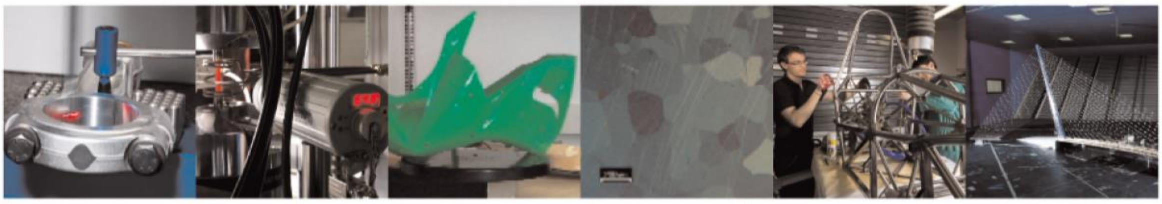

A branch-and-bound approach to schedule a no-wait flow shop to minimize the CVaR of the residual work content

Urgo, Marcello

A branch-and-bound approach to schedule a no-wait flow shop to minimize the CVaR of the residual work content.

Abstract

The aerospace industry ranks among the largest manufacturing industries in the world facing a significant growing phase as well as an increased competition. This paper addresses the scheduling of a set of jobs in a paced assembly line in presence of uncertainty affecting the availability of production resources, stemming from the assembly process in the aircraft manufacturing industry. The production problem is modeled as a *no-wait paced permutation flow shop* and solved providing a robust scheduling solution minimizing the conditional value-at-risk ($CVaR$) of the *residual work content*, i.e., the amount of workload that cannot be completed during the cycle time in the stations, due to a lack of available resources. A branch-and-bound approach is developed and applied to randomly generated instances as well as to an industrial problem related to the production of aircrafts.

Keywords: Stochastic scheduling, Robust scheduling, Conditional value at risk, Aircraft manufacturing

1. Introduction and Problem Statement

The aerospace industry ranks among the largest manufacturing industries in the world in terms of people employed (counting 2 million directly involved and 6 million considering the entire industrial ecosystem), and value of output, with the turnover of the top 20 companies reaching $ 500 billion in 2014 [1].

Although it can be considered a mature market, with almost $200 billion worth of global net orders in 2016 and more than 23000 aircrafts currently in service, commercial aviation is a dynamic and continuously growing industry. The two major civil aircraft manufacturers, Airbus and Boeing, are expecting a robust growth of the air travel demand for the future. More specifically, Airbus predicts a demand for almost 35000 new aircrafts in the next 20 years, whereof 60% shall contribute to the global fleet growth and 40% will replace old aircrafts [2]. On the other hand, Boeing estimates 41000 new deliveries in the period between 2017 and 2036, for a market value of $6.1 trillion in its global market forecasts [3].

At the same time, aircrafts production is historically characterized by high levels of backlog. During last years, the backlog even increased, reaching a global value of almost 10 equivalent years of production [4].

Grounding on these considerations, aircraft manufacturers are strongly focused to increase the efficiency of their production facilities to take advanced of the current business opportunity and face the competition of emerging manufactures in Asia and South America.

The focus of this work is aligned with this perspective, addressing the scheduling of the assembly process of aircrafts. More specifically, this work aims at providing a robust approach for the sequencing of the final assembly process of aircrafts, coping with the uncertainty possibly affecting production resources.

The final assembly process of modern civil aircrafts is traditionally operated in paced flow shops. Aircrafts enter the first station of the assembly line and proceed to the following according to a cycle time whose value is in the order of some days. In each station assembly operations are performed taking advantage of production resources, i.e., tools, fixtures, and workers. Although modern aircrafts are a rather standardized design, they are extremely complex and customized products counting more than 3000 operations to be accomplished in their assembly process. Most of these relates to structural components and are the same for a given model of aircraft. On the contrary, other operations depends on the specific customization selected and, hence, the actual workload assigned to the stations can vary. The assembling of complex make-to-order products is often modeled as the execution of one-of-a-kind project, paving the way to the adoption of project scheduling approaches able to cope with the variable characteristics of the jobs to be processed [5].

Due to the constraints imposed by the cycle time, this variability entails an intrinsic planning complexity and workload peaks have to be managed through overtime working hours or by shifting the completion of some operations to the next stations, disturbing the standard execution of the process. For this reason, being able of sequencing the aircraft with the objective of minimizing the deviation of the workload in the stations respect to the expected one is a major requirement. An additional complexity to this planning problem is added by the uncertainty typical in such complex process:

1. the availability of the workforce can vary over time due to personnel sickness, absenteeism or interference with the requirements coming from the other assembly processes;
2. assembly operations can require more effort than expected in case additional tests and verifications are required to match the stringent safety regulations of the aircraft industry;
3. the delay of components to be assembled could cause unpredicted delays.

