A Two-stage Stochastic Programming Model for the Resource Constrained Project Scheduling Problem under Uncertainty

Maria Elena Bruni, Luigi Di Puglia Pugliese, Patrizia Beraldi and Francesca Guerriero
Department of Mechanical, Energy and Management Engineering, Unical, Italy

Keywords: RCPSP, Uncertainty, Integer L-shaped, Stochastic Programming.

Abstract: Due to the increasing competitiveness of businesses, project planning and scheduling have become a challenging theme in the last years. In this paper, we propose a two-stage stochastic programming model for the resource constrained project scheduling problem, taking into account the stochasticity of activity durations. In this formulation, assuming that some activity duration scenarios are known, resource allocations are taken in the first stage, while scheduling decisions are postponed in the second stage. The resulting problem is a mixed integer problem with recourse, where binary variables appear in the first stage. In order to efficiently solve the problem, a decomposition algorithm is developed, based on the well-known integer L-shaped method. Detailed computational results are presented for a set of benchmark instances taken from the literature.

1 INTRODUCTION

The resource-constrained project scheduling problem (RCPSP) deals with the sequencing of activities that are usually related by precedence constraints and by the simultaneous use of scarce resources. The RCPSP has attracted the attention of scientists and practitioners, given also its importance in many fields (Bruni et al., 2011a,b), who have made considerable efforts to enhance the efficiency of the solution process. In the last decade, increasing financial pressures have put more emphasis on the project execution and, given the uncertainty characterizing the project environment, the role of deterministic static schedules has been challenged. Due to machine breakdowns, employee absenteeism and delay in materials supply, bad whether conditions and many other uncontrollable factors, one or more project activities may experience a delay, threatening the operational viability of the planned schedule and its implementation. To address these challenges, a flourishing stream of literature has focused on the RCPSP under uncertainty in which activity durations are assumed to be random variables. Two different approaches can be used depending on the genuine interpretation of the RCPSP under uncertainty and the way the uncertainty is tackled. Indeed, the problem can be viewed as a stochastic dynamic optimization problem, where decisions are made each time new information becomes available, or as a problem where a tentative plan, which can be changed during project execution, should be determined and agreed before knowing the realization of uncertainty. As far as the first option is concerned, the literature has focused on dynamic decision processes (called policies) that define, at completion of some activities, appropriate actions concerning the choice of a set of activities that should be executed next. The vast majority of the research in this area is concerned only with the expected makespan objective. Möhring and Stork (2000) following Igelmund and Radermacher studied the so-called linear preselective policies, that were exploited by Stork (2001) to develop a branch-and-bound procedure, to efficiently compute an optimal policy minimizing the expected makespan. Policies have also been used for determining predictive activity starting times, with the objective of minimizing costs related to positive and negative deviations of actual starting times, from the predicted ones, and to penalties/bonuses associated with late/early project completion. Deblaere et al. (2007) and Golenko et al. (1997) presented solution procedures for the RCPSP with random activities duration and the expected makespan as objective. Starting from the concept of critical chain introduced by Goldratt (Goldratt, 1997), in Rabbani et al. (2007) a new heuristic was presented for minimizing the expected project duration and its variance. Later, Ballestín (2007) developed regret-based biased random sampling procedures then embedded into a genetic algorithm, whereas Ballestín and Leus (2009) examined multiple possible objective

Bruni, M., Pugliese, L., Beraldi, P. and Guerriero, F.
A Two-stage Stochastic Programming Model for the Resource Constrained Project Scheduling Problem under Uncertainty.
DOI: 10.5220/0006612601940200
In *Proceedings of the 7th International Conference on Operations Research and Enterprise Systems (ICORES 2018)*, pages 194-200
ISBN: 978-989-758-285-1

functions for project scheduling with stochastic activity durations. For a special case involving only one renewable resource (the budget) a two-stage integer linear stochastic program has been proposed in Zhu et al. (2007), whereas Bruni et al. (2011a), proposed a chance-constrained based heuristic aiming at building a baseline schedule which is protected against possible disruptions.

