Noname manuscript No.
(will be inserted by the editor)

A Branch-and-Bound Approach for the Single Machine Maximum Lateness Stochastic Scheduling Problem to Minimize the Value-at-Risk

M. Urgo · J. Váncza

Received: date / Accepted: date

Abstract The research in the field of robust scheduling aims at devising schedules which are not sensitive – to a certain extent – to the disruptive effects of unexpected events. Nevertheless, the protection of the schedule from rare events causing heavy losses is still a challenging aim. The paper presents a novel approach for protecting the quality of a schedule by assessing the risk associated to the different scheduling decisions. The approach is applied to a stochastic scheduling problem with a set of jobs to be sequenced on a single machine. The release dates and processing times of the jobs are generally distributed independent random variables, while the due dates are deterministic. A branch-and-bound approach is taken to minimise the value-at-risk (VaR) of the distribution of the maximum lateness. The viability of the approach is demonstrated through a computational experiment and the application to an industrial problem in the tool making industry.

M. Urgo
Manufacturing and Production Systems Division, Department of Mechanical Engineering, Politecnico di Milano, Milano, IT
Tel.: +39-02-2399-8521
E-mail: marcello.urgo@polimi.it

J. Váncza
EPIC Center of Excellence for Production Informatics and Control, Institute for Computer Science and Control, Hungarian Academy of Sciences, HU
Department of Manufacturing Science and Technology, Budapest University of Technology and Economics, Budapest, HU E-mail: vancza@sztaki.mta.hu

1 Introduction and Problem Statement

Production scheduling in industry needs to cope with the occurrence of uncertainty, incomplete information and unexpected events that may stem from a wide range of sources, both internal and external. The estimation of the duration of production activities could be inaccurate, as well as their resource needs, the availability of machines, workers - or production resources in general - could vary, the supplying of raw materials or work-in-progress products could be late in relation to the scheduled time, new activities like rush orders or reworks could need to be executed with a higher priority (Cao et al., 2001; Alfieri et al., 2011; Makris and Chryssolouris , 2010; Mourtzis, et al., 2012; Nonaka, et al., 2012; Attia et al., 2014; Mogre et al., 2014). Hence, robust scheduling approaches have been developed, aiming at protecting the performance of a schedule by avoiding or mitigating the impact of uncertain events.

Many stochastic scheduling approaches typically address the uncertainty in the problem through a scalar performance indicator, e.g., the expected value. Minimising the expected value of an objective function provides a significant improvement compared to pure deterministic approaches but nevertheless it fails in comprehensively estimating the quality of the schedule from the stochastic point of view (Alfieri et al., 2012; Tolio and Urgo, 2013; Yin et al., 2014).
When we aim at minimizing the expected value of a scheduling objective function, e.g, the maximum lateness, we look at good performance in terms of respecting the due dates on average, but we fail in protecting the schedule against worst cases whose probability is low.
This is a key factor for managers aiming to maximize the expected profit but also avoid the impact of very unfavorable events potentially causing heavy losses. The research in the financial area has been largely addressing risk measures in the last years, focusing on the use of indicators able to consider the impact of uncertain events both in terms of their effect and of their occurrence probability as the *Value-at-risk* or the *Conditional Value-at-Risk* introduced by Rockafellar and Uryasev (2002).
In the scheduling area, on the contrary, risk measures are less investigated even if the concept of risk is often perfectly suitable to support scheduling decisions under uncertainty (Tolio and Urgo , 2007; Tolio et al., 2011). This is mainly due to the difficulty in calculating the distribution of scheduling objective functions in most of the scheduling problems (Radke et al., 2013).

We consider the scheduling of the production in an industrial environment producing tailor-made products. We put the focus on a single production phase, receiving the work-in-progress parts from the previous production step and delivering the worked parts to the following according to a production plan.
The arrival of the parts could undergo variations in relation to the uncertain events affecting the production, thus, we consider the release date a stochastic variable. Nevertheless, the delivery of the processed parts to the following production step must respect the due dates defined in the production plan, thus,

we consider them as deterministic. Moreover, as typical in the production of tailor-made products, the production process is often executed for the first time, due to the specific characteristics of the products. First, the machines need to be setup and the process is executed a first time, verified, checked and, if needed, adjusted. Hence, the remaining parts in the lot can be processed. We model the processing time as a stochastic variable to consider the variability of the setup and process adjustment phase. Moreover, we suppose that production lots are processed one at time, hence, a whole set of resources is modeled as a single machine processing a single job at time. This could seem a very restrictive hypothesis. Nevertheless, since the considered group of resources typically work on a single product (or product type) at a time (e.g., make-to-order shops working on a single job or batch at a time, multi-model transfer lines), then a single resource model is a reasonable modeling approach. Being a robust scheduling approach, the aim is to guarantee the reduction of the propagation of the uncertain events occurring at the considered production phase throughout the whole production plant. We propose a branch-and-bound algorithm taking as the objective function a risk measure associated to the maximum lateness, specifically, the minimization of the VaR of the *maximum lateness*. We restrict the analysis to static list policies with unforced idleness allowed (Pinedo, 2008, chap. 9.5), i.e., the machine can remain idle to wait for the release time of a scheduled job even if other jobs are ready to be processed.

The paper is organized as follows: Section 2 summarizes the current advances for the existing stochastic scheduling approaches while Section 3 provides the problem statement as well as a description of the VaR, the risk measure used. Section 4 describes the principles and characteristics of the proposed branch-and-bound solution method. Section 5 reports on the computational test result and Section 6 describes the industrial application case. Finally, Section 7 concludes the paper.

2 State of the Art

In its deterministic version, the considered scheduling problem is known as $1|r_j|L_{max}$ and has been recognized to be strongly NP-hard (Lenstra et al., 1977). A review of the existing solution approach for this scheduling problem can be found in Kellerer (2004) and (Pinedo, 2008, chap.9). Further approaches addressing similar scheduling problems are presented in Benmansour et al. (2012), Scholz-Reiter et al. (2013), Gafarov et al. (2014).

In a more general perspective, the addressed scheduling problem is a special case of the problem $1|r_j, prec|f_{max}$ addressing the optimization of a generic scheduling cost f_{max} that, in our case, is a function of the lateness. This scheduling problem has been widely addressed in the literature, e.g., Carlier (1982); Baker and Su (1974); Grabowski et al. (1986); Liu (2010); Chandra et al. (2014).

If we consider this class of scheduling problem and assume that all the jobs to be scheduled are available at time $t = 0$, thus ignoring the release times, the resulting scheduling problem $(1|L_{max})$ can be solved to optimality using the *earliest due date* (EDD) rule.

Referring to the stochastic counterpart, if processing times are arbitrarily distributed and due dates deterministic, the EDD rule minimizes the expected maximum lateness (Pinedo, 2008) in both non-preemptive static and dynamic scheduling problems, as well as in the preemptive dynamic version. This is a direct consequence of the fact that the EDD rule is optimal for the deterministic version of the scheduling problem. Hence, it provides an optimal solution for any sample of the processing times. Since this is valid for all the samples, then the EDD rule also minimizes the maximum lateness in expectation (Pinedo, 2008).

This also provides a result in relation to the distribution of the maximal lateness. In fact, given a schedule S^* with maximum lateness L^*, the probability of having $L_{max} \leq L^*$ must be less or equal to the value obtained with the EDD schedule. Hence, the cumulative distribution of the maximum lateness for the EDD schedule bounds from above all the cumulative distributions of the maximum lateness for any possible schedule. This behavior can be formalized in terms of stochastic order relations (Shaked and Shanthikumar, 2007; Ross, 1983, chap.9).

The application of rearrangement inequalities to scheduling problems have been addressed in Chang and Yao (1993). Using stochastic rearrangement inequalities, the authors obtain a solution for the stochastic counterpart of many classical deterministic scheduling problems. These results have been further exploited in Zhou and Cai (1997), Cai and Zhou (2005), Cai et al. (2007) and Wu and Zhou (2008).

