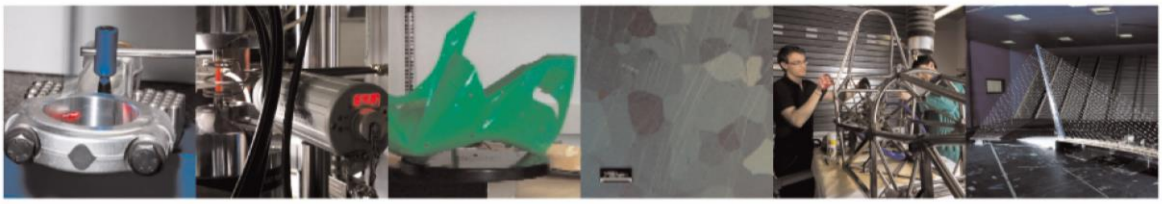


Scheduling Remanufacturing Activities for the Repair of Turbine Blades: An Approximate Branch and Bound Approach to Minimize a Risk Measure

Lei Liu, Marcello Urgo

DIPARTIMENTO DI MECCANICA ■ POLITECNICO DI MILANO
via G. La Masa, 1 ■ 20156 Milano ■ EMAIL (PEC): pecmecc@cert.polimi.it
http://www.mecc.polimi.it
Rev. 0

Scheduling remanufacturing activities for the repair of turbine blades: an approximate branch and bound approach to minimize a risk measure

Lei Liu, Marcello Urgo
Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy
lei.liu@polimi.it

Abstract. Refurbished products are gaining importance in many industrial sectors, specifically high-value products whose residual value is relevant and guarantee the economic viability of the remanufacturing at an industrial level, e.g., turbine blades for power generation. In this paper we address the scheduling of re-manufacturing activities for turbine blades. Parts entering the process may have very different wear state or presence of defects. Thus, the repair process is affected by a significant degree of uncertainty.

To cope with this, the proposed approach pursues robust schedules minimizing the risk associated to a timely completion time. An approximate branch and bound algorithm is developed grounding on the estimation of the lower bound of the makespan. The viability and efficiency of the approach is assessed through computational experiments grounding on the industrial case under study and a comparison is operated among alternative scheduling approaches.

Keywords: Remanufacturing, Scheduling under uncertainty, Risk, Branch and bound.

1 Introduction

Remanufacturing can be defined as "the rebuilding of a product to specifications of the original manufactured product using a combination of reused, repaired and new parts" [1]. Remanufacturing is a form of product recovery process entailing the repair or replacement of worn out components to obtain remanufactured products with the same customer expectations as new products.

The industrial viability of remanufacturing processes is specifically relevant for high-value products whose residual value is high [2, 3]. An example are turbine blades for power generation, whose individual price is close to a middle-class car, i.e. 10,000 euros, with F-class turbine engines needing about 400 blades for the different stages. While turbine blades are subjected to a very strenuous environment inside a gas turbine, such as high temperatures, stresses, and vibration, leading to blade failures, potentially destroying the engine, even though turbine blades are carefully designed to resist these conditions. The first stage of a gas turbine faces temperatures around 1370 °C, that can weaken the blades and make them more susceptible to creep and corrosion failures [4]. High stress from centrifugal force and fluid forces that can cause fracture, yielding, or creep failures of blades while vibrations can entail fatigue failures. The remanufacturing of turbine blades is a growing market in recent years, due to its potential to reduce cost and achieve sustainable production.

Nevertheless, planning and scheduling remanufacturing processes is more complex and difficult than traditional manufacturing due to uncertain factors. The complicating characteristics of remanufacturing environment are summarized in [2] in terms of the following aspects: uncertainty in the timing and the quantity of returns, balancing returns with demands, disassembly, uncertainty in materials recovered, reverse logistics, materials matching requirements, routing uncertainty and processing time uncertainty. Given the above-mentioned complexity and uncertainty, it is important for production managers to make effective production decisions [5].

In this paper, we address the scheduling of a portion of the remanufacturing process for turbine blades modelled as a two-machine permutation flow shop scheduling problem with stochastic processing times. The aim is at devising robust schedules to mitigate the impact of unfavourable scenarios. To this aim the objective function to minimize is the value-at-risk of the makespan, a measure of risk commonly used in the financial area [6, 7]. This class of risk measures (value-at-risk or conditional value-at-risk) are useful tools to find a trade-off between the expected performance while mitigating the impact of adverse events that may lead to poor performance of the objective function.

An approximate branch-and-bound approach is designed and implemented, exploiting bounding criteria for the objective function under study. A testing is provided on a set of instances generated according to parameters matching the main characteristics of the real industrial problem.

2 State of art

Production planning and scheduling in a remanufacturing environment can be way more complicated than in traditional manufacturing environments since processing routes and times may depend on the condition of the returned products. Research in the scheduling of remanufacturing environment has identified the combinations of finite capacity, part commonality and stochastic parameters as essential elements [8].

Variable processing times has been also labelled as the main source of uncertainty in remanufacturing environments [9]. The focus of this paper is on the very common class of two-machine flow-shop scheduling problems [10]. Nevertheless, the stochastic version of this problem, with processing times modelled in terms of stochastic distributions, has been much less studied than its deterministic counterpart, due to its complexity [11].

Most of the existing literature on the stochastic two-machine flow shop scheduling problem to minimize the makespan focus on minimizing its expected value, which can be denoted as $F2|prmu|E(C_{max})$. Gourgand et al. [12] review the majority of the revised works on stochastic two machines flow shop scheduling problems and they pointed that the stochastic two machine flow shop scheduling problems with specific features has been extensively studied. For example, for the processing times with exponential distribution and expected makespan minimization as the objective function, Talwar's rule is proposed [13] and it has been proved to be optimal [14]. For the case in which processing times of each job on machine 1 and machine 2 follow the same distribution, a dispatching rule to minimize the expected makespan is given [15], and three heuristic solutions are proposed based both on Talwar's rule and Johnson's rule [16].

Besides the expected makespan criterion, the absolute deviation robust scheduling problem of a two-machine flow shop, which can be denoted as $\mathrm{F2|prmu|R(C_{max})}$, was also considered [17]. In this study, they present a measure of schedule robustness that considers the poorest system performance over all potential realizations of job processing times and discuss two frameworks for structuring processing time uncertainty which are discrete processing time scenarios and continuous processing time intervals.