Although all of these factors are relevant, the availability of workforce is the prominent source of uncertainty since it has the highest impact on the management of the production. In this paper we address the scheduling of a set of jobs in a paced assembly line in presence of uncertainty affecting the availability of production resources, modeled as a random variable. The manufacturing system is modeled as a no-wait permutation flow-shop [6]. The proposed approach addresses the definition of a robust schedule minimizing the conditional value-at-risk ($CVaR$) of the *residual work content*, i.e. the amount of workload that cannot be completed during the cycle time in the stations, due to a lack of available resources. A branch-and-bound approach is developed to solve the described problem to optimality. The objective function used, the $CVaR$, is a measure of risk widely used in the financial research, e.g. in portfolio optimization [7] and is used as a robustness measure associated to a schedule. The proposed approach is tested on randomly generated instances and finally applied to an industrial case taken from the aeronautic industry.

2. State of the Art

Flow shop scheduling is a rather common research area. A wide survey on this class of problems is provided in [6] covering formalization and terminology aspects, theoretical results as well as optimization approaches and algorithms for different classes of problems, e.g., hybrid, permutation or no-wait flow shop. Additional surveys are [8] and [9] specifically addressing more than 400 scientific contributions for no-wait problems, i.e., systems where, for technological or other reasons, once a job is started, it must flow through every stage to completion without any delay in process, thus, without the possibility to wait in a buffer.

Since the general flow shop scheduling problem is $\mathcal{NP}$-hard (except for the simple case of 2-machine systems and some particular cases of the 3-machine systems [6]), most of the research efforts have been devoted to the development of approximated solution approaches with the aim to obtain near-optimal solutions in a reasonable computational time rather than solving them to optimality [10, 11, 12, 13].

Among the available solution approaches, the branch-and-bound is a common tool but, due to the complexity of the flow shop scheduling problem, only few contributions are available addressing flow shops with multiple machines. In [14], a branch-and-bound approach is used for a permutation flow shop scheduling problem whose branching scheme considers different subproblems representing a sub-set of jobs at the end of the complete sequence. Starting from this partial sequence, new sub-problems are derived by adding one more jobs before the already defined sequence. A depth-first strategy is pursued, prioritizing nodes with longer partial sequences to find a first complete solution as soon as possible. The same problem has been addressed in [15] developing new bounds able to cope with a number of jobs to schedule greater than 20. In [16], the special case of the no-wait flow shop problem with due dates is considered and a mixed integer programming model, two quadratic mixed integer programming models, two constraint programming models and a novel graph representation are provided. The aim is to identify a large number of infeasible solutions and support exact solution algorithms based on the enumeration of the remaining possible schedules.

All the papers cited so far aim to minimize a scalar function of the process completion time or, in some cases, a weighted sum of two or more of these functions, i.e., the total and maximum tardiness, lateness and earliness, the number of tardy or late or early jobs, etc. When addressing paced flow-shop systems, these objective functions are partially not applicable, since the flow time is univocally determined by the cycle time.

This specific class of problems has been addressed in [17], handling the problem of workforce planning in synchronous production systems. Specifically, a paced production system is considered where all the stations have the same production cycle and a feasible schedule is identified that minimizes a linear function of the workforce and total flow time costs. Heuristic algorithms are developed for problems with an arbitrary number of work stations. This work finds application in labor intensive production environments, typical in the automotive or aircraft industries. However, a relevant limitation of this approach in relation to the problem addressed in this paper is the set of decisions that can be taken. The approach in [17] considers the possibility of varying the assignment of the workforce to production cycles and stations. On the contrary, the approach we propose is aimed for short-term scheduling decisions and considers workforce availability as an input, since it has been planned at a higher level. An additional requirements is the fact that, at the execution phase, fluctuations of the availability of the workforce can occur, thus, scheduling approaches have to be able to cope with the associated uncertainty.

Stochastic and robust scheduling approaches have attracted a significant amount of research in the last years driven by the consideration that, in manufacturing processes, uncertainty can stem from different sources, both internal and external, affecting the performance of the production system.

In [18], the authors handle the problem of uncertainty affecting the processing times of the activities in a resource constrained project scheduling problem. Grounding on the fact that in uncertain contexts it is not possible to define probabilities for uncertain events, the authors propose a *minimax regret* approach aimed at the minimization of the largest possible difference in makespan between the selected solution and the optimal makespan for a given scenario. A further distinction must be done between proactive and reactive robust approaches. Proactive scheduling approaches incorporate information about uncertainty in a baseline schedule, pursuing *ex-ante* stability. Reactive approaches, on the contrary, only provide a proper strategy to modify the schedule when unexpected events occur.

