Int J Adv Manuf Technol (2012) 62:279–290
DOI 10.1007/s00170-011-3794-4

ORIGINAL ARTICLE

A two-stage stochastic programming project scheduling approach to production planning

Arianna Alfieri · Tullio Tolio · Marcello Urgo

Received: 17 December 2010 / Accepted: 17 November 2011 / Published online: 17 December 2011

Abstract In engineering-to-order (ETO) or manufacturing-to-order (MTO) systems producing highly customized items, the high level of customization, together with long flow times, forces the production plan to be defined before information on items customization, and details on the manufacturing activities are completely disclosed. Due to the partial available information, the production plan must provide a robust schedule of the activities and of the resources utilization, trying to incorporate a certain degree of anticipation of uncertainty. This paper proposes a two-stage stochastic programming project scheduling approach to support production planning in ETO/MTO system. The approach provides a baseline production plan together with a set of revisions of the plan to react to the occurrence of uncertain events. A scenario-based approach is used to model the changes affecting the characteristics of the activities to be processed. The proposed approach is tested on random-generated instances and on a real manufacturing system producing machining centers.

A. Alfieri
Production Systems and Economics Department,
Politecnico di Torino, Turin, Italy

T. Tolio
ITIA-CNR, Institute of Industrial Technologies
and Automation, National Research Council, Milan, Italy

M. Urgo (✉)
Manufacturing and Production Systems Division,
Mechanical Engineering Department,
Politecnico di Milano, Milan, Italy
e-mail: marcello.urgo@mecc.polimi.it

Keywords Production planning · Stochastic programming · Project scheduling

1 Introduction

This paper studies production planning under uncertainty in engineering-to-order (ETO) or manufacturing-to-order (MTO) production systems. The use of project scheduling approaches for production planning have been frequently addressed in the scientific literature [16, 23]. In particular, when hierarchical planning approaches are used, project scheduling can serve as a planning tool at aggregate levels [26–28] and are especially important and useful in MTO or ETO systems producing complex and highly customized items. In such environments, each item has its own characteristics, which are often tailored for a specific customer. Due to the complexity of such products, the single production activities often correspond to whole production phases; hence, to devise feasible short-term plans, routing and precedence relation issues must be taken into consideration already in the medium-term planning phase. The design, production, and delivery to the customer of each product is then a *one of a kind* activity that can be easily modeled as the execution of a project. In the production plant, different projects are executed together, competing for the same production resources (machines, workers, etc.).

In addition, when activities are executed by workers, the concepts of unary resources and activity duration must be reassessed. Given an assignment to a set of workers, the committed effort can vary over time according to workers availability or to the joint execution of different activities. Under this assumption,

either the resource used in each time period or the duration of the activity is not univocally defined causing traditional scheduling methods to be no more suitable. According to the literature, one of the proposed solutions is the variable intensity formulation of the resource-constrained project scheduling problem [2, 15] where an intensity variable is introduced defining the effort dedicated to process the activity in each time period.

Uncertain or incomplete data represent a not negligible factor in such production environments. High levels of customization and long flow times result in production plans to be defined before information on product customizations, and detailed production activities are completely disclosed. The influence of uncertainty on production systems performance can be severe, and it must be taken into consideration at the planning phase. Production planning must provide a robust schedule for the execution of activities and the utilization of resources, trying to incorporate a certain degree of anticipation of uncertain events. The term robustness is here refereed to the so-called quality robustness, that is, the insensitivity of the plan, in terms of target performance, to the occurrence of uncertain events. Hence, robustness of a production plan strictly relies on the possibility of modifying the schedule of the activities within the plant with little or no penalty in terms of objective function value.

In this paper, a two-stage stochastic programming approach is proposed to plan production in a MTO/ETO system affected by uncertainty. The proposed approach is based on a variable intensity formulation of the resource-constrained project scheduling problem and on a scenario modeling of uncertain events.

2 Literature review

Uncertainty in project scheduling problems can stem from different sources. Aytug et al. [4] define an extensive taxonomy of uncertainties at the planning level. Uncertainties can affect different characteristics of the problem: resource needs of activities, resource availability, need of rework activities, occurrence of new orders [22], release dates, and due dates. Scheduling approaches dealing with uncertainty are commonly divided in two main categories: approaches generating a baseline schedule before the effective starting of project execution and approaches in which the schedule is generated during project execution. Methods generating a baseline schedule are usually more suitable to be used in real manufacturing environments. In fact, as observed in [4], an activity plan/schedule serves several purposes: aims at optimizing a given objective function, can be used as capacity check for higher-level management decisions, and makes clear the connection among current and future activities.

Under uncertainty conditions, the baseline schedule must be robust, i.e., it has to be insensitive to the occurrence of uncertain events within a given range of magnitude. However, the robustness concept can be considered from two different points of view. The term *solution robustness* or *stability* is used to address insensitivity of the schedule in terms of activity start times [12, 13, 19]. The term *quality robustness*, instead, is used when referring to the property of a schedule to be insensitive in terms of the value of the objective function.

From the point of view of the stability, the available scheduling approaches can be classified according to the way they exploit information relative to uncertainty. According to this dimension, scheduling approaches can be partitioned in *reactive* and *proactive* approaches. Reactive scheduling approaches do not incorporate in the schedule information about uncertainty but simply provide a proper strategy to "repair" the schedule when something unexpected happens. They are therefore also referred to as *schedule repair approaches*. A particular case of reactive scheduling is the *minimal perturbation strategy*. In this case, the objective of the repair phase is the generation of a new schedule deviating as little as possible form the original one, i.e., aiming at the so-called *ex post stability* of the schedule. This type of approach can be found in [1, 6, 8]. Reactive scheduling approaches also address the problem of inserting new activities in a baseline schedule [3].

Proactive scheduling approaches, instead, try to incorporate information about uncertain events in the baseline schedule, so that it can be protected, as well as possible, against future disruptions, i.e., aiming at the so-called *(ex ante) stability*. Project scheduling approaches focused on stability are presented in [19, 20]. Given a variable activity duration, the authors define the stability in terms of the weighted deviation with respect to the baseline schedule when a single activity is perturbed. The authors present also an extension of the model in case resource constraints are considered. The schedule stability problem and its connection with makespan minimization have been studied in [34, 35]. An experimental analysis is carried out to evaluate several predictive–reactive resource-constrained project scheduling procedures under the objective of maximizing both scheduling stability and timely project completion.

