Project Scheduling with Success or Failure in Individual Activities

Roel Leus

KULeuven, Department of Decision Sciences and Information Management, Naamsestraat 69, 3000 Leuven,
Belgium, Roel.Leus@econ.kuleuven.be

Bert De Reyck

London Business School, Regent's Park, London NW1 4SA, United Kingdom, bdereyck@london.edu

We examine how to schedule projects in order to maximize their expected net present value when the project activities have a probability of failure and when an activity's failure leads to overall project termination. We formulate the problem, show that it is NP-hard, develop a branch-and-bound algorithm that allows to obtain optimal solutions and provide extensive computational results.

1   Introduction

An important feature of Research-and-Development (R&D) projects is that, apart from the commercial and market risks common to all projects, their constituent activities also carry the risk of technical failure. Therefore, besides projects overrunning their budgets or deadlines and the commercial returns not meeting their targets, R&D projects also carry the risk of failing altogether, resulting in time and resources spent without any tangible return. In this paper, we tackle the problem of scheduling the activities of an R&D project that is subject to technological uncertainty, i.e. in which the individual activities carry a risk of failure, and where an activity's failure results in the project's overall failure. The goal is to schedule the activities in such a way as to maximize the expected net present value of the project, taking into account the activity costs, the cash flows generated by a successful project, the activity durations and the probability of failure of each of the activities.

The model developed in this paper is useful for any R&D setting where activities carry a risk of failure, and is of particular interest to drug-development projects in the pharmaceutical industry, in which stringent scientific procedures have to be followed to ensure patient safety in distinct stages before a medicine can be approved for production. The project may need to be terminated in any of these stages, either because the product is revealed not to have the desired properties or because of harmful side effects. The failure of one of the stages results in overall project termination. As stated by [6], "If a drug candidate fails during the development phase it is withdrawn entirely from further testing. Unlike in the automobile industry, drugs are not modular products where a faulty stick shift can be replaced without throwing the entire car design away. In pharmaceutical R&D, drug design cannot be changed."

2   Problem formulation

We wish to maximize the expected net present value (NPV) of a project by constructing a project schedule specifying when to execute each activity. The final project payoff is only achieved when all activities are successful, and the project is terminated as soon as an activity fails. We focus on the case where all activity cash flows during the development phase are negative, which is typical for R&D projects. Activity success or failure is revealed at the end of each activity. Consequently, each activity will only be started if all the activities scheduled to finish earlier have a positive outcome. Therefore, in the objective function, the activity cash flows are weighted by the probability of joint success of all its scheduled predecessors. We make abstraction of resource constraints and

N	$= \{0, 1, \ldots, n\}$, the set of project activities; $N_i = N\backslash\{i\}$ $(i \in N)$ and $N_{0n} = N\backslash\{0, n\}$
c_i	cash flow of activity $i \in N_n$, non-positive integer; incurred at the start of the activity
C	integer end-of-project payoff, ≥ 0; received at the start of activity n
d_i	duration of activity $i \in N_n$, non-negative integer (positive for $i \in N_{0n}$)
p_i	probability of technical success (PTS) of activity $i \in N_n$
r	continuous discount rate
A	(strict) partial order on N, i.e. an irreflexive and transitive relation, representing technological precedence constraints
s_i	starting time of activity $i \in N$, ≥ 0; starting-time vector $\mathbf{s}$ is a schedule
δ	project deadline

Table 1: Definitions.

duration uncertainty, and consider the success probabilities of the different tasks as independent. The parameters that are used throughout the paper are defined in Table 1.

Without loss of generality, we assume activity 0 to be a dummy representing project initiation, with $c_0 = d_0 = 0$ and $p_0 = 1$, and $(0, i) \in A$ for all $i \in N_0$. Activity n represents project completion and is a successor of all other activities. Activities N_{0n} are referred to as *intermediate* activities; we assume that $d_i > 0$ for $i \in N_{0n}$. A deadline δ on the schedule length is imposed: we require that $s_n \leq \delta$. This deadline is needed because optimization will try to push activity start times to infinity if the optimal expected NPV of a particular problem instance is negative. A second reason for using a deadline is that it allows to examine the impact of schedule length on the quality of the schedule.

