Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia CIRP 19 (2014) 148 – 153
www.elsevier.com/locate/procedia

Robust Manufacturing Conference (RoMaC 2014)

A robust scheduling approach for a single machine to optimize a risk measure

Marcello Urgo[a,*], József Váncza[b,c]

[a] *Manufacturing and Production Systems Laboratory, Mechanical Engineering Department, Politecnico di Milano, Milano, Italy*
[b] *Fraunhofer Project Center for Production Management and Informatics, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary*
[c] *Department of Manufacturing Science and Technology, Budapest University of Technology and Economics, Budapest, Hungary*

[*] Corresponding author. Tel.: +39-02-2399-8521; fax: +39-02-2399-8585. *E-mail address:* marcello.urgo@polimi.it

Abstract

Robustness in scheduling addresses the capability of devising schedules which are not sensitive – to a certain extent – to the disruptive effects of unexpected events. The paper presents a novel approach for protecting the quality of a schedule by taking into account the rare occurrence of very unfavourable events causing heavy losses. This calls for assessing the risk associated to the different scheduling decisions. In this paper we consider 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. We present a branch-and-bound approach to minimize the Value-at-Risk of the distribution of the maximum lateness and demonstrate the viability of the approach through a series of computational experiments.

Selection and peer-review under responsibility of the International Scientific Committee of "RoMaC 2014" in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.

Keywords: Robust Scheduling, Risk, Manufacturing-to-Order Production

1. Introduction and Problem Statement

In real production environments, scheduling approaches have to deal with the occurrence of unexpected events that may stem from a wide range of sources, both internal and external. Production activities may require more time or resources than originally estimated, resources may undergo failures, materials may be unavailable at the scheduled time, release and due dates may change and new activities like rush orders or reworks could be inserted in the schedule. Robust scheduling approaches aim at protecting the performance of the schedule by anticipating to a certain degree the occurrence of uncertain events and, thus, avoiding or mitigating the costs due to missed due dates and deadlines, resource idleness, higher work-in-process inventory.

The vast majority of the stochastic scheduling literature considers the stochastic aspect of a problem in terms of a scalar performance indicator, e.g., the expected value. When addressing a scheduling problem, the capability of minimizing the expected value of an objective function provides a significant improvement respect to pure deterministic approaches. However, the expected value is not suitable to exhaustively model the quality of the schedule from the stochastic point of view [1,2].

As an example, minimizing the expected value of the maximum lateness aims at assuring an average good performance in terms of due date meeting but does not protect against the worst cases if their probability is low. Protection against worst cases is a natural tendency in management decisions. Plant managers who face uncertainty try to maximize the mean profit but also try to avoid the rare occurrence of very unfavourable situations causing heavy losses. To cope with this problem, the financial literature has proposed risk measures able to consider the impact of uncertain events both in terms of their effect and of their occurrence probability [3,4]. In the scheduling area, on the contrary, risk analysis and assessment are not so popular even if the concept of risk is often perfectly suitable to support scheduling decisions under uncertainty. Against its potential utility, the application of risk measures to scheduling problems has not be extensively addressed due to the difficulty in considering the objective function in terms of its stochastic distribution instead of a scalar performance indicator (i.e. expected value, variance) [5].

In this paper we consider a stochastic scheduling problem with a set of n jobs that must be sequenced on a single machine. This can model a single machine as well as a group

Selection and peer-review under responsibility of the International Scientific Committee of "RoMaC 2014" in the person of the Conference Chair Prof. Dr.-Ing. Katja Windt.
doi:10.1016/j.procir.2014.05.013

of resources, or a whole department. Although it could seem a restrictive hypothesis, a single resource model is applicable to several cases where a group of resources can only work on a single product or a single product type at a time (e.g., multi-model transfer lines, make-to-order shops working on a single job or batch at a time). The aim is at optimizing a risk measure of the maximum lateness using a branch-and-bound algorithm. The processing times p_j of the jobs are generally distributed independent random variables. The jobs are available after a release date r_j and have a due date d_j. The release dates r_j are also generally distributed independent stochastic variables while the due dates d_j are deterministic. The objective of the scheduling problem is to optimize a stochastic function of a given performance measure. In particular we focus on the *maximum lateness* $L_{max} = max\{L_j, j = 1, \cdots, n\}$, with $L_j = C_j - d_j, j = 1, \cdots, n$ where C_j is the completion time of job j under the given schedule. This objective function is likely to minimize a stochastic function of the maximum magnitude of the deviations with respect to the due dates, thus protecting the schedule from the impact of the worst cases.