Contrary to the stochastic project scheduling (Bruni et al. 2015), robust project scheduling assumes that the distribution of the uncertain activity duration is not known or only partially known. Robustness can be referred to either the project makespan (in this case it is referred as quality-robustness) or to possible deviations between the planned and realized starting times of the projected schedule (we call this solution robustness). The literature on robust project scheduling has mainly dealt with the development of effective and efficient proactive and reactive scheduling procedures. Proactive scheduling aims at generating robust baseline schedules, that incorporate some protection against possible disruptions, whereas reactive scheduling procedures can be invoked during the execution of the project, to repair the baseline schedule by deviating as little as possible from the original baseline schedule. For an extensive review of research in this field, the reader is referred to Demeulemeester and Herroelen (2011).

Robust optimization approaches have been recently proposed for the RCPSP, under general polyhedral uncertainty (Bruni et al., 2017a,b). Assuming that scenarios represent different realizations of the activity durations, a min-max absolute-regret problem is proposed by Artigues et al. (2011), to minimize the maximum absolute difference between the makespan obtained by the robust solution and the scenario dependent optimal solutions.

In this paper, we study a stochastic programming optimization approach for the RCPSP, assuming that the random variables are discretely distributed. We observe that such an assumption is quite general, since discrete distributions arise either naturally in many real-world applications, or as empirical approximations of the continuous ones derived, for example, by taking a Monte Carlo sample from a general distribution.

The two-stage stochastic RCPSP (TSRCPSP) is investigated by assuming that scenarios represent different realizations for the activity durations. The decision variables are divided into wait-and-see variables, that must be determined before the realization of the uncertain parameters, and here-and-now variables, that can adjust to the uncertain data when they become known. The objective is to find a schedule that minimizes the expected makespan over all scenarios. To solve the problem, we have designed and implemented an integer L-shaped decomposition approach, both in the single cut and the multi-cut versions.

The remainder of the paper is organized as follows: in Section 2, a formal definition of the TSRCPSP is given. Section 3 presents a detailed description of the proposed exact algorithm. Section 4 discusses the computational results obtained on a set of benchmark problems. Finally, some conclusions are drawn in Section 5.

2 PROBLEM FORMULATION

Project activity duration is typically unknown when scheduling decisions need to be taken; under uncertainty, specialized models able to cope with this uncertainty should be developed. Specifically, these models should allow to make some decisions about the schedule before the actual activities duration is known. Then, after the uncertainty is disclosed, a recourse action can be implemented to compensate deficiencies in the previously made schedules. Stochastic programming formulations extend and adapt deterministic models to allow this kind of schedule modifications. In these models, some decisions must be made in the first-stage under uncertainty. Then, in the second-stage, a recourse action can be made after observing the actual values of the random variables.

In real cases, it is a very common practice to decide upon the resource allocation well in advance, since often resources (e.g. expert staff) cannot be easily transferred between activities at short notice, (for instance in a multi-project environment), nor allocated without sufficient time lapse. In our model, we assume that the resource allocation is a static here-and-now decision, that can be made in advance, under uncertainty about the actual duration of activities. The starting times, on the contrary, can be decided later on.

Let define a project as a set of activities $V = \{0,\dots,n+1\}$ (where 0 and $n+1$ are two dummy activities representing the project start and end, respectively) that have to be scheduled. A set of renewable resources $K = \{1,\dots,m\}$, each with finite capacity R_k for all k belonging to K is used during the project execution. Precedence constraints between different activities within the project are modeled by a set of project arcs E such that $(i,j) \in E$ means that activity j has to start after the completion of activity i. Each activity $i \in V$ requires a non-negative amount r_{ik} and for the activities 0 and $n+1$ $r_{0k} = r_{n+1\,k} = R_k$ for all

$k \in K$.

The TSRCPSP can be defined on the basis of the flow network model, originally proposed by Artigues and Roubellat (2000). Different sets of variables are used. Variables f_{ijk} denote the number of units of resource k directly transferred from an activity i to an activity j. Binary variables y_{ij} assume the value 1 if activity j starts after the completion of activity i and 0 otherwise and define an extended set of precedence relations with the aim of resolving possible resource conflicts. Finally variables S_i define the starting times of each activity i.

We assume that the random vector, representing stochastic activity durations, has a finite support. Henceforth, we define Σ as the set of its possible realizations (scenarios) and we denote by $\{0, d_1^\sigma, \ldots, d_n^\sigma, 0\},\ \sigma \in \Sigma$ the durations of the activities under scenario σ occurring with probability p_σ.

