Activity Criticality Criteria and Uncertainty Mitigation in Project Management: A Robust Optimization 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 de Aragão
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

Large-scale construction projects are often plagued by extreme delays and cost overruns. Complex networks of suppliers, contractors and interdependencies among various areas are compounded by several sources of uncertainty and risk over the duration of such projects. As widely observed, such challenges are common across virtually all industries and geographies. In this work, we describe a computational study aimed at determining the activities of a project that should be the focus of risk-mitigation measures – in the sense that resource and effort should be put into place so as to ensure that their actual durations equal their original nominal estimates. We opt for a robust optimization approach that accounts for the combined effects of delays in multiple activities whose interdependencies are intricately defined by complex precedence relationships. Extensive computational experiments performed on benchmark instances of the RCPSP are reported.

KEYWORDS. Robust Optimization, Project Management, Risk Mitigation, IND - OR in Industry.

1. Introduction

Large-scale construction projects are often plagued by extreme delays and cost overruns. Complex networks of suppliers, contractors and interdependencies among various areas are compounded by several sources of uncertainty and risk over the duration of such projects. As widely observed, such challenges are common across virtually all industries and geographies.

In this work, we describe a computational study aimed at determining the activities of a project that should be the focus of risk-mitigation measures – in the sense that resource and effort should be put into place so as to ensure that their actual durations equal their original nominal estimates. As detailed ahead, instead of doing so by following traditional Monte Carlo simulation-driven procedures, we opt for a robust optimization approach that accounts for the combined effects of delays in multiple activities – whose interdependencies are intricately defined by complex precedence relationships.

The remaining of the paper is organized as follows: Section 2 defines the basic elements of the problem, provides an overview of relevant prior literature on the topic and discusses some of their limitations. Section 3 proceeds to present the robust optimization model used to assess the criticality of activities and the optimal set of uncertainty mitigation measures. In Section 4 we discuss the computational experiments performed on 480 *PSPLIB* instances for the *Resource Constrained Project Scheduling Problem* (RCPSP) (Kolisch and Sprecher, 1997) and Section 5 concludes with final remarks.

2. Context

The identification of the most critical activities of a project was first investigated by traditional methods such as CPM (Critical Path Method, (Kelley et al., 1959)) and PERT (Project Evaluation and Review Technique, (Fazar, 1959)). In such cases, a critical activity is defined as that for which any delay on its nominal duration leads to a delay of the project's completion date. In order to identify the critical activities, both techniques determine the so-called *critical path*, which is the longest path on the activity-on-arc (*AoA*) network representation of the project – critical activities essentially being then defined as the ones that lie on it.

The *AoA* network is a graph in which arcs (edges) represent the project's activities and nodes denote project's events (the start or end of an activity, for example). An arc representing an activity a links two nodes (events) (i, j) and has a weight which equals the duration of a. A (finish-to-start) precedence relationship between activities a and b ($i.e.$, the constraint that states activity b is only allowed to start once activity a has finished) is enforced by having the terminal node of the arc representing activity a coincide with the start node of the arc representing b, (Demeulemeester, 2002). As circular precedence relationships are not allowed, the *AoA* network is then a directed acyclic graph (*DAG*) and the critical path can be calculated in linear time by a longest-path algorithm (see (Cormen et al., 2001)).

The main difference between PERT and CPM is the detyermination of the durations of activities. PERT calculates the duration of each activity based on three different estimates: optimistic, most likely, and pessimistic durations. Using these three estimates, the method determines a beta probability distribution and then sets the duration of the corresponding activity to be equal to the mean of this distribution. CPM only requires a unique duration estimate for each activity, which is then used to construct the *AoA* network. These methods were very important to the development of research in project management, and they are still widely used in practice. Despite of their importance, its limitations are also discussed and recognized by the project management literature, (Moder et al., 1983).