Quality robustness is addressed in [18] where a project scheduling approach is proposed to cope with uncertainty affecting the availability of resources. A stochastic formulation of failures and repair is used, and a baseline schedule is provided through a recursive approach. This class of methods also contains approaches addressing the *stochastic resource-constrained project scheduling problem*, the aim of which is to schedule (project) activities so as to minimize the expected project duration [13]. For a deeper analysis of stochastic scheduling and related solution approaches, the reader can refer to [10, 11, 24, 25, 29].

However, although trying to incorporate uncertainty information in the baseline schedule, the described approaches do not provide a mechanism to modify the schedule at the occurrence of uncertain events. A challenging issue in robust scheduling is therefore trying to design approaches that retain both reactive and proactive scheduling benefits; this class of approaches are sometimes called *predictive–reactive*. Stochastic programming techniques seem the most promising tool to achieve this objective. They are able to exploit information about uncertainty in the future to define recourse actions that can be adopted to correct previously taken actions, thus reducing the impact on the target performance.

A particular type of stochastic programming approach is the two-stage (or multi-stage) approach. In two-stage (multi-stage) stochastic programming, the decision horizon is separated in two (more) parts, called stages. The first stage is associated to a set of unique decision variables (the so-called *first-stage variables*) and models the present, without considering how uncertainty will occur. The second (and following) stage is associated to a set of stage-related decision variables (the so-called *second-stage variables*) used to correct the first (or, in multi-stage, previous) stage decisions on the basis of uncertainty outcomes of the considered scenarios [5]. Scenarios are a way to model uncertainty outcomes in a discrete way [5].

The first use of stochastic programming to study the RCPSP dates back to [9]. In this paper, a multi-stage stochastic programming model is proposed which associates to each stage the decision variables related to the starting of different activities. In fact, if the activities have a discrete duration, then the stochastic decision process can be modeled as a multi-stage stochastic programming model. As previously described, rather than only providing a tool to schedule a project, the proposed method also allows an evaluation of the scheduling policy. A single stage stochastic programming approach addressing the robust resource loading problem is proposed in [36]. Further developments can be found in [21, 37]. Two-stage stochastic programming approaches are presented in [14, 31, 38]. In [31], a project scheduling approach is used to plan the utilization of outsourcing when rework activities are present. The cost of outsourcing resources is minimized in [14], where first-stage variables deal with the schedule while in the second stage the resource allocation is defined. Zhu et al. [38], instead, use stochastic programming to define due dates and to minimize lateness in case of uncertain activity durations. First-stage variables represent target times for the completion of the activities while in second stage the detailed schedule is devised.

3 A two-stage stochastic programming formulation for production planning

In two-stage stochastic programming approaches, the set of decisions can be divided in:

- A set of decisions that have to be taken before the observation of any of the uncertain elements in the problem. These decisions are called *first-stage decisions*, and the period when these decisions are taken is called *first stage*
- A set of decisions that can be taken after the occurrence of uncertain events. They are called *second-stage decisions*. The corresponding period is called *second stage*.

The definition of first and second stage, together with the corresponding variables, plays an important role in the stochastic programming models. In fact, besides being a simple classification, first- and second-stage variables define when decisions can be taken and their mutual influence.

In stochastic programming approaches, first and second stage can be easily defined if we can a priori separate the timeline in two sections—before and after the occurrence of uncertain events. As an example, the amount of products sold in a certain period could be an information used to take decisions concerning the type or quantity of good to be produced in the following periods. In this case, first-stage variables are used to define the quantity to produce in the first period, while second-stage variables model the quantity to produce in the following periods. In an analogous way, in the assets management problem, assets can be bought or sold on specific dates or in definite time windows. Hence, first-stage variables can represent the amount of each asset to be bought in the first period, while second-stage variable can be used to model the asset to sell or buy in the following period (or periods). In other words,

defining stages is relatively easy if time when uncertain events occur does not depend on first-stage actions.

In the application of stochastic programming to production planning problems, it seems natural to use *first-stage* variables to define the baseline production plan, i.e., a detailed schedule of the activities and the related resources allocation. *Second-stage* variables are instead used to represent modifications of the baseline production plan after the occurrence of uncertain events.

The above definition of first- and second-stage variables, although representing a natural interpretation of stages in the context of production planning, brings in the fact that the time an uncertain event occurs depends on the value of first-stage variables. In fact, most of the uncertainty in the planning problem strictly refers to the characteristics of the activities to be planned, i.e., their duration and/or work content. Since the production planning itself has to define the schedule of the activities, it also influences the possible occurrence in time of uncertain events. Consequently, it is not possible to know in advance which variables are in the first stage and which in the second stage without knowing the schedule of the activities, which is indeed the solution of the problem. Hence, in this case, it is not possible to completely formalize the two-stage stochastic programming problem, not being able to separate first- from second-stage variables.

To tackle this problem, a mechanism is needed to define if a certain part of the time horizon is before or after the occurrence of uncertain events. To this purpose, we define a boolean variable that assumes value 1 until the uncertain event has not yet occurred. From the time the uncertain event happens on, the variable (we call it *certainty mask* in the following) assumes value 0, thus indicating that the uncertain part of the time horizon starts. Given the definition of this *certainty mask*, if it is possible to link it to the occurrence of the uncertain events in the considered scenario, then first and second stage are univocally determined.

In the planning problem, we consider in the paper that the variables affected by uncertainty are the amount of resource needed to process the activities. The value of this (uncertain) amount can disclose: (1) before the execution of the activity it refers to, as in the case the release date of the activity is modified; (2) after the end of the activity execution, as in the case the activity processing time is longer due to rework needed after inspection or testing; and (3) during the execution of the activity.

To simplify the study, however, we assume that the disclosure of uncertainty affecting an activity occurs exactly when the execution of the activity starts. This hypothesis is not too constraining since it supposes that, when an activity is started, its execution details and, hence, the resources needed are univocally determined.

Referring to the previously defined *certainty mask*, the start time of the activity affected by uncertainty must be used to have the mask switching from the certainty to the uncertainty state. However, since the starting time of the uncertain activities is defined through the baseline schedule devised in the first stage, a mechanism must be embedded in the formulation to assure a proper definition of the certainty masks for the different scenarios.