The TSRCPSP may be formulated as follows.

$$\min_y Q(y) \tag{1}$$

$$y_{ij} = 1\ \forall (i,j) \in E, \tag{2}$$

$$y_{ij} + y_{ji} \leq 1\ \forall i,j \in V,\ i < j, \tag{3}$$

$$y_{il} \geq y_{ij} + y_{jl} - 1\ \forall i,j,l \in V, i \neq j \neq l, \tag{4}$$

$$f_{ijk} - min(r_{ik}, r_{jk}) y_{ij} \leq 0\ \forall i,j \in V, i \neq j,$$
$$i \neq n+1, j \neq 0,\ \forall k \in K, \tag{5}$$

$$\sum_{j \in V \setminus \{i,0\}} f_{ijk} = r_{ik}\ i \in V \setminus \{n+1\},\ \forall k \in K, \tag{6}$$

$$\sum_{i \in V \setminus \{j,n+1\}} f_{ijk} = r_{jk}\ j \in V \setminus \{0\},\ \forall k \in K, \tag{7}$$

$$f_{ijk} \geq 0\ \forall i,j \in V, i \neq j, i \neq n+1, j \neq 0,$$
$$\forall k \in K, \tag{8}$$

$$y_{ij} \in \{0,1\}\ \forall i,j \in V, i \neq j. \tag{9}$$

It should be noted that the constraints (3) and (4) are valid inequalities preventing cycles into the extended graph, and constraints (5), (6) and (7) are resource flow-conservation constraints, ensuring feasible resource allocations. Preprocessing constraints (2) impose the preexisting precedence constraints.

Hence, the solution of this first stage problem, gives a resource allocation, i.e., the sequence of the activities and the resource flow among them. It should be clear that it is often possible to make different resource allocation decisions, each represented by a different resource flow, for the same schedule, having serious impacts on the expected makespan of the schedule. For this reason, the TSRCPSP minimizes in the second stage the expected value of the makespan, $Q(y) = \sum_{\sigma \in \Sigma} p_\sigma Q(y, \sigma)$, resulting from the resource allocation decisions made in the first stage. For a given realization σ, $Q(y, \sigma)$ is the optimal solution of a scenario subproblem that can be obtained by solving the following model:

$$Q(y, \sigma) = \min S_{n+1}^\sigma \tag{10}$$

$$S_j^\sigma - S_i^\sigma - M y_{ij} \geq d_i^\sigma - M\ \forall i,j \in V$$
$$\forall \sigma \in \Sigma \tag{11}$$

$$S_0^\sigma = 0 \tag{12}$$

here M is large enough constant used to model the disjunctive precedence constraints (11).

For a fixed $y = \hat{y}$, the subproblem (10)–(12) is equivalent (for strong duality) to the following model.

$$Q(\hat{y}, \sigma) = \max \sum_{(i,j) | \hat{y}_{ij} = 1} d_i^\sigma \alpha_{ij}$$

$$\sum_{j \in V \setminus n+1} \alpha_{ji} + \sum_{j \in V \setminus 0} \alpha_{ij} = 0\ \forall i \in V \setminus \{0, n+1\}$$

$$\sum_{i \in V \setminus n+1} \alpha_{in+1} = 1$$

$$\sum_{i \in V \setminus 0} \alpha_{0i} = 1$$

$$\alpha_{ij} \leq \hat{y}_{ij}\ \ \forall i,j \in V$$

$$\alpha_{ij} \geq 0\ \ \forall i,j \in V$$

3 SOLUTION APPROACH

Problem (1)–(12) is a two-stage stochastic mixed integer model with continuous recourse, where the first-stage problem includes binary decision variables, whereas the recourse problem is a longest path problem. In general, solving two-stage stochastic programs is computationally demanding and decomposition techniques are usually used to solve the problem efficiently. Specifically, for the continuous recourse case, the L-shaped algorithm can be used to solve the two-stage stochastic programming problem based on Benders decomposition, a well-known technique for solving large scale MIP problems that exhibit a special structure. This approach decomposes the original problem into an integer master problem and one or more linear subproblems. In particular, after solving the master problem, the current optimal master solution is fed into the recourse problems. Subsequently, a series of cuts is generated based on the subproblems solutions and new constraints are added to the master problem which is then solved again. The above process is executed iteratively until an optimal solution is found. The L-shaped method was first proposed in Van Slyke and Wets (1969) for solving two-stage stochastic linear problems. Its main idea is to approximate the expected recourse function $Q(y)$ (whose evaluation requires the solution of all the second-stage

recourse problems) in the objective function of the two-stage stochastic problem by adding cuts into the master problem. The cuts are classified into two types known as optimality cuts and feasibility cuts. The former drive the master solution toward optimality, while the latter ensure the feasibility of the subproblems. In the case of complete recourse (which is also our case) it is not longer required to take care of feasibility issues.