One of the most important limitations of these methods when applied to real-world projects is the fact that they use deterministic estimates for the duration of activities. As discussed in (Gutjahr et al., 2000), the actual time required to complete an activity usually depends on a number of uncertain factors – such as the productivity of the resources employed to perform the activity – and potential sources of risks that may have an impact on its duration and are not fully known beforehand. A common strategy to dealing with this uncertain environment is to assign probability distributions to the durations of activities. Within this context, a project will likely not have a single deterministic critical path and the original concept of critical activities also needs to be revised. This encouraged researchers to create other activity criticality measures, (Demeulemeester, 2002).

The *path criticality index* (PCI) and the *activity criticality index* (ACI) were the first concepts developed to deal with criticality measures in project networks for which probability distributions are estimated for the durations of activities (stochastic networks), (Williams, 1992). The PCI is defined as the probability that a path p is of longest duration (i.e., the probability of p being a critical path), (Elmaghraby, 2000) while the ACI is the corresponding index for the activities, measuring the probability that each activity is part of a critical path (Williams, 1992). Other indexes, such as the *significance index* (see (Williams, 1992)) and the *cruciality index* (see (Demeulemeester, 2002)), try to capture the correlation between the duration of activities durations and total project duration.

3. Mathematical Model

The activity criticality indexes described in the previous Section assign importance to activities in a isolated fashion and contain no information on how the interdependence among multiple activities may lead to overall delays. While most of the methods developed to calculate these indexes are based on Monte-Carlo simulation (see (Dodin and Elmaghraby, 1985)), our proposed approach tries to capture the importance of the activities in a combined fashion using the robust optimization framework.

Essentially, the basic question that our model attempts to provide an answer to is: what are the α most important activities that one should guarantee to be performed within their nominal duration estimates (for example, by allocating extra resources or implementing risk-mitigation measures) in order to minimize project's duration T under the assumption that at most β activities will assume their worst-case durations? As detailed ahead, the β parameter relates to the general pessimism about the execution of the project's activities while the worst-case duration of each activity refers to a pessimistic estimate for its execution. Under such hypothesis, the model provides the optimal set of activities that should be the target of uncertainty mitigation and also an upper bound on the project's total duration T – which is arguably very useful for project managers.

We present next the details of the two-level optimization model and how it is then converted into a robust model – which, as discussed, is an adaptation of that introduced in (Flach and Mendes, 2013) to the context of criticality assessment.

$$\underset{y}{\text{Min}}\ \underset{z,u}{\text{Max}} \quad T \tag{1}$$

$$s.t.$$

$$T - \sum_{a\in A} \left(\bar{d}_a \cdot u_a + (1-y_a)\cdot \Delta_a \cdot z_a\right) \geq 0 \tag{2}$$

$$\sum_{a\in\delta_0^+} u_a = 1 \tag{3}$$

$$\sum_{a\in\delta_{n+1}^-} u_a = 1 \tag{4}$$

$$\sum_{a\in\delta_i^-} u_a - \sum_{a\in\delta_i^+} u_a = 0 \qquad \forall i \in \{1,...,n\} \tag{5}$$

$$\sum_{a\in A} z_a \leq \beta \tag{6}$$

$$z_a \leq u_a \qquad \forall a \in A \tag{7}$$

$$\sum_{a\in A} y_a \leq \alpha \tag{8}$$

$$y_a, z_a, u_a \in \{0,1\} \qquad \forall a \in A \tag{9}$$

where:

Parameters

- n : number of activities;
- $\bar{d}_a$: nominal duration of activity a;
- Δ_a : potential increase in duration of activity a associated with the uncertainty in its nominal duration estimate;
- β : number of activities that are assumed to take their worst-case duration $\bar{d}_a + \Delta_a$;
- α : number of activities for which uncertainty is mitigated;
- δ_i^+: set of arcs (activities) which start on node i.
- δ_i^-: set of arcs (activities) which end on node i.