3.1 Mathematical formulation

The two-stage stochastic programming formulation of the production planning problem described in Section 3 is here reported. The formulation is a time-indexed formulation where stages are based on the definition of *certainty mask* to allow variable duration of stages in the different scenarios. The decision variables that characterize the problem are the following:

- x_{jt}: continuous positive variable representing the percentage of work done on activity j in time bucket t (first stage)
- z_{jt}: binary variable defining the execution mask for activity j assuming value 1 if the activity j can be processed in time bucket t (first stage)
- η_{jt}: binary variable assuming value 1 if activity j is processed in time bucket t (first stage)
- $u_{t\sigma}$: certainty mask for scenario σ. It assumes value 1 until any activity affected by uncertainty in scenario σ starts (second stage)
- $\tilde{x}_{jt\sigma}$: continuous positive variable representing the percentage of work done on activity j in time bucket t in scenario σ (second stage)
- $\hat{x}_{j\sigma}$: continuous positive variable representing the ratio between the work content of activity j in scenario σ and in the baseline plan (second stage)
- $\tilde{z}_{jt\sigma}$: binary variable defining the execution mask for activity j, assuming value 1 if activity j can be processed in time bucket t in scenario σ (second stage)
- $\tilde{\eta}_{jt\sigma}$: binary variable assuming value 1 if activity j is processed in time bucket t in scenario σ (second stage)

While *certainty masks* are used to separate first- and second-stage variables, *execution masks* are used to allow the execution of activities in time buckets. They substitute the typical binary variables representing start times and/or end times in the classical time indexed formulation for planning problems. Each execution mask z_{jt} has value 1 at $t = 0$ and is constrained to have

non-increasing shape. Hence, mask z_{jt} assumes value 0 only after activity j has been completed. The use of an execution-mask-based formulation has shown to be more computationally efficient in a deterministic context with respect to a classical time indexed formulation [2]. Due to this, it has been considered for the extension to the stochastic case.

The work done on an activity, instead, refers to the part of the activity already completed. Since resources are shared among activities, the work done in a single time bucket on a single activity is not a priori given by the length of the time bucket, and hence, such variables are needed to correctly represent activity execution.

Using the above-defined variables, the following model can be devised:

$$\min \sum_{\sigma\in\Sigma}(\pi_\sigma \cdot C_{\max,\sigma}) \tag{1}$$

$$\text{s.t.} \quad C_{\max,\sigma} \geq t \cdot z_{jt\sigma} \qquad \forall j,t,\sigma \tag{2}$$

$$\sum_t x_{jt} = 1 \qquad \forall j \tag{3}$$

$$x_{jt} \leq B_j \eta_{jt} \qquad \forall j,t \tag{4}$$

$$x_{jt} \geq b_j \eta_{jt} \qquad \forall j,t \tag{5}$$

$$x_{jt} \leq B_j z_{jt} \qquad \forall j,t \tag{6}$$

$$z_{j(t-1)} \geq z_{jt} \qquad \forall j,t \tag{7}$$

$$x_{jt} \leq B_j(1 - z_{it}) \qquad \forall (i,j) \in \mathcal{P}, \forall t \tag{8}$$

$$\sum_j Q_{jk} x_{jt} \leq R_{kt} \qquad \forall k,t \tag{9}$$

$$\sum_{t=r_{js}}^{d_{js}} \tilde{x}_{jt\sigma} = \hat{x}_{jt\sigma} \qquad \forall j,t,\sigma \tag{10}$$

$$\tilde{x}_{jt\sigma} \leq B_j \tilde{\eta}_{jt\sigma} \qquad \forall j,t,\sigma \tag{11}$$

$$\tilde{x}_{jt\sigma} \geq b_j \tilde{\eta}_{jt\sigma} \qquad \forall j,t,\sigma \tag{12}$$

$$\tilde{x}_{jt\sigma} \leq B_j \tilde{z}_{jt\sigma} \qquad \forall j,t,\sigma \tag{13}$$

$$\tilde{z}_{j(t-1)\sigma} \geq \tilde{z}_{jt\sigma} \qquad \forall j,t,\sigma \tag{14}$$

$$\sum_i Q_{ik} \tilde{x}_{jt\sigma} \leq R_{kt} \qquad \forall k,t,\sigma \tag{15}$$

$$\tilde{x}_{jt\sigma} \leq B_j(1 - \tilde{z}_{it\sigma}) \qquad \forall (i,j) \in \mathcal{P}, t, \sigma \tag{16}$$

$$x_{jt} \leq B_j(1 - u_{t\sigma}) \qquad \forall \sigma, t, j \in U_\sigma \tag{17}$$

$$u_{t\sigma} \geq \left(1 - \sum_{h=0}^{t} \eta_{jh}\right) \qquad \forall \sigma, t, j \in U_\sigma \tag{18}$$

$$u_{t\sigma} \geq u_{j(t+1)} \qquad \forall \sigma, t, j \in U_\sigma \tag{19}$$

$$\tilde{x}_{jt\sigma} = x_{jt} \qquad \forall j,\sigma,t | u_{t\sigma} = 1 \tag{20}$$

$$\tilde{z}_{jt\sigma} = z_{jt} \qquad \forall j,\sigma,t | u_{t\sigma} = 1 \tag{21}$$

$$\tilde{\eta}_{jt\sigma} = \eta_{jt} \qquad \forall j,\sigma,t | u_{t\sigma} = 1 \tag{22}$$

Equation 1 is the objective function, which represents the expected makespan. In fact, π_σ and $C_{\max,\sigma}$ are the probability and the makespan of scenario σ. The makespan is defined by the completion time of the last activity to be completed, as defined by Eq. 2.

Constraints from Eqs. 3 to 8 define the baseline schedule. Constraints 3 assure that each activity is completely executed in the baseline plan. Constraints 4 and 5 limit the amount of activity that can be processed in a single time bucket through a minimum b_j and maximum B_j value. Moreover, they assure that, if the activity is executed, the η variables assume value 1. Constraints 6 assure that an activity is executed only if its mask z assumes value 1 while constraints 7 impose the non-increasing behavior for the z masks. Finally, Eq. 8 forces the correct execution of pair of activities linked by precedence relation ($\mathcal{P}$ is the set of precedence relations). If activity i must precede activity j, the last cannot start, and hence, x_{jt} cannot assume values greater than 0, if the first is not completed, i.e., $z_{it} = 0$. The minimum and maximum values in constraints 4 and 5 are mainly due to technological and/or economical reasons. In fact, it can be infeasible or non-economical to process a given activity, requiring given resources, more or less than given thresholds.