A proactive-reactive approach is proposed in [19] where a two-stage stochastic programming formulation is used to devise a robust production planning problems in a manufacturing-to-order system minimizing the expected value of the makespan over a set of scenarios. First-stage decisions define the baseline schedule whereas second-stage ones represent reactions to be taken after the occurrence of uncertain events. In this case, the expression *quality robustness* is used, i.e. the insensitivity of the plan, in terms of the target performance, to the occurrence of uncertain events [20]. Thus, a robust plan can undergo modifications with the aim at preserving the performance in terms of the value of the objective function.

A different class of robust approaches is aimed at the *solution robustness*, or *stability* [20], i.e., the insensitivity of the solution in terms of the start times of the activities.

Most of the described approaches consider as objective function the expected value of a scheduling performance indicator. Although this provides significant benefits in comparison with pure deterministic approaches, the expected value is not able to fully take into account the stochastic nature of the problem. A more powerful approach is being able of balancing the expected performance and the penalty incurred with the occurrence of less probable but high-impact scenarios. To this aim, the concept of risk is suitable to support a different class of scheduling approaches.

Financial research has been paying attention to the concepts of risk indicators and robustness for a long time and recent research trends have shown a renewed interest toward these aspects also in the scheduling domain.

The Conditional Value-at-Risk has been firstly introduced as a risk indicator by [7] as a criterion for the optimization of a portfolio of financial assets to mitigate the associated risk, pitting it against the already widely used Value-at-Risk. The advantages of CVaR over VaR are described and motivated and, in addition, an approach is described to minimize the conditional value at risk of a given random variable without the need to calculate the value-at-risk. This result is relevant from a practical point of view because of the considerable difficulties related to the calculation of the VaR, especially when scenarios are used to model the uncertainty.

In [21] the use of the CVaR is introduced as a criterion for stochastic scheduling problem proposing a scenario-based mixed integer programming formulation for general scheduling problems and solving it through a L-shaped algorithm and a heuristic approach tailored to the single machine total weighted tardiness problem. The VaR is also used to support robust scheduling in [22] and [23]. The authors implement a solution approach that exploit a branch-and-bound framework to optimize the maximum weighted tardiness.

3. Problem Formulation

The systems under study consists of an assembly line organized as a flowshop where no buffer exists between the stations. Although this could seem a rather restrictive hypothesis, it is rather common in many manufacturing environments due to technological or logistic reasons. An example are Manufacturing-To-Order (MTO) environments where, due to the dimension or characteristics of the processed products (e.g., a turbine or an aircraft), it is not feasible to have one or more of them waiting in a buffer. As previously described, such systems are commonly referred to as *nondelay* or *no-wait* flow shops [6]. Another example is food processing, where canning operations must immediately follow cooking to guarantee the freshness of the product. A direct consequence of this assumption is that, since no job can overtake another while processed, only permutation schedules are possible, i.e., the sequence of jobs for the first station is the same in all the following stations.

A special case for *no-wait* flow shops, and this is the one we focus on, are *synchronous* (or *paced*) lines. In these systems, a set of jobs is to processed on m stations arranged in a series so that each job visits all the stations in the same order. The flow time in each station is the same for all the jobs and represents the production *cycle* (or *period*) measured in units of time. Paced assembly lines are also defined as *synchronous* since jobs move from a station to the next one simultaneously at the end of every cycle.

Hence, we consider a set of jobs $\mathcal{J}$ to be processed in a *paced no-wait* flow shops consisting of m stations. The line operates according to a given production period or cycle c whose length is T time periods (e.g., hours or days). Thus, the whole planning horizon is partitioned in a set of cycles $\mathcal{C}$. The processing of the jobs in the stations entails the use of a set of resources $\mathcal{K}$. Each job J_j has a resource requirement $w_{j,i,k}$ representing the requirements of resource k for job j in station i. In general terms, $w_{j,i,k}$ is a random variable, since the real amount of a resource needed to process a job can be affected by uncertainty.

Due to the characteristics of the system under study, the scheduling of the jobs has to follow the structure of the cycles, hence, an arbitrary schedule or sequence S is defined in terms of the selection of the job to enter the first station of the line at each cycle time c.

The availability $a_{k,c}$ of each resource k in each cycle c is modeled as random variable. Each scheduled job entails a resource consumption $w_{j,i,k}$ in the cycle time c where it is going to be processed in station i, according to a sequence S. These resource requirements are subtracted from the resource availability and, since this is a random variable, the remaining availability of that resource is a random variable as well.

In case the availability of a resource is not sufficient to satisfy the requirements of a job in a given cycle c, the fraction of this requirements that cannot be satisfied represents work that will not be possible to complete in that cycle.