In Section 2 the present advances for the existing stochastic scheduling approach are summarized. Section 3 reports an outline of the risk measure used, the *Value-at-Risk* (VaR). Section 4 describes the principles and operation of the proposed branch-and-bound solution method.. Section 5 reports on the computational test result, while Section 6 concludes the paper.

2. State of the Art

The deterministic version of the considered stochastic problem is known as $1|r_i|L_{max}$ and has been recognized to be strongly NP-hard [6]. A review of the existing solution approach for this scheduling problem can be found in [7] and [8, chap.9]. If we do not consider the release times, the resulting scheduling problem ($1|L_{max}$) is rather simple and can be solved to optimality using the *earliest due date* (EDD) rule.

Referring to the stochastic counterpart, when considering a single machine scheduling problem with arbitrarily distributed processing times and deterministic due dates, the EDD rule still minimizes the expected maximum lateness [8]. This applies to non-preemptive static list and dynamic policies, as well as to preemptive dynamic policies. These results ground on the fact that the EDD rule minimizes the maximum lateness of the deterministic version of the problem. Hence, given any realization of the processing times, the EDD rule provides the optimal schedule and, since this happens for all the realizations, then the EDD rule minimizes the maximum lateness also in expectation [8].

This result has further implications on the maximum lateness distribution. Since the EDD schedule provides the optimal maximum lateness for any realization of the processing times, given a maximum lateness L^* and a schedule S^*, the probability of having $L_{max} \leq L^*$ must be less or equal to the value obtained with the EDD schedule. Due to this, 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 [9,10, chap.9].

The relationships between rearrangement inequalities and scheduling problems have been addressed in [11]. Using stochastic rearrangement inequalities, the author obtains a solution for the stochastic counterpart of many classical deterministic scheduling problems. These results have been rephrased and further exploited in [12–15].

It must be noticed that part of the stochastic scheduling literature addresses the problem of minimizing the maximum expected lateness $max(E[L])$. In this problem, using a stochastic function $E[L]$, the stochastic problem is reduced to a deterministic minimization [12]. On the contrary, considering the minimization of the expected value of the maximum lateness $E[L_{max}]$ retains the stochastic characteristics of the scheduling problem by regarding the whole distribution of the objective function.

A stochastic problem belonging to this class is analyzed in [15] where a set of jobs with deterministic process times and stochastic due dates are scheduled on a single machine to minimize the expected value of the maximum lateness ($E[L_{max}]$). The authors propose a dynamic programming algorithm and compare its performance to three different heuristic rules. The dynamic programming algorithm is also extended to cope with stochastic processing times and due dates. However, the provided results ground on the assumption that both the processing times and due dates are exponentially distributed.

Analogously to the deterministic case, when the release times are considered (either deterministic or stochastic), the problem becomes more difficult to solve. However, considering independent generally distributed release times and independent generally distributed processing times, if the due dates are deterministic, the EDD rule still minimizes L_{max} but only in the preemptive case [8]. Some further extensions are available but only assuming that the due dates are deterministic but both the release and processing times are exponentially distributed with the same mean [8].

Referring to the use of stochastic objective function other than the expected value, the most common is the variance. In fact, a trade-off between mean and variance is one of the most simple and common risk measure. A joint optimization of expectation and variance in a single machine scheduling problem has been proposed in [16]. Other common objective functions in the stochastic scheduling are the flow time and the completion time. Moreover, in a recent paper [17] provides closed form equations of mean and variance for a large set of scheduling problems. However, no algorithm, neither exact, nor heuristic, has been proposed for the maximum lateness single machine scheduling problem to optimize a stochastic objective function different from the expected value.

3. Risk Measures

Financial research has paid particular attention to the definition of risk measures to cope with uncertainty. In particular the study of extreme events, i.e., the tails of the distribution has received due attention. Risk measures as the *Value-at-Risk* are extensively used in portfolio management and a large amount of literature have been written on their mathematical properties and effectiveness in protecting assets investments.