Constraints 9 implement resource constraints, i.e., assure that, for each resource k and in each time bucket t, no more than the total available amount R_{kt} can be used. Here, the parameter Q_{jk} is the amount of resource k needed to execute activity j. A resource k could be used to model a single machine, a set of resources, i.e., a team of workers or a group of workers with specific skills.

Constraints 10–16 refer to second-stage variables. In particular, Eq. 10 provides that each activity is completely executed. It must be noticed that if in the considered scenario the duration of an activity is greater than the one considered in the definition of the baseline schedule, then $\hat{x}_{jt\sigma} \geq 1$. Parameters r_{js} and d_{js} are the earliest start time and latest finish time of activity j, determined by considering the duration of activity j in scenario σ. Constraints from Eqs. 11 to 16 are analogous to the first-stage constraints, but involve, obviously, second-stage variables.

Constraints from Eqs. 17 to 19 refer to the definition of certainty masks. To define the behavior of the certainty masks, the same mechanism of precedence

relation is used. Constraints 17 state that, until the certainty mask $u_{t\sigma}$ assumes value 1, the activity j affected by uncertainty cannot be processed ($x_{jt} \leq 0$). To force the certainty mask not assuming value 0 before the activity j has started, constraints 18 are used: If at time t activity j has not been processed yet (i.e., $1 - \sum_{h=0}^{t} \eta_{jh} = 1$), then mask $u_{t\sigma}$ must assume value 1. Finally, Eq. 19 defines the non-increasing behavior of the certainty masks. In constraints 17–19, U_σ is the set of activities affected by uncertainty in scenario σ.

Finally, constraints 20–22) are needed to respect the *non-anticipativity* condition in the stochastic programming formulation. In fact, they assure that, while the certainty mask $u_{t\sigma}$ assumes value 1, the scenario schedule must be equal to the baseline schedule. This means that second-stage decisions can be taken into account only after the uncertain event in the considered scenario has occurred.

4 Computational experiments

The two-stage stochastic programming formulation presented in Section 3.1 has been tested on randomly generated instances. To solve the stochastic programming model, CPLEX 12.1 was used on a XEON workstation (clock 3.0 Ghz, RAM 4.00 Gb).

4.1 Random instance generation

Randomly instances have been generated using RanGen2 [33], an activity network instance generator for project scheduling problems based on morphological indicators (activity networks can, in fact, have very different morphological structures). In [30, 32], several complexity measures are proposed to describe the morphological structure of a network while in [7], resource-related measures are presented. Among the morphological indices, we consider the following:

- I_1: *Size of the problem.* This index is equal to the number of nodes (i.e., activities) in the network, and it is a measure of the size of the network.
- I_2: *Serial or parallel indicator.* It measures how close a network is to a serial or parallel directed graph. When all activities are in parallel, $I_2 = 0$, while when all the activities are serially connected, $I_2 = 1$. Real networks contain a number of activities that can be executed in parallel and a number of serial precedences. The closer to 1 is the value of I_2, the larger the number of serial connections with respect to the parallel components of the network.

Among the resource-related measures, we consider:

- RU: *Resource density.* RU measures, for each activity, the number of resources it uses (not the amount used). The value RU varies between 0, if the activity needs no resource, to the maximum number of resources available, if the activity uses all the available resources. RU can only assume integer values.
- RC: *Resource constrainedness.* It computes, for each resource, the ratio between the average amount (over all activities that use the resource) required for the resource and its total availability. *RC* is zero if no activity uses the resource, while it approaches to 1 if all activities, requiring the resource, demand for a quantity close to the total availability. If *RC* is bigger than 1, the problem is resource-infeasible since, on average, it is required more than the available quantity of resource.

A pool of 2,000 instances was generated using the generation parameters reported in Table 1. The roles of I_1, I_2, RU, and RC are as described above while Res indicates the number of resources and n the percentage of the total number of activities in the instance with uncertain duration. Finally, to assure the complete randomness of the test instances, a set of 50 instances has been sampled, from the 2,000 randomly generated, to be used in the experiments.

RanGen2, however, generates instances for classical resource constrained project scheduling problems, i.e., instances with fixed activity durations and do not consider any representation for uncertainty. To use variable intensity formulation for activity execution and to consider uncertain duration of the activities, the generated instances were modified in the following way:

- The duration L_j of activity j is considered as the minimum duration, i.e., $B_j = 1/L_j$. The minimum percentage of activity processable in each time bucket is constrained to be at least the 5% of the activity ($b_j = 0.05$).
- A number n of activities in the instance are sampled and their durations considered stochastic variables. A symmetric triangular distribution is fitted considering the deterministic duration as the mode; the

Table 1 Parameter values for instance generation

I_1	10
I_2	0.50, 0.75
Res	1
RU	1
RC	0.5, 0.75
n	2 (20%)

minimum and maximum duration are defined as 75% and 125% of the mode.

- For each of the sampled activities, five scenarios are defined to model the triangular distribution of its duration. The discrete scenarios are generated using the *rounding method*, considering the discrete cumulative distribution function passing through the midpoints of the intervals [17]. According to this method, the support of the distribution is divided in five intervals with equal width h, and the cumulative distribution function passing through the midpoints x_i of the intervals is considered. Hence, each value x_i represents the interval $[x_i - h/2, x_i + h/2]$. The associated probability is defined as: $p[x_i] = F(x_i + h/2) - F(x_i - h/2)$ (Fig. 1).
- For each instance, 25 scenarios are generated. The number of activities in each instance is 10 (value of I_1 in Table 1) and 20% (value of n in Table 1) of them are considered potentially uncertain. The 25 scenarios are obtained considering all the combination (5×5) of the possible values of the two sampled activities.

4.2 Evaluation of the stochastic programming approach

The simplest approach to deal with uncertainty is to replace all random variables with their expected values. This modified problem is called *expected value problem* or *mean value problem*. Although the solution of such a simpler problem can be very far from the stochastic optimum, given the solution of the *expected value problem* (EV), it is possible to test its quality in each of the considered scenarios. This quality can be measured through the *expected result of using the EV solution* (EEV), i.e., the expected value of the performance of the EV solution at the occurrence of each of the scenarios and allowing second-stage decisions to be optimally taken.

