MISTA 2013

Optimal scheduling of aircrafts' engines repair process

Jorge Orestes Cerdeira · **Isabel Cristina Lopes** ·
Eliana Costa e Silva

1 Introduction

We address a real world scheduling problem concerning the repair process of aircrafts' engines by TAP - Maintenance & Engineering (TAP-ME). TAP-ME is the maintenance, repair and overhaul organization of TAP Portugal, Portugal's leading airline, which employs about 4000 persons to provide maintenance and engineering services in aircraft, engines and components. TAP-ME is aiming to optimize its maintenance services, focusing on the reduction of the engines repair turnaround time.

When an engine arrives at the TAP-ME Lisbon's base, the engine is dismantled in mini-modules, and each mini-module is decomposed in smaller parts which are cleaned and checked for defects. If defects are identified, the part is allocated for repair and a repair process is established, composed of an ordered sequence of tasks which are fully registered in the repair ID label. Repair operations may consist of manual repair jobs, such as welding or screwing, or machine repairs, such as shot peening or oven heating, which are carried out in a number of workstations. Defective parts are taken across the plant through the necessary workstations in order to have the repairs done.

There are several workcenters at the Lisbon maintenance unit, such as SOLD - where the welding jobs are performed; PINT - for the painting jobs; FORNO - the vacuum furnace; REP - where all other general repairs are made.

Some of the specific jobs in the FORNO, Plasma Shot and Shot Peening workcenters require setup times, which are known at the beginning of the repair process of that part, and do not depend on the sequence of the tasks to be performed in that machine. Hence this information can be included in the ID label of the part, by adding the setup time to the processing time of each task. All groups allow preemption-resume, except the FORNO, Plasma Shot

---

Jorge Orestes Cerdeira
FCT-UNL and ISA/UTL
E-mail: orestes@isa.utl.pt

Isabel Cristina Lopes
ESEIG-IPP/CIEFGEI
E-mail: cristinalopes@eu.ipp.pt

Eliana Costa e Silva
UPT, ESTGF-IPP and ALGORITMI
E-mail: eliana@upt.pt

and Shot Peening. This means that in these three workcenters, once a job process is started, it cannot be interrupted. In all other machines, if a job is interrupted as a result of machine or worker unavailability, the job will continue processing after the machine or worker becomes available.

The process is carried in a dynamic environment, as the human resources can be transferred between different workstations, if necessary. During its repair process, an engine part can go around several workstations, and may even return to some of these workstations. At the end of the repair process, the parts are once again checked and the engine is reassembled.

TAP-ME aims to develop a model for the repair process to minimize the total tardiness, while considering different clients priority, i.e., discriminating between delays on the deadlines accorded for the repair and maintenance of engines from different clients.

2 The developed model

The model we present for the TAP-ME repair process of aircrafts' engines is a mixed integer linear programming (MILP) formulation for the flexible job shop scheduling problem [1] [2] [3], which combines linear ordering variables for sequencing the operations, with assignment variables, and linear variables for the starting times of the jobs.

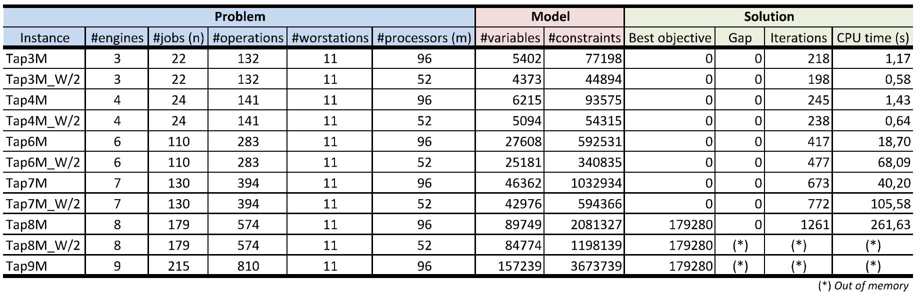
We identify each part $P \in \mathscr{P}$, where $\mathscr{P}$ is the set of parts, as a sequence of operations $P = (o, o', \ldots, o^{(k)})$, and denote by A_P the set of pairs of consecutive operations of P. We let t_o be the processing time of operation o, O_e be the set of operations of parts of engine $e \in E$, where E is the set of engines, and let L be a large number. We use binary variables a_o^m to assign each operation $o \in O$ to exactly one processor $m \in M_o$, where M_o is the set of processors that can process job o; linear ordering variables $x_{oo'}$ to describe the precedence relationships between pairs of operations (o, o') that are performed on a same processor; and continuous variables to define the starting times S_o of the operations o and the tardiness T_e of each engine $e \in E$. The model adequates the MILP formulation [4] [5] of the classic job shop scheduling problem, to handle flexibility due to the fact that there are operations o that can be executed, on the same workstation, by any processor of the set of processors M_o. The objective function is defined as a weighted sum of the tardiness regarding due dates D_e for engine e, with a parameter $\lambda_e \geq 0$ to express the relative priority assigned to engine e.