According to the notation used in [18], we consider a vector of decision variables x and a random vector y governed by a probability measure P on Y that is independent on x. The decision variables and random vectors x and y univocally determine the value of a performance indicator $z = f(x, y)$ with $f(x, y)$ continuous in x and measurable in y and such as $E[|f(x, y)|] \leq \infty$. Given x and the performance indicator z, we define the associated distribution function $\psi(x, \cdot)$ on $\mathbb{R}$ as:

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

Given the left limit of $\psi(x, \cdot)$ at ζ

$$\Psi(x, \zeta^-) = P(y|f(x, y) < \zeta) \tag{2}$$

if the difference $\Psi(x, \zeta) - \Psi(x, \zeta^-)$ is positive, then $\Psi(x, \cdot)$ has a probability 'atom' in ζ equal to $P\{f(x, y) = \zeta\}$.

As defined in [19] and using the notation in [18], given a risk level α, the *Value-at-Risk* (α-*VaR*) of a performance indicator z associated with the decision x is:

$$\zeta_\alpha(x) = min\{\zeta|\Psi(x, \zeta) \geq \alpha\} \tag{3}$$

A different case refers to *discrete distributions* as in scenario-based uncertainty models. In these cases the uncertainty is modeled through finitely many points $y_k \in Y$ and, consequently, $z = f(x, y)$ is concentrated in finitely many points and $\psi(x, \cdot)$ is a step function. Under these hypotheses, the definition of the *Value-at-Risk* in (3) must be rephrased [18]. Given x, if we assume that the different possible values of $z_k = f(x, y_k)$ with $P(z = z_k) = p_k$ can be ordered as $z_i < z_2 < \cdots < z_N$ and given k_α such that

$$\sum_{k=1}^{k_\alpha} p_k \geq \alpha \geq \sum_{k=1}^{k_{\alpha-1}} p_k \tag{4}$$

then the α-*VaR* is given by

$$\zeta_\alpha(x) = z_{k_\alpha} \tag{5}$$

The relationship between risk measures and stochastic ordering plays an important role in defining dominance rules. Referring to the *Value-at-Risk*, since it simply is a quantile of the objective function distribution, the stochastic dominance between two *cumulative distribution functions (cdf)* also implies a dominance between the respective *Value-at-Risk*, for any given α.

4. Solution Approach

We consider a single machine scheduling problem where a set of n jobs A, must be processed on a single machine. Let p_j denote the processing time of job $j \in A$ and s_j its starting time. Job preemption is not allowed, i.e., the processing of a job cannot be interrupted until its completion at time $c_j = s_j + p_j$. Each job is subject to a release date r_j and a due date d_j. We propose a branch-and-bound approach aiming at finding an optimal schedule minimizing the VaR of the maximum lateness L_{max}. We restrict the problem to static scheduling policies, i.e., when the optimal schedule is calculated, the information for all the jobs to be scheduled are available. In addition, unforced idleness is allowed, i.e., the machine is allowed to remain idle to wait for the release time of a specific job even if there are other jobs waiting for processing.

Referring to the stochastic characteristics of the scheduling problem, both the release times r_j and the processing times p_j of the jobs are independent stochastic variables with general discrete distributions. As a function of stochastic variables, the objective function is a stochastic variables itself whose distribution depends on the values of the stochastic variables p_j and r_j and on a set of decisions to be taken defining how the jobs are scheduled.

4.1. Branching scheme

The branching scheme is rooted at node (level 0) where no job has been scheduled. Starting from this node, the first job to schedule is selected, hence, there are n branches departing from this node going down to n new nodes (level 1). In general, at each node at level $k-1$ in the branching tree, the first $k-1$ jobs in the schedule are already sequenced and $n-k+1$ branches lead to a new node at level k with a different jobs scheduled next. Hence, at level k there are $n!/(n-k)!$ nodes [8].

4.2. Nodes evaluation

Let us consider two jobs $i, j \in A$, where the two jobs have stochastic processing times p_i and p_j described by their cumulative distribution functions $F_i(t) = P(p_i \leq t)$ and $F_j(t) = P(p_j \leq t)$ and the associated probability density functions $f_i(t) = P(p_i = t)$ and $f_j(t) = P(p_j = t)$.