The stochastic scheduling literature mostly addresses the problem of minimizing the maximum expected lateness $max(E[L])$. In this case, the stochastic problem is reduced to a deterministic minimization (Zhou and Cai, 1997). On the contrary, the minimization of the expected value of the maximum lateness $E[L_{max}]$ addresses the stochastic characteristics of the scheduling problem taking into consideration the whole distribution of the objective function.

A problem from this class is analyzed in Wu and Zhou (2008), considering a set of jobs with stochastic due dates and deterministic processing times to be scheduled on a single machine to minimize the expected value of the maximum lateness ($E[L_{max}]$). A dynamic programming algorithm is developed whose performance is compared to three heuristic rules. The authors also provide an extension of the dynamic programming algorithm to cope with stochastic processing times and due dates. However, the results presented ground on the assumption that both the due dates and processing times are exponentially distributed.

Cai et al. (2007) further extended the results in Chang and Yao (1993) proving that (i) if the processing times are independent random variables and can be likelihood-ratio ordered, (ii) the due dates are independent random variables and can be hazard-rate ordered, (iii) the orders are agreeable, then the maximum lateness is stochastically minimized sequencing the jobs in non-decreasing likelihood-ratio order of the processing times $\{p_i\}$ or, equivalently, in non-decreasing hazard-rate order of the processing times $\{d_i\}$. We recall that, if L_{max} is stochastically minimized, then its distribution stochastically dominates the L_{max} distributions of all the other schedules and also $E[L_{max}]$ is minimized (Chang and Yao (1993)).

When considering the release times (both deterministic and stochastic), the problem becomes more difficult to solve. However, considering independent generally distributed release times and processing times, if the due dates are given as deterministic, the EDD rule still minimizes L_{max} but only in the preemptive case (Pinedo, 2008). Some further extension are available but only assuming that the due dates are deterministic and both the release times and processing times are exponentially distributed with the same mean (Pinedo, 2008).

Further contributions to the single problem single machine scheduling and the due dates have been proposed in Baker and Trietsch (2009) and Baker and Trietsch (2014), addressing due dates as a decision problem and modeling safety times as a way to mitigate the probability of missing due dates.

Among the stochastic objective functions different from the expected value, the variance is the most common. In fact, a trade-off between mean and variance is one of the most simple and common risk measures. A joint optimization of expectation and variance in a single machine scheduling problem has been proposed in De et al. (1992). In the area of Resource Constrained Project Scheduling, many authors have addressed robustness related approaches. Exampleas are Artigues et al. (2013), considering stochastic processing times for the jobs and proposing a scenario-relaxation algorithm and heuristic or Fang et al. (2015), considering the stochastic resource-constrained project scheduling problem and proposing an algorithm exploiting a permutation-based local search. Flow time and completion time are also common performance indicators, Sarin et al. (2009) provide closed form equations of mean and variance for a large set of scheduling problems. However, neither exact nor heuristic algorithms have been proposed for the maximum lateness single machine scheduling problem to optimize a risk-related objective function.

Different approaches address the selection and/or assessment of the robustness of a scheduling solution through simulation techniques by Burdett and Kozan (2012, 2014) also addressing specific robustness considerations in terms of the shape of the distribution of the objective function and/or its sensitivity.

As stated before, risk measures have been largely addressed in the financial area, with the aim to cope with extreme events, i.e., the ones linked to the tails of a distribution. Risk measures such as the *value-at-risk* (VaR) are used in portfolio management and a large amount of literature have been written on their mathematical properties and effectiveness in protecting investment assets.

Stochastic scheduling using risk-related measures like the VaR has been recently addressed in the literature. Atakan et al. (2016) address a similar problem minimizing the VaR of a tardiness-related objective function, namely the Total Tardiness and the Total Weighted Tardiness for a single machine. The authors proposes a Lagrangian relaxation-based scenario decomposition method considering random processing times and deterministic due dates. The approach in Atakan et al. (2016) does not take into consideration release dates and adopt a scenario based formulation to deal with stochastic variables, differently from what the present work that explicitly considers the stochastic distributions of processing times and release dates.

3 Problem Formulation

We consider a set jobs A, with $|A| = n$ to be scheduled on a single machine. Let s_j be the starting time of job $j \in A$ and p_j the processing time. The preemption of jobs is not allowed, i.e., a job cannot be interrupted until it has been completely processed at time $c_j = s_j + p_j$. Each job has a a due date d_j and a release time r_j. The scheduling problem aims at minimising a function of the maximum lateness. In the present analysis we further restrict the analysis to discrete distributions for the realease dates and processing times and discrete values for the due dates. We limit the analysis to static list policies, assuming that all the information related to jobs to be scheduled are available and, moreover, we allow unforced idleness, i.e., the machine can wait for the release time of a specific job even if there are jobs waiting for processing.

Both the release times r_j and the processing times p_j are independent stochastic variables with general discrete distributions. The objective function is a stochastic variable itself whose distribution depends on the stochastic variables p_j and r_j and on the scheduling decisions.

According to the notation used in Rockafellar and Uryasev (2002), we consider a vector of decision variables $\mathbf{x}$ defining the schedule and a vector of random variables $\mathbf{y} = r_1, \ldots, r_n, p_1, \ldots, p_n$ governed by a probability measure P on Y that is independent of x. As an example x could define the positions of the jobs in the sequence or can be used to identify precedence relations between the jobs. The values of x and y univocally determine the performance indicator $z = f(x, y)$.

For a given set of values of the decision variables in x, we consider the resulting distribution function for z:

$$\Psi(x, \zeta) = P(f(x, y) \leq \zeta) \tag{1}$$

As defined in Artzner et al. (1999) and using the notation in Rockafellar and Uryasev (2002), the *value-at-risk* α (VaR_α) of the value of the performance indicator z associated with the decision x is:

$$\zeta_\alpha(x) = min\{\zeta | \Psi(x, \zeta) \geq \alpha\} \tag{2}$$

If the stochastic variables in y are discrete, $z = f(x, y)$ is concentrated in finitely many points and $\Psi(x, \cdot)$ is a step function. This applies to scenario-based models. In such cases the definition of the VaR in (2) must be rephrased

(Rockafellar and Uryasev, 2002).

Given x, if we assume that the different possible values of $z_k = f(x, y)$ can be ordered as $z_i < z_2 < \cdots < z_N$ so that $P(z = z_k) = p_k$ and k_α is an integer value such that:

$$\sum_{k=1}^{k_\alpha} p_k \geq \alpha \geq \sum_{k=1}^{k_\alpha - 1} p_k \tag{3}$$

then z_{k_α} is the VaR_α.

In the considered scheduling problem the objective function addresses the maximum lateness, i.e., $z = f_{L_{max}}(\mathbf{x}, \mathbf{y})$ where, the decision variable $\mathbf{x}$ models the selected schedule and $\mathbf{y}$ are the stochastic variables, i.e., the processing times $\mathbf{p}$, and the relase dates $\mathbf{r}$. Being a function of the stochastic variables $\mathbf{y}$, z is also a stochastic variable.

The cumulative distribution function of L_{max} is defined using the following:

$$F_z(\mathbf{x}, \zeta) = P(L_{max} \leq \zeta | \mathbf{x}) = P(\mathbf{y} | f_{L_{max}}(\mathbf{x}, \mathbf{y}) \leq \zeta) \tag{4}$$

Given this distribution, we will aim at minimizing the VaR introduced in (3) to guide the solution algorithm towards a schedule that could be considered optimal from a risk-related point of view.

The legitimacy of using the distribution defined in (4) together with the risk measures as defined in (2) needs further considerations. As stated in Rockafellar and Uryasev (2002), the definition of the VaR in Eq. (2) requires specific characteristics of the function $f(\cdot, \cdot)$. Since, in the considered case, $f(\cdot, \cdot)$ provides the value of a scheduling objective function in terms of a vector of random variables $\mathbf{y}$ and decision variables $\mathbf{x}$, continuity and convexity properties cannot be assured in general. However, limiting the analysis to the considered scheduling problem, since $\mathbf{x}$ defines the sequence of the jobs in a single machine scheduling problem, it is independent from the values of the stochastic variables in $\mathbf{y}$. Moreover, $f_{L_{max}}(\mathbf{x}, \mathbf{y})$ is continuous and non-decreasing in $\mathbf{y}$, being L_{max} a regular scheduling objective function.
In addition, since all the stochastic variables have discrete distributions, also the objective function distribution is discrete. Hence, the definition of VaR in (3) can be used, without any additional requirement on $f(\cdot, \cdot)$.