Table 2 Results (random-generated instances)

I2	RC	n_o	VSS		Solution time		Max delta
			Average	0.0664	Average	63.00	
0.5	0.5	20	Min	0.0000	Min	0.70	4 (0.0064)
			Max	0.4320	Max	299.20	
			Average	0.0458	Average	345.00	
0.5	0.75	20	Min	0.0000	Min	1.00	3 (0.0064)
			Max	0.2864	Max	2285.00	
			Average	0.0405	Average	49.11	
0.75	0.5	20	Min	0.0000	Min	31.34	4 (0.0576)
			Max	0.2688	Max	69.10	
			Average	0.0498	Average	68.40	
0.75	0.75	20	Min	0.0000	Min	0.10	2 (0.0288)
			Max	0.2496	Max	516.90	

The EEV can be used to estimate the benefits of modeling and solving a problem as a stochastic program instead of as an EV problem. Given the value of the objective function in the stochastic programming approach (i.e., the so-called recourse program RP), the *value of the stochastic solution* (VSS) can be defined as:

$$\text{VSS} = \text{EEV} - \text{RP}. \tag{23}$$

in other words, VSS represents the cost that is incurred if uncertainty is ignored.

4.3 Results

The results of the experiments are reported in Table 2. Columns $I2$, RC, and n_o reports the value of the $I2$ and RC indicators and the number of experiments done for that combination of values (20 experiments for all the combinations). Column VSS reports the average, minimum, and maximum value of VSS in the considered experiments. Column Solution Time reports the statistics for the time needed to solve the instances (in minutes). The last column, Max Delta, reports the maximum difference between the makespan obtained with the recourse programming (RP) and the expected

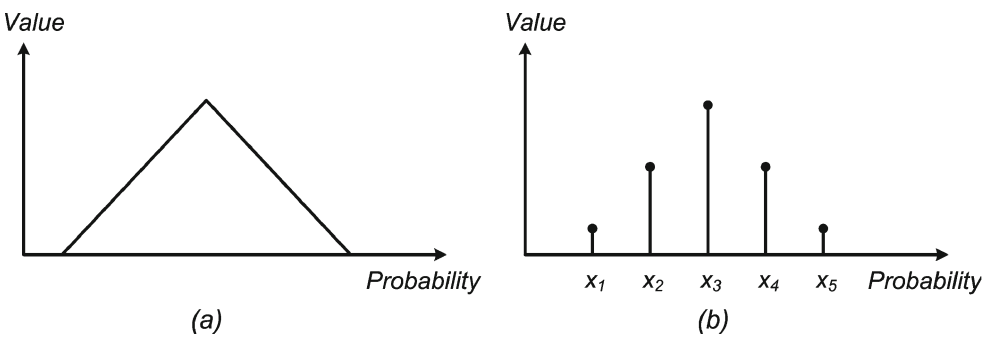

Fig. 1 Scenarios definition for activity duration

value (EV) approaches in the different scenarios. The number in parentheses is the probability associated to the scenario experiencing the maximum difference.

The results show that, as either $I2$ or RC increases, the average value of VSS decreases. This is also confirmed by the main effect plot for the value of VSS, reported in Fig. 2, where the points in the plot are the means of VSS for various values of $I2$ and RC. This trend reflects the characteristic of stochastic programming approaches. Such approaches can provide benefits when a large set of solutions is available and considering stochastic information allows to select the best solution assuring an average good performance on all the considered scenarios. In a scheduling perspective, many solutions are available when different schedules can be used to execute the activities. Multiple schedules are available when a set of activities can be executed in parallel (no precedence relations among them) since they could be executed in different sequences. At the same time, the availability of resources affects the possible feasible schedules. If enough resources are available to execute activities in parallel, then multiple schedules are possible; on the contrary, when scarce resources allow only a small number of activities (only one in the worst case) to be executes at the same time, possible schedules are significantly less.

However, these observations do not exhaustively represent the problem. In fact, the importance of being able to select the best decision in a stochastic environment strongly relies on the magnitude of the consequences of a wrong decision. In the considered problem, a good schedule, given the stochastic information, should leave the widest possibility to react to uncertain event in the most different ways. However, the effectiveness of such reactions is strongly influenced by the characteristics of the scheduling problem. The first influencing factor is parameter $I2$. If some activities can be executed in parallel, at the occurrence of an uncertain event, it could be possible to freely change the execution of some of them to react. On the contrary, when only a sequential execution of activities is allowed, the degrees of reaction are limited. Hence, when the reaction to the occurrence of uncertain events is constrained by precedence relations and availability of resources, the capability of choosing the best schedule gains importance. These observations are confirmed by the interaction plot of VSS, reported in Fig. 3. It can be seen that, when $I2 = 0.5$, a decrease in resources availability causes a difficulty in implementing a proper reaction to uncertain events; hence, the average value of VSS decreases. When $I2 = 0.75$, instead, the trend is opposite, that is, the average value of VSS increases as the value of RC increases. This is due to the fact that, as the network of activities tends to a serial structure, the available options to correct the schedule decreases and, consequently, being able to select the best schedule in the early stages becomes more important.

It must be noticed that Figs. 2 and 3 report the absolute value of VSS. A better understanding of the magnitude of the influence for the two factors can be obtained normalizing the value of VSS with respect to the value of the objective function in the corresponding recourse programming solution (RP). Normalized values are reported in Figs. 4 and 5. Basically this normalization allows to consider the fact that, as the resource availability decreases, the value of makespan increases. In particular, Fig. 5 shows that, given $I2 = 0.5$, increasing RC to 0.75 implies a reduction of the normalized VSS of about 35% (from 0.001806 to 0.001170) while reaching the same value of RC when $I2 = 0.75$ causes