Despite these advances on the two-machine flow shop scheduling problems with processing time uncertainty, minimizing the expected makespan fails in estimating the quality of the schedule in a stochastic point of view [18], the absolute deviation robust approach pays too much importance on particularly worst-case scenario, which in reality, may be unlikely to occur, therefore, using this approach tends to be too conservative. Considering the limitations of $\mathrm{E(C_{max})}$ and $\mathrm{R(C_{max})}$criterion, in this paper, we propose a performance evaluation criterion based on a risk measure, the value-at-risk, defining the problem $\mathrm{F2|prmu|VaR(C_{max})}$ [19].

Risk measures have been broadly used in the financial area such as the portfolio management field to hedge against uncertainties and deal with extreme scenarios, but they are rarely addressed in manufacturing scheduling field, and most of the existing literature apply the risk measure approach to the single machine scheduling [20], make to order production [21, 22, 23] and assembly line scheduling [24]. Tolio et al. [25] propose a branch and bound method to solve a single resource scheduling problem in a hard metal tools production environment aiming at minimizing the risk of maximum tardiness. Sarin et al. [26] formulate a scenario-based mixed-integer program formulation for minimizing conditional value-at-risk for the single machine and parallel machine scheduling problem with total weighted tardiness objective function. Atakan et al. [27] address a single machine scheduling problem aims at minimizing the value-at-risk of the total tardiness and the total weighted tardiness. Chang et al. [28] present a robust optimization model for the single machine scheduling problem with random job processing time with mean and covariance information to find an optimal schedule by minimizing the conditional value-at-risk of total flow time. Urgo et al. [19] apply a branch-and-bound approach for the stochastic single machine scheduling problem with uncertain processing time and release time aims at minimizing the value-at-risk of maximum lateness. Kasperski et al. [29] discuss a wide class of single machine scheduling problems with uncertain job processing times and due dates, and applied risk measure approaches (VaR and CVaR) to get a solution. Meloni et al. [30] evaluate the conditional value-at-risk of makespan for a resource constrained project scheduling problem in which for each activity only the interval for its integer valued duration is known.

3 Problem formulation

We consider a simplification of the whole remanufacturing process for turbine blades into a two-machine permutation flow shop scheduling problem. This simplification can be used to focus on a critical subpart of the whole process, or to be applied to an aggregate process considering macro-machines instead of the detailed activities.

We consider a set of *n* jobs, {1, 2,. . . ,n}, to be processed on two machines in series. The routing of the jobs through the shop is given and the processing time of an operation *i* of job j, p_{ij} $(i = 1{,}2;\ j = 1, \dots, n.)$ is modelled as an independent stochastic variable with a discrete distribution. Because of stochastic processing times, the makespan is also a stochastic variable depending on p_{ij}, as well as on scheduling decisions. The aim of the scheduling approach is to minimize the VaR (value-at-risk) of the makespan. The formulation of the scheduling problem and the associated objective function is operated according to the notations proposed in [31].

A schedule decision vector $\boldsymbol{x}$ defines the positions of the jobs in the sequence while a vector of random variables $\boldsymbol{y} = [p_{11}, \dots, p_{n2}]$ defines the stochastic processing times. These variables are governed by a probability measure $\boldsymbol{P}$ on $\boldsymbol{Y}$ and are independent of scheduling decision $\boldsymbol{x}$. The probability distribution of the makespan, $f_{C_{max}}(\boldsymbol{x}, \boldsymbol{y})$, depends on the values of $\boldsymbol{x}$ and $\boldsymbol{y}$. And the cumulative density distribution $F_{C_{max}}(\boldsymbol{x}, \zeta)$ can be denoted as Eq. (1).

$$F_{C_{max}}(\boldsymbol{x}, \zeta) = P(C_{max} \leq \zeta \mid \boldsymbol{x}) = P(\boldsymbol{y} | f_{C_{max}}(\mathbf{x}, \mathbf{y}) \leq \zeta) \quad (1)$$

Given that the scheduling decision vector $\boldsymbol{x}$ is independent from the values of the stochastic variables in $\boldsymbol{y}$, and C_{max} is a regular scheduling objective function [32], then $C_{max}(\boldsymbol{x}, \boldsymbol{y})$, which is also a stochastic variable, is continuous and non-decreasing in $\boldsymbol{y}$. and the value-at-risk α (VaR_{α}) of C_{max}, associated with a schedule decision $\boldsymbol{x}$, denoted as $\zeta_{\alpha}(\mathbf{x})$, is defined according to Eq. (2). This expression can be exploited for the design of an algorithm looking for a schedule to minimize the value-at-risk of the makespan.

$$\zeta_{\alpha}(\boldsymbol{x}) = min(\zeta | F_{C_{max}}(\boldsymbol{x}, \zeta) \geq \alpha) \quad (2)$$

4 Solution approach

In this section, we propose an approximate branch-and-bound approach for the minimization of the value-at-risk of the makespan in a stochastic two-machine flow shop scheduling problem. The approach includes a branching scheme and the estimation of proper lower and upper bounds for both partial and complete solutions.

4.1 Branching scheme

As described in Sec. 3, vector $\boldsymbol{x}$ contains schedule decisions, specifically $\boldsymbol{x_k}$ denotes the index of the job in the *k*-th position of the sequence. The branching tree is defined starting from the root node, where no job has been sequenced. From this node (level 0), n branches depart, one for each job that can be the next in the sequence, pointing to n nodes (level 1). This scheme is repeated, at each level k ($k \leq n$), jobs occupying the
k-th position in the partial sequences are defined, up to level *n*, where the nodes define a complete schedule. A node is pruned if its lower bound is higher than the incumbent solution.

The described branching scheme has the advantage of being simple, although entailing possible disadvantages. This will be discussed when commenting the results of the experiments in Sec. 5.1. Alternative branching schemes, already proposed in the literature [33, 34] will not be used in this work but taken into consideration for future improvement of the approach.

4.2 Approximate evaluation of a complete schedule (leaf node)

Leaf nodes in the branching tree are associated with a complete sequencing of the *n* jobs. An approximate evaluation of the node is operated through the estimation of bounds for the value-at-risk α (VaR_{α}) of the makespan for the schedule.