We define the *residual work content* as the global amount of workload that cannot be completed in the stations due to a lack of resources. Let us define the *residual resource availability*, $ra_{k,c}(S)$, as the amount of resource k still available in cycle c given the schedule S. If we consider a schedule with a single job j entering the first station in the line at cycle p, its processing will entail a consumption of resources in all the m stations in the line. Namely, j will be processed in station 1 at cycle c, in station 2 at cycle $c+1$ and in the last station at cycle $c+m-1$. The resource consumption of job j at cycle c, according to a schedule S where it is sequenced in position p is:

$$rc_{j,k,c}(S) = \begin{cases} -w_{j,c-p+1,k} & \text{if } c \in \{p, \dots p+M-1\} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The resource availability $ra_{k,c}(S)$, modeling the amount of resource available in each cycle after the jobs in S have been processed, can be calculated subtracting the resource consumption calculated in Equation 1 to the amount of the available resources:

$$ra_{k,c}(S) = a_{k,c} - \sum_{j \in J} rc_{j,k,c}(S) \tag{2}$$

Since $a_{k,c}$ is a random variable and $rc_{j,k,c}(S)$ a constant, this reduces to summing the resource consumption of all the scheduled jobs in cycle c and then right-shifting $a_{k,c}$ according to this quantity. This residual availability can assume negative values, when the available resources are not enough the satisfy the requirements of the scheduled jobs. As previously declared, the fraction of the cumulative resource requirements exceeding the resource availability represents work that will not be accomplished in the cycle time and requiring overtime work or being completed at the end of the assembly process. We define the *residual work content* for a given resource k in a given cycle c as:

$$rwc_{k,c}(S) = \min(-ra_{k,c}(S), 0) \tag{3}$$

Thus $rwc_{k,c}(S)$ is a random variable assuming values in the positive domain only (if unaccomplished work is likely to occur). The *total residual work content* is defined as the cumulative amount of the uncompleted work for all the cycles and obtained through a convolution of the $rwc_{k,c}(S)$ for all $c \in \mathcal{T}$:

$$trwc_k(S) = \underset{c \in \mathcal{T}}{\circledast} \; (rwc_{k,c}(S)) \tag{4}$$

The *total residual work content* $trwc_k(S)$ is a random variable and, aiming at mitigating its impact on the assembly system, we pursue a robust schedule by minimizing its conditional value-at-risk ($CVaR$)

The cumulative distribution function of $trwc_k(S)$ is defined using the following:

$$F_{trwc_k}(S, \zeta) = P(trwc_k \leqslant \zeta | S) \tag{5}$$

As defined in [24] and using the notation in [7], the *value-at-risk* α (VaR_α) of the value of the performance indicator z associated with the decision x is:

$$\zeta_\alpha(S) = min\{\zeta | F_{trwc_k}(\mathbf{S}, \zeta) \geqslant \alpha\} \tag{6}$$

The $\alpha - CVaR$ of (5) associated to a schedule S is the mean of the α-tail distribution of $trwc_k(S)$:

$$F\alpha_{trwc_k}(S, \zeta) = \begin{cases} 0 & \text{for} \zeta < \zeta_\alpha(S), \\ [F\alpha_{trwc_k}(S, \zeta) - \alpha] \, / \, [1 - \alpha] & \text{for} \zeta \geqslant \zeta_\alpha(S), \end{cases} \tag{7}$$

Given this distribution, we aim at minimizing the $CVaR$ to guide the solution algorithm towards a schedule that could be considered optimal from a risk-related point of view.

The described problem formulation is resumed in the following sets, parameters, variables:

Sets/indices

$\mathcal{J}$: set of jobs to be scheduled, indexed by j.

M : set of stations in the flow shop, indexed by i.

$\mathcal{K}$: set of resources, indexed by k.

T : set of cycle times, indexed by c.

Paramenters

$a_{k,c}$: availability of resource k in cycle c, is modelled through a stochastic distribution.

$w_{j,i,k}$: resource consumption of job j in relation to resource k and station i, is a deterministic value.

Decision variables

S : a schedule.

$ra_{k,c}(S)$: residual availability of resource k in cycle c given the schedule S, is a random variable.

$rc_{j,k,c}(S)$: resource consumption of job j in relation to resource k and cycle time c given the schedule S, is a deterministic variable.

$rwc_{k,c}(S)$: residual work content for resource k in cycle c given the schedule S, is a random variable.

$trwc_k(S)$: total residual work content of resource k given the schedule S, is a random variable.

$F_{trwc_k}(S, \zeta)$: cumulative distribution function for $trwc_k(S)$.

$F\alpha_{trwc_k}(S, \zeta)$: $\alpha - CVaR$ of $trwc_k(S)$.

