HEURISTIC PROCEDURES FOR PROBABILISTIC PROJECT SCHEDULING

Patrizia Beraldi [(a)], Maria Elena Bruni [(b)], Francesca Guerriero[(c)], Erika Pinto[(d)]

Dipartimento di Elettronica, Informatica e Sistemistica, University of Calabria,
Via P. Bucci 41C, 87030 Rende (Cosenza), Italy
[(a)] beraldi@unical.it, [(b)] mebruni@unical.it, [(c)]guerriero@unical.it[(c)]epinto@unical.it

ABSTRACT

In this paper we analyze the resource-constrained project scheduling problem under uncertainty. Project activities are assumed to have known deterministic renewable resource requirements and probabilistic activity durations described by random variables with a given density function. We develop heuristic algorithms for building a schedule with protected starting times, obtained using a buffering mechanism guided by probabilistic information.

Keywords: Stochastic project scheduling.

1. INTRODUCTION

The resource-constrained project scheduling problem (RCPSP) consists in minimizing the duration of a project, subject to the finish-start, zero-lag precedence constraints and the resource constraints. In its deterministic version the RCPSP, assumes complete information both on the resource usage and activities duration and determines a feasible baseline schedule, i.e. a list of activity starting times minimizing the makespan value. The role of the baseline schedule has been widely recognized in (Mehta and Uzsoy 1998; Möhring and Stork 2000), and it stems in supporting project decision makers providing a basis for planning external activities and facilitating resource allocation. Notwithstanding its importance, the planned baseline schedule in real contexts may have little, if some value, since project execution may be subject to severe uncertainty and then may undergo several types of disruptions as described in (Zhu, Bard and Yu 2005). In this paper we limit ourselves to represent uncertainty with stochastic activity durations. We shall refer to the insensitivity of planned activity start times to uncertain events as stability or solution robustness.

The stability of the program depends on what extent project managers consider uncertainty as a key features of project behaviors. In this paper we propose new heuristic procedures for generating a predictive schedule which exhibits acceptable solution robustness in the presence of multiple and frequent activity disruptions. We observe, that very few research papers explicitly consider probabilistic information in solution methods. We should mention here, the works (Van de Vonder, Demeulemeester and Herroelen 2008; Lambrechts, Demeulemeester and Herroelen 2008) for the case of uncertain resource availability. Besides this, our approach differs from the cited paper in some important aspects. In our paper we consider the stochastic programming framework and, in particular, the probabilistic paradigm in the form of joint probabilistic constraints. At the best of our knowledge none of the methods proposed in the literature consider joint probabilistic constraints. Even with some limitation, our scheduling approach can be tailored to reflect the level of risk that an individual decision maker is willing to bear in hedging against processing time uncertainty.

The remainder of the paper is organized as follows. In Section 2, we present a review of the relevant literature on project scheduling under uncertainty. In Section 3 we describe our scheduling methodology for generating robust baseline schedules. Section 4 is devoted to the presentation of the benchmark heuristics used to assess the efficacy of the newly developed heuristics and of the design of computational experiments. Results are analyzed in Section 5 and conclusions are presented in Section 6.

2. RELATED WORK

The methodologies for stochastic project scheduling basically view the project scheduling problem as a multistage decision process. Since the problem is rather involved and an optimal solution is unlikely to be found, scheduling policies are used for defining which activities to start at random decision points through time, based on the observed past and the a priori knowledge about the processing time distributions. (Igelmun and Radermacher 1983), propose a set of preselective scheduling policies. These policies, roughly speaking, define, for each possible resource conflict, a preselected activity that is postponed from a set of activities that cannot be executed together due to resource conflicts.

A branch-and-bound algorithm is developed in order to compute optimal preselective policies, and computational tests are reported for small instances. (Möhring and Stork 2000), introduce a new class of scheduling policies, called linear preselective policies intended to minimize the makespan for the RCPSP with stochastic activity durations. This new class combines the benefits of preselective policies and priority policies and is based on both determining sets of activities that cannot be executed together due to resource conflicts and on choosing an activity to be postponed according to a priority list. (Stork 2000), compares four scheduling policies for minimizing the makespan for the RCPSP with stochastic activity durations. Dominance rules and lower bounds are developed and then embedded into a branch-and-bound algorithm.

Research on heuristic procedures for solving the stochastic RCPSP is an active field of research. (Tsai and Gemmill 1998), propose a tabu search which makes use of a reduced neighborhood based on feasible swaps that can be executed on the current feasible sequence.

Given a feasible schedule, the makespan is computed by sampling repeatedly activity durations in order to obtainan estimate of the expected makespan. (Golenko- Ginzburg and Gonik 1997), develop a heuristic procedure operating in stages, where the decision to schedule the next activity is based on the precedence constraints and current resource availability.

A multiple knapsack problem minimizing expected project duration is proposed to solve resource conflicts. In a follow-up work, (Golenko-Ginzburg and Gonik 1998), apply a similar heuristic for a project scheduling problem where the duration of an activity is a random variable that depends on the amount of resources assigned to that activity. (Herroelen and Leus 2004), develop mathematical programming models for the generation of stable baseline schedules in a project environment without resource consumption.

They minimize the expected weighted sum of the absolute deviations between the planned and the actually realized activity starting times when exactly one activity duration disruption is expected to take place during project. (Tavares, Ferreira and Coelho 1998), study the risk of a project as a function of the uncertainty of the duration and the cost of each activity.

The authors make use of a buffering mechanism to increase the earliest activity start times. In (Rabbani, Fatemi, Ghomi, Jolai and Lahiji 2007), a newly developed resource-constrained project scheduling method in stochastic networks is presented which merges the critical chain concepts with traditional resource-constrained project scheduling methods.

The objective function takes into account the expected project duration and its variance. Since the developed model is a stochastic optimization model which cannot be solved in the general case, this paper suggests a heuristic algorithm where ready activities at each decision point are supplied by available resources on the basis of assigned priority level. These priority levels are the activities contribution in reducing the expected project duration and its variance. Therefore, the activities with the greatest probability to be on the critical chain and the greatest correlation with the project variance are fed-in first.