The critical path is the longest sequence of activities in a schedule, an activity on the critical path cannot be started until its predecessor activity is complete. In a two-machine flow shop problem, the critical path starts from the first operation of the first job scheduled (on the first machine) and ends with the last operation of the last job scheduled (on the second machine). Inspired by [17] on the identification of the critical path, for a given sequence, we consider all the *n* possible critical paths. Hence, we can calculate the cumulative distribution function for each of the paths. For path *k* (*k* <= *n*), the operation on the first machine of job *k* (say $O_{k,1}$) is the last one in the critical path on the first machine, then the possible critical path Π_k can be described as Eq. (3).

$$\boldsymbol{\Pi}_k = \left(O_{1,1}, O_{2,1}, \ldots, O_{k,1}, O_{k,2}, O_{k+1,2}, \ldots, O_{n,2}\right) \quad (3)$$

where $O_{j,i}$means the operation of the *j*-th job in the schedule on *i*-th machine.

Thus, the calculation of the C_{max} for path $\Pi_k$is a sum of the $n + 1$ operations, the processing time of each operation is modeled by its cumulative distribution function, thus the makespan C_{max} which can be represented as the convolution of the processing times of all operations is also a stochastic variable.

For any two subsequent operations i, j with stochastic processing times p_i and p_j modeled by their cumulative distribution functions $F_i(t) = P(p_i \leq t)$ and $F_j(t) = P(p_j \leq t)$, the cumulative distribution function of the sum of the two stochastic processing times $F_{i+j}(t)$ is the convolution of $F_i(t)$ and $F_j(t)$, and it can be represented in Eq. (4) and Eq. (5).

$$F_{C_i}(t) = F_i(t) \tag{4}$$

$$F_{C_j}(t) = F_{C_i}(t) \oplus F_j(t) = F_{i+j}(t) = F_i(t) \oplus F_j(t) \tag{5}$$

where C_i and C_j means the completion time of operation i and operation j, respectively, and $\oplus$ denotes the convolution sum operation.

Thus, given a full path from the first operation of the first job sequenced to the second operation of the last job, the cumulative distribution function of its completion time is

$$F_{C_{\Pi}}(t) = F_{\Pi^1}(t) \oplus F_{\Pi^2}(t) \oplus \ldots \oplus F_{\Pi^n}(t) \oplus F_{\Pi^{n+1}}(t) \tag{6}$$

where $\mathbf{\Pi}$ denotes the considered path, $|\mathbf{\Pi}| = \mathrm{n} + 1$, $\mathbf{\Pi}^i$ denotes the i-th operation in this path, and $C_{\mathbf{\Pi}}$ means the completion time of path $\mathbf{\Pi}$.

For any given schedule, *n* possible critical paths $(\mathbf{\Pi}_1, \mathbf{\Pi}_2, \ldots, \mathbf{\Pi}_n)$ and their corresponding $F_{C_{\mathbf{\Pi}}}(t)$ calculated.

An upper and lower bound of the distribution of the makespan can be obtained by Eq. (7) and Eq. (8), respectively, approximating the real distribution (see Fig. 1) [35],

$$F_{C_{max}}^{UB}(t) = min\{F_{\Pi_1}(t), F_{\Pi_2}(t), \ldots, F_{\Pi_n}(t)\} \tag{7}$$

$$F_{C_{max}}^{LB}(t) = F_{\Pi_1}(t) \cdot F_{\Pi_2}(t) \cdot \ldots \cdot F_{\Pi_n}(t) \tag{8}$$

where $\mathbf{\Pi}_\mathrm{i}$ denotes the i-th possible critical path we obtained as described above.

From the upper and lower bounding distributions obtained, we can derive the corresponding lower and upper bound value of the VaR of the makespan (Eq. (2)).

The estimation of the real cumulative distribution function of the makespan is way much difficult. In fact, it entails considering the correlation among the different possible paths. Its exact calculation entails complex integration procedures [36, 37], while approximated solutions usually rely on Monte Carlo simulation [38]. Both the approaches are not suitable to be embedded in a branch-and-bound algorithm due to the high computational effort and consequent low performances in terms of solution time.

To address this issue, we will ground on an approximate evaluation of complete schedules based on the upper bound distributions described in Eq. (7), from which the lower bound of the VaR can be estimated.

Thus, during the search process of the branch-and-bound algorithm, the obtained lower bound will be considered as the incumbent solution. Note that this is an approximate evaluation of the VaR in leaf nodes and this could lead to better solutions to be pruned. Nevertheless, a more conservative approach using the lower bound distribution (Eq. (8)) and the upper bound of the VaR would entail very long solution times and the frequent situation of not being able to identify a solution due to the impossibility for the branch-and-bound algorithm to close the gap.

To assess the impact of this approximation, in the experiment section, the distance between the lower bound value and upper bound value will be assessed for the proposed solutions, to demonstrate that the approximation is reasonable. Moreover, a Monte Carlo simulation is adopted to further assess the accuracy of this approximation with respect to the actual value of the VaR of the makespan for the obtained schedules.

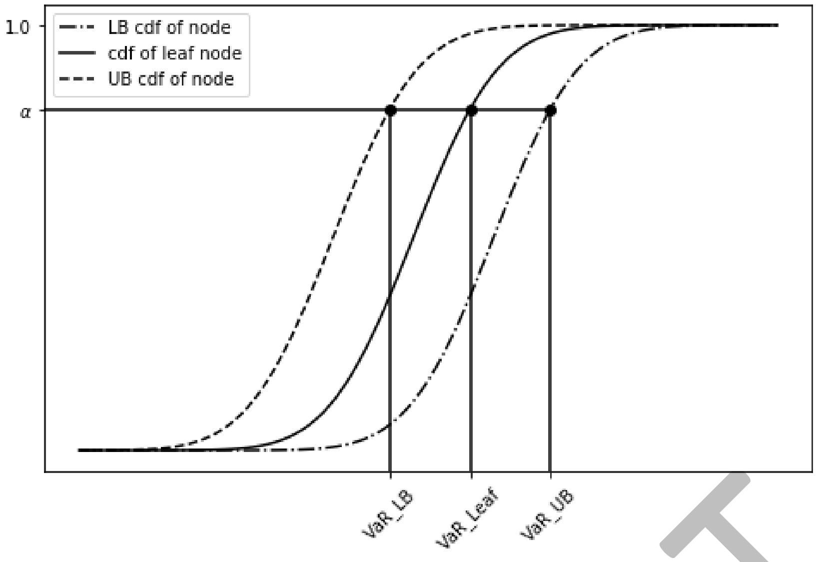


Fig. 1. Calculation of upper and lower bounds for the VaR at each node

4.3 Evaluation of a partial schedule

For nodes in the branching tree that are not representing a complete schedule, a lower bounding procedure is proposed. Grounding on the branching scheme, nodes associated with a partial schedule have *s* jobs already sequenced in the first *s* positions, while the remaining *n-s* jobs have to be still sequenced. Similarly to what described for leaf nodes, the possible *s* critical paths involving the already scheduled jobs are evaluated and an upper bound distribution of the maximum of them computed, (see Eq. (7)).