As demonstrated in (Ma and Wong, 2010), first-order stochastic dominance also implies a dominance between the respective VaRs for any α, this will be exploited for the definition of dominance rules in a branch-and-bound approach.

4 Solution Approach

The problem is solved by the classical branch-and-bound method. The proposed method includes an appropriate branching scheme, a method for evaluating nodes and calculating upper and lower bounds, as well as the definition of a set of dominance rules.

4.1 Branching scheme

As described in Section 3, we consider the vector of decision variables $\mathbf{x}$ where x_k defines which job is scheduled in position k. This implies a branching scheme starting from the root node (level 0) where no job has been sequenced. From this node, n branches depart, one for each job in the list that can be the next in the sequence. If we consider a node at the $k-1$ level of the branching tree, the partial schedule provides the sequence of the first $k-1$ jobs while $n-k+1$ branches are connected to new nodes at level k, each one with a different job to be scheduled as the next one in the sequence. At each level k there are $n!/(n-k)!$ nodes (Pinedo, 2008). The proposed branching scheme is clearly simple and other schemes have been proposed in the literature, demonstrating better performance in addressing the deterministic version of the scheduling problem (Carlier, 1982; Grabowski et al., 1986; Liu , 2010; Chandra et al., 2014), Nevertheless, as shown in the following paragraph, their application to the stochastic version of the problem is not straightforward and, moreover, dominated in terms of computational time due to the need of recomputing the cumulative distribution function of the schedule at each node. On the contrary, the proposed simple branching scheme has the advantage of reducing the number of convolution operations (see Section 4.2) and, consequently, the time needed to evaluate a single node in the tree.

4.2 Nodes evaluation

Let us consider two subsequent jobs $i, j \in A$ with stochastic processing times p_i and p_j modeled by their cumulative distribution functions $F_i(t) = Prob(p_i <= t)$ and $F_j(t) = Prob(p_j <= t)$ and the associated probability density functions $f_i(t) = Prob(p_i = t)$ and $f_j(t) = Prob(p_j = t)$.

The sum of the processing times of the two jobs is a stochastic variable and its cumulative distribution function $F_{i+j}(t)$ is the convolution of $F_i(t)$ and $F_j(t)$ (Agrawal and Elmaghraby, 2001).

$$
\begin{aligned}
F_{i+j}(t) &= F_i(t) * F_j(t) \\
&= \int_0^t F_i(t-s) dF_j(s) \\
&= \int_0^t F_i(t-s) f_j(s) ds
\end{aligned} \tag{5}
$$

Provided that the execution of the two activities starts at time 0, the cumulative distribution functions of the completion times of jobs i and j (c_i and c_j)

can be defined as:

$$F_{c_i}(t) = F_i(t) \tag{6}$$
$$F_{c_j}(t) = F_{c_i}(t) * F_j(t) = F_{i+j}(t) = F_i(t) * F_j(t) \tag{7}$$

The release time for job j can be modeled as an additional job k with processing time r_j to be executed before j. Job k has no resource request, it is not interfering with the jobs competing for the machine but just enforcing a temporal constraint. Hence, job j can be executed only after both job k and i have been completed. Provided that job j is started as soon as possible, the cdf for its start time (s_j) and completion time (c_j) can be calculated as:

$$F_{s_j}(t) = F_{c_i}(t) \cdot F_{r_j}(t) \tag{8}$$

$$F_{c_j}(t) = F_{s_j}(t) * F_j(t) \tag{9}$$

Given the cdf of the completion time of j and its due date d_j, the cdf of the lateness L_j can be calculated as:

$$F_{L_j}(t) = F_{c_j}(t + d_j) \tag{10}$$

Provided the cdf of the lateness for all the considered jobs, the cdf of the maximum lateness is:

$$F_{L_{max}}(t) = \prod_{j \in A} F_{L_j}(t) \tag{11}$$

and, hence, all the previous described risk measures can be calculated.

Grounding on this formulation, it is possible to calculate the distribution of the maximum lateness for those nodes where the schedule is completely defined, i.e., the leaves of the branching tree. On the contrary, for the other nodes, only a partial schedule is defined, containing a subset of the jobs ($S \in A$). For the already scheduled jobs, the maximum lateness cdf can be calculated according to the equations described above. For the unscheduled jobs ($A \setminus S \in A$), the contribution to the objective function cannot be calculated. The aim of the following steps is to define a way to estimate an upper bound and a lower bound for this contribution. Given a job in the set of not scheduled jobs, $j \in A \setminus S$, a lower bound for its lateness can be obtained assuming it starts immediately after the already scheduled jobs (A^S) or, if more constraining, after its release time r_j (Figure 1). Starting from the cdf of the completion time of the already scheduled jobs $F_{c_S}(t)$ and the cdf of the release time $F_{r_j}(t)$, the cdf of the earliest start time and completion time for j are:

$$F^{LB}_{s_j}(t) = F_{c_S}(t) \cdot F_{r_j}(t) \tag{12}$$

$$F^{LB}_{c_j}(t) = F_{s_j}(t) * F_j(t) \tag{13}$$

A lower bound for the cdf of the lateness L_j can be calculate accordingly:

$$F^{LB}_{L_j}(t) = F^{LB}_{c_j}(t + d_j) \tag{14}$$

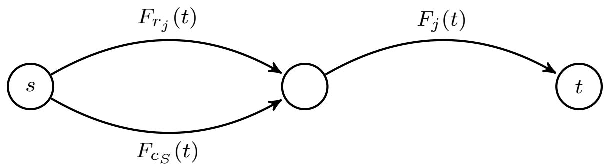

Fig. 1 Scheduling scheme for the lower bound completion time of job $j \in A \setminus S$.

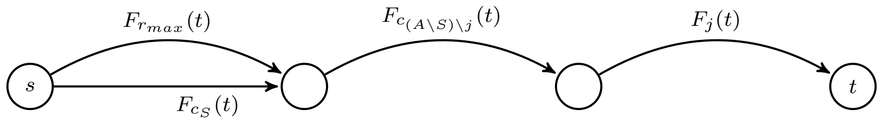

Fig. 2 Scheduling scheme for the upper bound completion time of job $j \in A \setminus S$.

while the lower bound for the maximum lateness is:

$$F^{LB}_{L_{max}}(t) = \prod_{j \in S} F_{L_j}(t) \prod_{j \in A \setminus S} F^{LB}_{L_j}(t) \tag{15}$$

An upper bound for the lateness L_j of the not scheduled job $j \in A \setminus S$ can be obtained assuming that it will be sequenced as the last job in the schedule. If we consider the not yet scheduled jobs but j and ignore their release times, we can calculate the cdf of the sum of their processing times $F_{(A \setminus S) \setminus j}(t)$ as the convolution of all the cdfs $F_k(t)$ with $k \in A \setminus S$ and $k \neq j$. Since the sequence of the jobs in $A \setminus S$ is undefined, the contribution of the release times cannot be calculated. However, a worst case for a job j can be defined, according to the scheduling scheme in Figure 2, considering the distribution of the maximum release time among the jobs to schedule:

$$F_{r_{max}}(t) = \prod_{k \in A \setminus S} F_{r_k}(t) \tag{16}$$

and then assuming that the batch of jobs to be scheduled different from j are executed after this release time. Hence, the upper bound cdf of the completion time for all the jobs but j is:

$$F^{UB}_{c_{A \setminus j}}(t) = (F_{c_S}(t) \cdot F_{r_{max}}(t)) * F_{(A \setminus S) \setminus j}(t) \tag{17}$$