Decision variables

- T : project's completion date;
- z_a : decision variable that indicates ($z_a = 1$) activity a assumes its worst-case duration $\bar{d}_a + \Delta_a$;
- u_a: decision variable that indicates ($u_a = 1$) activity a lies on the path with maximum duration;
- y_a : decision variable that indicates the implementation ($y_a = 1$) of a risk-mitigation measure associated with activity a so as to ensure its duration equals $\bar{d}_a$;

There are not readily available techniques that allows us to solve the problem above in its current form so in order to circumvent such difficulty, we again follow the approach discussed in (Flach and Mendes, 2013). The proposed reformulation essentially determines the enumeration os all paths from source node 0 to sink node $n + 1$ – which denote, respectively, the initial and final milestones of the project – and the replacement of the inner maximization problem by the set of constraints (13).

$$\underset{y}{\text{Min}} \quad T \tag{10}$$

$$s.t. \tag{11}$$

$$T \geq t_p \qquad \forall p \in P \tag{12}$$

$$t_p \geq \underset{\sum z_a \leq \beta}{\underset{z}{\text{Max}}} \sum_{a \in p} \left(\bar{d}_a + (1 - y_a) \cdot \Delta_a \cdot z_a\right) \qquad \forall p \in P \tag{13}$$

$$\sum_{a \in A} y_a \leq \alpha \tag{14}$$

$$y_a, z_a \in \{0, 1\} \qquad \forall a \in A \tag{15}$$

where P is the set of all paths from source node 0 to sink node $n + 1$ and t_p is the duration of a particular path p.

Though the model cannot yet be solved in a straigthforward manner, it allows for the application of the techniques presented in (Bertsimas and Sim, 2004): the maximization problem on the right-hand side of each of the constraints defined in (13) may be substituted by the objective function of its dual while the dual feasibility constraints are incorporated into the outer minimization problem. Once this transformation is carried out, we are left with a mixed integer linear programming problem which can be solved by commercially available solvers.

4. Computational Results

We performed our experiments on project networks extracted from the 480 *PSPLIB* instances of the *Resource Constrained Project Scheduling Problem* (RCPSP) with $n = 30$ jobs (or activities), (Kolisch and Sprecher, 1997). From this set of instances we used the information about the duration d_a and precedence relationships and arbitrarily defined worst-case duration estimates of each activity a.

For every instance, we solved our model for each combination of α and β where $\alpha \leq \beta \leq n$. Tests were performed on a Intel Core i5-3360M PC with 4 cores of 2.80GHz and 8 GB of RAM. The model was implemented using the programming language Python and solved by IBM(R) ILOG(R) CPLEX(R) 12.5.0.0.

In Figure 1, we analyze how the increase on the number of mitigated activities α impacts the delay relative to the original project duration, given a fixed assumption on the number of activities which take on their worst-case duration – in particular, the chart displays results for each combination of $(\beta, \alpha) \in \{(10, 2), (10, 4), (10, 6), (10, 8)\}$. Given that T represents the original total project duration and T_r represents the robust estimate of total project duration (i.e., the total duration of the project as per the result of the optimization model), the y-axis represents the delay expressed in percentage points ($D = 100 * \frac{(T_r - T)}{T}$) for each of the instances represented on the x-axis (sorted in ascending order of delay). One