Thus, for the sequenced jobs, we can estimate the makespan distribution with Eq. (9), in which *j* denotes the job index, L_j denotes the location of job *j* in this schedule, *S* is the assigned job index set, C_k means the makespan of scheduled jobs in the path whose last critical job is job *k* and $\oplus$ means the convolution sum operation.

$$F_{C_{k,a}}(t) = \left(\oplus F_{(j,1)}(t)\ for\ j \in S\ and\ L_j <= k\right) \oplus \left(\oplus F_{(j,2)}(t)\ for\ j \in S\ and\ L_j >= k\right) \quad k = 1, \dots, s \tag{9}$$

And the upper bound distribution for the longest path can be computed by Eq. (10).

$$F^{UB}_{C_{max,a}}(t) = min\left\{F_{C_{1,a}}(t), F_{C_{2,a}}(t), \dots, F_{C_{k,a}}(t)\right\} \tag{10}$$

With respect to the unscheduled jobs, the minimum time needed to complete them can be computed in terms of the sum of their operations to be executed on the second machine, see Eq. (11).

$$F_{C_b}(t) = \oplus F_{(j,2)}(t)\ for\ j \in A/S \tag{11}$$

Thus, the upper bound distribution for this node can be calculated with Eq. (12).

$$F^{UB}_{C}(t) = F^{UB}_{C_{max,a}}(t) \oplus F_{C_b}(t) \tag{12}$$

Grounding on this upper bound distribution (Eq. (12)), the lower bound of the VaR can be derived (Eq. (2)) and assigned to this node.

5 Computational results

The proposed branch-and-bound algorithm has been coded using C++ and taking advantage of the Bob++ [39] and Boost [40] libraries. To investigate its computational performance, a set of experiments have been designed ad executed. All the experiments ran on a MacBook Pro with a 2.6 GHz Intel Core i5 and 8 GB of RAM.

5.1 Generation of the test instances

A set of instances have been generated to assess the performance of the proposed approach. A generation approach proposed in [17] has been used, taking into consideration knowledge related to the remanufacturing of the turbine blades, namely:

a. The variability of the processing times is related to the unpredictable degree of wear and presence of defects.
b. The mode of the processing time can be lower or higher than the expected value. This is due to the fact that, if a blade results to be too severely damaged during the repair process, it will be discarded. Hence, the duration of the process can be less than the ideal one.
c. Few data are usually available, due to the fact that the repair process takes months and remanufacturing activities have been established only in recent years.

Thus, the test instances have been generated based on the following rules:

1. the processing times of the jobs on each machine follow discrete triangular distributions generated by defining a lower limit value a, a mode value c and an upper limit value b;
2. the lower limit value of the processing time for each job j = 1, 2,...,n on each machine i = 1,2 is drawn from a uniform distribution of integers on the interval $a_{ji} \in [10\beta_i, (10 + 40\alpha_1)\beta_i]$, where $[\beta_1, \beta_2] = [[1.0,1.0], [1.0,1.2], [1.2,1.0]]$, and $\alpha_1 = (1.0,0.6,0.2)$
3. the upper limit value of the processing time for each job j on machine i is randomly drawn from a uniform distribution of integers on the interval $b_{ji} \in [a_{ji}, a_{ji}(1 + \alpha_2)]$, where $\alpha_2 = (0.2,0.6,1)$
4. the mode value of processing time is drawn from a uniform distribution of integers on the interval $c_{ji} \in [a_{ji}, b_{ji}]$

A further consideration has to be provided with respect to the dominance of the possible dominance of the first or second machine in the flow shop. Stochastic dominance is a type of comparison between random variables based on some properties i.e., a random variable dominates another with respect to some stochastic property [32]. Thus, with the aim at focusing on scheduling problems where the sequencing is not dominated by the performance of the first machine, the following rules have been used to discard instances likely to exhibit this dominance:

a. The number of jobs whose processing time of the operation on the second machine is stochastically dominated (first order) by the processing time of the operation on the first machine must less or equal than 50% jobs.
b. The sum of processing times of the operations on the first machine must be stochastically dominated (first order) by the sum of the processing times on the first machine. For simplicity, we can simplify this rule as follows: sum of the second machine operation's lower interval value ($\sum a_{j2}$) is larger than sum of the first machine operation's lower interval value ($\sum a_{j1}$), the sum of the second machine operation's upper interval value ($\sum b_{j2}$) is larger than the sum of the first machine operation's upper interval value ($\sum b_{j1}$), here a and b denotes the lower and upper interval value in a triangular distribution.

The generated instances are used to run the optimization algorithm with different risk levels α (1, 5 and 10%) and instances with n = 10, 20, 30 jobs have been considered. Namely, 10 instances are generated for each combination of the possible number of jobs n, the risk level α, and the parameters used to generate the instances (α_1, α_2 and $[\beta_1, \beta_2]$), for a total of 2430 instances.

To assess the performance of the proposed approach, the following aspects have been analysed:

1. The performance of the algorithm is evaluated in terms of the solution time (time to find the optimal solution), how it scales as the dimension of the problem increases, and the accuracy of the approximation in the evaluation of leaf nodes evaluation.

2. The comparison with other stochastic scheduling approaches, specifically, the minimization of the maximum regret.

5.2 Performance of the algorithm

The results in Table 1 show the performance of the branch and bound algorithm in terms of solution time, fraction of the leaf nodes visited in the evaluated nodes are represented as well. The average, minimum, maximum, and standard deviation are reported for each combination of job number and risk level. Note that the fraction of the evaluated nodes in the branching tree of the total number of nodes in a enumeration tree are all almost 0% in all the instances, which we don't present in the table.

Considering all the instances, the algorithm is able to find the optimal solution in an average time of 3.798 s, ranging from a minimum value of 0.03 s to a maximum value of 80 s. Moreover, the average number of leaf nodes visited is about only 1.32% among all the evaluated nodes, which proves the efficiency of the lower bound.