When resource availability constraints are considered, (Leus and Herroelen 2004), assuming the availability of a feasible baseline schedule, present exact and approximate formulations of the robust resource allocation problem, proposing for its solution a branch and-bound algorithm. The so-called resource flow network (Artigues and Roubellat 2000) is used to represent the flow of resources across the activities of the project network. In (Deblaere, Demeulemeester, Herroelen and Van de Vonder 2007) is presented a procedure for allocating resources to the activities of a given baseline schedule in order to maximize its stability in the presence of activity duration variability. The authors propose three integer programming based heuristics and one constructive procedure for resource allocation, thereby avoiding the use of stochastic variables. In (Van De Vonder, Demeulemeester, Herroelen and Leus 2006; Van de Vonder, Demeulemeester, Herroelen and Roel Leus 2005) a modi_cation of the ADFF heuristic presented in (Herroelen and Leus 2004; Herroelen and Leus 2004) is proposed in order to prohibit resource conflicts. In order to obtain a precedence and resource-feasible schedule, the resource flow-dependent float factor (RFDFF) heuristic uses information coming from the resource flow network in the calculation of the so called activity dependent oat factor. In (Van de Vonder, Demeulemeester and Herroelen 2008), multiple algorithms are introduced to include time buffers in a given schedule while a prede_ned project due date remains respected. While the virtual activity duration extension heuristic presented in (Van de Vonder, Demeulemeester and Herroelen 2008), relies on the standard deviation of the duration of an activity in order to compute a modified duration, the starting time criticality (STC) heuristic tries to combine information on activity weights and on the probability that activity cannot be started at its scheduled starting time.

Within the stochastic programming context, a two-stage integer linear stochastic model is proposed in (Zhu, Bard and Yu 2007) to determine target times in the first stage followed by the development of a detailed project schedule in the second stage with the aim of minimizing the cost of project completion and expected penalty incurred by deviating from the specified values.

Temporal protection is used against machine failure in (Gao 1995). The durations of activities requiring resources prone to breakdown are extended to provide extra time for protection. The protection equals the original duration augmented with the duration of breakdowns that are expected to occur during activity execution, based on breakdown statistics.

In (Lambrect, Demeleumester, Herroelen 2008) the case of uncertain resource availability due to a breakdown is tackled. The objective is to build a robust schedule that meets the project deadline and minimizes the schedule instability cost. In the paper it is shown that protection of the baseline schedule may provide significant performance gains over the use of deterministic scheduling approaches.

For an extensive review of research in this field, the reader is referred to (Herroelen and Leus 2004; Herroelen and Leus 2005).

3. SCHEDULING ACTIVITIES EXPLOITING PROBABILISTIC INFORMATION

The deterministic RCPS may be stated as follows. Let consider a project represented by a directed acyclic graph G = (N,V) and opt for activity on the node format (Wiest and Levy 1977). Each node in the set N corresponds to a single project activity and each arc in the set V corresponds to a precedence relation between each pair of activities. Each activity $j \in$ N, has to be processed without interruptions requiring a constant amount of resource r_{jk}, for each renewable resource type k, k = 1..., K. Each renewable resource is assumed to have a constant per period availability equal to a_k.

Let $S_t \subseteq$ V be the set of activities that are in progress at time period t. The RCPSP can be formulated as:

$$\min s_n$$

$$s_j \geq s_i + d_i \quad (i, j) \quad V \tag{1}$$

$$\sum_{i:i \in S_t} r_{ik} \geq \alpha_k$$

where d_i is the deterministic duration of activity i and s_i the planned starting time of activity i. Two kinds of constraints subsist among activities: precedence feasibility constraints which force activity j to be started only when all its immediate predecessor activities have been processed, and renewable resource constraints which prevent activities to exceed limited capacities of resources. The problem becomes even more involved in the stochastic RCPSP, where the durations of activities are not known in advance, but are instead represented by random variables $_j$ j N with known cumulative probability distribution function .

Given the uncertainty in activity duration, the decision maker may be inclined to solve several deterministic programs involving different values of the uncertain problem parameters or to replace the random variables by their expected value. As a matter of fact, the well known PERT model replaces randomness with a certainty equivalent in the form of expected value. However, none of these approaches can be considered satisfactory in face of uncertainty.

$$\min s_n \tag{2}$$

$$P(s_j \geq s_i + \quad) \qquad (i, j) \quad V \tag{3}$$

$$\sum_{i:i \in} \quad \alpha_k \tag{4}$$

We observe that although the chance constraints paradigm has been tacitly accepted by the research community for over 30 years, recently, its validity as a tool for a point estimate of the project makespan or for the estimation of the complete cumulative density function of the makespan has been questioned. It is also well recognized that the chance constrained approach fails to give reasonable hints about the criticality of a path or activities, thereby destructuring critical chain approaches.

Indeed, the value taken by the starting time of an activity j, s_j is a function of a random variable and a decision variable s_i which is itself stochastic, its value depending on the starting times of preceding activities.

The appropriateness of a static chance constrained model, should at least be questioned. Nevertheless, a dynamic formulation in the form of a multistage recourse programming problem, may involve a huge number of scenarios, overwhelming form a computational point of view. These motivations are at the basis of the simplification of the problem on which our heuristics rely, based on a decoupling of the dynamic aspect of the problem from its probabilistic nature. In particular, the solution of the problem is viewed as a bilevel hierarchical process, where the temporal dependence is treated on the first level, whereas, stochasticity is introduced in a second level.

In the foregoing we shall present two different heuristics based on the general principle of joint probabilistic constraints.

3.1. The Joint Probabilistic Constraints heuristic

The Joint Probabilistic Constraints heuristic (JPCH) is inspired by heuristic algorithms for the deterministic RCPSP, but embeds the joint probabilistic constraints paradigm in its scheme. The schedule is constructed in two phases. In the first phase, a precedence feasible priority list is constructed following an ordering criterion. In the second phase, this priority list is transformed into a precedence and resource feasible schedule sequentially adding activities to the schedule until a feasible complete schedule is obtained.

In each step, g, or decision point t_g, the activities in the priority list, that are also part of the set E_g, containing all eligible activities which can be precedence and resource feasibly started at t_g, with better ranking are selected to be started at t_g and inserted in S_g. The set of already scheduled activities is denoted by A_g.