The general scheme of the L-shaped approach, in this case, is the following.

- Set $LB := -\infty$; $UB := +\infty$ $t = 0$.
- while $UB > LB$ do
  - Solve the master problem obtaining y^*, f^*.
  - Assign to LB the optimal objective function value of the master problem.
  - Solve the subproblems for each scenario $\sigma \in \Sigma$ and let $Q(y^*)^t$ be the optimal expected makespan.
  - Update $UB = min(UB, Q(y^*)^t)$.
  - Add an optimality cut and update $t := t + 1$.
  - end while
- Return LB

By defining η as an additional variable, the master problem in our case assumes the following form:

$$\min \eta \quad (13)$$
$$s.t. (2) - (9) \quad (14)$$

3.1 Optimality Cuts

At a given iteration of the algorithm, once the master problem has been solved and the corresponding optimal solution obtained, we should solve all the scenario related subproblems (10)–(12). Let $Q(y*, \sigma)^t$, for all $\sigma \in \Sigma$ the optimal second stage objective function, representing the makespan under scenario σ. Let π^{*t}_σ the optimal longest path for the scenario σ. Let denote with $\pi^{*t} = \cup_{\sigma \in \Sigma} \pi^{*t}_\sigma$ the subset of arcs belonging to the union of the optimal longest paths for each scenario. Given a finite global lower bound L of the TSRCPSP, and defining $Q = (Q(y*)^t - L)$, the following valid cut can be added to the master problem formulation:

$$\eta \geq Q \sum_{(i,j)|(i,j)\in\pi^{*t}} y_{ij} - [Q(|\pi^{*t}| - 1) - L] \quad (15)$$

Since there is a finite number of such cuts the algorithm converges to an optimal solution to the TSRCPSP.

Proposition 1. *The cut (15) is a valid optimality cut.*

Proof. The quantity $\sum_{(i,j)\in\pi^{*t}} y_{ij}$ is always less than or equal to $|\pi*|$, taking exactly the value $|\pi^{*t}|$ when $y_{ij} = 1, \forall (i,j) \in \pi^{*t}$. In this case, the right-hand side takes the value $Q(y*)^t$, otherwise the right-hand side takes a value less than or equal to L. Since the first stage decision variables $\mathbf{y}$ are binary, there is only a finite number of feasible first stage solutions and therefore, the number of cuts that can be added to the master is finite. □

We have also implemented a multicut L-shaped algorithm (Birge and Louveaux, 1988). In general, the multicut L-shaped algorithm has less major iterations than the L-shaped algorithm. However, solving the master problem requires more computation time. In particular, by defining an additional set of free variables η^σ and denoting with $Q_\sigma = Q(y*, \sigma)^t - L$, the following optimality cut can be added for each $\sigma \in \Sigma$:

$$\eta_\sigma \geq Q_\sigma \sum_{(i,j)|(i,j)\in\pi^{*t}_\sigma} y_{ij} - [Q_\sigma(|\pi^{*t}_\sigma| - 1) - L] \quad (16)$$

Optimality cuts (16) ensure that the value of each variable η^σ is larger than or equal to the optimal value of its corresponding second-stage problem for each scenario.

Proposition 2. *The cut (16) is a valid optimality cut.*

Proof. The proof is omitted since trivial. □

In this case, the master problem objective function is $\min \sum_{\sigma\in\Sigma} p_\sigma \eta^\sigma$.

4 COMPUTATIONAL EXPERIMENTS

This Section reports on the computational experiments carried out to assess and compare the performance of the proposed approaches. The algorithms have been coded in Java and run on a PC with 16 GB RAM and 2.50 GHz Intel Core i7 - 4710 HQ CPU. The numerical results have been collected on 120 instances generated from the benchmark deterministic instances of the PSPLIB (Kolisch and Sprecher, 1997) (available at http://www.om-db.wi.tum.de/psplib) with 30 activities. The instances differ essentially for three main parameters, namely the network complexity ($NC \in \{1.50, 1.80, 2.10\}$) , the resource factor ($RF \in \{0.25,\ 0.50\}$) and the resource strength ($RS \in \{0.70,\ 1.00\}$). The first parameter measures the average number of non-redundant arcs per nodes including the dummy activities, the second reflects the average percentage of different resource types for which a non-dummy activity has a non-zero resource

demand and the last one concerns the strength of the resource constraints.