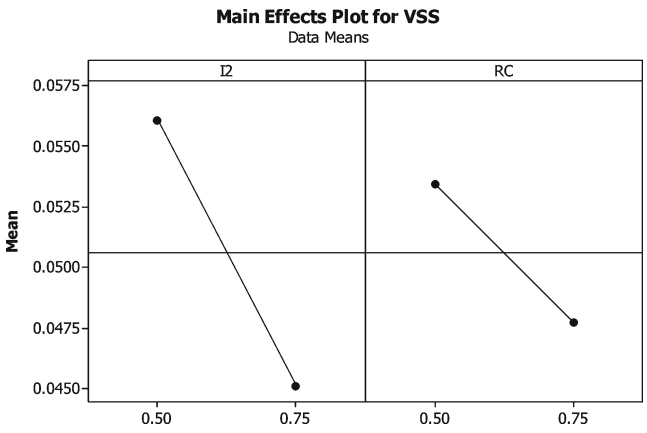

Fig. 2 Main effects plot for VSS

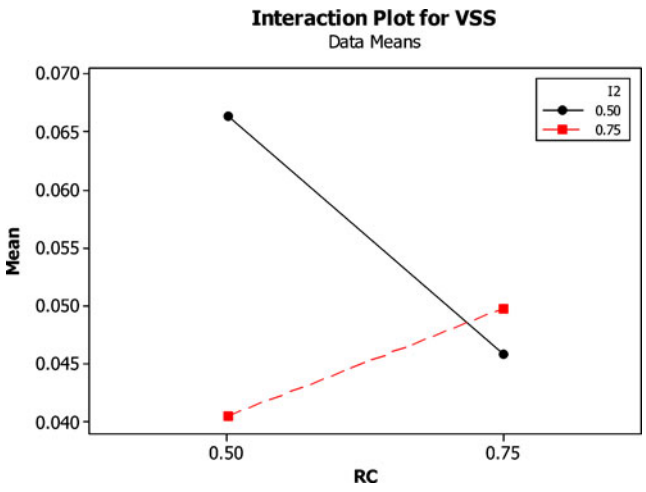

Fig. 3 Interaction plot for VSS

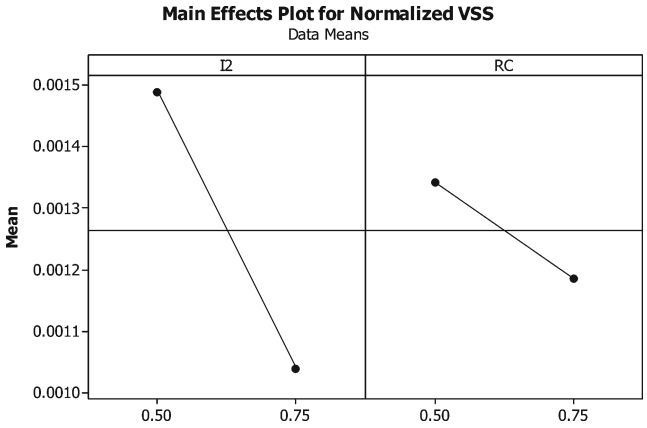

Fig. 4 Main effects plot for normalized VSS

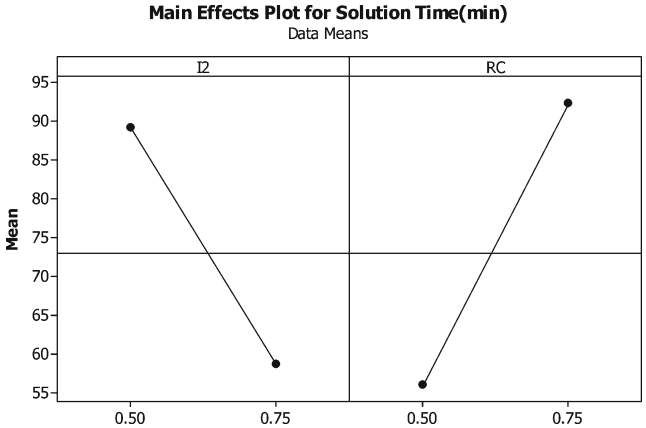

Fig. 6 Main effects plot for solution time

an increase about 37% (from 0.000878 to 0.001202). Hence, the influence of the two factors is comparable.

The values of VSS obtained in the experiments are rather low. Hence, if VSS is used to measure the benefit achievable using a stochastic programming approach, the performance could be considered extremely poor. However, VSS is an average value and, due to this, the contribution of unfavorable cases causing a significant increase of the makespan must be weighted by their occurrence probability, which is usually low. To highlight the magnitude of unfavorable cases, the last column in Table 2 (Max Delta) reports the maximum difference in the value of the makespan between the stochastic programming (RP) and the expected value approach (EV), together with the probability of the associated scenario, in parentheses. It can be seen that the maximum difference ranges between 2 and 4 weeks. But due to the low occurrence probability, its impact on VSS is not so large. It is clear that, even if VSS assumes

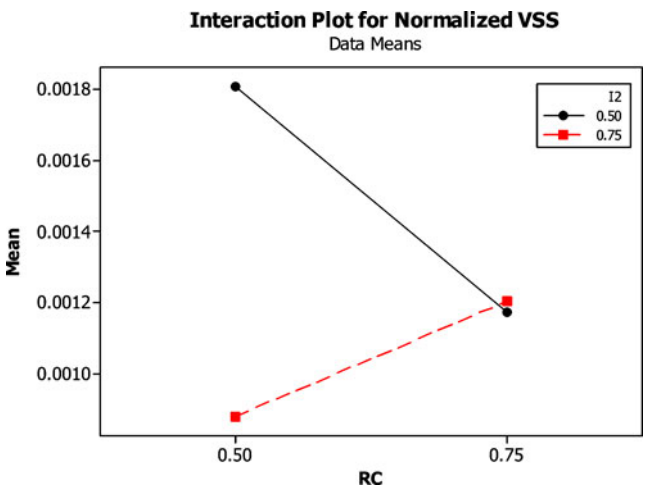

Fig. 5 Interaction plot for normalized VSS

low values, being able to avoid a delay of 4 weeks in the delivery of a product to a customer could have strong impact on the performance of a production system. Hence, the benefits of using a stochastic approach could be important.

Column Solution Time in Table 2 reports the average, minimum, and maximum time needed to solve the different instances. The results show a rather variable solution time ranging between less than 1 min and some hours. From Fig. 6, it can be noticed that the solution time decreases as $I2$ increases and it is lower when RC is low. A first explanation to this trend can be provided linking the solution time to the number of possible solutions that must be considered. In fact, as the network of activities tends to a chain, the possible solutions are less. A second explanation relies in the fact that, as the RC increases, the makespan increases and, given a larger time horizon, the number of variables in the mathematical programming problem is larger too.

Finally, it must be noticed that the instances have been solved using a commercial solver; hence, its performance should only be used as an estimation of the order of magnitude of the solution time. Ad hoc solution algorithms could be developed to reduce computational time. However, although the solution time could reach the value of some hours, the application of the approach is still viable due to the fact that production plans are updated with a frequency of one or more weeks.

5 Industrial case

To demonstrate the viability of the developed method from a practical point of view, a real industrial environment producing machining centers has been considered. A machining center is a computer numerical

controlled machine integrated with an automatic tool changer, and it often has equipment for pallet or part handling (Fig. 7).