An algorithmic description of the Joint Probabilistic Constraints heuristic is given below:

- Initialization $g = 0$; $t_g = 0$; $A_0 = \{\emptyset\}$, $S_0 = \{\emptyset\}$.
- While $|A_g \cup S_g| \leq |N|$ do
  $g{:}=g+1$;
  $t_g = \min_{j \in A_g} c_j$;
- Calculate the residual resource availability and the set A_g and E_g.
- Use the priority rule to select the activities to be included in the set $S_g \subseteq E_g$.

- Calculate the completion times of activities in S_g.
- end while

Let us analyze in greater detail the problem of determining the completion times c_j of the set S_g of activities scheduled at decision point t_g, i.e., the starting times of activities to be scheduled at next decision point t_{g+1}. At each iteration g, the completion times are the solution of the following problem:

$$\min C \tag{5}$$
$$C \geq c_j \quad \forall j \in S_g \tag{6}$$
$$P\,(c_j \geq t_g + \hat{d}_j \quad \forall j \in S_g\ ) \geq \alpha \tag{7}$$

where C is the next decision point t_{g+1}. In problem (5-7) the probabilistic constraints are jointly imposed on all the activities in S_g. This ensure that the probability of disrupting the starting time of successor activities is kept above the prescribed probability level α. As evident, this time buffering mechanism is used with the aim of absorbing potential disruptions caused by activity shifts. Buffer sizes are computed on the basis of the joint probability that activities in S_g disrupt subsequent activities. We observe that not considering joint probabilistic constraints would lead, at decision point t_g, to a probability disruption equal to the sum of the individual probability disruptions of activities in S_g.

The proposed heuristic, on the contrary, imposes a joint probability of disruption α for all the activities in S_g. We further remark that, considering the expected duration increase of activities i would result in disruption probability for each activity of at least 50% which can be unacceptable in some contexts.

3.2 The resource allocation heuristic

Rather than using a priority rule for deciding the set of activities to be included in S_g the resource allocation heuristic (RAH) determines at each decision point t_g, both the starting times and the resources allocated to each activity. Activities are not ordered in a list and if E_g contains more than one element a competition has to be arranged to choose the optimal subset of activities S_g that can be supplied by available resources. Decisions, at each t_g are made on the basis of the solution of the following subproblem.

$$\max \textstyle\sum_{i \in E_g} \beta_i - \gamma * C$$

$$C \geq c_i \; \forall\, i \in E_g \mid \beta_i = 1$$

$$P(c_i \geq t_g + \hat{d}_i \; \forall (i) \in E_g \mid \beta_i = 1) \geq \alpha \tag{8}$$

$$\textstyle\sum_{i: i \in E_g} r_{ik} * \beta_i \geq r_k(t_g) \; \forall k$$

where $r_k(t_g)$ is the residual resource availability at time t_g. The objective function tries to balance two conflicting objectives, the resource and the time allocation decisions. The two objective functions are weighted through the parameter γ, which is adjusted dynamically in order to find a good balance between conflicting objectives.

An algorithmic description of the resource allocation heuristic is given below:

- Initialization $g = 0$; $t_g = 0$; $A_0 = \{\emptyset\}$, $S_0 = \{\emptyset\}$.
- While $|A_g \cup S_g| \leq |N|$ do
- g:=g+1.
- $t_g = \min_{j \in A_g} c_j$.
- Calculate E_g, and update the residual resource availability and the set A_g. Set β = 0:01 $S_g = \{\emptyset\}$
- While no activities have been selected ($S_g = \{\emptyset\}$):
  - Solve the competition problem (8). Let β*,C*, c* be the optimal solution.
  - If $\beta_i^* = 0$; $\forall\, i \in E_g$ β:= β/10.
  - Otherwise, $S_g := \{i \in E_g \mid \beta_i^* = 1\}$.
  - Set the completion times of activities in S_g $c_i := c_i^*$
- end while

Decision variables of problem (8) are the chosen activities to be supplied by resources and the resource capacities assigned to those activities. The choice of the activities to be supplied with available resources at each decision point should also reduce the remaining projects duration as much as possible. Therefore, whilst the first term of the objective function tries to maximize the resource consumption at each decision point, the second term aims at reducing the partial makespan. Problem (8) has to be solved at each decision point, when at least more than one activity is ready to be operated and the residual available amount of resources is not zero.

3.3 Remark

Both the joint probabilistic constraints and the resource allocation heuristics involve the solution of a model under joint probabilistic constraints. In the following, we show how to transform them, in the case of independent random variables, into deterministic equivalent problems.

With this aim, let consider the following problem:

$$P(\hat{a}_i^T x \leq \hat{b}_i \; \forall\, i = 1,\ldots,m) \tag{9}$$

where the probabilistic constraints are jointly imposed on m separate constraints involving the random variables $_i$. Under the independence assumption among the random variables $_i$, the probabilistic constraints (9) can be rewritten as

$$\prod_{i=1,\dots,m} P(a_i^T x \le$$

and equivalently, denoting with F_i the marginal probability distribution function of the continuous random variable b_i, as

$$\prod_{i=1,\dots,m} F_i (a$$

By taking logarithm we can rewrite the constraints as follows:

$$\sum_{i=1,\dots,m} \ln F_i (c_i^T x) \ge \ln \alpha.$$

Since the logarithm is an increasing function and $0 < F_i \le 1$, this transformation is legitimate.

Furthermore, for log-concave distribution functions, convexity of the constraints is preserved. Fortunately, the class of log-concave random variables includes several commonly used continuous, univariate probability distributions as for example the Uniform, Normal, Exponential, Beta, Weibull, Gamma, Pareto, and Gompertz distributions. We observe that also in the case of discrete distributions, problems with joint probabilistic constraint can be reduced to deterministic equivalent problems that we shall report hereafter, for the sake of completeness. Let us introduce an integer vector z_i, whose entries are defined as $z_i = a_i^T x \quad i = 1\dots,m$. In the case of log-concave marginal distribution, it is possible to rewrite z_i in a 0 - 1 formulation. If $l_i + k_i$ is a known upper bound, where l_i represents the α-fractile of the distribution function F_i, z_i can be written as

$$z_i = l_i + \sum_{k=1\dots K} z_{ik}$$

and the probabilistic constraints as

$$\sum_{i=1,\dots,m} \sum_{k=1\dots k_i} a_{ik} z_{ik}$$

where

$$= \ln F_i(l_i + k) - \ln F_i(l_i + k - 1)$$

and

$$= \ln \quad - \ln F(l).$$

4. COMPUTATIONAL EXPERIMENTS

In this section we shall present the results of the computational study carried out with the aim of assessing the performances of the algorithms introduced in the previous section. We have compared our algorithm with the approaches found in the literature closer to our work (the STC and the RFDFF heuristics) and with a set of newly created benchmark heuristics, which rely on the use of separate chance constraints.