Table 1. Results

Number of jobs	Risk level		Mean	Min	Max	SD
10	1	Solution time (s)	0.0829	0.0042	0.6154	0.0795
		% Leaf nodes	2.7	0	5.6	1.4
	5	Solution time (s)	0.0865	0.0047	0.5782	0.0827
		% Leaf nodes	2.7	0	13.5	1.6
	10	Solution time (s)	0.0845	0.004	0.561	0.0806
		% Leaf nodes	2.8	0	7.1	1.4
20	1	Solution time (s)	1.493	0.0155	9.3982	1.4789
		% Leaf nodes	0.9	0	5.9	0.5
	5	Solution time (s)	1.5522	0.0162	10.1651	1.5905
		% Leaf nodes	0.8	0	2	0.4
	10	Solution time (s)	1.7567	0.0254	9.0531	1.4728
		% Leaf nodes	0.8	0	3	0.4
30	1	Solution time (s)	9.2934	0.0462	83.2805	10.1144
		% Leaf nodes	0.4	0	1.3	0.2
	5	Solution time (s)	9.6959	0.0439	50.5855	8.5714
		% Leaf nodes	0.4	0	0.7	0.2
	10	Solution time (s)	10.1369	0.0458	54.6699	9.1156
		% Leaf nodes	0.4	0	0.8	0.2

In this algorithm, the solution time strictly depends on the number of evaluated nodes, Fig. 2 shows the correlation between the solution time and the number of evaluated nodes for instances with different number of jobs. Further, for each evaluated node, the convolution calculation is the most time-consuming part, Fig. 3 shows that the instances with larger job number will cost more time per node, this is straightforward since calculations with cumulative density functions having larger support intervals in these instances.

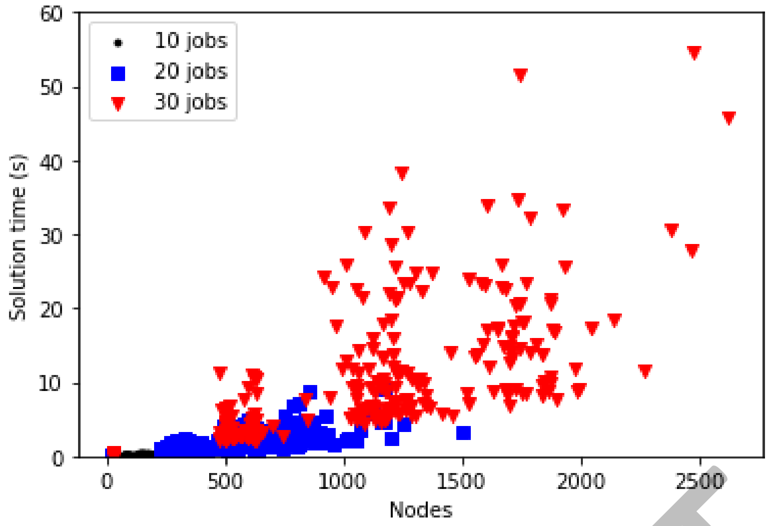


Fig. 2 Solution time respect to the number of visited nodes

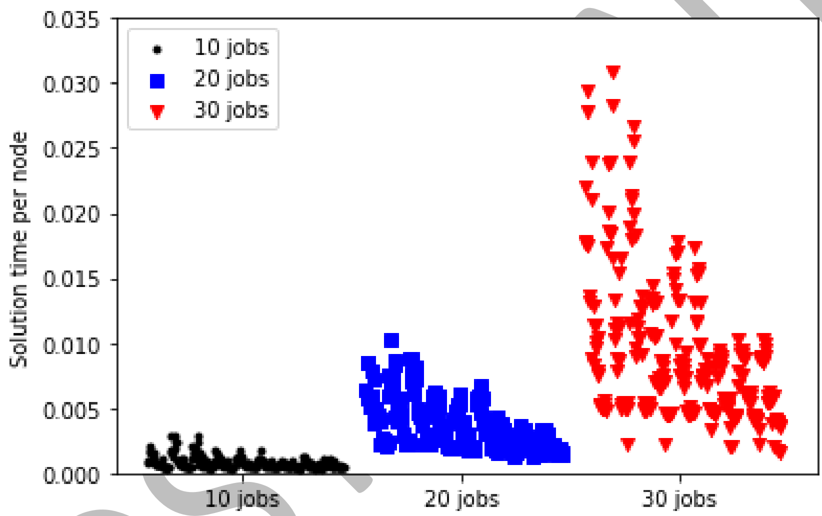


Fig. 3 Solution time spent in each of the evaluated nodes

To assess the relevance of the risk level and number of jobs on the performance of the algorithm an analysis has been carried out using the ANOVA. The results are presented in Table 2, showing that instances with different number of jobs, entail significant differences in the solution time, while there is no statistical evidence that the algorithm performance is affected by the selected risk levels for the VaR. The boxplot of the solution time of the algorithm with respect to both the factors are presented in Fig. 4.

Table 2. Two factors ANOVA results

	F-value	p-value
Job Number	734.4	10^{-25}
Risk Level	0.94	0.39
Job Number: Risk Level	0.42	0.79

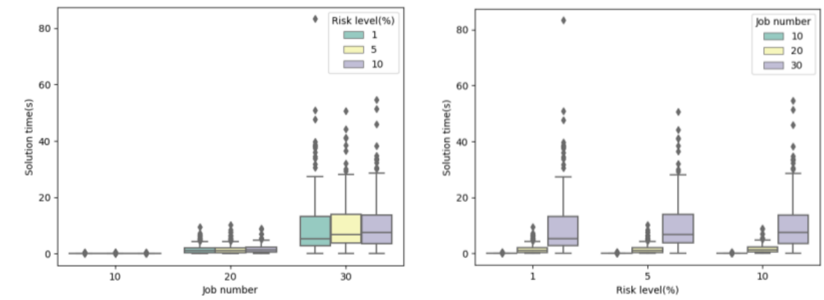


Fig. 4 Boxplot of solution time on two factors

To validate the accuracy of the approximation on the leaf node evaluation, for all the instances, we collect the interval length between the lower bound VaR value and the upper bound VaR value of the solution schedule, and we conclude Eq. (13), this means the relative distance from our solution VaR value to the real best possible VaR value is no larger than 1%.

$$\frac{UB-LB}{LB} * 100\% <= 1\% \quad (13)$$

In addition, to estimate the error caused by using the lower bound of the VaR as the real value, for the optimal schedules obtained while solving the test instances, Monte Carlo simulation has been used to estimate the actual value of the VaR. Thus, Eq. (14) is used to estimate the relative error, where VaR_{simu} is the VaR obtained thorough the Monte Carlo simulation, $VaR_{B\&B}$ is the values obtained through the approximated branch and bound algorithm. For all the instances solved, the relative was not larger than 0.6 %.

$$\delta = \frac{|VaR_{simu} - VaR_{B\&B}|}{VaR_{BB}} * 100\% \quad (14)$$

5.3 Comparison with a minimax regret approach

To evaluate the benefits of a scheduling approach based on the VaR, an alternative robust scheduling approaches, minimizing the maximum regret of the makespan [17] has been implemented and tested on the same instances.