4. Branch and bound algorithm

As described in Section 3, we consider the vector of decision variables S where S_l defines which job is scheduled in position l. If these decision variables are addressed sequentially, a branching scheme is obtained with a root node (level 0) where no job has been sequenced. From this node, n branches depart, one for each job that can be the next in the sequence. In general, a node at level $l - 1 < n$ of the branching tree represents a partial schedule containing a sequence with the first $l - 1$ jobs. From this node, $n - l + 1$ branches depart, each one modeling a different job to be sequenced next. At each level l the branching tree contains $n!/(n - l)!$ nodes [25].

4.1. Nodes evaluation

At each node, a lower and an upper bound of the target performance (the *total residual work content*) are calculated to determine the most promising branches and prune the dominated ones. At each node in the branching tree, a set of jobs $\mathcal{Q} \in \mathcal{J}$ has been already scheduled while the remaining jobs in $\mathcal{J} \backslash \mathcal{Q}$ are still to be sequenced. Namely, the contribution to the objective function provided by a job j already scheduled in position p (or, equivalently, entering the assembly line at cycle time p) can be calculated according to Equations (1), (2) and (3). A different approach has to be used for the contribution of unscheduled jobs. For these, a lower bound of the resource consumption can be estimated hypothesizing that it will be equal to the lowest resource request among the unscheduled jobs $\underline{w}_{\mathcal{J} \backslash \mathcal{Q},i,k}$. In the same way, an upper bound can be calculated as the highest resource request among the unscheduled jobs $\overline{w}_{\mathcal{J} \backslash \mathcal{Q},i,k}$. Hence, given a partial schedule S, with $\mathcal{Q}$ being the set of jobs already scheduled, the upper bound of the resource consumption of a job j is:

$$\overline{rc}_{j,k,c}(S) = \begin{cases} -w_{j,c-p+1,k} & \text{if } j \in \mathcal{Q}, \quad c \in \{p, \dots p + M - 1\} \\ -\overline{w}_{\mathcal{J} \backslash \mathcal{Q},c-p+1,k} & \text{if } j \in \mathcal{J} \backslash \mathcal{Q}, \quad c \in \{p, \dots p + M - 1\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

while the lower bound is:

$$\underline{rc}_{j,k,c}(S) = \begin{cases} -w_{j,c-p+1,k} & \text{if } j \in \mathcal{Q}, \quad c \in \{p, \dots p+M-1\} \\ -\underline{w}_{\mathcal{J}\backslash\mathcal{Q},c-p+1,k} & \text{if } j \in \mathcal{J}\backslash\mathcal{Q}, \quad c \in \{p, \dots p+M-1\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Grounding on this, in each node, the lower ($trwc_k^{LB}(S)$) and upper ($trwc_k^{UB}(S)$) bound of the residual work content can be calculated by randomly assigning the entering of the unscheduled jobs in the line in the unallocated cycles, and then using Equations (2), (3) and (4).

4.2. Dominance rules

The definition of the dominance rules among the nodes of the tree gorunds on the formalization of the problem in Section 3, with the availability of the resources in each cycle $ra_{k,c}(S)$ being a random variable. On the contrary, the resource requirements for each job is a deterministic value. Let us consider a given sequence $\hat{S}$ and imagine to add a new job j in the sequence. This job will consume part of the available resources and, thus, the consequent resource availability $ra_{k,c}(\hat{S} + \{j\})$ could only be diminished in comparison with the previous one ($ra_{k,c}(\hat{S})$). This consideration entails the possibility to define be a first-order stochastic dominance of the distribution of the residual resource availability for schedule $\hat{S}$ respect to $\hat{S} + \{j\}$, i.e.,

$$ra_{k,c}(\hat{S}) \geqslant ra_{k,c}(\hat{S} + \{j\}) \tag{10}$$

Notice that this dominance is linked to the fact that, every time a new job is scheduled, the distribution of $ra_{k,c}$ does not change in shape but it is shifted to the right.

Furthermore, referring to the residual work content, every time a new job is scheduled,

$$F_{trwc_k}(\hat{S} + \{j\}, \zeta) \geqslant F_{trwc_k}(\hat{S}, \zeta) \quad \forall \quad \zeta \tag{11}$$

Due to the fact that the shape of F_{trwc_k} never changes but it is only shifted as new jobs are scheduled, the position of the CVaR respect to the distribution never changes and, as the distribution is shifted, it is shifted as well. For this reason, the dominance provided by the upper and lower bounds for the *total residual work content* can be exploited to prune the nodes in the tree.

