# Optimal Activity Crashing in Project Management
# An Endogenous Uncertainty Stochastic Programming Approach

**Bruno Flach**
IBM Research – Brazil
Av. Pasteur 138-146, Rio de Janeiro, CEP 22290-240, RJ, Brazil
bflach@br.ibm.com

**Carlos Raoni Mendes**
IBM Research – Brazil, Pontifícia Universidade Católica do Rio de Janeiro
Av. Pasteur 138-146, Rio de Janeiro, CEP 22290-240, RJ, Brazil
craoni@br.ibm.com

**Marcus Poggi**
Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225 – RDC, CEP 22451-900, RJ, Brazil
poggi@inf.puc-rio.br

## ABSTRACT

In this work, we describe the development of a stochastic programming model aimed at determining an optimal activity crashing plan within a project management context. In particular, we are interested in selecting a subset of actions to be carried out in order to minimize the sum of costs incurred due to activity crashing and those related with a project's (stochastic) completion date. A particular feature of our approach is the incorporation of decision-dependent uncertainties into the model and solution procedure, which characterizes it as part of a difficult and rarely studied class of problems known as stochastic programs with endogenous uncertainty.

**KEYWORDS. Stochastic Programming, Project Management, Endogenous Uncertainty, MP - Probabilistic Models.**

## 1. Introduction

Significant delays and over-budget costs are often a reality across several different types of projects. Various important economic activities rely on the development of massive infrastructure which requires huge investments, the combination of distinct skills and expertise and the performance of possibly thousands of activities. That is the case, for example, in the construction of power plants for energy generation, offshore platforms for oil exploration and production and refineries that process raw materials into useful products. From a project manager's viewpoint, the situation is worsened by the fact that there are complex interdependecies among activities, various contractors carrying out different segments of the project and a potentially large number of risk factors that might interfere with original plans.

Within this context, activity crashing refers to the allocation of (extra) resources to a given task so as to minimize its expected duration or, alternatively, maximize the likelihood of its completion within a certain time frame. Such resources may include, for example, the assignment of additional or more skilled workers, outsourcing to subcontractors and/or the use of extra equipment to perform a given amount of workload. Obviously, the allocation of additional resources requires investments that must be carefully assessed in order to properly evaluate the trade-off between costs involved in activity crashing and those that may result from delays in project completion and, possibly, associated penalties.

In this work, we describe the development of a stochastic programming model aimed at determining an optimal activity crashing plan - meaning the subset of actions to be carried out in order to minimize the sum of costs incurred due to activity crashing and those related with the project (stochastic) completion date. A particular feature of our approach is the incorporation of decision-dependent uncertainties into the model and solution procedure, which characterizes it as part of a difficult and rarely studied class of problems known as stochastic programs with endogenous uncertainty.

The remaining of the paper is organized as follows: section 2 analyzes previous work on related problems and highlights our contributions, section 3 describes the mathematical model of the problem, identifies some of the difficulties that prevent its solution by traditional methods and presents the solution methodology – first introduced in (Flach and Poggi, 2010) – and section 4 presents computational results that illustrate the application of the proposed methodology. Finally, section 5 details the conclusions and directions of current / future work.

## 2. Literature Review

Identify the most critical activities of a project (i.e., the activities that requires a special attention in order to successfully complete the project) is one of the most important roles on project management research. The first works on this subject were related with two important techniques, the critical path method (CPM), and the program (or project) evaluation and review technique (PERT), (Kelley and Walker, 1959) and (Fazar, 1959) respectively. Both techniques are based on the identification of the project's critical path, the set of activities that prevent the project to finish earlier, (Kelley, 1961). Any delay on these critical path activities causes a delay on the project completion date. The main difference between PERT and CPM is that PERT estimates the duration of each activity based on the mean of a beta probability distribution determined by three estimates, optimistic, most likely and pessimistic duration, while CPM only requires a unique duration estimate for each activity. Besides its recognized importance these methods are limited, as discussed in (Moder and Davis, 1983).