The assembling of a machining center is a rather complex activity. A great number of components must be assembled together onto the machining center structure. Some components are the main components of the machining center, e.g., the spindle, the turning table, the controlled axes, and their activation equipment. Other components, like small mechanical components (screws, nuts, clamps, lids, and so on), or components of the pneumatic and hydraulic systems are considered ancillary. Moreover, most of the assembled components must be electrically wired.

The assembling process of the machining center is defined considering the assembling of the different groups of components. Hence, each activity to be planned represent a whole assembling phase whose execution can take days. The precedence relations among these assembling phases are defined identifying the groups of components that must have been already installed before the installation of another given group.

Even if a standard configuration for a machining center type exists, changes frequently occur to satisfy specific customer needs. Most of the machining centers are frequently specifically designed for the customers, and this is a common practice for European (and in particular Italian) machining center manufacturers. After the customized parts have been completely designed, a large set of components is assigned to external suppliers, while only high precision manufacturing activities for critical components are executed internally. At the end, all the parts and ancillary components are assembled together, tested, and then partially disassembled and delivered to the customer.

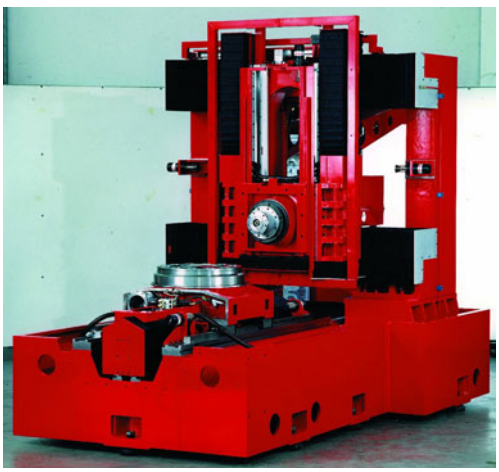

Fig. 7 Machining center structure with preassembled components installed

To reduce the lead time, it is common practice that the production of a machining center is planned or also started even if the design of the customized parts is not completed. Hence, when the production plan is devised, the resource needs for some activities could be partially unknown. This incomplete information is a significant source of uncertainty which can have serious impact on the performance of the production plan.

The stochastic programming approach developed in Section 3.1 has been applied to plan the production of a set of machining centers. A set of aggregate production activities have been considered: *structure preparation*, *structure painting*, *assembling autonomous components*, *assembling*, *wiring*, *testing*, *metrological testing*, *disassembling*, and *delivery*. The production activities for a given machining center are executed by a group of operators. The operators can be considered unspecialized, since all of them are uniformly skilled to execute all the different production activities. To model the uncertainty related to the incomplete information, a set of scenarios has been defined to represent the uncertain resource requested to execute an activity. In particular, scenarios have been defined only for the *assembling* and *wiring* activities, being the most relevant and critical activities.

The scenarios are built over historical production records through the following steps:

- A single type of machining center is considered, and all the production records for that type are taken into consideration.
- Production records for the *assembling* and *wiring* are considered to compute the total amount of working hours dedicated to these activities for different machining centers.
- A triangular continuous distribution is fitted on the data.
- For each of the two considered uncertain activities, five scenarios are generated to represent the triangular distribution. All the combination are considered using the same approach as in the previous experiments on random generated instances.

The resource availability is defined considering the average number of operators usually dedicated to the production of a single machining center. The planning horizon is about 30 weeks with the resolution of 1 week.

The stochastic programming approach is used to plan the production of a single machining center with the objective function to minimize the expected value of the makespan and considering a definite availability of the resources. This resembles the plan of the execution of a new order considering the already existing load in the production plan. First, the recourse programming

Table 3 Characteristics of the industrial instance and results

I_1	10
I_2	0.67
Res	1
RU	1
RC	0.8
n	2 (20%)
RP	24.368
EEV	24.585
VSS	0.217

model is applied to devise the production plan and the associated value of the objective function (RP). The characteristics of the industrial instance are reported in the upper part of Table 3. The bottom part of the table summarizes the results of the application of the stochastic approach and its evaluation to the industrial instance. It can be noticed that this instance is quite similar to the previous ones, except for the value of RC which is greater than the maximum value of 0.75 considered before. However, the value of VSS is in line with the previous results.

Besides this test, a comparison with a traditional planning method has been done. First, the stochastic model is solved to find the optimal production plan. The plan is used to simulate the execution of each of the considered orders. During the execution of the plan, unexpected events occurs. At this occurrence, the plan can be revised (i.e., another plan is defined taking into account the changes in the resource request of the activity). This is similar to the evaluation of the expected value plan. The differences are that the executed plan is the one obtained using the recourse programming approach, and it is evaluated on the considered orders instead of on the scenarios.

The same procedure has been applied using the traditional planning approach. In this case, first a deterministic production plan has been devised considering the expected value of the triangular distribution. Then, the execution of this production plan is simulated considering the same set of orders as before. When uncertain activities occur, the plan is modified accordingly.

Table 4 Results (industrial case)

Job order	RP	Det
J1	25	25
J2	26	26
J3	24	24
J4	18	19
J5	29	29
J6	24	24
J7	18	18
J8	17	18
J9	27	28
Average	23.11	23.33

The results of these last experiments are reported in Table 4. For each of the considered real jobs (orders), the value of the makespan in the stochastic and in deterministic approach, found as described above, are reported. The average makespan values (both stochastic and deterministic), computed overall the nine considered orders (J1, J2,..., J9), are presented in the last line of the table. It is possible to notice that using a stochastic programming approach leads to an average benefit (in terms of makespan reduction) of about 1%. This improvement is rather small, but looking at the single orders, it is possible to see that improvements are either 0 or 1 (week) that in percentage means 0% or something between 3% and 6%, depending on the instance makespan. Notice also that improvements different from 0% happen in two cases over the nine considered, i.e., in about 22.2% of the cases. The average computational time, for this set of orders, was about 60 s.

Considering the solution time and the achievable improvements (in the real context, a reduction in the makespan of 1 week can prevent the company from paying high delay penalty to the customer), the stochastic approach seems to be a better tool, with respect to the traditional mean value approach, to deal with production planning in such an intricate context.