The computational experiments have been carried out on a set of benchmark problems with 30 and 60 activities randomly selected from the project scheduling problem library PSPLIB (Kolisch and Sprecher 1997).

We have replaced the deterministic duration of each activity by two probability distributions, one continuous and one discrete, taking d_i as their mean.

In particular, we have tested the Uniform distribution $U(0.75d_i;\ 2.85d_i)$ and the Poisson distribution with mean d_i and activity durations are assumed to be independent.

Extensive simulation has been used to evaluate all procedures on robustness measures and computational efficiency. For every network instance, 1000 scenarios have been simulated by drawing different actual activity durations from the described distribution functions.

Using these simulated activity durations, the realized schedule is constructed by applying the following reactive procedure. An activity list is obtained by ordering the activities in increasing order of their starting times in the proactive schedule. Ties are broken by increasing activity number. Relying on this activity list, a parallel schedule generation scheme builds a schedule based on the actual activity durations. We opted for railway execution never starting activities earlier than their prescheduled start time in the baseline schedule. Effectively, this type of constraint is inherent to course scheduling, sports timetabling and railway and airline scheduling, or when activity execution cannot start before the necessary resources have been delivered.

4.1 Separate chance-constraints based heuristics

Schedule generation schemes are the core of most heuristic solution procedures for the deterministic RCPSP. The best-known Schedule generation scheme order all activities according to a priority list and at every decision point select the next activities to start based on this priority list. Mainly two types of schedule generation scheme are used to build a schedule from an activity list, the Parallel Schedule generation scheme and the serial schedule generation scheme. These schedule generation schemes construct a precedence-resource feasible sequence through a stepwise increase of a partial schedule. The serial method at each step, selects an activity from the set of activities eligible for scheduling and assigns to it the earliest possible start time according with both precedence and resource constraints. Selection of activities from the set of activities eligible for scheduling is performed according to a priority rule. The parallel method is based on a time incrementation procedure since at every decision time (t = 0 and the completion times of activities), it starts as many unscheduled activities as possible in accordance with the precedence and resource constraints selecting activities according to a priority rule. We use activity-based priority policies as benchmark heuristics, considering, instead of the deterministic durations, the α-fractiles of activities. We shall refer to these heuristics in the following as Parallel Separate chance-constraints based heuristics (PSCCBH) e Serial Separate chance-constraints based heuristics (SSCCBH).

Regardless the schedule generation scheme applied, the resulting schedule depends on the ordering criterion adopted. The following static priority rules for generating the priority list have been tested in the computational experiments (Kolisch and Hartmann 1999). Some of them have been proposed by the authors.

- The MinC rule orders the activities by increasing value of their resource requirement.
- The MinD rule orders the activities by increasing value of their α-fractile.
- The MaxC rule orders the activities by decreasing value of their resource requirement.
- The MaxD*C rule orders the activities Job ordered by decreasing value of their alfa-duration*resource requirement.
- The LST rule(Kolisch, Sprecher and Drexl 1995) orders the activities by increasing value of their latest starting time.
- The LFT (Davis and Patterson 1975) orders the activities by increasing value of their latest finish time.
- The MTS (Alvarez-Valdes and Tamarit 1989) orders the activities by decreasing value of the number of their successors.

5. ANALYSIS OF THE RESULTS

A total of 13 scheduling procedures are evaluated. Algorithms 1-4 are the PSCCBH whereas algorithms 5-8 are the SSCCBH with the first four priority rule listed in Section 4.1. Algorithms from 9 to 11 are the PSCCBH with the rules LST, LFT and MTS respectively, whereas Algorithm 12 is a SSCCBH with the LST priority rule. The JPCH Section 3.1 is executed considering the four priority rules (JPCH 1{4). All the algorithms, but the RFDFF and the STC heuristics, have been tested for 5 α values {0:8; 0:85; 0:9; 0:95; 0:99}.

The computational experiments were performed in a PC Pentium III, 667 MHz, 256 MB RAM. All procedures were coded in AIMMS language (Bisschop and Roelofs 2007) and the subproblems solved woth Cplex 10.1 and Conopt. We show in Tables 1-4 the average results calculated over all networks and executions for each approach. The complete set of the numerical results are fully reported in (Beraldi, Bruni, Guerriero and Pinto 2007). The quality of the algorithm on every problem instance has been evaluated by the following measures: average tardiness (Tavg) over all networks and executions, average timely project completion probability (TPCP) over all networks and executions, average number of jobs over all networks and executions, whose starting time in the actual schedule differs from the baseline schedule (delayed) and CPU time in seconds (time).