4.3. Application example

A description of the branch-and-bound approach is provided through an example of the evaluation of a node. Let us consider a set of jobs to be scheduled in a flow shop consisting of 3 stations. The aim of the approach is to find the schedule that minimizes the $CVaR_\alpha$, with $\alpha = 0.05$, of the residual work content. As illustrated in Figure 1, let us consider the evaluation of a node with a partial schedule $S := (-, 1, -, 2, -, -, -, ...)$, with job 1 being processed in the first station at cycle 2 and job 2 at cycle 4 respectively, and let us consider a generic resource k. For these two jobs, their resource request $w_{j,i,k}$ is fitted according to the station and cycle where these jobs are processed according to S. Since S is incomplete, for the undecided positions in the sequence, an upper bound (Figure 1, below) and lower bound (Figure 1, above) of the resource requests ($\overline{rc}_{j,k,c}(S)$, $\underline{rc}_{j,k,c}(S)$) are considered. Then, the distributions of the residual work content ($rwc_{k,c}$) is computed both in the upper (Figure 1, below) and lower bound case (Figure 1, above). Operating a convolution on these distributions, then the upper and lower bound distribution of the total residual work content ($trwc_k^{UB}$, $trwc_k^{LB}$) are calculated. Starting from these distributions, upper and lower bounds of the conditional value-at-risk ($CVaR_\alpha^{UB}$,$CVaR_\alpha^{LB}$) can be obtained. According to what described in Section 4.2, the distribution of the total residual work content for any leaf rooted by the considered node lies between the upper and lower bound distributions and, consequently, also the $CVaR_\alpha^{leaf}$ of that leaf (complete schedule) is bounded by the upper and lower bounds obtained.

Figure 1: Example evaluation of a node in the branching tree.

5. Industrial Application

A first testing of the branch-and-bound approach has been carried out for validation. To this aim 10 9-jobs instances have been generated with the following characteristics:

- the resource availability of the stations within the cycle time is triangularly distributed whose parameters (a, b, c) are randomly sampled;
- the workload requirement for a job j in station m is a deterministic quantity, whose value is randomly sampled from a discrete uniform distribution.

The described instances have been solved through the branch-and-bound approach obtaining an optimal sequence for the jobs. Hence, a complete enumeration of the solutions is operated and the best sequence found compared with the optimal one obtained through the branch-and-bound approach. For all the tested instances the algorithm was able to found the optimal solution.

To assess the viability of the described branch-and-bound approach, it has been applied to the scheduling of the final assembly process of narrow-body, short-haul aircrafts (Figure 2).

Figure 2: The Airbus A320 final assembly line in Tianjin (courtesy of Airbus).

A commercial aircraft, as stated before, is an extremely complex and highly customizable product. Starting from the base aircraft model, customers select the desired customization through a set of available options. Among these, the most relevant ones entailing additional workload at the assembly phase are: in-flight entertainment devices, systems for mobile communications providing internet access and mobile telephone services, additional galleys, lavatories and stowages, additional fuel tanks to increase the operational flexibility of the aircraft, cargo loading systems to carry containers and pallets on the aircraft, rear galley and lavatory configuration for people with reduced mobility.

The assembly line implementing this process is partitioned in different segments. Our focus is on the one managed with a cycle time, which carry out all the main assembly activities for the aircrafts, i.e., the structural parts, the cabin and the customization options. This assembly line consists of five stations capable to process different jetliner models and equip them with all the possible custom features. Workers and tools are available in the stations to execute the assembly process. The tools (e.g., fixture, handling devices) are always available since they constitute a specific equipment for each station and have a negligible failure rate. On the contrary, the availability of the workers is affected by stochastic fluctuations due to personnel sickness and absenteeism. The resource requirements of the aircraft to be assembled depend on the specific model and customization but is considered deterministic. In fact, at the time of scheduling, the orders are completely defined and the workload to be accomplished in each station of the lines to assemble the aircraft known.

The approach has been tested considering the historical data of the availability of the workers in a whole year as well as the pool of orders (in total 68) assembled in the same period. The availability of the workers per working day has been modeled fitting a triangular distribution on the available historical data.

The performance of the branch-and-bound approach is evaluated in terms of the time to find the optimal solution and the fraction of explored nodes respect to the total number of nodes in the branching tree. A set of 20 9-jobs problem instances have been created by sampling the jobs to be processes from the a pool of 68 real orders assembled in the line ̀in a given period. The branch-and-bound algorithm has been used to minimize the Conditional Value-at-Risk of the Residual Work Content. Moreover, different value of risk levels has been tested, namely $\alpha = 0.05$ and $\alpha = 0.10$, for a total of 80 experiments. The branch-and-bound algorithm has been completely coded in `C++` using the `BoB++` library [26, 27] and the Boost library [28]. The experiments have been performed using 8 parallel threads on an Intel 4-Core i7 Processor 7700-HQ running at 3.4 GHz and 16 GB of DDR4 SDRAM.

