Maintenance planning on French military aircraft operations

Franco Peschiera, Alain Haït, Olga Battaïa, Nicolas Dupin

ISAE-SUPAERO, Université de Toulouse, France

September 24, 2018
1 Problem

2 State of the art

3 Complexity analysis

4 Model

5 Results and perspectives

6 References
Problem
Problem (informally)

Assign both missions and maintenance operations to a fleet of aircraft in order to maximize availability and minimize costs. Missions have fixed start and end times and have particular needs in terms of aircraft and time.
Problem (example)
Problem

Problem

- A series of $j \in J$ tasks are planned along a horizon divided into $t \in T$ periods.

A series of $j \in \mathcal{J}$ tasks are planned along a horizon divided into $t \in \mathcal{T}$ periods.

Each task requires $R_j$ of resources for $H_j$ hours and needs to be assigned for a minimum of $MT_j$ consecutive periods.
Problem

- A series of $j \in \mathcal{J}$ tasks are planned along a horizon divided into $t \in \mathcal{T}$ periods.
- Each task requires $R_j$ of resources for $H_j$ hours and needs to be assigned for a minimum of $MT_j$ consecutive periods.
- Resources $i \in \mathcal{I}$ require recurrent preventive maintenance operations.

A series of $j \in J$ tasks are planned along a horizon divided into $t \in T$ periods.

Each task requires $R_j$ of resources for $H_j$ hours and needs to be assigned for a minimum of $MT_j$ consecutive periods.

Resources $i \in I$ require recurrent preventive maintenance operations.

A maintenance operation takes exactly $m$ periods and restores the resource’s remaining usage time to exactly $H^M$ units.
State of the art
State of the art (1)

- In Cho (2011), US Air Force aircraft were assigned daily operations over a year to minimize the number of maintenances.
State of the art (1)

- In Cho (2011), US Air Force aircraft were assigned daily operations over a year to minimize the number of maintenances.
- In Kozanidis (2008), Greek aircraft had monthly assignments of maintenances and flight hours in order to maximize the availability and final state of squadrons.
State of the art (1)

- In Cho (2011), US Air Force aircraft were assigned daily operations over a year to minimize the number of maintenances.
- In Kozanidis (2008), Greek aircraft had monthly assignments of maintenances and flight hours in order to maximize the availability and final state of squadrons.
- In Verhoeff, Verhagen, and Curran (2015), monthly assignments were done and several objectives were taken into account: availability, serviceability and final state.
State of the art (1)

- In Cho (2011), US Air Force aircraft were assigned daily operations over a year to minimize the number of maintenances.
- In Kozanidis (2008), Greek aircraft had monthly assignments of maintenances and flight hours in order to maximize the availability and final state of squadrons.
- In Verhoeff, Verhagen, and Curran (2015), monthly assignments were done and several objectives were taken into account: availability, serviceability and final state.
- In Seif and Yu (2018), a generalization of Kozanidis (2008) was done in order to deal with different types of maintenances and an heterogeneous fleet.
Asignments to missions are done instead of assignments of flight hours.
State of the art (2)

- Assignments to missions are done instead of assignments of flight hours.
- Missions need to be assigned for a minimal duration.
Asignments to missions are done instead of assignments of flight hours.

Missions need to be assigned for a minimal duration.

Because of these differences, a new model is proposed to deal with this new problem.
<table>
<thead>
<tr>
<th>Problem</th>
<th>State of the art</th>
<th>Complexity analysis</th>
<th>Model</th>
<th>Results and perspectives</th>
<th>References</th>
</tr>
</thead>
</table>

Franco Peschiera, Alain Haït, Olga Battaïa, Nicolas Dupin
ISAE-SUPAERO, Université de Toulouse, France
Maintenance planning on French military aircraft operations

Complexity analysis
Complexity analysis


- $P$ tasks with duration $u_p$ and start time $s_p$. 
Complexity analysis


- $P$ tasks with duration $u_p$ and start time $s_p$.
- $E$ employees with a set of tasks $P_e \in P$ that each can perform.
Complexity analysis


- $P$ tasks with duration $u_p$ and start time $s_p$.
- $E$ employees with a set of tasks $P_e \in P$ that each can perform.
- Minimize the number of required employees to do all tasks.
Reduction to the ‘Fixed interval scheduling’ in Smet (2015) **proves NP-completeness.**

- $P$ tasks with duration $u_p$ and start time $s_p$.
- $E$ employees with a set of tasks $P_e \in P$ that each can perform.
- Minimize the number of required employees to do all tasks.
Complexity analysis


- $P$ tasks with duration $u_p$ and start time $s_p$.
- $E$ employees with a set of tasks $P_e \in P$ that each can perform.
- Minimize the number of required employees to do all tasks.
Model
Model: variables

The following variables are used in the model.