The time needed to process the two jobs in a serie (first i and then j) is a stochastic variable and its cumulative distribution function $F_{i+j}(t)$ is the convolution of $F_i(t)$ and $F_j(t)$ with $*$ being the convolution operator [20]:

$$\begin{aligned} F_{i+j}(t) &= F_i(t) * F_j(t) \\ &= \int_0^t F_i(t-s)dF_j(s) \\ &= \int_0^t F_i(t-s)f_j(s)ds \end{aligned} \tag{6}$$

Provided that the schedule starts at time 0, the cumulative distribution functions of the completion times of jobs i and j (F_{c_i} and F_{c_j}) can be defined as:

$$\begin{aligned} F_{c_i}(t) &= F_i(t) \\ F_{c_j} &= F_{c_i}(t) * F_j(t) = F_{i+j}(t) = F_i(t) * F_j(t) \end{aligned}$$

If we consider a stochastic release time for job j, it can be modeled as an additional job k with processing time r_j to be executed before j. Hence, job j can be executed only after both job k and i have been completed. Provided that job j starts as soon as possible, the cdf for its start time (s_j) and completion time (c_j) of can be calculated as:

$$F_{s_j}(t) = F_{c_i}(t) \cdot F_{r_j}(t) \tag{7}$$

$$F_{c_j}(t) = F_{s_j}(t) * F_j(t) \tag{8}$$

Hence, 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:

$$F_{L_j}(t) = F_{c_j}(t - d_j) \tag{9}$$

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

$$F_{L_{max}}(t) = \prod_{j \in A} F_{T_j}(t) \tag{10}$$

and all the previous described risk measures can be calculated.

This provides a way to calculate the cumulative distribution function of the maximum lateness in the leaves of the branching tree where the schedule is completely defined. In the nodes of the branching tree, on the contrary, only a subset of the jobs ($A^S \in A$) has been scheduled. For these activities the maximum lateness cdf can be calculated using the steps above. The execution of the remaining jobs ($A \backslash A^S \in A$) has not been sequenced yet and, hence, the cdf of the maximum lateness of the complete schedule cannot be univocally calculated. However, an upper and lower bound of the cdf can be provided. Given a not yet scheduled jobs in $j \in A \backslash A^S$, 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. Given the cdf of the completion time of the already scheduled jobs $F_{c_{A^S}}(t)$ and the cdf of the release time $F_{r_j}(t)$, the cdf of the earliest start time and completion time for j are:

$$F^{LB}_{s_j}(t) = F_{c_{A^S}}(t) \cdot F_{r_j}(t) \tag{11}$$

$$F^{LB}_{c_j}(t) = F_{s_j}(t) * F_j(t) \tag{12}$$

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

$$F^{LB}_{L_j}(t) = F^{LB}_{c_j}(t - d_j) \tag{13}$$

while the lower bound for the maximum lateness is:

$$F^{LB}_{L_{max}}(t) = \prod_{j \in A^S} F_{L_j}(t) \prod_{j \in A \backslash A^S} F^{LB}_{L_j}(t) \tag{14}$$

An upper bound for the lateness L_j of a not jet scheduled jobs $j \in A \backslash A^S$ can be obtained assuming that it will be sequenced as the last job in the schedule according to the following scheme. If we leave out the release times of the not yet scheduled jobs, we can calculate the cdf of the sum of their processing times $F_{A \backslash A^S}(t)$ as the convolution of all the cdfs $F_j(t)$ with $j \in A \backslash A^S$. However, leaving out the contribution of the release times is not reasonable but, when the sequence of the jobs in $A \backslash A^S$ is not given, their influence cannot be assessed. A worst case can be defined considering the distribution of the maximum release time among the jobs to schedule:

$$F_{r_{max}}(t) = \prod_{j \in A \backslash A^S} F_{r_j}(t) \tag{15}$$

and then assuming that all the jobs to be scheduled are executed after this release time.

$$F^{UB}_{s_{A \backslash A^S}}(t) = F_{c_{A^S}}(t) \cdot F_{r_{max}}(t) \tag{16}$$

$$F^{UB}_{c_{A \backslash A^S}}(t) = F^{UB}_{s_{A \backslash A^S}}(t) * F_{A \backslash A^S}(t) \tag{17}$$