A time limit of 20 minutes has been imposed for both algorithms. We have assumed that each activity either may have a normal duration or it can be delayed. In this case the duration assumes a peak value. In practical settings, it is unlikely that all the activities exhibit longer durations than expected. Henceforth, we have considered single disruption scenarios, that is in each scenario only one activity can be disrupted. Consequently, the number of scenarios is 30. The numerical results are collected in Tables 1–4.

Columns NC, RF, and RS correspond to the characteristics of the problem instance. The total computational times and the time spent for solving the subproblems, in seconds, are indicated in the table with headers *time* and *timeSP*, respectively. The average number of iterations-calculated over the instances solved within the time limit (column #*solved*)-is reported in column *iter*. The number of cuts is reported in columns #*cuts*.

The behavior of the Benders approach is strongly influenced by the characteristics of the instances, as it happens also in the deterministic case. In the next Sections,we present a detailed analysis of the computational performance of the proposed approaches depending on the network characteristics.

4.1 Sensitivity to the Network Characteristics

For the sake of comprehension, let us consider the parameters *NC*, *RF*, and *RS* separately.

Parameter *NC*. Tables 1 and 2 show the average results over all solved instances varying *NC*, for the single cut and the multi-cut algorithm, respectively.

Table 1: Average computational results over all the instances solved within the time limit varying *NC*, single cut.

NC	*time*	*iter*	#*cuts*	*timeSP*	#*solv*
1.5	197.74	53.60	52.60	0.011	30
1.8	227.52	73.48	72.48	0.016	27
2.1	140.25	66.66	65.66	0.017	32

From the analysis of the results, we can observe that, on average, 75% of the instances are solved to optimality within the time limit. As can be observed, the total computational time is similar for both algorithms, but the number of cuts that should be added into the master problem is higher for single cut than for the multi-cut version. In general, the subproblems do not represent a computational challenge since they can be solved in polynomial time. This behaviour is observed for all values of *NC*.

Table 2: Average computational results over all the instances solved within the time limit varying *NC*, multi-cut.

NC	*time*	*iter*	#*cuts*	*timeSP*	#*solv*
1.5	192.90	24.90	23.90	0.005	25
1.8	229.68	38.65	37.65	0.014	27
2.1	145.13	35.70	34.70	0.044	26

Parameter *RF*. Referring to the values of *RF*, Tables 3 and 4 highlight the strong relation between the average portion of the resources used and consumed and the behaviour of the proposed solution approach. Indeed, the lower the value of *RF*, the higher the number of solved instances to optimality.

Table 3: Average computational results over the instances solved within the time limit, single cut.

RF	*time*	*iter*	#*cuts*	*timeSP*	#*solv*
0.25	63.78	26.44	25.44	0.01	59
0.5	426.68	138.83	137.83	0.03	30

Table 4: Average computational results over the instances solved within the time limit, multi-cut.

RF	*time*	*iter*	#*cuts*	*timeSP*	#*solv*
0.25	128.63	14.92	13.92	0.00	47
0.5	288.78	64.09	63.09	0.05	31

The single cut version is more efficient on problem with a small *RF*, whereas when the *RF* increases the computational time increases as well. The multi-cut version of the algorithm is able to solve one more instance with $RF = 0.5$. The higher is the value of the *RF*, the harder are the instances to be solved, since when very few resource types are required by the activities (low values for RF) it is easy to take good resource allocation decisions.

The last part of the Section is devoted to the analysis of the impact of the resources strength. The higher the value of RS, the higher the number of instances solved to optimality, revealing the difficulty of the instances with low *RS* values. The analysis of the results shows that the most time consuming instances for the single-cut algorithm are those with $RS = 1$, whereas for the multi-cut version is the opposite. On the other hand, the multi-cut version is able to solve the instances with $RS = 1$ more efficiently than the single-cut algorithm, with an average time of around 148 seconds.

This might be due to the fact that problems with $RS = 1$ have less resource conflicts than problems with lower values of *RS*. We notice that this has a di-

Table 5: Average computational results over the instances solved within the time limit, single-cut.