While the upper bound cdfs of the completion time of job j is:

$$F^{UB}_{c_j}(t) = F^{UB}_{c_{A \setminus j}}(t) * F_j(t) \tag{18}$$

An upper bound for the cdf of the lateness L_j can be calculate as:

$$F^{UB}_{L_j}(t) = F^{UB}_{c_j}(t + d_j) \tag{19}$$

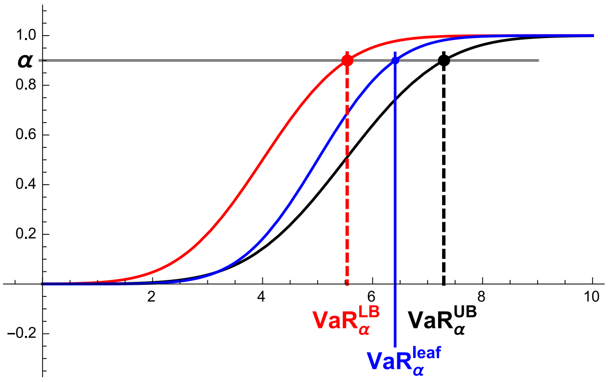

Fig. 3 Calculation of lower and upper bounds for the VaR at each node.

and the upper bound for the maximum lateness is:

$$F^{UB}_{L_{max}}(t) = \prod_{i \in S} F_{L_i}(t) \cdot \prod_{j \in A \setminus S} F^{UB}_{L_j}(t) \tag{20}$$

4.3 Dominance rules

The aim of the proposed approach is to minimise maximum lateness, a *regular* performance measure. In scheduling, such performance measures are functions non-decreasing in the completion times of the jobs c_i and, consequently, also non-decreasing in their processing times p_i (Pinedo, 2008, chap.2).

At each node in the branching tree, the lower bound cdf represents a schedule where unscheduled jobs are executed immediately after the already scheduled ones. For all the successors of this node, the effective starting time of an unscheduled job j would never be earlier than this, thus, the completion time of an unscheduled job can only increase or remain the same compared to its parent nodes.

Due to the regularity of the considered objective function, for a given sampling of the processing times and release dates, the cdf of a node is greater or equal to the cdf of any successor nodes for any value of the objective function. This is also the definition for the first-order stochastic dominance and, consequently, the lower bound cdf effectively provides a lower bound for the VaR of the maximum lateness respect to all the successor nodes. A dual reasoning can be done considering the upper bound cdfs, leading to the fact that the upper bound cdf in a node is stochastically dominated by all the upper bound cdfs of its successor nodes and the cdf in a leaf of the tree stochastically dominates all the upper bound cdfs of its parent nodes.

In the end, as shown in Figure 3 the cdf of a leaf of the tree (solid) always lies in the region bounded by the lower bound (dotted) and upper bound (dashed) cdfs of any of its parent nodes. For these reasons the lower and upper bound cdfs can be used to calculate the bounds of the VaR, providing a comparison criteria among the nodes of the search tree.

4.4 Application example

A description of the branch-and-bound approach is provided through an example. Let us consider a set of four activities to be scheduled as described in Table 1. All the processing time are modeled through discrete triangular distributions while the release dates are modeled through discrete uniform distributions. The due dates are deterministic. The aim of the approach is to find the schedule that minimizes the VaR_α of the maximum lateness L_{max} with $\alpha = 0.95$. As illustrated in Figure 4, the algorithm starts from the root node r (with no scheduled job) with four different branches, associated to the each of the four different jobs to be scheduled, i.e., $\{1, 2, 3, 4\}$. Four different nodes, $\{a, b, c, d\}$, in the branching tree are created and evaluated. The evaluation of a node follows the approach described in Section 4.2 to provide a lower and upper bound cumulative distributions for the given schedule. As illustrated in Section 4.3, these distributions also provide a lower and upper bound for the VaR_α as shown in Figure 4 where, for each node, a graph of the two bound distributions defines the region where the distributions of the solutions departing from it will lay and, consequently also the bounds for the $VaR_{0.95}$. Given the evaluation of the nodes in the first level of the branching tree, the branch-and-bound algorithm chooses the most promising node (node c) and proceed to the evaluation of the branches departing from it, i.e., the schedules having job 3 as the first in the sequence and then one of the three remaining jobs as the second. Notice that, after the evaluation of the nodes in the first level of the tree, some branches can be pruned. Specifically the lower bounds for the nodes b and d ($VaR_{0.95}^{LB_b} = 902$ and $VaR_{0.95}^{LB_d} = 965$) are higher than the upper bound of the best node selected ($VaR_{0.95}^{UB_c} = 822$), hence, both node b and d are pruned (single tick on the arc in Figure 4). On the contrary, the lower bound obtained in node a is lower than the upper bound in node c and, consequently, the branches departing from node a remain in the list of the ones to be possibly explored. The same is done for the nodes in the second level of the tree, e, f and g resulting in node e being the most promising option to be further explored ($VaR_{0.95}^{LB_e} = 105$ and $VaR_{0.95}^{UB_e} = 409$). The graph associated to node e in Figure 4 shows the new bounding distributions as well as the ones of the parent node c. As expected, as the schedule is defined, the accuracy of the estimation increases and the distance between the bounding distributions decreases. Also in this case, the new bounds allow the pruning of some nodes. The lower bound of node a ($VaR_{0.95}^{LB_a} = 492$) is now higher than $VaR_{0.95}^{UB_e}$ causing it to be pruned (double tick on the arc in Figure 4). Also the lower bounds of nodes f and g are higher than $VaR_{0.95}^{UB_e}$ and both f and g are pruned as well. The algorithm proceeds following the branches departing from the only remaining node e considering the scheduling of alternatively job 2 or 4 as the third one in the sequence. Notice that, once the third job in the sequence

Table 1 Exemplary scheduling instance.

Job	Process Time	Release Date	Due Date
0	∼ Triangular(63,108,118)	∼ Uniform(414,514)	513
1	∼ Triangular(105,161,173)	∼ Uniform(780,870)	827
2	∼ Triangular(50,90,99)	∼ Uniform(120,224)	201
3	∼ Triangular(105,160,171)	∼ Uniform(818,938)	920

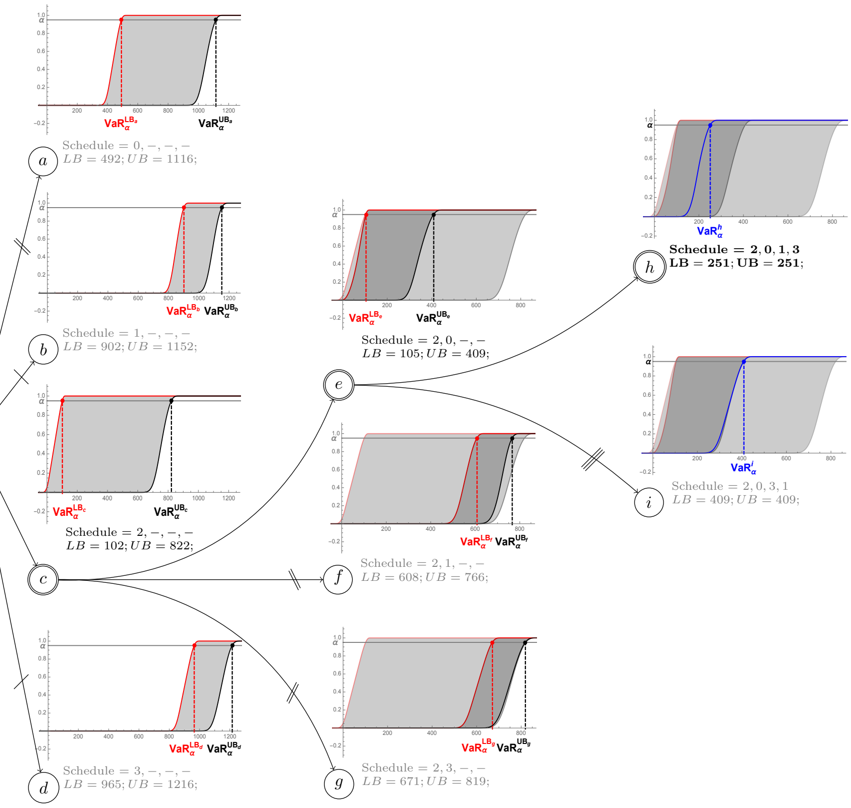

Fig. 4 Exemplary application of the branch-and-bound approach.

is decided, the whole sequence is also determined. Hence, nodes h and i are leaves of the tree. The branch-and-bound approach evaluates them and selects node h as the optimal solution with schedule $\{3, 1, 2, 4\}$ and $VaR^h_{0.95} = 251$. The application of the approaches results in the evaluation of only 9 nodes in the tree (2 of them are complete schedules) while 5 are pruned during the exploration of the tree.