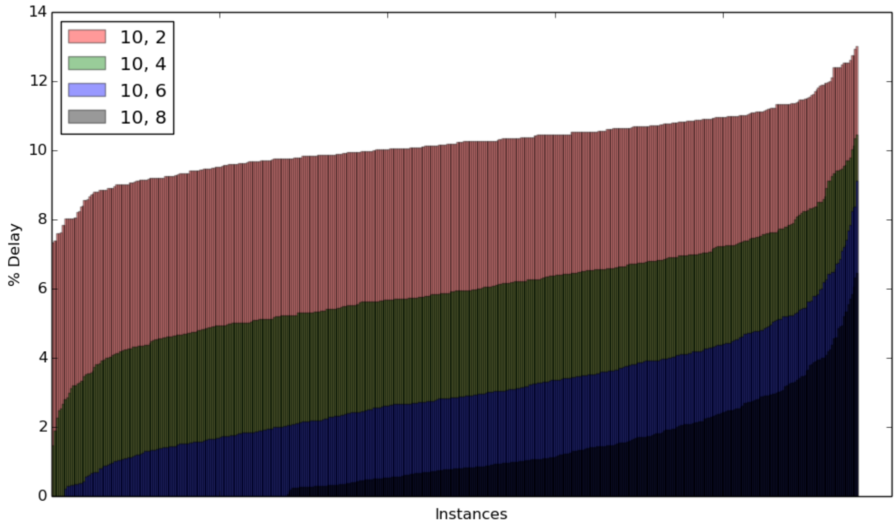


Figure 1. Ordered total percent delay by instance for $\beta = 10$ and $\alpha = \{2, 4, 6, 8\}$.

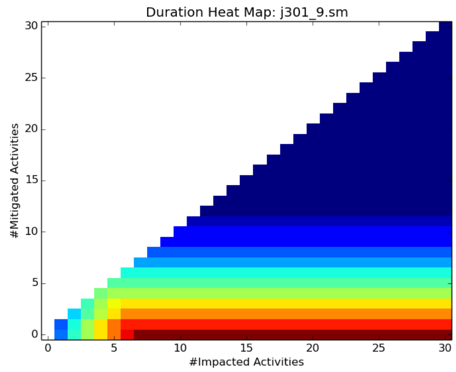
can notice a significant decrease on delays as α increases, which suggests how important it is to correctly select the set of activities to have theirs risks mitigated.

Another important fact noticed in Figure 1 is that for the instances which present higher values of D, increasing α has a progressively lower impact on the actual decrease of the delay. This is evidence that in complex project networks, it becomes increasingly difficult to mitigate the impact of simulataneous delays in multiple activities. To analyze this fact and the quality of our approach, we selected two representative instances: *j301_9.sm* and *j3025_9.sm*. The first instance (*j301_9.sm*) represents the instances where it's easier to determine the best combination of activities to mitigate, that is, the instances where a traditional critical path analysis could be used with success to perform this task. On the other hand, instance *j3025_9.sm* represents the case of complex networks, where the traditional critical path analysis would be expected to perform poorly.

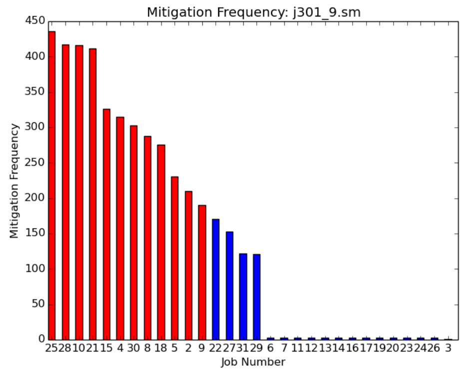
Instance	β	α	T	T_r	D	*Mitigated Activities*
j301_9.sm	10	2	42	47.02	11.95%	[**25**, **28**]
j301_9.sm	10	4	42	45.60	8.59%	[**10**, **21**, **25**, **28**]
j301_9.sm	10	6	42	44.51	5.97%	[**8**, **10**, **15**, **21**, **25**, **28**]
j301_9.sm	10	8	42	43.41	3.36%	[**4**, **8**, **10**, **15**, **18**, **21**, **25**, **28**]
j3025_9.sm	10	2	50	56.27	12.55%	[5, **28**]
j3025_9.sm	10	4	50	54.70	9.41%	[4, 5, **7**, **28**]
j3025_9.sm	10	6	50	54.17	8.35%	[5, **7**, 10, **12**, 26, **28**]
j3025_9.sm	10	8	50	53.23	6.47%	[4, 5, **7**, 10, **12**, 26, **28**, **29**]

Table 1. Results for instances j301_9.sm and j3025_9.sm with $\beta = 10$ and $\alpha = \{2, 4, 6, 8\}$.