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

$$F^{UB}_{L_j}(t) = F^{UB}_{c_{A \backslash A^S}}(t - d_j) \tag{18}$$

while the upper bound for the maximum lateness is:

$$F^{UB}_{L_{Max}}(t) = \prod_{j \in A^S} F_{T_j}(t) \prod_{j \in A \backslash A^S} F^{UB}_{L_j}(t) \tag{19}$$

4.3. Dominance rules

In the considered scheduling problem, the aim is at minimizing the maximum lateness. The maximum lateness is a *regular* objective function, i.e., a function non-decreasing in $C_1, \ldots, C_n$ -where C_i denotes the completion time of job *i*- and, due to the absence of unforced idleness, also non-decreasing in $p_1, \ldots, p_n$,.

At each node in the branching tree, the lower bound cdf represents a schedule where the not yet sequenced jobs are executed immediately after the already scheduled ones. If we schedule an additional job *j*, the not yet scheduled jobs must be shifted to start at earliest after job *j* is processed. Due to this, the completion time of the not scheduled jobs increases or, at least, has the same value as in the ancestor node. Since the objective function is regular, given a certain sample of the activity durations and release times, the values of the cdf of the ancestor must be greater or equal to the value of any of the successor nodes.

Hence, at each node in the branching tree, the lower bound cdf effectively provides a bound on the lower bound cdf of all the successor nodes, even more, the lower bound cdf stochastically dominates the lower bound cdfs of all the successor nodes. Moreover, since at the leaves of the tree the upper and lower bound cdfs collapse in a single curve, then this curve is also stochastically dominated by the lower bound cdfs of all its ancestor 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 three stochastically dominates all the upper bound cdfs of its ancestor nodes.

In the end, the cdf in a leaf of the tree always lies in the region bounded by the lower bound and upper bound cdfs of any of its ancestors. For these reasons the lower and upper bound cdfs can be used to calculate bounds for the considered risk measures providing a comparison criteria among the nodes of the search tree.

5. Testing

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 reach the optimal solution while the second addresses the comparison between the performance of the algorithm and other simpler approaches. Usually this comparison is done considering two algorithms aiming at the same objective function, but we adopted a different approach. The underlying idea is the observation that taking into consideration the distribution of the objective function introduces a significant complexity in the problem. Hence, besides evaluating the time needed to find the optimal solution, it is also interesting investigating the benefits coming from the use of a more complex approach compared to a simpler one. In this case we used as a comparison the solution provided by the *Earliest Due Date (EDD)* rule, a really simple rule that is not optimal but can be applied in a really fast way.

5.1. Experimental setup

To test the proposed approach we generated a set of instances consisting of 10 jobs with processing times distributed according to a discrete triangular distribution, release dates modelled through a discrete uniform distribution and deterministic due dates. In total, 160 instances have been randomly generated and solved considering the *VaR* and the different risk levels (5% and 25%), for a total of 320 experiments. The algorithm has been coded in C++ using the BoB++ library [21,22] and executed on 8 parallel threads on a workstation equipped with a double Intel Quad-Core X5450 processor running at 3.00 GHz and 8 Gb of RAM.

5.2. Results

The results in Table 1 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 fraction of the nodes of the complete branching tree visited during the search. For each combination, the minimum, maximum, average values and the standard deviation are reported.

Table 1. Solution time (in seconds).

Risk	Variable	Min.	Max.	Avg.	StDev.
5%	Solution time	0.810	114.780	7.350	14.030
	% nodes	0.017	2.438	0.126	0.266
25%	Solution time	0.547	97.625	6.042	9.156
	% nodes	0.012	1.785	0.111	0.190

The results in Table 1 show that the algorithm was able to find the optimal solution in an average time of about 6.7 seconds, with a variability ranging from a minimum value of 0.547 seconds to a maximum value of 114.781 seconds. Moreover, the average number of nodes visited during the search is about 0.12% of the total number of nodes in the branching tree (notice that the total number of nodes is equal to $\sum_{k=1}^{n} \frac{n!}{(n-k)!}$ and for $k = 10$ is equal to 6235300 nodes). In addition, the results seem to show a slightly increase of the solution time when dealing with a risk level of 5%. This behavior is reasonable since, as the considered quantile resides in the tail of the distribution, the value for different schedules are packed together in a strict range and the effectiveness of the bounding and pruning rules is decreased.