The performance of the proposed algorithm are resumed in Table 1. The average time to find an optimal solution is about 4830 seconds, ranging from a

Table 1: Performance of the B&B approach.

		Solution Time [s]		**Pruning Efficiency [%]**	
Risk Level	0.05	**Avg**	4810	**Avg**	0.581
		Min	4614	**Min**	0.583
		Max	5033	**Max**	0.589
	0.10	**Avg**	4850	**Avg**	0.581
		Min	4613	**Min**	0.581
		Max	5220	**Max**	0.589

minimum of 4613 to a maximum of 5220 seconds. Also the fraction of the total number of nodes in the tree explored to find the optimum has been considered, labeled with the name *Pruning Efficiency*. Namely, the algorithm needs to evaluate 58% of the nodes on average. For both the solution time and the pruning efficiency, the risk level used seems not having any significant effect. This behavior is confirmed by the graph in Figure 3 showing the relationship between the solution time and the number of evaluated nodes for all the tests instances solved, whereas the boxplot in Figure 4 highlights the variability ranges of the solution time fo the different risk levels considered.

Figure 3: Scatterplot of the solution time against the number of explored nodes to find a solution.

Figure 3 also confirms a reasonable behavior, i.e., the dependence of the solution time from the number of nodes to be evaluated. In fact, every time a new node is explored, a set of convolution operations have to be executed, according to what described in Section 4.2.

Figure 4: Scatterplot of the solution time against the number of explored nodes to find a solution.

To assess the effectiveness of the proposed approach, a set of 68 jobs has been taken into consideration and the branch-and-bound algorithm used to find the optimal solution in terms of the CVaR. The solution required a significant amount of time (about 15 hours). The sequence obtained has been evaluated calculating the distribution of the residual work content to obtain the minimum and maximum value, the mean, the VaR and the CVaR. A characterization of the approach has been executed in two steps. First, the same problem has been solved using a deterministic approach, i.e., assuming the availability of the resources deterministic and looking for the schedule minimizing the total deviation of the requested amount of resource respect to the availability of the resource over the whole scheduling horizon. The deterministic scheduling approach has been implemented through a MIP formulation. In addition, the performance of the schedule provided by the branch-and-bound algorithm has been also compared with the one obtained using the schedule used by the company in the same planning period. For these two solutions, the calculation of the distribution of the residual work content (under the hypothesis that the resource availability is uncertain) has been carried out exploiting the same branch-and-bound algorithm. The generation of the branching tree has been constrained to be limited to the nodes (and the single leaf) representing a given solution.

The result of the comparison are reported in Table 2, values are anonymized because of their commercially sensitive nature, they are expressed as a percentage of the highest value of the residual work content found out by adopting the real schedule used in the company. The best results in terms of the minimization of the CVaR are achieved with the proposed stochastic branch-and-.bound approach. As expectable, the most consistent comparative advantage of using the proposed approach (that however implies a higher computational effort) lies in the reduction of the worst cases impact, i.e. in the lowering of 0.05-VaR, 0.05-CVaR and maximum value of the distribution.

Furthermore, the complete distribution for the solution obtained through the proposed aproach and the real schedule implemented in the company are reported in Figure 5 showing how the minimization of the CVaR actually provides a better solution with regards to this, by shaping the distribution of the residual work content so that the right tail is smaller. Nevertheless, at the same

Table 2: Comparison of the B&B solution with the deterministic and the real ones.

	CVaR	Det	Real
min	19.18%	17.30%	23.41%
max	90.88%	95.20%	100.00%
mean	46.58%	51.07%	55.47%
5%-VaR	65.60%	69.94%	75.95%
5%-CVaR	71.33%	76.40%	80.57%

time, the whole distribution is shifted towards the left causing also the mean value and the VaR to be reduced. Again, the residual work content in Figure 5 is expressed in relative terms due to confidentiality of the industrial data. Hence, a residual work content of 0.50 is an amount of hours that is 50% of the maximum amount registered. This reference value is the same for the two distributions.

Figure 5: Distribution of the residual work content obtained with the minimization of the $CVaR$ (top) and the real sequence (bottom).

6. Conclusions

This work proposed a robust scheduling approach for a no-wait flow shop under resource availability uncertainty and cycle time. The motivation for this approach stem from an industrial case in the aircraft manufacturing industry. To pursue robustness, the aim is to minimize a function of the risk associated to the residual work content, i.e., the amount of work that cannot be accomplished in the cycle time. The risk function used was the conditional value at risk. To cope with this problem, a branch-and-bound approach was designed, implemented and tested showing promising results although the computational times are rather high. Moreover, with regards to the industrial application. considerable improvements have been obtained with respect to the scheduling approach currently. Further improvements of the approach will pursue different branching schemes to foster the rapid identification of dominated or dominating scheduling decisions and reduce the solution time.