The study of the activity crashing problems (see section 1) was a natural extension of the research motivated by the development of the traditional project management methods CPM and PERT. An important problem of this class is the deterministic discrete time-cost problem (DDTCP): given the project's set of activities with deterministic durations and precedence relations, a set of discrete 0-1 activity crashing opportunities (crashing measures) with costs associated, a project due date and a penalty function for delays on project's completion date, what is the optimal crashing plan that minimizes the project's cost? In (Walter J. Gutjahr, 2000), they show that DDTCP is a NP-Hard problem. Dynamic Programming and Branch and Bound are the main methods applied to produce practical solutions to the DDTCP (see (Panagiotakopoulos, 1977) and (Hindelang and Muth, 1979)).

Another research branch of deterministic activity crashing problems deals with the time-cost trade-off as different objective functions. This approach results in multi-objective problems, where does not exist a single solution that simultaneously optimizes each objective, so the optimization searches for a set of the so called non-dominated solutions (see (Roy and Vincke, 1981)). The work of (Doerner et al., 2008) apply different metaheuristic methods to solve one problem of this class. In (Rahimi and Iranmanesh, 2008) they explore the multi-objective flexibility and add another trade-off dimension (adding a new objective function), the quality, and also apply a metaheuristic method to solve the problem.

One weakness of modeling activities durations as deterministic values is that the time required to complete an activity is only known with certainty after its termination, (Gutjahr et al., 2000). A common approach to deal with this uncertainty is to model activities durations as independent random variables. As an example, classic PERT models the duration of each activity by a beta probability distribution determined by the three estimates described before. Despite PERT assume that activities durations are random variables, the method itself only uses deterministic values (the beta distribution means). The first works to directly approach the uncertainty used basically Monte-Carlo simulation (see (Slyke and Richard, 1963)) to estimate project duration and cost.

This uncertainty nature of activities durations is also studied on activity crashing problems. As an example, the stochastic discrete time-cost problem (SDTCP) is the stochastic version of DDTCP, the only difference between them is that on SDTCP activities durations are modeled as independent random variables. In (Gutjahr et al., 2000) they propose an application of the stochastic branch and bound method (see (Norkin and Ruszczyński, 1998)) to the SDTCP. Another common approach to solve SDTCP is to combine Monte-Carlo simulation with heuristic methods, as in (Walter J. Gutjahr, 2000).

Stochastic activity crashing problems are not limited to the discrete case, some authors work with the assumption that activities can be continuously crashed. In (Rahimi and Seifi, 2009) they solve a problem of this class using a sequential quadratic programming (SQP) framework. Other authors claim that the use of mathematical programming methods for activity crashing in a real project scenario is impractical. In (Ashok and Jibitesh, 2011) they list a set of difficulties of activity crashing in the maintenance of a thermal power plant, and propose a qualitative analysis framework to identify the best activities to crash.

It should be clear that an activity crashing optimization analysis for a real project scenario depends on the underlying problem due to multiple modeling decisions. One modeling decision example is between discrete or continuous crashing. The main contribution of our work is to provide a flexible mathematical modeling approach that can be used to solve many different stochastic activity crashing problems, depending only on how do you model your first- and second-stage decisions, as we explain in detail on next section.

## 3. Mathematical Model and Proposed Methodology

Mathematically, the problem is formulated by assuming we are given an activity network which describes the precedence relationships among different tasks of the project. Such network is represented by a graph and the costs associated with its edges denote the duration of each particular activity. As previously mentioned, the objective is to determine the optimal set of activities to be crashed in such as way as to minimize total incurred costs – given by the sum of the costs involved in crashing selected activities (crashing costs are denoted by $r_a$) and those related with the project's completion date – which may include penalties (benefits) associated with project completion after (before) a particular target deadline, as represented by function $\mathcal{H}(\cdot)$ assumed to be convex piecewise linear in the project duration. First-stage constraints might include, for example, budget limitations or minimum / maximum investment levels in any particular activity or group of activities and second-stage constraints might be as simple as those that define the project's duration associated with each scenario.

$$\text{Min} \quad \sum_{a \in A} r_a x_a + \sum_{s \in S} p_s \cdot \mathcal{H}\left(y_s\right) \tag{1}$$

s.t.