A second type of results aims at comparing the optimal solution obtained with the branch-and-bound approach with the solution obtained with a simple scheduling rule, i.e., the *Earliest Due Date (EDD)* rule. To compare the two solutions, first the EDD rule is used to obtain a schedule. Hence, the schedule is evaluated with the exact approach to calculate the real *VaR* associated. This value is then compared with the *VaR* of the optimal schedule provided by the branch-and-bound approach. The results are summarized in Table 2.

Table 2. Comparison with the EDD rule.

Risk	Variable	Min.	Max.	Avg.	St. dev.
5%	% △ EDD	0	228.570	6.43	24.37
25%	% △ EDD	0	98.50	5.09	15.52

The results show that the proposed approach perform on average between 5.09% and 6.43% better respect to a simple rule as the EDD. Performing better means that the solution provided by the EDD rule has a *VaR* different from that associated to the considered risk level. To better explain, let us want to find a schedule S_{opt} minimizing the 5%-*VaR* of the maximum lateness and that 5%-$VaR_{S_{opt}}$ = 30 days. This means that S_{opt} assures that the probability of having a maximum lateness grater that 30 days is 5%, and this probability is greater for all the other possible schedules. If the EDD rule provides us a different optimal schedule S_{EDD} we know that the associated 5%-*VaR* will be greater. Hence, if we adopt the schedule S_{EDD}, given the risk of 5%, we are exposing ourselves to a maximum lateness greater than that we would have adopting the optimal schedule S_{opt}. If the *% difference vs EDD* is equal to 10%, then under the adoption of S_{EDD}, the 5%-*VaR* is equal to 33 days, that is greater than 30 days. Hence, the probability of having a maximum lateness greater than 30 days is more than 5%.

It must be noticed that, for some instances, the branch-and-bound approach and the EDD rule provide the same value of the objective function (although not always the same schedule). For some other instances, instead, the difference is greater, reaching a maximum value of 228.57%, and exactly these extreme cases are the main justification to the adoption of stochastic approaches in place of those based on expected values.

6. Discussion

In this paper we presented a branch-and-bound stochastic scheduling approach to minimize a stochastic function of the maximum lateness. The proposed approach deals with a single machine stochastic scheduling problems with jobs characterized by a stochastic generally distributed discrete processing time, a stochastic generally distributed discrete release time and a deterministic discrete due date.

Since the aim is at guaranteeing a robust schedule capable of providing protection against the occurrence of low probability extremely unfavorable events, a measure of risk is used in the stochastic objective function, in particular, the *Value-at-Risk*.

The performance of the proposed branch-and-bound approach is reasonably fast in term of time to find the optimal solution. Needles to say that the dimension of the solved instances (10 jobs) is rather small and, as the number of jobs increases, also the solution time will do, and certainly more than linearly. However the parallel capabilities of the implementation easily permit to exploit the benefits of new multi core architecture or the execution on high performance calculation environments.

Clearly, the adoption of more powerful and complex computation systems must found a justification in the potentially achievable benefits. To this aim, an average benefit of about 7% respect to the adoption of a simple not optimal rule as the EDD seems low. However, as always happens when assessing the benefits of more complex stochastic approaches, the measure provided by the comparison of the average performance must not be considered reliable.

The need of a stochastic approach arises when expected value approaches are no more suitable. Two stochastic distributions with the same expected values could be significantly different, primarily in the shape and weight of the tails. Stochastic approaches aim at exploiting this difference and, hence, when compared in terms of expected values cannot exhibit significantly good performance. As stated before, the benefits of a stochastic approach reside in the capability of distinguishing the shape of different distributions, thus being able to assess the effects of events unlikely to occur but having a high impact on the targeted performance. From this perspective, an average difference of 6.69% respect to the EDD rule is not so important as a maximum difference of about 230% is.