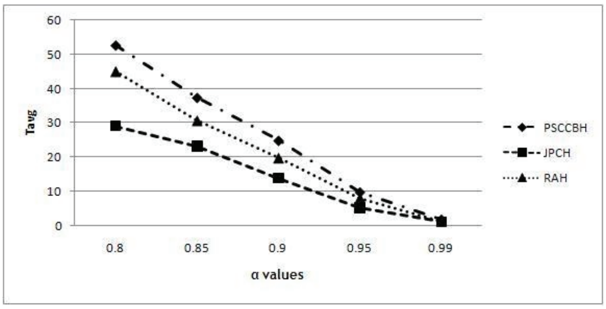

Figure 1: Tardiness versus α values

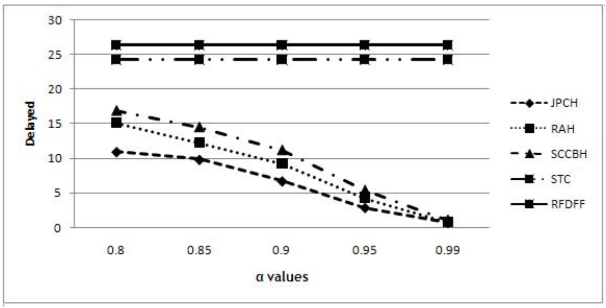

Figure 2: Delayed versus α values

For the sake of clarity, although RFDFF and STC heuristics construct exactly the same schedule whatever the risk averseness of the decision maker, we have reported the results of the RFDFF and STC heuristics for all the probability levels tested. As could be expected,

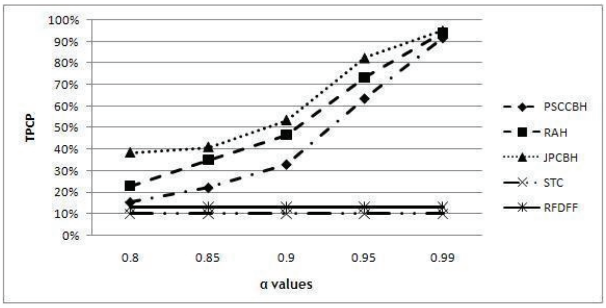

Figure 3: TPCP versus α values

proactive scheduling procedures, which use probabilistic information, always seem to outperform procedures that do not. It can be easily observed that even the simple benchmark heuristics 1-12 perform better than STC and RFDFF. In this respect, a note of caution is in order. The performances of the STC and RFDFF heuristics may have been biased by the high disruption probability associated with each activity. We have experimentally observed that, in the 1000 project simulations, almost all the activities showed a duration higher than their expected value, denoting a highly variable environment. This high variability is confirmed by the TPCP which is, as evident, on average very low.

JPCH 1-4 rank best among the heuristics. The gain in performances is more evident for increasing _ values, reflecting a more conservative strategy hedging against more disruptions scenarios. The performances of the algorithms 1-12 are clearly indistinguishable, and depend on the ordering criterion adopted.

Unfortunately, there is no unitary evidence of one criterion over the others. For these heuristics, the number of activities whose actual starting time exceed the planned starting time (reported in the column delayed) is quite satisfactory, especially for increasing α values. In particular, their performances are comparable with those of JPCH 1-4 for very high probability levels.

The RAH performs better than algorithms 1-12 but shows worse performances than JPCH 1-4.

With the aim of assessing the variation of the performance measures with respect to the probability level α we show in Figures 1, 2 and 3 the tardiness, the number of activities delayed and the timely project completion probability for different α values, for the same test problems. The average performance over the twelve separate chance constraints based heuristics has been considered for comparison. The tardiness of the STC and RFDFF heuristics has not been reported, since it was very high. As evident, the schedule performances deteriorate with decreasing α values. Similar behavior has been observed for all the other tests. This result is expected, since there is clearly a correlation between the schedule robustness and the values of α

Algorithms from 1 to 12 are computationally very cheap. This is due to the simple schedule construction procedures based on the parallel and serial schedule generation scheme. We observe that RFDFF and STC are slower due to the fact that the procedures have to be executed a number of times, until a creation of a deadline feasible schedule is allowed.

The computational time of JPCH 1-4 and of procedures STC and RFDFF is comparable for most instances.

However, we should observe that for larger networks JPCH 1-4 show worse computational times than STC and RFDFF heuristics. This is due to the extra effort required for solving, at each decision point, the probabilistic model. We notice, in addition, that depending on the nature of the probabilistic model to be solved (nonlinear continuous or linear integer), a different computational effort is required. In our experiments, it is evident that more CPU time is required to solve the integer deterministic equivalent problem related to discrete random variables. We observe also that for the continuous distribution function considered, the RAH is overwhelming from a computational point of view (no results are reported in Tables 3 and 4.

This is due to the mixed integer nonlinear nature of the problem to be solved at each iteration of the RAH.

As a final remark, we observe that our heuristic does not consider, in its present form, possible weights associated to activities representing the marginal cost of starting the activity earlier or later than planned in the baseline schedule. We remark, that the ordering criterion could be adapted to reflect the relative importance of the activities. Nevertheless, there is no absolute consensus on how to estimate such costs.