For each instance, the optimal solution minimizing the maximum regret of the makespan (s_2) is obtained and compared with the scheduled obtained through the proposed approximate branch-and-bound approach (s_1) aiming at the minimization of the VaR. Thus, we expect the VaR of s_2 to be larger or equal to the one of s_1, while the max regret value of s_1 is likely to be larger or equal than the one of s_2.

Besides these reasonable considerations, a possible way to compare the two schedules is calculating the probability for schedule s_1 to incur in a maximum value of the makespan potentially higher than s_2. Thus, given max_1 the maximum possible value of the makespan for s_1 and max_2 the maximum possible value for s_2, the probability of having $max_1 \geq max_2$ has been evaluated numerically grounding on the distribution of the makespan for the two schedules (Fig. 5). Grounding on these results, we conclude that

$$P(max_1 \geq max_2) \approx 0 \quad (15)$$

which means that the solution obtained with the proposed approach is not worse than the minimax regret approach in terms of the maximum makespan. The detailed results of the experiments are not reported since almost all of the results are equal to 0.

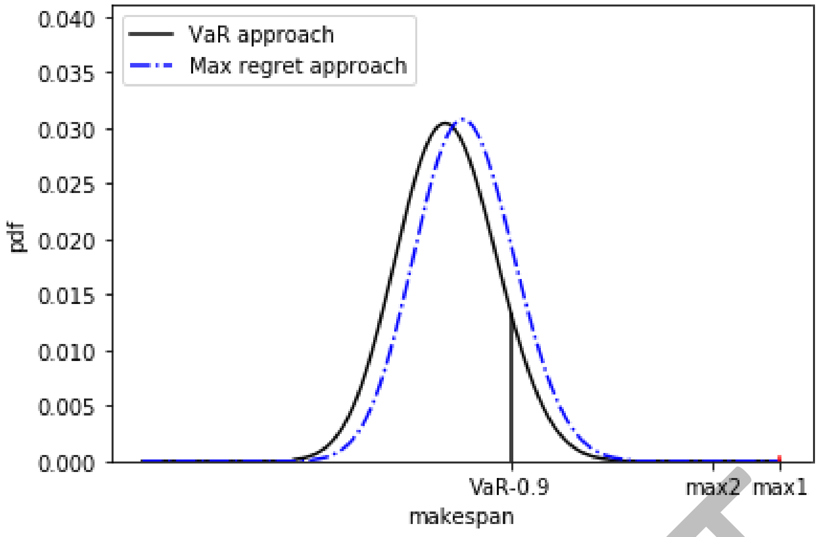


Fig. 5 pdf of objective function values under two approaches

A second comparison of the two approaches has been carried out in terms of the solution time. The results for 10 jobs instances for both the approaches are reported in Fig. 6a (solution time) Fig. 6b (solution time deviation), as well as the gap value of the solution time between these two approaches in Fig. 7. For more than 90% of the instances with 20 or 30 jobs, it was not possible to get the results in one hour under the minimization of the maximum regret approach, this is in line with the results in [17], they only give results for instances with up to 15 jobs. The comparison is done in terms of:

$$deviation = t_1 - t_2 \tag{16}$$

and

$$Gap = \frac{t_1 - t_2}{t_2} * 100\% \tag{17}$$

where t_1 denotes the solution time obtained with the proposed approach and t_2 the solution time of the max regret approach.

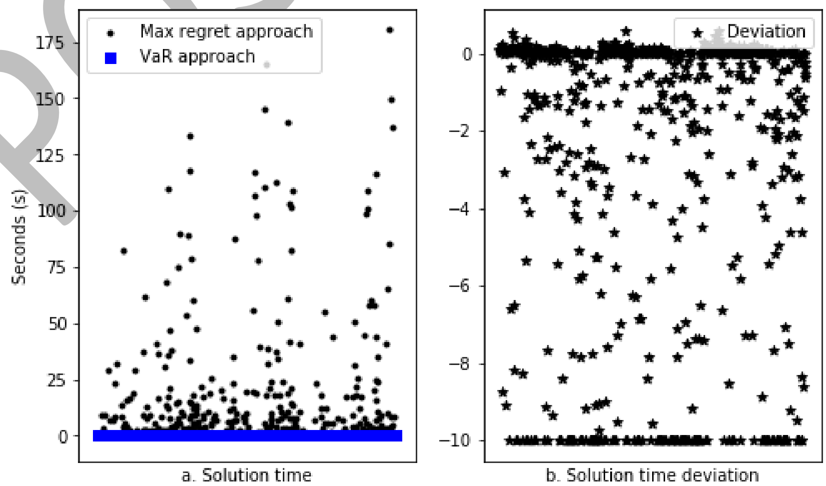


Fig. 6 The comparison of solution time of two approaches

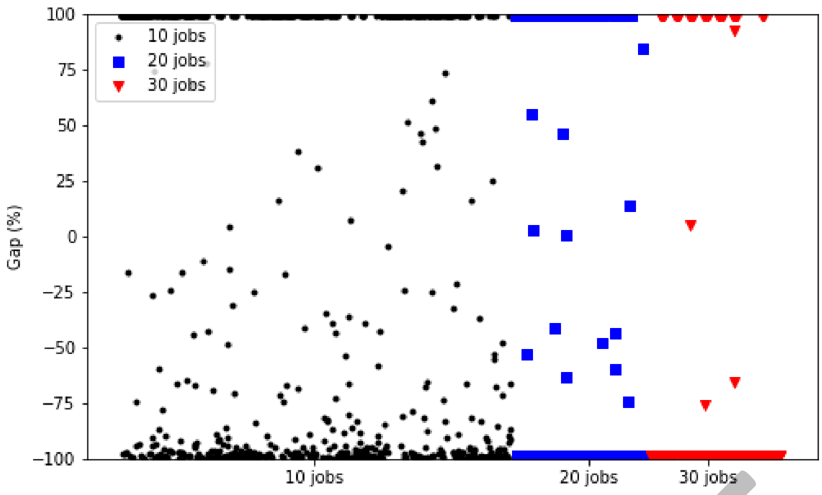
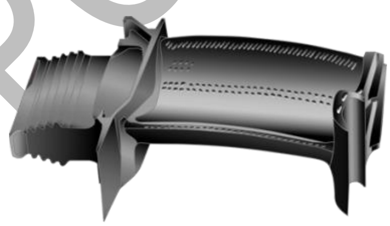