Moreover, when dealing with lateness-related objective functions, the impact of a variation in the value of the object function is always 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, the benefits of the proposed approach can be better exploited when dealing with scheduling problem with a small number of jobs and where the impact of low probability extreme unfavorable events is significant. Possible application are the implementation of robust approaches within a more complex production system or the negotiation of due dates.

Further research will target the extension of the approach to different scheduling problems possibly through the introduction of different calculation methods able to provide an estimation of the objective stochastic distribution in return of a faster calculation.

Acknowledgements

This research has been partially funded by the EU FP7 projects VISIONAIR - Vision and Advanced Infrastructure for Research, Grant no. 262044 and RobustPlaNet - Shock-robust Design of Plants and their Supply Chain Networks, Grant no. 609087.

References

[1] Alfieri A, Tolio T, Urgo M A two-stage stochastic programming project scheduling approach to production planning. International Journal of Advanced Manufacturing Technology, 62(1-4):279–290
[2] Tolio T, Urgo M. Design of flexible transfer lines: A case-based reconfiguration cost assessment. Journal of Manufacturing Systems 2013; 32 (2):1–5.
[3] Tolio T, Urgo M. A Rolling Horizon Approach to Plan Outsourcing in Manufacturing-to-Order Environments Affected by Uncertainty. CIRP Annals - Manufacturing Technology 2007; 56 (1):487–490.
[4] Tolio T, Urgo M, Váncza J. Robust production control against propagation of disruptions. CIRP Annals - Manufacturing Technology 2011; 60(1):489– 492.
[5] Radke AM, Tolio T, Tseng MM, Urgo M. A risk management-based evaluation of inventory allocations for make-to-order production. CIRP Annals - Manufacturing Technology 2013; 62(1):459–462.
[6] Lenstra, J, Kan, AR, Brucker, P. Complexity of machine scheduling problems. Annals of Discrete Mathematics 1977;1:343 – 362.
[7] Kellerer, H. Minimizing the maximum lateness. In: Handbook of Scheduling: Algorithms, Models and Performance Analysis. Chapman & Hall/CRC Computer & Information Science Series; Chapman and Hall/CRC; 2004, p. –.
[8] Pinedo, ML. Scheduling - Theory, Algorithms, and Systems. Springer New York; 2008.
[9] Shaked, M., Shanthikumar, JG. Stochastic Orders. Springer Series in Statistics; Springer New York; 2007.
[10] Ross, SM. Stochastic Processes. 2nd ed.; Holden Day; 1983.
[11] Chang, CS, Yao, DD. Rearrangement, majorization and stochastic scheduling. Mathematics of Operations Research 1993;18(3):658–684.
[12] Zhou, X, Cai, X. General stochastic single-machine scheduling with regular cost functions. Mathematical and Computer Modelling 1997;26(3):95 – 108.
[13] Cai, X, Zhou, X. Single-machine scheduling with exponential processing times and general stochastic cost functions. Journal of Global Optimization 2005;31(2):317–332.
[14] Cai, X, Wang, L, Zhou, X. Single-machine scheduling to stochastically minimize maximum lateness. Journal of Scheduling 2007;10(4):293–301.
[15] Wu, X, Zhou, X. Stochastic scheduling to minimize expected maximum lateness. European Journal of Operational Research 2008;190(1):103 – 115.
[16] De, P, Ghosh, JB, Wells, CE. Expectation-variance analyss of job sequences under processing time uncertainty. International Journal of Production Economics 1992;28(3):289 – 297.
[17] Sarin, S, Nagarajan, B, Jain, S, Liao, L. 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 2009;17(4):400–416.
[18] Rockafellar RT, Uryasev S, 2002. Conditional value-at-risk for general loss distributions. Journal of Banking & Finance 26 (7), 1443 – 1471.
[19] Artzner P, Delbaen F, Eber JM, Heath D, 1999. Coherent measures of risk. Mathematical Finance 9 (3), 203–228.
[20] bibinfoauthorAgrawal, M. K. , Elmaghraby, S. E., On computing the distribution function of the sum of independent random variables. Computers & Operations Research 2001;28(5):473 - 483.
[21] Bob++: A framework to solve combinatorial optimization problems. URL: `https://software.prism.uvsq.fr/bobpp/`
[22] Djerrah A, Le Cun B, CungVD, Roucairol, C., 0-0 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.