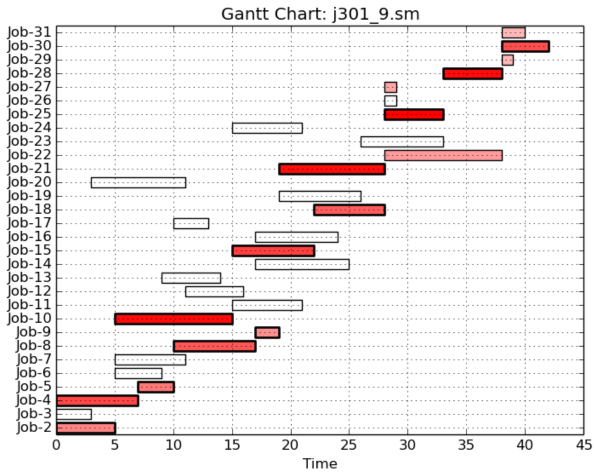
Table 1 details the results of the abovementioned selected instances. In the last column, *Mitigated Activities*, we show the set of activities' (or jobs') which are selected as part of the optimal solution obtained by our model for the corresponding instance and values of β and α. The activities in bold are those that are part of a critical path ($i.e.$, the activities that would traditionally be defined as critical). We first notice that, as α increases, decreases in delay for *j301_9.sm* are higher than those for *j3025_9.sm*. Second, all the activities on the solutions for *j301_9.sm* are critical activities, while for *j3025_9.sm* only half of them are. These results show that in complex cases the traditional critical path analysis cannot adequately capture the combined effects of worst-case durations, highligthing the main advantage of our approach.



(a) Duration Heat Map



(b) Mitigation Frequency



(c) Gantt Chart

Figure 2. Result graphics for the j301_9.sm instance.

We next analyzed all the results obtained for each combination of $\alpha \leq \beta \leq 30$ for both instances. In order to accomplish that, we built three different charts to summarize the results of all the different combinations for which we solved the model. Figures 2 and 3 present the charts for *j301_9.sm* and *j3025_9.sm*, respectively:

1. The **Duration Heat Map** chart captures the duration result of our model for each combination of $\alpha \leq \beta \leq 30$. The x-axis represents the number of impacted activities (β) while the y-axis represents the number of mitigated activities (α). Colors are assigned based on the corresponding duration in each point (combination of α and β) following a heat-scale ranging from dark-blue (no delay) to dark-red (maximum delay).

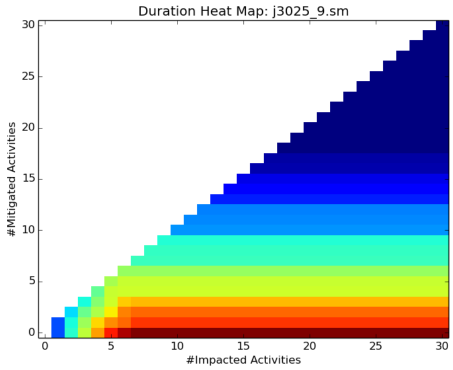
2. The **Mitigation Frequency** chart displays the number of times that each activity (job) is on the optimal solution of the tested combinations ($\alpha \leq \beta \leq 30$), that is, the frequency that the activity was selected to be mitigated. On the x-axis we have the activity numbers ordered by frequency, while on the y-axis we have the corresponding frequencies. Red bars indicate critical activities, while blue bars are used for the non-critical ones. This is somewhat analogous to the evaluation performed in the calculation of an activity's criticality index but aims at providing evidence as to how the traditional analysis could be expected to perform.
3. **Gantt Chart** shows the original schedule of the activities, respecting the precedence relations and with its durations represented by the bar sizes. The critical activities are highlighted with borders in bold. The redness of each activity is assigned proportionally to its mitigation frequency.

The duration heap map of *j301_9.sm* (Figure 2(a)) shows that on a large portion of the tested combinations the mitigation was capable of avoid any delay. Actually, for all combinations with $\alpha \geq 12$ it's possible to maintain the original project duration. As we could expect, 12 is exactly the number of critical activities. Another important fact is showed by the mitigation frequency graphic 2(b): all the critical activities have a higher mitigation frequency than any non-critical activity.