$$
\begin{array}{lrll}
\min & \multicolumn{3}{l}{\sum_{e \in E} \lambda_e T_e} \\
\text{s.t.} & \sum_{m \in M_o} a_o^m = 1, & \forall o \in O \\
& S_{o'} \geq S_o + t_o, & \forall (o,o') \in A_P, \forall P \in \mathscr{P} \\
& T_e \geq S_o + t_o - D_e, & \forall o \in O_e, \forall e \in E \\
& x_{oo'} + x_{o'o} \geq a_o^m + a_{o'}^m - 1, & \forall o \neq o' \in O, \forall m \in M_o \cap M_{o'} \\
& x_{oo'} + x_{o'o} \leq 1, & \forall o \neq o' \in O : M_o \cap M_{o'} \neq \emptyset \\
& S_{o'} \geq S_o + t_o - L(1 - x_{oo'}), & \forall o \neq o' \in O : M_o \cap M_{o'} \neq \emptyset \\
& S_o \geq S_{o'} + t_{o'} - L(1 - x_{o'o}), & \forall o \neq o' \in O : M_o \cap M_{o'} \neq \emptyset \\
& a_o^m \in \{0,1\}, & \forall o \in O, m \in M_o \\
& x_{oo'} \in \{0,1\}, & \forall o \neq o' \in O : M_o \cap M_{o'} \neq \emptyset \\
& S_o \geq 0, & \forall o \in O \\
& T_e \geq 0, & \forall e \in E
\end{array}
$$

We can also accommodate in the proposed model the existence of non working periods and preemption-resume operations. Non working periods (e.g., lunch breaks, night rest breaks, non working days) can be easily handled defining, for each non working period of some processor (machine or worker) m, a dummy operation $d \in O_m$ with processing time t_d equal to the duration of the non working period, and assigning to variable S_d, the starting time of operation d, the starting time of the non working period.

3 Computational results

We experimented our model on data provided by TAP-ME[1] respecting an ordinary week at the Lisbon unit. The model was implemented in AMPL modelling language and solved using Gurobi 3.0.3. Figure 1 shows the results obtained with several instances extracted from the data provided for that week. We tested with all the available processors and with only half of the processors in each workstation. Larger instances were not solved due to memory limits (Intel Core i5-3450CPU@3.10GHz 8Gb RAM).



Problem						Model		Solution			
Instance	#engines	#jobs (n)	#operations	#worstations	#processors (m)	#variables	#constraints	Best objective	Gap	Iterations	CPU time (s)
Tap3M	3	22	132	11	96	5402	77198	0	0	218	1,17
Tap3M_W/2	3	22	132	11	52	4373	44894	0	0	198	0,58
Tap4M	4	24	141	11	96	6215	93575	0	0	245	1,43
Tap4M_W/2	4	24	141	11	52	5094	54315	0	0	238	0,64
Tap6M	6	110	283	11	96	27608	592531	0	0	417	18,70
Tap6M_W/2	6	110	283	11	52	25181	340835	0	0	477	68,09
Tap7M	7	130	394	11	96	46362	1032934	0	0	673	40,20
Tap7M_W/2	7	130	394	11	52	42976	594366	0	0	772	105,58
Tap8M	8	179	574	11	96	89749	2081327	179280	0	1261	261,63
Tap8M_W/2	8	179	574	11	52	84774	1198139	179280	(*)	(*)	(*)
Tap9M	9	215	810	11	96	157239	3673739	179280	(*)	(*)	(*)

(*) *Out of memory*

Fig. 1 Results of the MILP model for real instances.

For the sake of brevity, in Figure 2, we only show a fragment, concerning the first 480 minutes, of the solution found using real deadlines and priorities of the first instance. For example, PART10 undergoes operation "ILIMP_LIMPAR_PECA" on processor "ILIMP_13" from minute 0 to 60, followed by operation "IIDF_EFECT_DETECCAO_FENDAS" on processor "IIDF_1" from minute 60 to 120. We can also observe that on the workcenter REP at minute 0 start, simultaneously, three operations on three different processors (IREP_1, IREP_9, IREP_20). All parts of the three engines were processed long before the deadlines.

As referred above, we could not run the model for the whole week instance. For large instances, other approaches, including heuristic strategies, should be conceived. We plan to exploit this soon.

Acknowledgements We are grateful to TAP Maintenance & Engineering for having authorize us to report computational results on real data that they provided. The problem was initially presented at ESGI'86 held in Porto, May 2012. We thank Ana Bidarra and Francisco Correia, from TAP-ME, for their support during the ESGI'86 and in later contacts. We also would like to thank Manuel Cruz and Pedro Freitas of the ESGI86's organization for all the assistance.

---

[1] The id reference of the engines and parts, as well as the names of the repair tasks were modified, in order to respect our confidentiality agreement with TAP-ME.

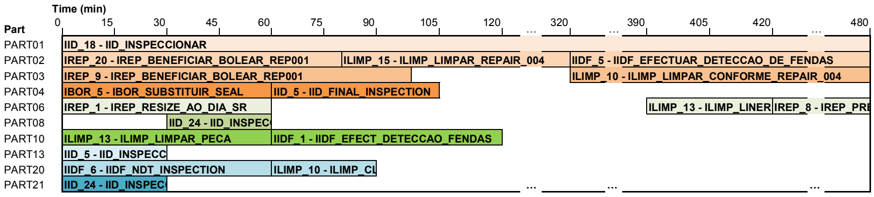


Fig. 2 Fragment of the solution, for the first 480 minutes, found for the three engines (MM1 - Parts 01 to 05; MM2 - Parts 06 to 11; MM3 - Parts 12 to 22). Each part is represented in a different color and it is indicated, for each period of time, the processor and the operation.

References

1. Brucker, P., Schlie, R.: Job-shop scheduling with multi-purpose machines. Computing **45**(4), 369–375 (1990)
2. Chan, F.T.S., Wong, T.C., Chan, L.Y.: Flexible job-shop scheduling problem under resource constraints. International Journal of Production Research **44**(11), 2071–2089 (2006)
3. Fattahi, P., Saidi Mehrabad, M., Jolai, F.: Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing **18**(3), 331–342 (2007)
4. Manne, A.: On the job shop scheduling problem. Operations Research **8**, 219–223 (1960)
5. Unlu, Y., Mason, S.J.: Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. Computers and Industrial Engineering **58**(4), 785 – 800 (2010)