$$Ax \leq b \tag{2}$$

$$W_s y_s = h_s \qquad \forall s \in S \tag{3}$$

$$p_s = \prod_{a \in A} (p^N_{as} + (p^C_{as} - p^N_{as}) \cdot x_a) \qquad \forall s \in S \tag{4}$$

$$x \in \{0,1\}^{|A|}; y \in \mathbb{R}^+ \tag{5}$$

where:

- $\xi_{as}$ - duration of activity $a$ in scenario $s$;
- $p^N_{as}$ - probability of the duration of activity $a$ in scenario $s$, given that no investment is made on its crashing (i.e., $P(\xi_a = \xi_{as} | x_a = 0)$);
- $p^C_{as}$ - probability of the duration of activity $a$ in scenario $s$, given that a investment is made on its crashing (i.e., $P(\xi_a = \xi_{as} | x_a = 1)$);
- $p_s$ - continuous variable equal to the probability of scenario $s$;
- $x_a$ - binary variable which is equal to $1$ if an investment is to be made on crashing activity $a$, $0$ otherwise;
- $y_s$ - vector of second-stage variables associated with scenario $s$.

The formulation presented above characterizes a mixed-integer nonlinear program (MINLP) whose exact solution is not to be expected from traditional solution methods.

Specifically, three obstacles make this problem particularly hard: *(i)* non-linearity associated with the product of first- (scenarios' probabilities) and second-stage variables (critical path associated with each particular scenario of activities' durations), *(ii)* non-linearity associated with the computation of the probability of each scenario and *(iii)* impossibility of using traditional scenario generation techniques as they rely on the *a priori* knowledge of the random variables' probability distributions.

The application of the reformulation scheme proposed in (Flach and Poggi, 2010) to the activity crashing problem, presented next, provides an alternative to determining a provably optimal solution to the problem:

$$\text{Min} \quad \sum_{a \in A} r_a x_a + \frac{1}{|S|} \sum_{s \in S} g_s \left( \frac{\hat{p_s}}{p_s^{INI}} \right) \tag{6}$$

$$s.t. \quad Ax \leq b \tag{7}$$

$$w_s = \sum_{a \in A} (ln(p_{as}^N) + (ln(p_{as}^C) - ln(p_{as}^N)) \cdot x_a), \forall s \in S \tag{8}$$

$$\hat{p_s} \geq \alpha_k + \beta_k \cdot w_s, \quad \forall s \in S, \forall k \in K \tag{9}$$

$$\hat{p} \in \mathbb{R}^+, w \in \mathbb{R} \tag{10}$$

$$x \in \{0,1\}^{|A|} \tag{11}$$

where:

- $g_s$ - value of the optimal solution of the second-stage problem defined for each scenario $s$;
- $K$ - set of linear constraints that approximate the exponential function;
- $\alpha_k, \beta_k$ - coefficients of the $k$-th segment used to approximate the exponential function;
- $w_s$ - continuous variable equal to the natural logarithm of the probability of scenario $s$;
- $\hat{p_s}$ - continuous variable equal to the approximation of the probability of scenario $s$.
- $p_s^{INI}$ - probability of sampled scenario $s$, calculated based on the initial probability distribution of the availability of activity duration, i.e. $p_s^{INI} = \prod_{a \in A} p_{as}^N$ .

As in (Flach and Poggi, 2010), the fact that only a small number of constraints which provide an approximation to the exponential function ($\hat{p_s} \geq \alpha_k + \beta_k \cdot w_s$) will be binding at the optimal solution allows for the development of a cut generation algorithm that progressively adds cuts to the problem until a solution with gap less than or equal to a small constant $\epsilon$ is found:

**Algorithm 1** Cut Generation Algorithm

Initialize the set of cuts $K = \emptyset$, the best solution $best = \emptyset$, the lower bound $LB = -\infty$, the upper bound $UB = +\infty$ and define the maximum percentage error $\epsilon$
**while** $|(UB - LB)/UB| > \epsilon$ **do**
    $sol(x_a^*, w_s^*) =$ Solve model $P$ with the currently defined set of cuts $K$
    $LB = v(sol)$ ▷ Set $LB$ to the objective value of the current $P$ solution ($sol$)
    $origV = \sum_{a \in A} r_a x_a^* + \frac{1}{|S|} \sum_{s \in S} g_s \left( \frac{exp(w_s^*)}{p_s^{INI}} \right)$ ▷ Calculate objective value of $sol$ in the original problem
    **if** $origV < UB$ **then**
        $UB = origV$
        $best = sol$
    **end if**
    **for** each scenario $s \in S$ **do**
        Add cut $\alpha_k = exp(w_s^*) \cdot (1 - w_s^*)$ and $\beta_k = exp(w_s^*)$ to cut set $K$
    **end for**
**end while**
**return** $best$

## 4. Experiments and Computational Results

We focused our experiments on the stochastic activity crashing problem under the following assumptions:

- **Activity network**: the activity network follows the PERT model, where we have a set of activities $A$, precedence relations between them and activities durations are independent random variables modeled by a beta probability distribution parametrized by the three duration estimates – optimistic (op), most likely (ml) and pessimistic (pe).
- **Activity crashing**: the crashing effect in the probability distribution of an activity is assumed to turn the most likely estimate to be equal to the optimistic one, so the distribution changes from $PERT(op, ml, pe)$ to $PERT(op, op, pe)$. Figure 1 shows an example of activity crashing probability distribution change, the almost half-triangular shape represents the probability distribution of the activity duration when this activity has been subject to a crashing investment, while the other curve represents the original distribution. This figure was generated by sampling 10000 values of each distribution and building the respective histograms.
- **First-stage constraints**: the only constraint considered on first-stage was a budget constraint. Given a total budget $b$ to invest on the crashing plan and the crashing resources cost of each activity ($r_a$) we have the constraint $\sum_{a \in A} r_a \cdot x_a \leq b$.
- **Objective function**: the objective considered was the minimization of project completion date, so for each scenario $s$, $g_s$ is equivalent to the critical path size (or the time required to complete the project) for the scenario $s$.

Computational tests were performed to analyze the performance of the proposed algorithm. All tests were conducted on a Intel Core i5-3360M PC with 4 cores of 2.80GHz and 8 GB of RAM. Models and algorithms were implemented using python programming language and solved by IBM(R) ILOG(R) CPLEX(R) 12.5.0.0.

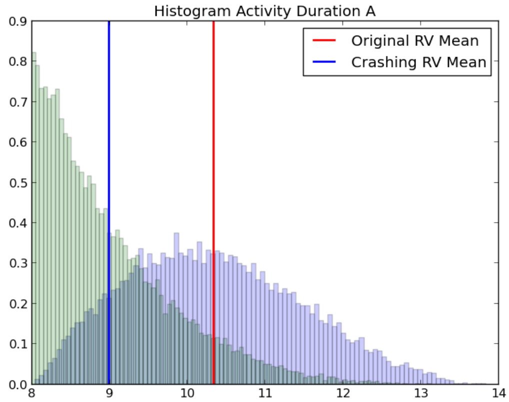


**Figure 1. Pre- and post-crashing duration histograms of an activity with 8, 10 and 14 as duration estimates.**

The tests were performed on the 10 activities instance with activity network described on table 1. This instance was inspired by an instance extracted from PSPLib, (Kolisch and Sprecher, 1997). We also used unit cost for the crashing activities costs and a budget of three units. This is equivalent to look to the best subset of three activities to crash in order to minimize the expected value of project completion date.

| *Activity* | *Optimistic* | *Most Likely* | *Pessimistic* | *Predecessors* |
|---|---|---|---|---|
| A | 8 | 10 | 14 | |
| B | 16 | 20 | 28 | A |
| C | 4 | 5 | 7 | B, H |
| D | 23.2 | 29 | 40.6 | C |
| E | 23.2 | 29 | 40.6 | D, G, H |
| F | 12 | 15 | 21 | A |
| G | 4 | 5 | 7 | C, F |
| H | 12 | 15 | 21 | A |
| I | 12 | 15 | 21 | D |
| J | 12 | 15 | 21 | I |