5 Testing

The branch-and-bound algorithm has been completely coded in `C++` using the `BoB++` library (Bob++, 2012; Djerrah et al., 2006) and the Boost library (Boost, 2013). The computational experiments have been executed on 16 parallel threads on a workstation equipped with an Intel Eight-Core Xeon Processor E5-2650v2 running at 2.6 GHz and 64 GB of RAM.

To test the proposed algorithm, two aspects have been taken into consideration. The first one concerns the performance of the algorithm in terms of time needed to solve a given instance to optimality. In addition, it is also relevant to evaluate the performance of the algorithm compared to other existing approaches. Usually, this comparison is done considering two algorithms aiming at the same objective function but we adopted a different approach because the distribution of the objective function introduces a significant complexity in the problem. Even though the new approach may require more solution time, it has some benefits that can be demonstrated in comparison with a more simple method. We use as a comparison the solution provided by the *Earliest Due Date (EDD)* rule, a simple rule that is non-optimal but can be applied in a really fast way.

To assess the performance of the algorithm on a wide set of scheduling problems, the test instances have been generated by varying the number of jobs and their characteristics, i.e., processing times, release dates and due dates. The processing times of the jobs follow a discrete triangular distribution and have been generated defining the mean value, the coefficient of variation and the skewness. The release times follow a discrete uniform distribution and have been generated defining the mean value and width (or half-width). The due dates are deterministic and have been generated considering their *strictness*.

To generate the test instances the following procedure has been used:

1. define the number of jobs (10 and 20).
2. for each job j, the average process time $\bar{p}_i$ is sampled from a discrete uniform distribution between 50 and 150 for 50% of the instances and between 25 and 75 for the remaining ones;
3. for each job j, the coefficient of variation of the processing time is sampled for 50% of the instances from a discrete uniform distribution between 0.4 and 0.6 and from a discrete uniform distribution between 1.4 and 1.6 for the remaining ones;
4. for each job j, the skewness of the processing time is randomly assigned the value -0.5, 0 or 0.5;
5. for each job j, the average release time $\bar{r}_j$ is sampled from a discrete uniform distribution between 1 and 10 times the average processing time;
6. for each job j, the half-width of the release time is sampled for 50% of the instances from a discrete uniform distribution between 40 and 60 and from a discrete uniform distribution between 120 and 160 for the remaining ones;
7. for each job j, the deterministic value of the due date d_j is sampled for 50% of the instances from a discrete uniform distribution between 0 and

Table 2 Results.

Number of jobs	Risk level		Solution Time [s]	Visited Nodes	% Visited Nodes
10	1%	Mean	0.4388	1539	0.0247
		Min	0.0250	258	0.0041
		Max	1.8830	3377	0.0542
		StDev	0.3164	612	0.0094
	5%	Mean	0.4485	1557	0.0250
		Min	0.0300	201	0.0032
		Max	1.9850	4196	0.0669
		StDev	0.3334	612	0.0098
	25%	Mean	0.4619	1602	0.0257
		Min	0.0250	254	0.0041
		Max	2.2210	4239	0.0680
		StDev	0.3442	654	0.0105
20	1%	Mean	70.9	76203	0.000*
		Min	3.0	10326	0.000*
		Max	3489.5	5033503	0.000*
		StDev	171.9	236625	0.000*
	5%	Mean	75.3	80519	0.000*
		Min	5.1	16521	0.000*
		Max	3474.2	5009033	0.000*
		StDev	169.7	230398	0.000*
	25%	Mean	80.2	84080	0.000*
		Min	7.5	20192	0.000*
		Max	3246.5	4709357	0.000*
		StDev	161.0	210799	0.000*
	Total	Mean	38.00	40958	0.0125
		Min	0.03	201	0.000*
		Max	3489.48	5033503	0.0680
		StDev	124.30	164720	0.0144

50 and from a discrete uniform distribution between 150 and 200 for the remaining ones;

8. the generated instances are used to run the optimization algorithm with different risk levels α (1%, 5% and 25%).

In total, 620 instances have been generated for the different number of jobs and solved considering different risk levels, for a total of 3840 experiments.

5.1 Results

The results in Table 2 show the performance of the algorithm in terms of the time (in seconds) needed to find the optimal solution (Solution time). The table also reports the number and fraction of the nodes of the complete branching tree visited during the search. Both classes of results are detailed with respect to the risk measure used and the risk level considered, moreover, for each combination, the minimum, maximum, average values and the standard deviation are reported.

Further on, Table 2 shows that, considering the whole set of instances, the algorithm was able to find the optimal solution in an average time of 38 seconds, with a variability ranging from a minimum value of 0.03 seconds to a maximum value of 3489.48 seconds. Moreover, the average number of nodes visited during the search is about 0.013% of the total number of nodes in the branching tree (notice that the total number of nodes is equal to $\sum_{k=1}^{n-1} \frac{n!}{(n-k)!}$). Considering the results in relation to the different numbers of jobs in the test instances, we see that, in case of 10 jobs, the algorithm is able to find the optimal solution in an average time of 0.45 seconds, ranging from a minimum of 0.0250 to a maximum of 2.2210 seconds. To find the optimal schedule it was necessary to analyse an average of 1566 nodes of the whole branching tree containing 6235300, thus needing the analysis of 0.0251% of them. When considering the 20-job instances, the solution time predictably increases. On average 75.46 seconds are needed to solve an instance to optimality, ranging from a minimum of 3.02 to a maximum of 3489.48 seconds. The branch-and-bound algorithm explores on average 80267 nodes out of $4.18041 \cdot 10^{17}$, i.e., less than 10^{-10}% considered as 0.000 in Table 2. The number of nodes analysed to solve the 20-job instances ranges from a minimum of 10326 to a maximum of 5 033 503.

In addition, although the results seem to show a slight increase of the solution time as the considered risk level increases, there is no statistical evidence to state that the solution time is affected by the different risk levels.

Clearly, the solution time strictly depends on the number of nodes evaluated during the search. This is confirmed in Figure 5, showing a linear correlation between the two values. The deviation from this linear correlation can be easily explained considering that, given the same number of evaluated nodes, the time needed to evaluate a given node depends on the specific scheduling problem. In particular, the amplitude of the support of the distributions influences the time needed to perform the convolution, since the combinations of values to be computed is greater. Furthermore, the time needed to perform the convolution calculations also depends on the support of the distribution and, hence, on the length of the duration of the activities and, in fact, looking at the solution time for the different class of instances (with shorter and longer average duration of the jobs), it is clear that if the support of the distribution is larger, then the algorithm could require more to find the optimal solution. This hypothesis is supported in Figure 6, reporting the average solution time spent in each node in relation to the number of jobs to schedule. The graph clearly shows that the average solution time increases when the number of jobs is greater and proportionally to the average duration of the jobs. The results confirm also that convolution is the most time consuming component of the algorithm: handling schedules both of more and longer jobs requires calculations with cdfs having larger support. Finally, this is confirmed in Figure 7 showing that, as the average duration of the jobs decreases, the solution time is significantly smaller, thus allowing the proposed algorithm to be easily used on even larger instances.