References

[1] R. S. Lineberger, A. Hussain, 2018 Global aerospace and defense industry outlook , Tech. rep., Deloitte (2018).

[2] Airbus, Global Market Forecast 2018-2037, Tech. rep. (2018).

[3] Boeing Commercial Airplanes, Current Market Outlook 2017– 2036, Tech. rep. (2017).

[4] Deloitte, Global Commercial Aerospace Industry - Aircraft order backlog analysis, Tech. rep. (Jul. 2016).

[5] A. M. Radke, T. Tolio, M. M. Tseng, M. Urgo, A risk management-based evaluation of inventory allocations for make-to-order production, CIRP Annals - Manufacturing Technology (62) (2013) 459–462.

[6] H. Emmons, G. Vairaktarakis, Flow Shop Scheduling, Vol. 182 of International Series in Operations Research & Management Science, Springer US, Boston, MA, 2013.

[7] R. T. Rockafellar, S. Uryasev, Conditional value-at-risk for general loss distributions, Journal of Banking Finance 26 (2002) 1443–1471.

[8] N. G. Hall, C. Sriskandarajah, A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process, Operations Research 44 (3) (1996) 510–525.

[9] A. Allahverdi, A survey of scheduling problems with no-wait in process, European Journal of Operational Research 255 (3) (2016) 665–686.

[10] T. Aldowaisan, A. Allahverdi, New heuristics for no-wait flowshops to minimize makespan, Computers & Operations Research 30 (8) (2003) 1219–1231.

[11] J. M. Framinan, M. S. Nagano, J. V. Moccellin, An efficient heuristic for total flowtime minimisation in no-wait flowshops, The International Journal of Advanced Manufacturing Technology 46 (9-12) (2009) 1049–1057.

[12] D. Laha, J. N. D. Gupta, A Hungarian penalty-based construction algorithm to minimize makespan and total flow time in no-wait flow shops, Computers & Industrial Engineering 98 (2016) 373–383.

[13] M. Nawaz, E. E. Enscore Jr, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega 11 (1) (1983) 91–95.

[14] Y.-D. Kim, Minimizing total tardiness in permutation flowshops, European Journal of Operational Research 85 (3) (1995) 541–555.

[15] C.-S. Chung, J. Flynn, Ö. Kirca, A branch and bound algorithm to minimize the total tardiness for m-machine permutation flowshop problems, European Journal of Operational Research 174 (1) (2006) 1–10.

[16] H. Samarghandi, M. Behroozi, On the exact solution of the no-wait flow shop problem with due date constraints, Computers & Operations Research 81 (2017) 141–159.

[17] G. L. Vairaktarakis, X. Cai, C.-Y. Lee, Workforce planning in synchronous production systems, European Journal of Operational Research 136 (3) (2002) 551–572.

[18] C. Artigues, R. Leus, F. T. Nobibon, Robust optimization for resource-constrained project scheduling with uncertain activity durations, Flexible Services and Manufacturing Journal 25 (1-2) (2014) 175–205.

[19] A. Alfieri, T. Tolio, M. Urgo, A two-stage stochastic programming project scheduling approach to production planning, The International Journal of Advanced Manufacturing Technology 62 (1-4) (2012) 279–290.

[20] M. Lambrecht, E. Demeulemeester, W. Herroelen, Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities, Journal of Scheduling 11 (2) (2007) 121–136.

[21] S. C. Sarin, H. D. Sherali, L. Liao, Minimizing conditional-value-at-risk for stochastic scheduling problems, Journal of Scheduling 17 (1) (2013) 5–15.

[22] T. Tolio, M. Urgo, J. Vancza, Robust production control against propagation of disruptions, CIRP Annals - Manufacturing Technology 60 (1) (2011) 489–492.

[23] M. Urgo, J. Vancza, 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.

[24] P. Artzner, F. Delbaen, J. M. Eber, D. Heath, Coherent Measures of Risk, Mathematical Finance 9 (3) (1999) 203–227.

[25] M. L. Pinedo, Scheduling Theory Algorithms and Systems, Springer, 2010.

[26] B. L. Cun, F. Galea, P. Vander-Swalmen, T. Menouer, Bob++: A framework to solve combinatorial optimization problems., http://www.prism.uvsq.fr/~blec/bobpp/ (2012).

[27] A. Djerrah, B. Le Cun, V.-D. Cung, C. Roucairol, Bob++: Framework for solving optimization problems with branch-and-bound methods., in: D. Kurlander, M. Brown, R. Rao (Eds.), 15th IEEE International Symposium on High Performance Distributed Computing, ACM Press, 2006, pp. 369–370.

[28] Boost c++ libraries, http://www.boost.org (2017).