Fig. 7 Gap between the solution time of two approaches

For about half of the 10 jobs instances and a small fraction of instances with 20 and 30 jobs, the max regret approach has computational advantage over VaR approach, since the calculation of the VaR requires the estimation of the whole distribution of the objective function, while the minmax regret just relies on extreme values. Nevertheless, considering the whole picture, the VaR approach can solve the generated instances in a reasonable time. While the proposed selecting rule is not efficient for max regret approach, computational effort of this approach is highly influenced the parameter α_2, i.e. the variability of individual job processing times, which measured by the difference between $\underline{p}_{ij}$ and $\overline{p}_{ij}$ [17].

6 Industrial application

The proposed scheduling approach has been preliminary tested in a test case derived from the scheduling of remanufacturing activities for turbine blades (Fig. 8) [41, 42]. In this class of processes, a blade undergoes a dissembling, thus defects are removed by means of a machining process, a reconstruction of the original shape by adding material through laser welding, and rework the blade, finally they are reassembled and sent back to the customer.



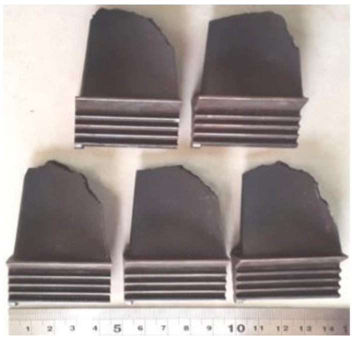


Fig. 8 Examples of blades in a gas turbine

The defect removal and laser-based additive processes are largely impacted by the uncertainty related to the state of the blade, thus, the processing times could vary according to the level of damage and the machining parameters to be used. To preliminary test the proposed approach, 3 sets of instances have been defined: 10 instances for each set, with number of jobs equal to 6, 12 and 18, respectively. Based on historical data from the industrial plant, a discrete triangular distribution is fitted to the data related to the processing times. The results are reported in Table 3, showing that the proposed approach is able to find a solution in less than 2 seconds, even in instances where the number of jobs is 18. In all the experiments, the fraction of leaf nodes actually explored is 7% on average, thus the algorithm is able to focus the search on a reduced set of promising solutions. Since the average duration of the operation in the industrial case is about 6 days, smaller in comparison with the value used in the previous experiments (larger than 30 days), the support of the associated distribution is smaller and, thus, the time needed for convolution operations is smaller. In fact, the solution time for 18 jobs instances seems smaller in comparison to the experiments in Section 5.2. This shows that the proposed approximate algorithm provides an effective scheduling tool for remanufacturing activities of turbine blades.

7 Conclusions

In this paper, an approximate branch-and-bound algorithm been described and demonstrated for a two-machine flow shop stochastic scheduling problem to minimize the value-at-risk of the makespan. The proposed approach has a direct application in the scheduling of remanufacturing activities of turbine blades. The efficiency and advantages of this approach have been analysed in comparison with an alternative stochastic scheduling approach. Further research will address the exact evaluation of the value of the objective function in leaf nodes as well as advanced branch schemes to overcome possible low performance in presence of domination of the first or second machine.

Table 3. Results of the application of the industrial case

Number of jobs	Risk level		Mean	Min	Max	SD
6	1	Solution time (s)	0.0084	0.0073	0.011	0.0014
		% Leaf nodes	14.35	12	18.18	1.98
	5	Solution time (s)	0.0061	0.0049	0.0078	0.001
		% Leaf nodes	14.9	8.69	18.51	2.85
	10	Solution time (s)	0.0067	0.0053	0.012	0.0019
		% Leaf nodes	14.79	8.69	17.64	2.49
12	1	Solution time (s)	0.1135	0.0666	0.1646	0.0332
		% Leaf nodes	4.29	2.99	5.33	0.73
	5	Solution time (s)	0.1031	0.0565	0.1482	0.0291
		% Leaf nodes	4.46	3.38	5.94	0.77
	10	Solution time (s)	0.1022	0.0755	0.1511	0.0268
		% Leaf nodes	4.95	3.01	6.15	1.06
18	1	Solution time (s)	0.6785	0.4442	0.8396	0.1381
		% Leaf nodes	2.7	1.65	3.78	0.62
	5	Solution time (s)	0.6575	0.3568	0.9735	0.2459
		% Leaf nodes	2.84	2.01	3.93	0.68
	10	Solution time (s)	0.6634	0.2865	1.1296	0.2435
		% Leaf nodes	3.1	2.23	3.81	0.57

8 Acknowledgements

We thank Ansaldo Energia for the support in the definition of the requirements in relation to the planning and scheduling of remanufacturing activities for turbine blades.

References

1. Johnson, M. R. & McCarthy I. P. (2014) Product Recovery Decisions within the Context of Extended Producer Responsibility. Journal of Engineering and Technology Management 34, 9-28
2. Tolio, Tullio, et al. "Design, management and control of demanufacturing and remanufacturing systems." CIRP Annals 66.2 (2017): 585-609.
3. Diani, Marco, and Marcello Colledani. "Cyber-Physical Systems formalization in de-and remanufacturing and application to size reduction stage." Procedia CIRP 91 (2020): 741-746.
4. Bhagi LK, Rastogi V, Gupta P (2013).Fractographic investigations of the failure of L-1 low pressure steam turbine blade. Case Studies in Engineering Failure Analysis, 1(2), pp.72–78
5. Gong, Guiliang, et al. "Remanufacturing-oriented process planning and scheduling: mathematical modelling and evolutionary optimisation." International Journal of Production Research 58.12 (2020): 3781-3799.
6. Jorion, Philippe. Value at risk: the new benchmark for managing financial risk. The McGraw-Hill Companies, Inc., 2007.
7. Campbell, Rachel, Ronald Huisman, and Kees Koedijk. "Optimal portfolio selection in a Value-at-Risk framework." Journal of Banking & Finance 25.9 (2001): 1789-1804.
8. Morgan, Shona D., and Roger J. Gagnon. "A systematic literature review of remanufacturing scheduling." International Journal of Production Research 51.16 (2013): 4853-4879.
9. Guide, V. Daniel R., Mark E. Kraus, and Rajesh Srivastava. "Scheduling policies for remanufacturing." International Journal of Production Economics 48.2 (1997): 187-204.
10. Johnson, Selmer Martin. "Optimal two-and three-stage production schedules with setup times included." Naval research logistics quarterly 1.1 (1954): 61-68.