A different set of considerations have been provided to compare the solution obtained with the branch-and-bound algorithm against a schedule obtained with the *Earliest Due Date (EDD)* rule. First the EDD rule is used and the associated VaR calculated. This value is compared with the VaR of the

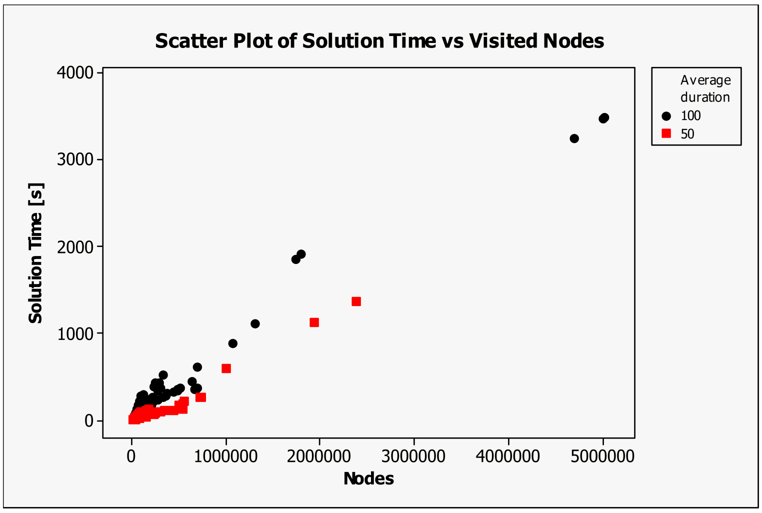

Fig. 5 Scatter plot of the solution time respect to the number of visited nodes.

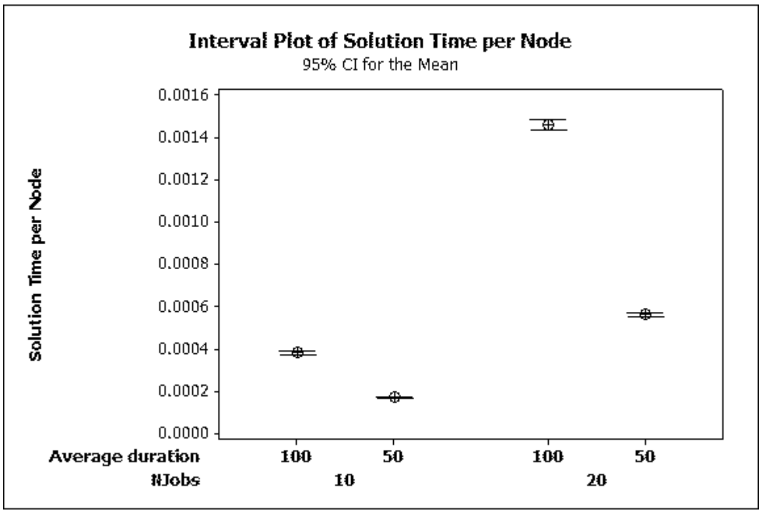

Fig. 6 Interval plot of the solution time [s] spent in each of the evaluated nodes.

optimal schedule coming from the branch-and-bound approach. The results are summarised in Table 3.

If we compare them in terms of the VaR, the proposed approach performs on average 2% better respect to a simple rule like the EDD, i.e., the schedule obtained with the EDD has a higher VaR for a given risk level. As an example, let us image to use the proposed approach to define an optimal schedule S_{opt} minimizing the $VaR_{5\%}$ of the maximum lateness and that $VaR_{5\%}^{S_{opt}} = 30$

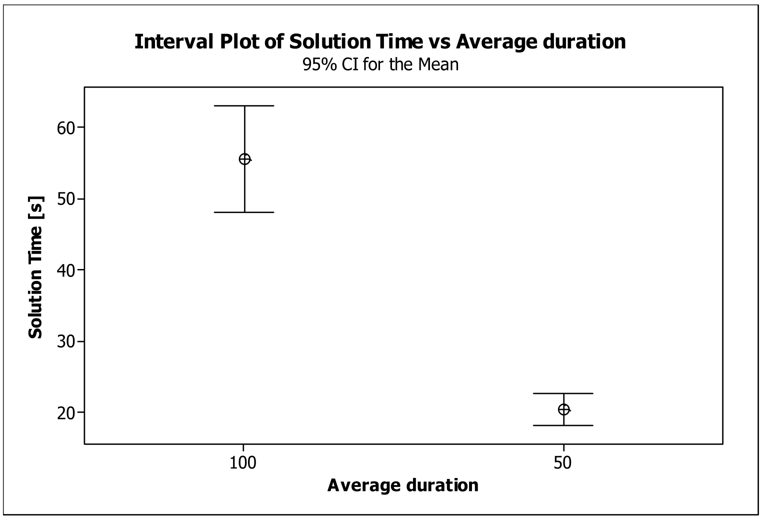

Fig. 7 Interval plot of the solution time [s] in relation to the dimension of the support of the distribution of the processing times.

Table 3 % difference in comparison with EDD rule.

Average duration	Number of jobs	Risk level	Mean	Min	Max	StDev
50	10	1%	3.39	0.000	100.00	8.41
		5%	5.18	0.000	300.00	22.56
		25%	5.66	0.000	228.57	18.84
	20	1%	2.15	0.000	24.03	24.03
		5%	2.28	0.000	26.09	26.09
		25%	2.50	0.000	30.85	30.85
100	10	1%	0.71	0.000	16.02	16.02
		5%	0.72	0.000	16.53	16.53
		25%	0.71	0.000	17.59	17.59
	20	1%	0.60	0.000	9.48	9.48
		5%	0.61	0.000	9.84	9.84
		25%	0.59	0.000	10.61	10.60
	Total		2.09	0.000	300.00	9.27

days. This means that S_{opt} assures that the probability of having a maximum lateness grater that 30 days is 5% and, since it is optimal, any other schedule would have a higher value of the $VaR_{5\%}$. In the 40.31% of the experiments, the branch-and-bound approach and the EDD rule provide the same value of the objective function (although not always the same schedule).

For the remaining 59.69% of the experiments, the EDD rule provides a different schedule S_{EDD} with $VaR_{5\%}^{S_{EDD}} \geq VaR_{5\%}^{S_{opt}}$ and, in the worst case, reaches a maximum value of 300%. If, as an example, $VaR_{5\%}^{S_{opt}} = 30$ days and the *% difference vs EDD* is equal to 10% then, using S_{EDD}, the $VaR_{5\%}^{S_{EDD}} = 33$ days, thus, the probability of having a maximum lateness greater than 30 days is more than 5%.

Fig. 8 Sintered carbide tools (courtesy of Ceratizit).

Notice that the difference is higher if the duration and number of jobs is smaller. Although a difference of 2% on average could be considered too small to justify the proposed approach, it must be considered that stochastic approaches based on risk measures exists to protect the addressed performance in the worst cases, rather than addressing the average ones. As shown in Table 3 the difference in the worst case reaches the value of 300%. These extreme cases alone provide the main justification to the adoption of the proposed scheduling approach.

6 Industrial Application

The viability of the proposed scheduling approach has also been tested in a real industrial environment producing tailor-made sintered carbide tools widely used in manufacturing today (Figure 8). Carbides are composite materials consisting of a hard material and a comparatively soft binder metal. They are used to work with hard materials (e.g., titanium) or to achieve high cutting speed. They are also applied to produce traditional tools or to act as tools in other processes (e.g., extrusion, drawing, etc.). A significant part of the production of sintered carbide tools is devoted to tailor-made tools produced in small lots or even as a single unit.

The production of carbide tools is a powder-metallurgical process whose principal ingredient is a mixture of powders (tungsten carbide, cobalt, nickel, iron). As shown in Figure 9, in the first step of the production process, the powder is pressed to obtain a near net shape sample. Then, the obtained parts are machined according to the desired final condition and sintered applying high temperature ($1300 - 1500°$C) and sometimes also significant pressure to have a homogeneous and dense carbide with a high level of hardness. During the sintering process the volume of the parts is significantly reduced (up to 50%). To obtain the desired final shape and the required finishing they must undergo a grinding process. Finally, they are assembled with the other components to form the tool.