**Table 1. Test instance based on an instance from PSPLib**

Once a crashing plan has been determined (i.e., variables $x_a$) the probability distribution of each activity is known (i.e., $PERT(op, ml, pe)$ for activities with $x_a = 0$ and $PERT(op, op, pe)$ for activities with $x_a = 1$), which then completely defines the probability distribution of a project's total duration – though it might be very difficult to devise a closed formula for it. In order to overcome this difficulty and consistently estimate the distribution of project duration, we used Monte-Carlo simulation. The process consists in sampling duration scenarios (i.e., vectors with a duration for each activity, sampled from the corresponding probability distributions of activities' durations) and calculating the associated project completion date for each one of them, then allowing us to determining its mean, standard deviation and plotting a histogram.

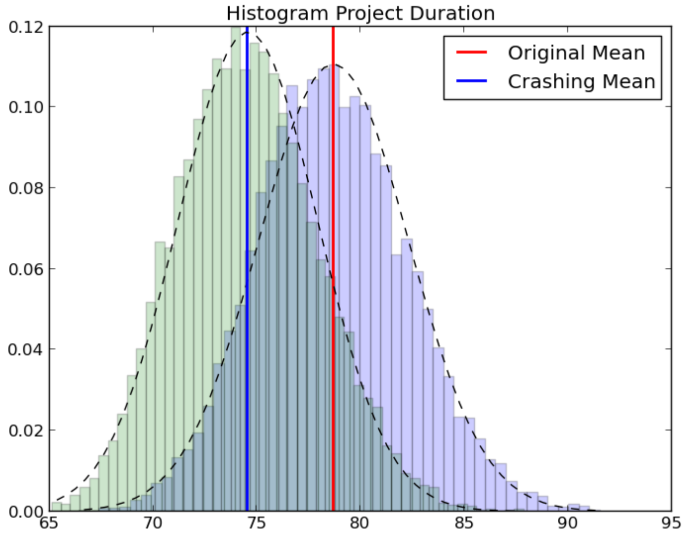


**Figure 2. Probability distributions of project completion date pre- and post-crashing decisions.**

To set a benchmark for our algorithm, we performed a brute force analysis running the Monte-Carlo simulation, previously described, with $N_1 = 10000$ scenarios for each possible crashing plan (i.e., each subset of three activities to crash). Then we ranked each plan by the simulation mean value. A similar analysis is also done in the works for the SDTCP, (Gutjahr et al., 2000) and (Walter J. Gutjahr, 2000). The best plan according to this analysis was to invest in crashing the activities B, E and I, wich results in an estimated mean of $73.34$ for the project completion date. Without any crashing investment the estimated mean, also from the Monte-Carlo simulation, was $78.69$.

Regarding our method, we used an initial sample of $|N_2| = 1000$ scenarios to build the model, sampled from the initial distributions, and then we executed our code to solve this model. The method found as best crashing plan to invest in crashing activities A, E and I. Figure 2 displays the probability distribution of project completion date for the crashing plan found by our method for the instance in case, crashing of activities A, E and I, and the original probability distribution without any crashing. These histograms were obtained also by Monte-Carlo simulation with $N = 10000$ scenarios. The estimated crashing mean of the solution found had a value of $74.5$, which has a GAP of only $1.58\%$ from the solution found by brute force (i.e. $(74.5 - 73.34)/73.34$), but obtained using considerably less computational time.

Giving a close look in both solutions, brute force and our approach, we notice that they disagree only on the first activity to crash, B and A respectively. That disagreement does not result in great impact on the expected value of the project completion date, as showed on the previous paragraph. But an interesting aspect to notice on both solutions is that they both agree on the crashing investment in activity E. This is a tricky decision, because E is not on the critical path of the network formed with only most likely durations (i.e., A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ I $\rightarrow$ J, see Figure 3). The importance of E appears only considering the uncertain nature of activity durations, the use of traditional methods will not even consider invest in E. This result also enforces the quality of our approach, showing that

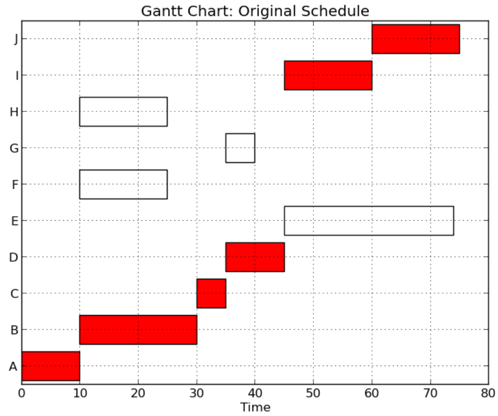


**Figure 3. Schedule of instance from table 1 considering most likely durations. Critical path activities shaded.**

our method can identify this kind of complex decisions that are required in an uncertainty environment.