Table 1: Results on 30 nodes test problems with discrete duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	Tavg	TcP	delayed	time	Tavg	TcP	delayed	time	Tavg	TcP	delayed	time	Tavg	TcP	delayed	time	Tavg	oMn	delayed	time
1	52,12	0,17	17,09	0,05	37,89	0,20	14,83	0,06	26,55	0,30	11,82	0,04	9,78	0,68	5,40	0,05	1,69	0,93	1,10	0,05
2	53,04	0,15	16,74	0,05	36,47	0,20	14,34	0,06	22,49	0,36	10,64	0,05	8,98	0,63	5,17	0,05	1,66	0,91	1,12	0,06
3	58,04	0,14	17,53	0,05	38,73	0,19	14,64	0,07	28,11	0,30	12,21	0,05	10,07	0,62	5,61	0,06	1,73	0,91	1,14	0,06
4	51,37	0,15	16,90	0,05	37,89	0,23	14,83	0,06	24,38	0,35	11,26	0,05	8,80	0,68	5,04	0,07	1,69	0,92	1,16	0,06
5	53,38	0,16	16,79	0,23	37,19	0,25	14,38	0,24	25,82	0,36	11,15	0,22	10,74	0,62	5,81	0,24	1,65	0,94	1,07	0,24
6	57,19	0,15	17,30	0,23	37,85	0,23	14,36	0,26	25,80	0,33	11,25	0,21	10,96	0,59	5,79	0,26	1,90	0,91	1,19	0,28
7	50,39	0,17	16,60	0,26	37,13	0,25	14,02	0,29	25,20	0,35	11,06	0,23	9,09	0,67	4,82	0,27	1,87	0,91	1,19	0,27
8	53,63	0,17	16,88	0,17	38,42	0,23	14,52	0,27	25,10	0,37	11,01	0,21	10,00	0,64	5,34	0,25	1,72	0,92	1,12	0,25
9	52,10	0,14	17,13	0,05	35,51	0,24	14,47	0,04	23,38	0,30	11,09	0,04	9,80	0,59	5,82	0,04	1,64	0,91	1,11	0,08
10	50,62	0,15	16,99	0,05	36,45	0,22	14,61	0,05	23,78	0,33	11,21	0,04	9,56	0,65	5,35	0,04	1,52	0,94	1,03	0,09
11	50,37	0,14	16,97	0,05	38,16	0,18	15,34	0,04	24,18	0,28	11,31	0,05	9,30	0,62	5,49	0,05	1,61	0,92	1,11	0,12
12	50,20	0,14	16,64	0,31	36,58	0,22	14,43	0,31	22,70	0,31	11,03	0,31	9,21	0,63	5,50	0,30	1,54	0,91	1,08	0,40
13	45,05	0,23	15,16	0,41	30,70	0,35	12,32	0,35	19,75	0,47	9,29	0,35	8,06	0,74	4,34	0,35	1,22	0,94	0,79	0,35
JPCH-1	29,94	0,46	10,87	0,41	17,32	0,59	8,07	0,37	12,23	0,58	5,68	0,37	5,09	0,86	2,73	0,37	1,11	0,95	0,72	0,37
JPCH-2	32,33	0,28	11,95	0,41	28,58	0,32	11,70	0,37	17,80	0,41	8,78	0,37	5,45	0,81	3,04	0,37	1,23	0,95	0,90	0,37
JPCH-3	27,68	0,44	10,99	0,41	22,37	0,45	9,51	0,37	12,52	0,64	6,29	0,37	4,93	0,81	2,88	0,37	1,10	0,96	0,71	0,37
JPCH-4	26,30	0,35	9,97	0,41	24,25	0,27	10,01	0,37	12,87	0,52	6,14	0,37	5,31	0,82	2,95	0,37	1,00	0,95	0,68	0,37
STC	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41
RDFF	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41

Table 2: Results on 60 nodes test problems with discrete duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time
1	170,64	0,01	41,3	0,17	126,35	0,02	41,4	0,16	73,51	0,12	33,0	0,15	33,38	0,43	19,0	0,156	5,80	0,85	4,1	0,16
2	154,63	0,01	39,2	0,17	114,36	0,03	39,6	0,17	73,05	0,14	31,9	0,15	29,63	0,45	16,5	0,174	5,76	0,87	3,9	0,16
3	165,49	0,01	40,0	0,17	127,90	0,03	41,2	0,18	79,37	0,12	33,4	0,17	35,35	0,38	19,6	0,161	5,92	0,84	4,1	0,16
4	156,20	0,01	39,7	0,18	120,15	0,04	40,6	0,18	71,62	0,15	31,6	0,17	31,39	0,43	17,4	0,156	5,35	0,87	3,7	0,17
5	150,72	0,02	37,4	0,76	107,25	0,09	36,6	0,75	71,33	0,18	29,7	0,78	31,10	0,46	16,6	0,723	5,24	0,88	3,4	0,74
6	155,62	0,02	38,1	0,61	114,95	0,07	38,4	0,62	69,17	0,20	29,3	0,59	30,46	0,48	15,6	0,575	5,06	0,88	3,3	0,59
7	148,88	0,03	37,7	0,82	111,81	0,08	37,5	0,82	68,06	0,15	29,5	0,80	29,77	0,49	16,0	0,736	5,50	0,87	3,5	0,75
8	141,47	0,04	37,1	0,61	107,49	0,08	36,6	0,64	67,81	0,20	29,0	0,63	28,03	0,51	15,0	0,623	5,63	0,87	3,7	0,62
9	161,14	0,01	44,4	0,12	120,34	0,02	40,8	0,11	77,18	0,08	34,0	0,10	32,02	0,47	18,5	0,103	6,33	0,84	4,4	0,10
10	160,62	0,01	44,6	0,12	118,85	0,02	41,0	0,10	76,95	0,08	34,1	0,12	31,87	0,41	18,1	0,108	6,11	0,83	4,3	0,11
11	160,01	0,00	44,4	0,14	119,13	0,03	40,9	0,11	72,05	0,09	32,8	0,12	30,96	0,35	18,3	0,111	5,46	0,86	4,0	0,10
12	151,12	0,01	43,5	0,74	107,12	0,04	38,5	0,66	72,42	0,12	32,3	0,72	28,81	0,51	16,5	0,641	5,34	0,86	3,8	0,67
13	115,80	0,10	37,0	2,29	74,61	0,31	30,3	2,29	41,74	0,51	20,8	2,29	24,41	0,63	13,5	2,286	3,93	0,90	2,7	2,29
20	32,64	0,40	10,9	2,40	26,79	0,47	8,9	2,40	22,18	0,62	6,9	2,40	31,25	0,84	8,8	2,403	30,20	0,97	7,9	2,40
21	33,96	0,41	11,8	2,40	37,08	0,46	12,1	2,40	38,65	0,62	11,9	2,40	41,02	0,81	11,2	2,403	29,52	0,96	7,7	2,40
22	27,02	0,40	9,9	2,40	32,45	0,57	10,9	2,40	25,80	0,63	7,4	2,40	28,82	0,80	8,4	2,403	26,53	0,97	7,4	2,40
23	22,86	0,38	9,0	2,40	25,37	0,56	9,3	2,40	21,56	0,68	7,1	2,40	21,50	0,82	6,8	2,403	26,72	0,97	7,0	2,40
STC	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,286	385,82	0,00	52,3	2,29
RDFF	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,403	193,29	0,03	58,5	2,40