In the industrial application, we take into consideration the production of tailor-made tools, produced in a specific shop area in the plant and, in particular, the grinding phase. The grinding process requires a setup of the machines to produce the specific tailor-made tool. Such tools are often produced for the first time, hence, the process needs to be adjusted to achieve the final characteristics of the products. If the analysis is restricted to a specific class of tools (drawing dies), the grinding area works on a single lot at a time (Figure 9).

To formalise the industrial application we deal with the scheduling of jobs at the grinding area as illustrated in the right column in Figure 9. The job to schedule represents a lot of a tailor-made tool to be produced. The lot arrives from the previous production step, i.e., the sintering phase. In order to model the possible deviations of the arrival of the lots at the grinding shop, we consider stochastic release times. As described before, the grinding shop processes one lot at time, hence, we can model its behaviour as a single machine. Moreover, the grinding process entails a setup of the machines and is often adjusted to achieve the desired specifications. For these reasons we model the processing time of the lots with a stochastic distribution. The finished lots must be delivered to the following production step according to the production plan. Aiming at reducing the propagation of local schedule disruption throughout the plant, we assign the lots a deterministic due date according to the production plan and aim at minimizing the VaR of the maximum lateness.

A set of 3 test instances have been defined on the basis of historical data from the plant. The analysis of the data showed that the different jobs at the grinding shop can be classified into 8 classes according to the characteristics of the tool to produce. For each of the classes, the difference between the standard duration and the actual duration of the processing time has been analysed to define the distribution for the different classes of products. Finally, a triangular distribution has been fitted on these data. The distributions of the release dates have been defined calculating the range of variability of the actual release dates compared to the planned ones, for each class of products, in the last six months of production. Each instance has been constructed considering a set of 30 jobs and using the real production plan to provide the expected release date, the due date and the class of the jobs. The latter is used to sample the distribution of the processing time. Each of the considered instances provides the set of jobs that are going to be executed in about 2 weeks.

The results are reported in Table 4 demonstrating that the algorithm is able to provide an estimation of the VaR that is more accurate than the one provided by the EDD rule of 6.63% on average, ranging from a minimum of 1.79% to a maximum of 11.69%. Moreover, it was able to find the optimal solution analysing on average 21000 nodes out of a total of $4.55779 \cdot 10^{32}$. Also in this case the fraction of evaluated nodes is close to 0. Notice that, even if the

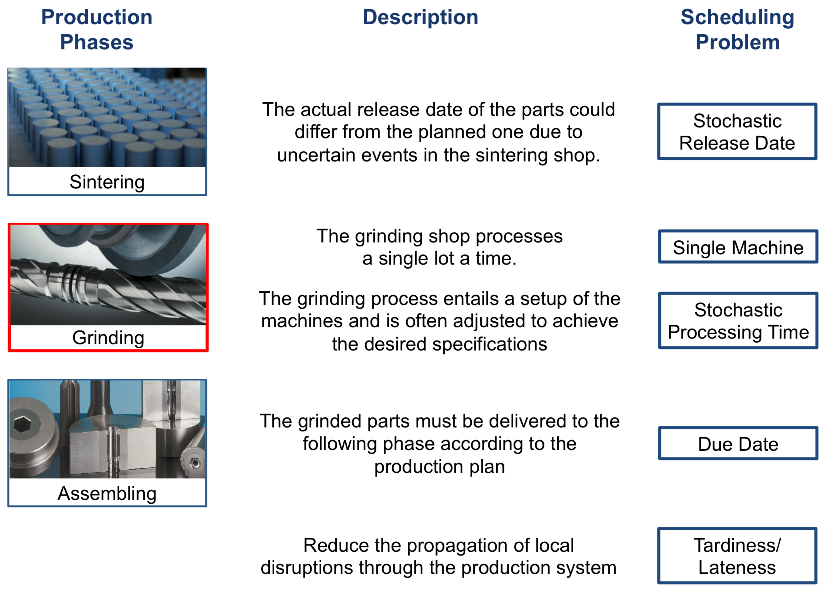

Fig. 9 Formalization of the industrial case and of the associated scheduling problem.

Table 4 Results of the application to the industrial case.

Number of jobs	Risk level		Visited Nodes	% Visited Nodes	% difference vs EDD
30	1%	Mean	21568	0.000*	6.60
		Min	10213	0.000*	1.79
		Max	34399	0.000*	11.25
		StDev	12160	0.000*	4.73
	5%	Mean	21591	0.000*	6.64
		Min	9876	0.000*	2.27
		Max	35523	0.000*	11.66
		StDev	12966	0.000*	4.60
	25%	Mean	20529	0.000*	6.63
		Min	10629	0.000*	1.79
		Max	33072	0.000*	11.69
		StDev	11453	0.000*	4.12

number of jobs to schedule was 30, the solution time was less than 1 second for all the instances, due to the fact that, since the average duration of the jobs is about 5 hours, the support of the associated distribution is significantly smaller than the one used in the previous experiment (on average 100).
Hence, since the time needed for the convolution operations is smaller, the algorithm is able to schedule 30 jobs in a very short time. This provides a way of using the algorithm also for large instances, at the price of reducing the resolution used to estimate the processing times and the release dates and, consequently, reducing the accuracy of the cdf's estimation.

7 Conclusions

A branch-and-bound approach for stochastic scheduling to minimize a stochastic function of the maximum lateness has been described and demonstrated in this paper. The proposed approach is targeted to a single machine scheduling problem with deterministic discrete due dates and generally distributed discrete processing times and release times.

Since the aim is to guarantee a robust schedule capable of providing protection against the occurrence of low probability but extremely unfavourable events, a measure of risk is used in the stochastic objective function; in particular the *value-at-risk* has been considered.

The performance of the proposed branch-and-bound approach is reasonably fast in terms of the time needed to find the optimal solution. Clearly the dimension of the solved instances (up to 20 jobs) is not large and, as the number of jobs increases, the solution time will increase as well, certainly more than linearly. However, the parallel capabilities of the implementation allow to easily exploit the benefits of new multi core architectures or the execution on high performance calculation environments (16 parallel threads on an Intel Eight-Core Xeon Processor E5-2650v2).

Notice that the completion time, and hence also the lateness, is a sum of stochastic variables and, due to the central limit theorem, as the number of jobs increases, the distribution of the sum of their processing times converges to a normal distribution and the proposed exact calculation using multiple convolution steps can be approximated with the normal distribution hypotheses. Due to this specific property, the proposed exact solution approach finds its most favourable application with a "small" number of activities, i.e., not much greater than 20.

Moreover, the solution time also depends on the dimension of the support of the considered distributions and, consequently, on the number of points where it assumes a value. For this reason, using a rough estimation of the distributions with a small number of points (shifts instead of hours) could be used to deal with a larger number of jobs.

Clearly the adoption of more powerful but complex approaches must find a justification in the potentially achievable benefits. To this aim, an average benefit of about 2% with respect to the adoption of a simple dispatching rule like the EDD could be considered low. However, as it always happens when assessing the benefits of stochastic approaches, their primary goal is to protect the schedule in the worst cases rather than in the average ones. More specifically, a risk-based stochastic approach mainly focuses on the capability of distinguishing the shape of different distributions, thus being able to assess the effects of events unlikely to occur but with a high impact on the targeted performance. From this perspective, although an average difference of 2% respect to the EDD rule seems not so relevant, a maximum difference of about 300% really matters.

As typical in lateness-related objective functions, the impact of a greater lateness is strictly related to the type of contract between the customer and the supplier. Depending on the kind of penalties agreed, even a small deviation from a negotiated maximum lateness could have a high impact.

In conclusion, prime candidates for this novel method are scheduling problems with relatively small number of jobs whose execution can be extremely unfavourably affected by stochastic, low probability events. Further research will address the application to different scheduling problems as well as its exploitation in the negotiation of due dates.