- $m_{it}$: maintenance start $|I| \times |T|
- $a_{jti}$: task assignment $|T| \times |J_t| \times |I_j|
- $a^s_{jti}$: task assignment start $|T| \times |J_t| \times |I_j|
- $u_{it}$: monthly usage time $|I| \times |T|
- $rut_{it}$: remaining usage time $|I| \times |T|$
Model: main constraints

\[
\text{Min } \sum_{t \in T, i \in I} m_{it} \times H^M - \sum_{i \in I} r_{ut_i | T|} \tag{1}
\]

The objective is to minimize the total number of maintenances and, at the same time, maximize the end status of the resources.
Model: main constraints

\[
\text{Min } \sum_{t \in \mathcal{T}, i \in \mathcal{I}} m_{it} \times H^M - \sum_{i \in \mathcal{I}} r_{\text{ut}} |\mathcal{T}| \tag{1}
\]

The objective is to minimize the total number of maintenances and, at the same time, maximize the end status of the resources.

\[
\sum_{t' \in \mathcal{T}_t} \sum_{i \in \mathcal{I}} m_{it'} + N_t \leq C_{\text{max}} \quad t \in \mathcal{T} \tag{2}
\]

\[
\sum_{i \in \mathcal{I}_j} a_{jti} \geq R_j \quad j \in \mathcal{J}, t \in \mathcal{T}_j \tag{3}
\]

\[
\sum_{t' \in \mathcal{T}_t} m_{it'} + \sum_{j \in \mathcal{J}_t \cap \mathcal{O}_i} a_{jti} \leq 1 \quad t \in \mathcal{T}, i \in \mathcal{I} \tag{4}
\]
Model: maintenances

\[ m_{it'} + m_{it} \leq 1 \quad \text{for} \quad t \in T, t' \in T_t^m, i \in I \quad (5) \]

\[ \sum_{t' \in T_t^M} m_{it'} \geq m_{it} \quad \text{for} \quad t \in T, i \in I \quad (6) \]
Model: other constraints

- Minimum cluster availability.
Model: other constraints

- Minimum cluster availability.
- Minimum duration for task assignment.
Model: other constraints

- Minimum cluster availability.
- Minimum duration for task assignment.
- Hour consumption of resources depending on tasks and maintenances.
Results and perspectives
Example of a solution
Results on previous instances

All instances are solved to 0.02% of optimality in less than one hour. Previously this was not the case.

| id | $|J|$ | $|T|$ | assign | gap (%) | time (s) |
|----|----|----|-------|--------|---------|
| I_0 | 9  | 11 | 310   | 0.01   | 29.79   |
| I_1 | 9  | 21 | 650   | 0.02   | 173.11  |
| I_2 | 9  | 31 | 990   | 0.02   | 409.66  |
| I_3 | 9  | 41 | 1249  | 0.03   | 836.08  |
| I_4 | 10 | 11 | 530   | 0.02   | 37.66   |
| I_5 | 10 | 21 | 1070  | 0.01   | 208.25  |
| I_6 | 10 | 31 | 1610  | 0.01   | 331.90  |
| I_7 | 10 | 41 | 2069  | 0.02   | 1731.50 |
| I_8 | 11 | 11 | 1080  | 0.01   | 76.18   |
| I_9 | 11 | 21 | 2120  | 0.02   | 3332.95 |
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
## Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
  - 2 - 5 resources needed
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
  - 2 - 5 resources needed
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
  - 2 - 5 resources needed

Instances have **30,000 - 80,000 variables** and **20,000 - 50,000 constraints**.
Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
  - 2 - 5 resources needed

- Instances have **30,000 - 80,000 variables** and **20,000 - 50,000 constraints**.
- Runs of 30 minutes have been done on 30 randomly generated cases per scenario.
Experiments

Random instances were generated to test the performance following the known instances guidelines.

- 60 - 200 resources.
- 90 periods.
- 3 - 9 tasks:
  - 12 - 36 periods long
  - 2 - 5 resources needed

- Instances have $30,000 - 80,000$ variables and $20,000 - 50,000$ constraints.
- Runs of 30 minutes have been done on 30 randomly generated cases per scenario.
- CPLEX was used in an i7, quadcore Ubuntu 18.04 workstation.
From feasible small instances, a mean gap of 14.17% was found after 30 min.
Statistics

- From feasible small instances, a mean gap of 14.17% was found after 30 min.
- From feasible large instances, a mean gap of 37.6% was found after 30 min.
Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
## Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
- A deeper analysis on the detailed solving process.
Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
- A deeper analysis on the detailed solving process.
- Benchmarking of solvers (gurobi, CBC, choco and CPO).
Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
- A deeper analysis on the detailed solving process.
- Benchmarking of solvers (gurobi, CBC, choco and CPO).
- Heuristics for large scale instances.
Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
- A deeper analysis on the detailed solving process.
- Benchmarking of solvers (gurobi, CBC, choco and CPO).
- Heuristics for large scale instances.
- The inclusion of new possibilities, such as the planned storage of resources.
Perspectives

- More extreme (bigger) scenarios: more tasks and more resources.
- A deeper analysis on the detailed solving process.
- Benchmarking of solvers (gurobi, CBC, choco and CPO).
- Heuristics for large scale instances.
- The inclusion of new possibilities, such as the planned storage of resources.
- A stochastic sensibility analysis, including taking account of stochasticity in the input data.
References