Table 3: Results on 30 nodes test problems with continuous duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	time	Tavg	TcP	jobrit	ex Norob
1	27,99	0,27	13,04	0,04	14,07	0,44	9,27	0,04	7,11	0,59	6,24	0,06	2,59	0,84	2,48	0,04	0,00	1,00	0,00	0,04
2	27,74	0,26	12,81	0,05	13,54	0,45	9,02	0,05	7,57	0,62	6,45	0,05	2,65	0,84	2,50	0,04	0,00	1,00	0,00	0,04
3	32,14	0,23	14,26	0,09	17,12	0,36	10,98	0,05	8,20	0,56	7,00	0,06	2,51	0,85	2,37	0,05	0,00	1,00	0,00	0,04
4	26,57	0,28	12,36	0,04	13,99	0,41	9,39	0,06	6,81	0,60	5,93	0,05	2,34	0,85	2,22	0,05	0,00	1,00	0,00	0,05
5	30,39	0,25	13,55	0,21	14,96	0,41	9,70	0,21	7,18	0,60	6,18	0,21	2,40	0,84	2,28	0,19	0,00	1,00	0,00	0,19
6	30,42	0,25	13,63	0,18	13,96	0,44	9,39	0,20	8,05	0,58	6,86	0,20	2,45	0,84	2,34	0,17	0,00	1,00	0,00	0,18
7	29,93	0,25	13,70	0,19	15,60	0,38	10,09	0,23	8,52	0,52	7,18	0,20	2,18	0,86	2,11	0,19	0,00	1,00	0,00	0,20
8	28,82	0,25	12,90	0,17	14,28	0,41	9,42	0,19	8,06	0,56	6,82	0,19	2,38	0,86	2,26	0,17	0,00	1,00	0,00	0,18
9	27,32	0,24	12,98	0,04	14,58	0,38	9,56	0,05	7,34	0,59	6,35	0,05	2,22	0,88	2,13	0,04	0,00	1,00	0,00	0,05
10	26,22	0,29	12,74	0,05	14,20	0,42	9,53	0,05	7,56	0,58	6,52	0,05	2,30	0,88	2,19	0,04	0,00	1,00	0,00	0,05
11	27,59	0,26	13,20	0,05	14,57	0,38	9,68	0,04	7,39	0,60	6,40	0,05	2,40	0,82	2,30	0,04	0,00	1,00	0,00	0,05
12	26,63	0,26	12,88	0,25	13,65	0,40	9,11	0,26	7,36	0,64	6,37	0,26	2,25	0,86	2,16	0,25	0,00	1,00	0,00	0,26
20	12,09	0,85	5,83	0,15	3,26	0,90	2,33	0,16	1,66	0,94	1,42	0,15	0,55	0,97	0,47	0,16	0,00	1,00	0,00	0,15
21	26,92	0,69	12,04	0,37	14,70	0,74	8,36	0,37	7,30	0,85	5,35	0,38	1,77	0,94	1,53	0,37	0,00	1,00	0,00	0,36
22	22,31	0,71	9,78	0,23	9,90	0,83	6,47	0,23	5,77	0,88	4,47	0,23	1,58	0,95	1,20	0,24	0,00	1,00	0,00	0,24
23	28,00	0,64	11,64	0,20	15,57	0,77	8,74	0,22	6,41	0,87	4,82	0,19	1,73	0,93	1,42	0,20	264,41	1,00	0,00	0,22
STC	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21
RDFF	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58

Table 4: Results on 60 nodes test problems with continuous duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	Tavg	TPCP	jobrit	CPU	Tavg	TPCP	jobrit	CPU	Tavg	TPCP	jobrit	CPU	Tavg	TPCP	jobrit	CPU	Tavg	TPCP	jobrit	CPU
1	96,71	0,04	38,1	0,11	53,55	0,18	30,7	0,10	25,82	0,40	21,1	0,11	8,29	0,69	7,8	0,10	0,00	1,00	0,0	0,11
2	95,43	0,03	37,1	0,11	49,51	0,20	28,3	0,13	23,00	0,40	18,8	0,11	7,12	0,73	6,7	0,13	0,00	1,00	0,0	0,11
3	103,35	0,05	39,3	0,12	52,45	0,16	30,0	0,11	24,25	0,38	19,9	0,11	8,44	0,74	8,0	0,11	0,00	1,00	0,0	0,12
4	85,00	0,08	35,2	0,12	49,61	0,20	28,2	0,11	26,21	0,40	20,9	0,12	8,02	0,73	7,4	0,11	0,00	1,00	0,0	0,11
5	96,06	0,06	36,4	0,50	47,36	0,22	26,9	0,50	22,02	0,43	17,9	0,50	7,82	0,75	7,3	0,50	0,00	1,00	0,0	0,50
6	94,35	0,06	36,2	0,38	47,20	0,21	26,6	0,39	23,62	0,41	19,0	0,39	7,22	0,76	6,7	0,39	0,00	1,00	0,0	0,40
7	96,90	0,07	37,5	0,53	50,21	0,19	28,7	0,54	23,53	0,42	19,4	0,51	7,71	0,75	7,2	0,54	0,00	1,00	0,0	0,50
8	92,49	0,07	35,1	0,40	46,32	0,21	26,8	0,40	24,69	0,43	19,7	0,40	6,47	0,78	6,0	0,40	0,00	1,00	0,0	0,41
9	90,56	0,05	37,5	0,10	49,72	0,19	28,8	0,10	25,17	0,33	20,8	0,10	7,91	0,72	7,4	0,10	0,00	1,00	0,0	0,10
10	91,61	0,05	37,6	0,10	50,91	0,17	29,3	0,12	24,31	0,38	20,2	0,11	7,56	0,71	7,1	0,12	0,00	1,00	0,0	0,11
11	87,77	0,04	36,9	0,10	50,97	0,17	29,4	0,11	24,51	0,39	20,3	0,10	8,17	0,75	7,7	0,11	0,00	1,00	0,0	0,12
12	94,36	0,04	37,5	0,73	45,34	0,18	26,7	0,74	21,24	0,35	1233,6	0,74	7,93	0,68	7,4	0,74	0,00	1,00	0,0	0,75
A	5,53	0,96	3,1	0,35	3,77	0,92	2,3	0,51	0,81	0,99	0,8	0,84	0,38	1,00	0,4	0,51	0,00	1,00	0,0	0,86
B	27,67	0,95	14,2	1,56	14,18	0,96	9,0	1,57	6,54	0,98	5,6	1,95	1,90	0,99	1,8	0,51	0,00	1,00	0,0	0,86
C	50,76	0,85	16,3	1,05	13,51	0,96	9,4	1,30	6,56	0,98	5,9	1,95	1,49	0,99	1,5	1,57	0,00	1,00	0,0	1,96
D	50,52	0,86	15,1	1,02	16,15	0,95	10,7	1,30	5,66	0,97	5,3	1,09	2,35	0,99	2,3	1,30	0,00	1,00	0,0	1,58
STC	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78
RFDFF	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78

6. CONCLUSIONS

This paper presents heuristic procedures for solving project scheduling problems under uncertainty. The heuristics exploits probabilistic information on random activities duration within the framework of joint probabilistic constraints. In the proposed algorithm, the temporal aspect of the problem is treated at a higher level, whereas the probabilistic aspect is tackled at decision points, when activities are supplied by available resources. This hierarchical view of the problem has allowed to develop effective heuristics for projects with high variability with the aim of obtaining a schedule with good performances.