Acknowledgements

This research has been developed in the framework of the XVII Executive Programme of Scientific and Technological Cooperation between the Republic of Hungary and the Republic of Italy and GINOP-2.3.2-15-2016-00002 grant.

References

Agrawal, M. K., Elmaghraby, S. E., 2001. On computing the distribution function of the sum of independent random variables. Computers & Operations Research 28 (5), 473–483.

Alfieri, A., Tolio, T., Urgo, M., 2011. A project scheduling approach to production planning with feeding precedence relations. International Journal of Production Research 49 (4), 995–1020.

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

Artigues, C., Leus, R., Talla Nobibon, F., 2013 Robust optimization for resource-constrained project scheduling with uncertain activity durations. Flexible Services and Manufacturing Journal 25 (1), 175–205.

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., 1999. Coherent measures of risk. Mathematical Finance 9 (3), 203–228.

Atakan, S., Bülbül, K., Noyan, N., 2016. Minimizing value-at-risk in single-machine scheduling. Annals of Operations Research 248, 25–73.

Attia, E.-A., Duquenne, P., Le-Lann, J.-M., 2014. Considering skills evolutions in multi-skilled workforce allocation with flexible working hours. International Journal of Production Research 52 (15), 4548–4573.

Baker, K.R., Su, Z.S., 1974. Sequencing with due dates and early start times to minimize tardiness. Naval Research Logistics Quarterly 21, 171–176.

Baker, K.R., Trietsch, D., 2009. Safe scheduling: Setting due dates in single-machine problems. European Journal of Operational Research 196 (1), 69–77.

Baker, K.R., Trietsch, D., 2014. Safe Scheduling. In INFORMS Tutorials in Operations Research, 79-101.

Benmansour, R., Allaoui, H., Artiba, A., 2012. Stochastic single machine scheduling with random common due date. International Journal of Production Research 50 (13), 3560–3571.

Bob++, 2012. A framework to solve combinatorial optimization problems. http://www.prism.uvsq.fr/ blec/bobpp/

Boost, 2013. Boost C++ libraries. http://www.boost.org.

Burdett, R.L., Kozan, E., Strickland, C., 2012. A practical approach for identifying expected solution performance and robustness in Operation Research applications. ASOR Bulletin 31(2), 1–28.

Burdett, R.L., Kozan, E., 2014. Performance profiling for predictive train schedules. Journal of Rail Transport Planning and Management 4, 98-114.

Cai, X., Wang, L., Zhou, X., Oct. 2007. Single-machine scheduling to stochastically minimize maximum lateness. Journal of Scheduling 10 (4), 293–301.

Cai, X., Zhou, X., 2005. Single-machine scheduling with exponential processing times and general stochastic cost functions. Journal of Global Optimization 31 (2), 317–332.

Cao, Q., Patterson, J.W., Griffin, T.E., 2001. On the operational definition of processing time uncertainty. International Journal of Production Research 39 (13), 2833–2849.

Carlier, J., 1982. The one-machine sequencing problem. European Journal of Operational Research 11, 42–47.

Chandra, C., Liu, Z., He, J., Ruohonen, T., 2014. A binary branch and bound algorithm to minimize maximum scheduling cost. Omega 42, 9–15.

Chang, C.-S., Yao, D. D., 1993. Rearrangement, majorization and stochastic scheduling. Mathematics of Operations Research 18 (3), 658–684.

De, P., Ghosh, J. B., Wells, C. E., 1992. Expectation-variance analysis of job sequences under processing time uncertainty. International Journal of Production Economics 28 (3), 289–297.

Djerrah, A., Le Cun, B., Cung, V.-D., Roucairol, C., 2006. Bob++: Framework for solving optimization problems with branch-and-bound methods. In: High Performance Distributed Computing, 2006 15th IEEE International Symposium on., pp. 369–370.

Fang, C., Kolisch, R., Wang, L., Mu, C., 2015. An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem. Flexible Services and Manufacturing Journal 27 (4), 585–605.

Gafarov, E.R., Dolgui, A., Werner, F., 2014. A new graphical approach for solving single-machine scheduling problems approximately. International Journal of Production Research 52 (13), 3762–3777.

Grabowski, J., Nowicki, E., Zdrzałka, S., 1986. A block approach for single-machine scheduling with release dates and due dates. European Journal of Operational Research 26, 278–285.

Kellerer, H., 2004. Minimizing the maximum lateness. In: Handbook of Scheduling: Algorithms, Models and Performance Analysis. Chapman & Hall/CRC Computer & Information Science Series. Chapman and Hall/CRC.

Lenstra, J., Kan, A. R., Brucker, P., 1977. Complexity of machine scheduling problems. Annals of Discrete Mathematics 1, 343–362.

Liu, Z. 2010 Single machine scheduling to minimize maximum lateness subject to release dates and precedence constraints. Computers & Operations Research 37, 1537?1543.

Ma, C., Wong, W.K., 2010. Stochastic dominance and risk measure: A decision-theoretic foundation for VaR and C-VaR. European Journal of Operational Research 207 (2), 927–935.

Makris, S., Chryssolouris, G., 2010. Customer's behaviour modelling for manufacturing planning. International Journal of Computer Integrated Manufacturing 23 (7), 619–629.

Mogre, R., Wong, C.Y., Lalwani, C.S., 2014. Mitigating supply and production uncertainties with dynamic scheduling using real-time transport information. International Journal of Production Research 52 (17), 5223–5235.

Mourtzis, D., Doukas, M., Psarommatis, F., 2012. Design and planning of decentralised production networks under high product variety demand. Procedia CIRP, 3 (1), 293–298.

Nonaka, Y., Erdos, G., Kis, T., Nakano, T., Váncza, J., 2012. Scheduling with alternative routings in CNC workshops. CIRP Annals - Manufacturing Technology 61 (1), 449–454.

Pinedo, M. L., 2008. Scheduling - Theory, Algorithms, and Systems. Springer New York.

Radke, A. M., Tolio, T., Tseng, M. M., Urgo, M., 2013. A risk management-based evaluation of inventory allocations for make-to-order production. CIRP Annals - Manufacturing Technology 62 (1), 459–462.

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

Ross, S. M., 1983. Stochastic Processes, 2nd Edition. Holden Day.

Sarin, S., Nagarajan, B., Jain, S., Liao, L., May 2009. Analytic evaluation of the expectation and variance of different performance measures of a schedule on a single machine under processing time variability. Journal of Combinatorial Optimization 17 (4), 400–416.

Scholz-Reiter, B., Hildebrandt, T., Tan, Y., 2013. Effective and efficient scheduling of dynamic job shops - Combining the shifting bottleneck procedure with variable neighbourhood search. CIRP Annals - Manufacturing Technology 62 (1), 423–426.

Shaked, M., Shanthikumar, J. G., 2007. Stochastic Orders. Springer Series in Statistics. Springer New York.

Tolio,T., Urgo, M., 2007. A rolling horizon approach to plan outsourcing in manufacturing-to-order environments affected by uncertainty. CIRP Annals - Manufacturing Technology 56 (1), 487–490.

Tolio, T., Urgo, M., 2013. Design of flexible transfer lines: A case-based reconfiguration cost assessment. Journal of Manufacturing Systems 32 (2), 325–334.

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

Yin, Y., Wu, W-H., Cheng, T.C.E., Wu, C-C., 2014. Due-date assignment and single-machine scheduling with generalised position-dependent deteriorating jobs and deteriorating multi-maintenance activities. International Journal of Production Research 52 (8), 2311–2326.

Wu, X., Zhou, X., 2008. Stochastic scheduling to minimize expected maximum lateness. European Journal of Operational Research 190 (1), 103 – 115.

Zdrzalka, S., Grabowski, J., 1989. An algorithm for single machine sequencing with release dates to minimize maximum cost. Discrete Applied Mathematics 23, 73-89.

Zhou, X., Cai, X., 1997. General stochastic single-machine scheduling with regular cost functions. Mathematical and Computer Modelling 26 (3), 95–108.