11. Emmons, Hamilton, and George Vairaktarakis. Flow shop scheduling: theoretical results, algorithms, and applications. Vol. 182. Springer Science & Business Media, 2012.
12. Gourgand, Michel, Nathalie Grangeon, and Sylvie Norre. "A review of the static stochastic flow-shop scheduling problem." Journal of decision systems 9.2 (2000): 1-31.
13. Talwar, P. P. "A note on sequencing problems with uncertain job times." Journal of the Operations Research Society of Japan 9.3-4 (1967): 93-97.
14. Cunningham, Andrew A., and Sujit K. Dutta. "Scheduling jobs, with exponentially distributed processing times, on two machines of a flow shop." Naval Research Logistics Quarterly 20.1 (1973): 69-81.
15. Pinedo, Michael. "Minimizing the expected makespan in stochastic flow shops." Operations Research 30.1 (1982): 148-162.
16. Baker, Kenneth R., and Dan Trietsch. "Three heuristic procedures for the stochastic, two-machine flow shop problem." Journal of Scheduling 14.5 (2011): 445-454.
17. Kouvelis, Panos, Richard L. Daniels, and George Vairaktarakis. "Robust scheduling of a two-machine flow shop with uncertain processing times." Iie Transactions 32.5 (2000): 421-432.
18. Tolio, Tullio, and Marcello Urgo. "Design of flexible transfer lines: A case-based reconfiguration cost assessment." Journal of Manufacturing Systems 32.2 (2013): 325-334.
19. Urgo, M., and József Váncza. "A branch-and-bound approach for the single machine maximum lateness stochastic scheduling problem to minimize the value-at-risk." Flexible Services and Manufacturing Journal 31.2 (2019): 472-496.
20. Filippi, Carlo, Gianfranco Guastaroba, and Maria Grazia Speranza. "Conditional value-at-risk beyond finance: a survey." International Transactions in Operational Research 27.3 (2020): 1277-1319.
21. Alfieri, Arianna, Tullio Tolio, and Marcello Urgo. "A two-stage stochastic programming project scheduling approach to production planning." The International Journal of Advanced Manufacturing Technology 62.1 (2012): 279-290.
22. Radke, Andreas M., et al. "A risk management-based evaluation of inventory allocations for make-to-order production." CIRP Annals 62.1 (2013): 459-462.
23. Buergin, Jens, et al. "Robust assignment of customer orders with uncertain configurations in a production network for aircraft manufacturing." International Journal of Production Research 57.3 (2019): 749-763.
24. Urgo, Marcello. "A branch-and-bound approach to schedule a no-wait flow shop to minimize the CVaR of the residual work content." Computers & Industrial Engineering 129 (2019): 67-75.
25. Tolio, Tullio, Marcello Urgo, and József Váncza. "Robust production control against propagation of disruptions." CIRP annals 60.1 (2011): 489-492.
26. Sarin, Subhash C., Hanif D. Sherali, and Lingrui Liao. "Minimizing conditional-value-at-risk for stochastic scheduling problems." Journal of Scheduling 17.1 (2014): 5-15.
27. Atakan, Semih, Kerem Bülbül, and Nilay Noyan. "Minimizing value-at-risk in single-machine scheduling." Annals of Operations Research 248.1-2 (2017): 25-73.
28. Chang, Zhiqi, et al. "Distributionally robust single machine scheduling with risk aversion." European Journal of Operational Research 256.1 (2017): 261-274.
29. Kasperski, Adam, and Paweł Zieliński. "Risk-averse single machine scheduling: complexity and approximation." Journal of Scheduling 22.5 (2019): 567-580.
30. Meloni, Carlo, and Marco Pranzo. "Expected shortfall for the makespan in activity networks under imperfect information." Flexible Services and Manufacturing Journal 32.3 (2020): 668-692.
31. Rockafellar, R. Tyrrell, and Stanislav Uryasev. "Conditional value-at-risk for general loss distributions." Journal of banking & finance 26.7 (2002): 1443-1471.
32. Pinedo, Michael. Scheduling. Vol. 29. New York: Springer, 2012.
33. Carlier, Jacques. "The one-machine sequencing problem." European Journal of Operational Research 11.1 (1982): 42-47.
34. Gmys, Jan, et al. "A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem." European Journal of Operational Research 284.3 (2020): 814-833.
35. Ludwig, Arfst, Rolf H. Möhring, and Frederik Stork. "A computational study on bounding the makespan distribution in stochastic project networks." Annals of Operations Research 102.1 (2001): 49-64.
36. Schmidt, Craig W., and Ignacio E. Grossmann. "The exact overall time distribution of a project with uncertain task durations." European Journal of Operational Research 126.3 (2000): 614-636.

37. Demeulemeester, Erik Leuven, and Willy S. Herroelen. Project scheduling: a research handbook. Vol. 49. Springer Science & Business Media, 2006.
38. Ragsdale, C. (1989), The Current State of Network Simulation in Project Management Theory and Practice, Omega, 17(1), 21-25.
39. Djerrah, A., et al. "Bob++: Framework for solving optimization problems with branch-and-bound methods." 2006 15th IEEE International Conference on High Performance Distributed Computing. IEEE, 2006.
40. Boost, Boost C++ Libraries, http://www.boost.org/(2020).
41. Wikipedia page, https://en.wikipedia.org/wiki/Turbine_blade, last accessed 2021/03/04.
42. Kolagar, A. M., et al. "Failure analysis of gas turbine first stage blade made of nickel-based superalloy." Case studies in engineering failure analysis 8 (2017): 61-68.