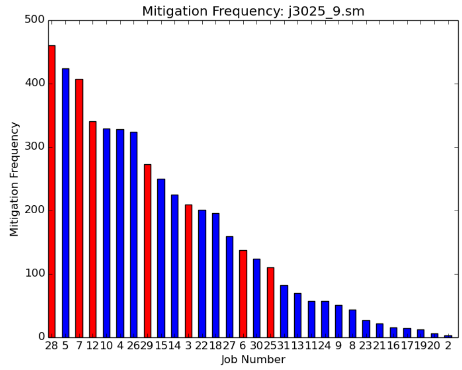
By looking at the graphs for instance *j3025_9.sm* in Figure 3, we also notice the same behaviour that was showed in table 1 for this particular instance. The color variation showed by its duration heat map (Figure 3(a)) as α increases is smoother than the one for the *j301_9.sm*, which shows that the impact of increasing the number of mitigated activities in *j3025_9.sm* is not as effective as in *j301_9.sm*. In fact, in *j3025_9.sm*, only for $\alpha \geq 18$ it is possible to guarantee that there the project will not be delayed – which is surprising given that there are only 7 critical activities. The mitigation frequency graphic (Figure 3(b)) also shows evidence of the complexity of this particular instance: many of the non-critical activities have mitigation frequencies that comparable – and often higher – than those of critical ones. In this case, focusing mitigation measures to avoid delays exclusively on critical activities would not be a good strategy to ensure shortest project duration. Finally, it is noteworthy that the comparison of the two project networks provides no evidence as to which project is more complex, which again suggests that our proposed approach might prove useful across projects with diverse characteristics.

5. Conclusions

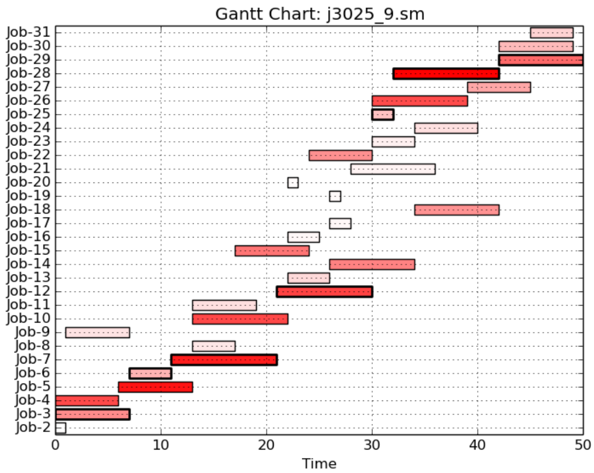
In this work, we described the application of a robust optimization framework to the assessment of the criticality of activities that compose a (potentially large-scale) project. Computational experiments and extensive analysis were performed on 480 benchmark instances of the *Resource Constrained Project Scheduling Problem* from the *PSPLIB* library. Results suggest that our approach is capable of handling complex project networks and provide non-intuitive solutions in which traditional methods would perform poorly, thus warranting future research on possible extensions.



(a) Duration Heat Map



(b) Mitigation Frequency



(c) Gantt Chart

Figure 3. Result graphics for the j3025_9.sm instance.

References

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to algorithms*. MIT press.

Demeulemeester, E. L. (2002). *Project scheduling: a research handbook*, volume 102. Springer.

Dodin, B. M. and Elmaghraby, S. E. (1985). Approximating the criticality indices of the activities in pert networks. *Management Science*, 31(2):207–223.

Elmaghraby, S. E. (2000). On criticality and sensitivity in activity networks. *European Journal of Operational Research*, 127(2):220–238.

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

Flach, B. and Mendes, C. R. (2013). Strategic project planning: Optimal contracting and resource allocation. In *6th International Conference on Management and Control of Production and Logistics*. IFAC.

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.

Kelley, J., E., J., 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., Phillips, C., and Davis, E. (1983). Project management with cpm, pert and precedence diagramming.

Williams, T. (1992). Criticality in stochastic networks. *Journal of the Operational Research Society*, pages 353–357.