6 Conclusions

In this paper, a two-stage stochastic programming formulation of the resource constraint project scheduling problem was proposed as an approach for production planning under uncertainty. One of the innovative issues presented was the possibility of defining a two-stage stochastic programming formulation where stages are not a priori defined in time. The stochastic programming approach was tested on randomly generated instances and on an industrial case producing machining centers.

The results on both random-generated instances and industrial case show that the explicit consideration of uncertainty through a stochastic approach can lead to a non-negligible advantage in term of plan effectiveness, with respect to a mean value approach. The side effect is represented by computational time that could be high. However, they remain affordable, considering the usual updating frequency for production plans. Moreover, ad hoc algorithms can be developed to reduce computational times.

Acknowledgements The authors thanks MCM S.p.A for their support in the definition of the real case. This research was partially funded by the "XVII Executive Programme of Scientific and Technological Cooperation between the Republic of Hungary and the Republic of Italy."

References

1. Akturk MS, Gorgulu E (1999) Match-up scheduling under a machine breakdown. Eur J Oper Res 112(1):81–97
2. Alfieri A, Tolio T, Urgo M (2011) A project scheduling approach to production planning with feeding precedence relations. Int J Prod Res 49(4):995–1020
3. Artigues C, Roubellat F (2000) A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. Eur J Oper Res 127(2): 297–316
4. Aytug H, Lawley MA, McKay K, Mohan S, Uzsoy R (2005) Executing production schedules in the face of uncertainties: a review and some future directions. Eur J Oper Res 161: 86–110
5. Birge JR, Louveaux F (1997) Introduction to stochastic programming. Springer, New York
6. Calhoun KM, Deckro RF, Moore JT, Chrissis JW, Van Hove JC (2002) Planning and re-planning in project and production scheduling. Omega 30(3):155–170
7. Demeulemeester E, Vanhoucke M, Herroelen W (2003) Rangen: a random network generator for activity-on-the-node networks. J Sched 6(1):17–38
8. ElSakkout H, Wallace M (2000) Probe backtrack search for minimal perturbation in dynamic scheduling. Constraints 5(4):359–388
9. Fernandez, A R Armacost, Pet-Edwards J (1998) A model for the resource constrained project scheduling problem with stochastic task durations. In: Proceedings of the 6th industrial engineering research conference
10. Golenko-Ginzburg D, Gonik A (1997) Stochastic network project scheduling with non-consumable limited resources. Int J Prod Econ 48(1):29–37
11. Golenko-Ginzburg D, Gonik A (1998) A heuristic for network project scheduling with random activity durations depending on the resource allocation. Int J Prod Econ 55(2):149–162
12. Herroelen W, Leus R (2004) Robust and reactive project scheduling: a review and classification of procedures. Int J Prod Res 42(8):1599–1620
13. Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165:289–306
14. Keller B, Bayraksan G (2010) Scheduling jobs sharing multiple resources under uncertainty: a stochastic programming approach. IIE Trans 42(1):16–30
15. Kis T (2005) A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. Math Program 103(3):515–539
16. Klein R (2000) Project scheduling with time-varying resource constraints. Int J Prod Res 38(16):3937–3952
17. Klugman SA, Panjer HH, Willmot GE (2004) Loss models, from data to decisions. Wiley series in probability and statistics. Wiley, New York
18. Lambrechts O, Demeulemeester E, Herroelen W (2008) Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. J Sched 11(2):121–136
19. Leus R (2003) The generation of stable project plans—complexity and exact algorithms. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium
20. Leus R, Herroelen W (2004) Stability and resource allocation in project planning. IIE Trans 36(7):667–682
21. Leus R, Wullink G, Hans E, Herroelen W (2003) A hierarchical approach to multi-project planning under uncertainty. Research report 0346, Department of Applied Economics. Katholieke Universiteit Leuven, Leuven, Belgium
22. Makris S, Chryssolouris G (2010) Customer's behaviour modelling for manufacturing planning. Int J Comput Integr Manuf 23(7):619–629
23. Márkus A, Váncza J, Kis T, Kovács A (2003) Project scheduling approach for production planning. CIRP Ann 52(1): 359–362
24. Moehring RH (2000) Scheduling under uncertainty: bounding the makespan distribution. Technical report, vol 700. Technische Universitaet Berlin
25. Mokhtari H, Aghaie A, Rahimi J, Mozdgir A (2010) Project timecost trade-off scheduling: a hybrid optimization approach. Int J Adv Manuf Technol 50:811–822
26. Monostori L, Erdős G, Kádár B, Kis T, Kovács A, Pfeiffer A, Váncza J (2010) Digital enterprise solution for integrated production planning and control. Comput Ind 61(2): 112–126
27. Neumann K, Schwindt C (1997) Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production. OR Spectrum 19(3):205–217
28. Neumann K, Schwindt C, Zimmermann J (2003) Project scheduling with time windows and scarce resources. Springer, New York
29. Stork F (2001) Stochastic resource-constrained project scheduling. Ph.D. thesis, Tecnicsche Universitaet Berlin
30. Tavares LV, Ferreira JA, Coelho JS (1999) The risk of delay of a project in terms of the morphology of its network. Eur J Oper Res 119(2):510–537
31. Tolio T, Urgo M (2007) A rolling horizon approach to plan outsourcing in manufacturing-to-order environments affected by uncertainty. CIRP Ann 56(1):487–490
32. Vanhoucke M, Coelho J, Tavares LV, Debels D (2004) On the morphological structure of a network. Faculty of Economics and Business Administration, Ghent University, Belgium
33. Vanhoucke M, Coelho J, Debels D, Maenhout B, Tavares LV (2008) An evaluation of the adequacy of project network generators with systematically sampled networks. Eur J Oper Res 187(2):511–524
34. Van de Vonder S, Demeulemeester E, Herroelen W, Leus R (2005) The use of buffers in project management: the trade-off between stability and makespan. Int J Prod Econ 97(2):227–240
35. Van de Vonder S, Demeulemeester E, Herroelen W (2007) A classification of predictive–reactive project scheduling procedures. J Sched 10(3):195–207
36. Wullink G (2005) Resource loading under uncertainty. PhD thesis, Beta Research School for Operations Management and Logistics, Enschede, The Netherlands
37. Wullink G, Gademann A, Hans E, VanHarten A (2005) Scenario-based approach for flexible resource loading under uncertainty. Int J Prod Res 42(24):5079–5098
38. Zhu G, Bard J, Yu G (2007) A two-stage stochastic programming approach for project planning with uncertain activity durations. J Sched 10(3):167–180