## 5. Conclusions

In this work, we described the development of a stochastic programming model aimed at determining an optimal activity crashing plan within a project management context. The problem we considered is part of a difficult and rarely studied class of problems known as stochastic programs with endogenous uncertainty, where decisions affect the probability distributions of random variables.

Results presented in section 4 have encouraged our modelling approach. For illustration purposes we dealt with a small instance of the problem, which have allowed us to evaluate the quality of the obtained solution by comparing it with a brute force analysis combined with a Monte-Carlo simulation – which obviously cannot be expected to be computationally feasible in real instances of the problem. Another interesting result – particular to the instance used as an example – was the identification of the importance of investing in activities that may not lie on the project's original critical path, which would probably not be determined by traditional methods still used in practice.

Results motivate future research on the proposed method in order to allow for the solution of larger instances of the problem. Improving the solution algorithm and the initial scenarios sampling to build the model are two promising research directions currently being explored.

## References

**Ashok, M., B. S. and Jibitesh, M.** (2011). Activity crashing in shutdown maintenance through qualitative assesment: a case study. *Advances in Production Engineering & Management*, 6(4):239–248.

**Doerner, K., Gutjahr, W., Hartl, R., Strauss, C., and Stummer, C.** (2008). Nature-inspired metaheuristics for multiobjective activity crashing. *Omega*, 36(6):1019 – 1037.

**Fazar, W.** (1959). Program evaluation and review technique. *The American Statistician*, 13(2):10.

**Flach, B. and Poggi, M.** (2010). On a class of stochastic programs with endogenous uncertainty: Theory, algorithm and application. Technical report.

**Gutjahr, W. J., Strauss, C., and Wagner, E.** (2000). A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing*, 12(2):125–135.

**Hindelang, T. J. and Muth, J. F.** (1979). A dynamic programming algorithm for decision cpm networks. *Operations Research*, 27(2):225–241.

**Kelley, J. E.** (1961). Critical-path planning and scheduling: Mathematical basis. *Operations Research*, 9(3):296–320.

**Kelley, Jr, J. E. and Walker, M. R.** (1959). Critical-path planning and scheduling. In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, IRE-AIEE-ACM '59 (Eastern), pages 160–173, New York, NY, USA. ACM.

**Kolisch, R. and Sprecher, A.** (1997). Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216.

**Moder, J.J., P. C. and Davis, E.** (1983). Project management with cpm, pert and precedence diagramming.

**Norkin, V., P. G. and Ruszczyński, A.** (1998). A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83(1-3):425–450.

**Panagiotakopoulos, D.** (1977). A cpm time-cost computational algorithm for arbitrary activity cost functions. *Infor*, 15(2):183–195.

**Rahimi, M. and Iranmanesh, H.** (2008). Multi objective particle swarm optimization for a discrete time, cost and quality trade-off problem. *World Applied Sciences Journal*, 4(2):270–276.

**Rahimi, M. and Seifi, A.** (2009). An embedded simulation approach for optimal activity crashing in stochastic networks. In *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, pages 1827–1831.

**Roy, B. and Vincke, P.** (1981). Multicriteria analysis: survey and new directions. *European Journal of Operational Research*, 8(3):207–218.

**Slyke, V. and Richard, M.** (1963). Monte carlo methods and the pert problem. *Operations Research*, 11(5):839–860.

**Walter J. Gutjahr, Christine Strauss, M. T.** (2000). Crashing of stochastic processes by sampling and optimisation. *Business Process Management Journal*, 6(1):65 – 83.