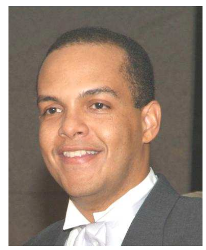
Pernesta, A., Nyberg, M., & Warnquist, H. (2012). Modeling and inference for troubleshooting with interventions applied to a heavy truck auxiliary braking system. *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 705-719.

Sarac, A., Batta, R., & Rump, C. (2006). A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, vol. 175, issue 3, pp. 1850-1869.

Vomlelová, M. (2003). Complexity of decision-theoretic troubleshooting. *International Journal of Intelligent Systems*, vol. 18, issue 2, pp. 267-277.

BIOGRAPHIES

Wlamir Olivares Loesch Vianna holds a bachelor's degree on Mechanical Engineering (2005) from Universidade de São Paulo (USP), Brazil, and Master Degree on Aeronautical Engineering (2007) from Instituto Tecnológico de Aeronáutica (ITA), Brazil. He is with Empresa Brasileira de Aeronáutica S.A (EMBRAER), São José dos Campos, SP, Brazil, since 2007. He works as a Development Engineer of a R&T group at EMBRAER focused on PHM technology applications in aeronautical systems.



Leonardo Ramos Rodrigues holds a bachelor's degree in Electrical Engineering from Universidade Federal do Espírito Santo (UFES, 2003), Brazil, a Masters's Degree (2008) and a Doctorate's Degree (2013) in Aeronautical Engineering from Instituto Tecnológico de Aeronáutica (ITA, Brazil). He is with EMBRAER S.A. since 2006, working as a Development Engineer focusing on PHM technology applications. His current research interests are the application of health monitoring techniques for electronic components and the usage of PHM information for inventory optimization.

Takashi Yoneyama is a Professor of Control Theory with the Electronic Engineering Department of ITA. He received the bachelor's degree in electronic engineering from Instituto Tecnológico de Aeronáutica (ITA), Brazil, the M.D. degree in medicine from Universidade de Taubaté, Brazil, and the Ph.D. degree in electrical engineering from the University of London, U.K. (1983). He has more than 300 published papers, has written four books, and has supervised more than 70 theses. His research is concerned mainly with stochastic optimal control theory. He served as the President of the Brazilian Automatics Society in the period 2004-2006.