RS	*time*	*iter*	*#cuts*	*timeSP*	*#solv*
0.7	131.26	52.68	51.68	0.01	34
1	220.01	71.53	70.53	0.02	55

Table 6: Average computational results over the instances solved within the time limit, multi-cut.

RS	*time*	*iter*	*#cuts*	*timeSP*	*#solv*
0.7	252.13	33.44	32.44	0.04	27
1	148.62	33.09	32.09	0.01	51

rect impact on the master problem, where a low number of extra precedences need to be added in order to eventually solve resource conflicts. Since the master it is easy to be solved, the multi-cut version turns out to be more efficient than the single cut on average. For the instances not solved to optimality within the time limit, the gap is below 1% for both the single cut and the multi-cut L-shaped algorithm.

5 CONCLUSIONS

In this paper, we have studied the RCPSP under uncertainty. In summary, the main contributions of this paper are the following. First, we have proposed a two stage stochastic programming approach, where resource allocation is a first stage decision, whereas the starting times are second stage decisions. An integer L-shaped algorithm has been tested on a set of benchmark instances generated from deterministic benchmark instances.

The analysis of the results shows that the single-cut version provides better results than the multi-cut one. This consideration would suggest the use of different cut aggregation strategies. The design of these strategies together with the definition of tailored approaches for the master problem solution is the subject of ongoing research.

REFERENCES

Artigues C and 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, Leus R, Nobibon FT (2013) Robust optimization for resource-constrained project scheduling with uncertain activity durations. Flexible Services and Manufacturing Journal 25(12):175205

Ballestín F (2007) When it is worthwhile to work with the stochastic RCPSP? J Sched 10:153–166

Ballestín F and Leus, R (2009) Resource-constrained project scheduling for timely project completion with stochastic activity durations. Prod Oper Manag 18:459–474

Birge J and Louveaux F (1988)A multicut algorithm for two-stage stochastic linear programs. European Journal of Operational Research 34:384–392

Bruni ME, Di Puglia Pugliese L, Beraldi P, Guerriero F, (2017a) An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. Omega 71: 66–84

Bruni ME, Di Puglia Pugliese L, Beraldi P, Guerriero F, (2017b) A computational study of exact approaches for the robust resource-constrained project scheduling problem. Submitted

Bruni ME, Beraldi P, Guerriero F. (2015) The stochastic resource-constrained project scheduling problem. Handbook on Project Management and Scheduling Vol. 2:811–835

Bruni ME, Beraldi P, Guerriero F, Pinto E (2011a) A heuristic approach for resource constrained project scheduling with uncertain activity durations. Comput Oper Res 38:1305–1318

Bruni ME, Beraldi P, Guerriero F, Pinto E (2011b) A scheduling methodology for dealing with uncertainty in construction projects. Eng Computation 28(8):1064–1078

Deblaere F, Demeulemeester E, Herroelen W, VandeVonder S (2007) Robust resource allocation decisions in resource-constrained projects. Decision Sci 38:1–37

Demeulemeester E and Herroelen W (2011) Robust Project Scheduling. Foundations and Trends in Technology, Information and Operations Management: Vol. 3: No. 34, pp 201-376.

Goldratt EM (1997) Critical Chain. The North River Press, Great Barrington

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 55:149–162

Kolisch R and Sprecher A (1997) Psplib - a project scheduling problem library. European Journal of Operational Research 96(1): 205–216

Laporte G, Louveaux FV (1993) The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters, 13(3): 133-142

Möhring RH and Stork F (2000) Linear preselective strategies for stochastic project scheduling. Math Method Oper Res 52(3):501–515

Rabbani M, Fatemi Ghomi SMT, Jolai F, Lahiji NS (2007) A new heuristic for resource-constrained project scheduling in stochastic networks using critical chain concept. Eur J Oper Res 176:794–808

Stork F (2001) Stochastic resource-constrained project scheduling. PhD Dissertation, Technische Universität Berlin, Berlin, Germany.

Tavares LV, Ferreira JAA and Coelho JS (1998) On the optimal management of project risk. European Journal of Operational Research 107:451–469

Van Slyke RM and Wets RJ (1969) L-shaped linear programs with applications to optimal control and stochastic programming. J. Applied. Math 17(4):638–663

Zhu G, Bard JF, Yu G (2007) A two-stage stochastic programming approach for project planning with uncertain activity durations. J Sched 10:167–180