7. REFERENCES

Alvarez-Valdes, R., Tamarit, J.M., 1989. Heuristic algorithms for resource constrained project scheduling: A review and an empirical analysis. *Advances in project scheduling*, 113-134. Slowinski, R. and J. Weglarz (eds.). Elsevier, Amsterdam.

Artigues, C., Roubellat, F., 2000. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, 127, 297-316.

Artigues, C., Michelon, P., Reusser, S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149, 249-267.

Aytug , H., Lawley , M.A., McKay, K. , Mohan, S., and Uzsoy, R., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161, 86-110.

Ballestin, F., 2007. When it is worthwhile to work with the stochastic RCPSP?. *Journal of Scheduling*, 10, 153-166.

Beraldi, P., Bruni, M.E., Guerriero, F., Pinto, E., 2007 Heuristic procedures for the robust project scheduling problem, Technical Report, ParCoLab, DEIS, University of Calabria.

Bisschop, J., Roelofs, M., 2007. AIMMS. 3.7 Users guide. Paragon Decision Technology B.V., The Netherlands. Optimization, Inc., Incline Village, NV, 2006.

CPLEX, ILOG CPLEX 6.5: Users Manual, CPLEX Optimization, Inc., Incline Village, NV, 1999.

Drud, A. S. CONOPT: A System for Large Scale Nonlinear Optimization, Reference Manual for CONOPT Subroutine Library, 69p, ARKI Consulting and Development A/S, Bagsvaerd, Denmark (1996).

Davis, E., Patterson, J., 1975. A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Management Science*, 21, 944-955.

Deblaere, F., Demeulemeester, E., Herroelen, W., Van de Vonder, S., 2007. Robust Resource Allocation Decisions in Resource-Constrained Projects. *Decision Sciences*, 38, 1-37.

Golenko-Ginzburg, D., Gonik, A., 1998. A heuristic for network project scheduling with random activity durations depending on the resource allocation. *International Journal of Production Economics*, 55, 149-162.

Golenko-Ginzburg, D., Gonik, A., 1997. Stochastic network project scheduling with non-consumable limited resources. *International Journal of Production Economics*, 48, 29-37.

Herroelen, W., Leus, R., 2004. The construction of stable project baseline schedules. *European Journal of Operational Research*, 156, 550-565.

Herroelen, W., Leus, R., 2004. Robust and reactive project scheduling: a review and classification of procedures. *International Journal of production Research*, 42(8), 1599-1620.

Herroelen, W., Leus, R., 2005. Project scheduling under uncertainty - survey and research potentials. *European Journal of Operational Research*, 165(2), 289306.

Igelmund, G., Radermacher, F.J., 1983. Algorithmic approaches to preselective strategies for stochastic scheduling problems. *Networks* (13), 29-48.

Kolisch, R., Sprecher, A., 1997. PSPLIB a project scheduling problem library. *European Journal of Operational Research*, 96, 205-216.

Kolisch, R., Sprecher, A., Drexl, A., 1995. Characterization and generation of a general class of resource constrained project scheduling problems. *Management Science*, 41(10), 1693-1703.

Kolisch, R., Hartmann, S., 1999. Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis J. Weglarz (editor): Project scheduling: Recent models, algorithms and applications, 147-178, Kluwer, Amsterdam, the Netherlands, 1999.

Leus, R., 2003. The generation of stable project plans. Ph.D. Thesis, Department of Applied Economics, Katholieke Universiteit Leuven, Belgium.

Leus, R., Herroelen, W., 2004. Stability and resource allocation in project planning. *IIEE Transactions*, 36, 667-682.

Mehta, S.V., Uzsoy, R.M., 1998. Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation*, 14(3), 365-378.

M hring, R.H. and Stork, F., 2000. Linear preselective policies for stochastic project scheduling. *Mathematical Methods of Operations Research*, 52, 501-515.

Rabbani, M., Fatemi Ghomi, S.M.T., Jolai, F., Lahiji, N.S., 2007. A new heuristic for resource-constrained project scheduling in stochastic networks using critical chain concept. *European Journal of Operational Research*, 176, 794-808.

Stork, F., 2000. Branch-and-bound algorithms for stochastic resource-constrained project scheduling, Research Report. 702/2000. Technische Universitat Berlin

Tavares, L.V., Ferreira, J.A.A., Coelho, J.S., 1998. On the optimal management of project risk. *European Journal of Operational Research* 107, 451-469.

Tsai, Y.W., Gemmil, D.D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. *European Journal of Operational Research* 111, 129-141.

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. *International Journal of Production Economics*, 97, 227-240.

Van De Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R., 2006. The trade-off between stability and makespan in resource-constrained project scheduling, *International Journal of Production Research*, 44(2), 215-236.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., 2007. A classi_cation of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10, 195-207.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., 2008. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, 189, 723-733.

Wang, J. (2005). Constraint-based schedule repair for product development projects with time-limited constraints. *International Journal of Production Economics*, 95, 399-414

Wiest, J.D., Levy, F. K., 1977 A Management Guide to PERT/CPM with GERT/PDM/DCPM and Other Networks, Prentice-Hall, New Jersey.

Zhu, G., Bard, J.F., Yu, G., 2005. Disruption management for resource-constrained project scheduling. *Journal of the Operational Research Society*, 56, 365-381.

Zhu, G., Bard, J.F., Yu, J.F., 2007. A two-stage stochastic programming approach for project planning with uncertain activity durations. *Journal of Scheduling* 10, 167-180.