In order to formulate the problem we wish to solve, we define the additional variables

$$q_i(\mathbf{s}) = \prod_{\substack{k \in N: \\ s_k + d_k \leq s_i}} p_k$$

associated with activities $i \in N_0$. Remark that $q_n(\mathbf{s})$ is a constant, independent of the schedule; we write $q_n \equiv q_n(\mathbf{s})$. q_i represents the probability that activity i is executed, and thus needs to be paid for. We now formally state the *R&D-Project Scheduling Problem* or RDPSP:

$$\max \qquad g(\mathbf{s}) = q_n C e^{-r s_n} + \sum_{i=1}^{n-1} q_i(\mathbf{s}) c_i e^{-r s_i} \tag{1}$$

subject to

$$
\begin{aligned}
s_i + d_i &\leq s_j & \forall (i, j) \in A \\
s_n &\leq \delta \\
s_i &\geq 0 & \forall i \in N
\end{aligned}
$$

In the objective function $g()$, each activity cash flow c_i is weighted with two factors, namely with $q_i(\mathbf{s})$, the probability of joint success of all predecessors in time, and with a discount factor $e^{-r s_i}$, dependent on the starting time s_i of activity i.

3 Properties

Theorem 1. *If $r = 0$ and $\delta \geq \sum_{i \in N_n} d_i$ then an optimal feasible schedule exists without activities scheduled in parallel.*

The proofs of the theorems appear in [4]. Intuitively, the theorem says that when money has no time value, it is a dominant choice to perform all tasks sequentially. Theorem 1 allows us to establish ties with the literature on sequential testing. We define problem LCT ('*least-cost testing*') as problem RDPSP whose solution space is restricted to schedules that impose a complete order on N; remark that LCT is not a sub-problem of RDPSP since we restrict the set of solutions and not the input parameters.

Without dummy start and end (and so without final project payoff), a number of special cases of LCT with $r = 0$ can be solved in polynomial time. If $A = \varnothing$ then each schedule that sequences the activities in non-increasing order of $c_i/(1 - p_i)$ is optimal. One of the earliest references for this result seems to be [8]; another source is [2]. A polynomial-time algorithm for LCT also exists when $G(N, A)$ consists of a number of *parallel chains* (see [3]). Based on [9] it can be shown that the problem is also solvable in polynomial time when $G(N, A)$ is *series-parallel*.

The foregoing results carry over to RDPSP when $\delta \geq \sum_{i \in N_n} d_i$ and $r = 0$. However, the incorporation of precedence constraints taking the form of an arbitrary acyclic digraph $G(N, A)$ results in an NP-hard problem:

Theorem 2. RDPSP *is* NP-*hard in the ordinary sense, even if $r = 0$, $C = 0$, $\forall i \in N_{0n} : d_i = 1$, and $\delta \geq \sum_{i \in N_n} d_i$.*

Corollary 1. LCT *is ordinarily* NP-*hard under the same conditions.*

This corollary settles what is said to be an open problem in [9] and [11].

4 A branch-and-bound algorithm

For an arbitrary relation E on N, define $\mathcal{S}(E) = \{\mathbf{s} \in \mathbb{R}_{\geq}^{n+1} : s_i + d_i \leq s_j, \forall (i, j) \in E\}$, which is a convex polyhedron ($\mathbb{R}_{\geq}$ denotes the set of positive real numbers). $\mathcal{S}(E)$ is non-empty if and only if the corresponding precedence graph $G(N, E)$ is acyclic. The set of feasible schedules for RDPSP is $\{\mathbf{s} \in \mathcal{S}(A) : s_n \leq \delta\}$. Clearly, if $A \subseteq E$ then $\mathcal{S}(E) \subseteq \mathcal{S}(A)$. If $A \subseteq E$ and $G(N, E)$ is acyclic, we say that E is a *feasible extension* of A. For a given schedule $\mathbf{s}$, we define the schedule-induced strict order $R(\mathbf{s}) = \{(i, j) \in N \times N | i \neq j \wedge s_i + d_i \leq s_j\}$, which corresponds to the precedence constraints implied by $\mathbf{s}$ (see e.g. [1, 10]).

RDPSP is solved in two phases. In the first phase we produce a feasible extension E of A, which yields values

$$y_i(E) = \prod_{(k,i) \in E} p_k$$

for activities $i \in N_{0n}$. We then substitute values $y_i(E)$ for $q_i(\mathbf{s})$ in the objective function (1) for each $i \in N_{0n}$, and optimize $g(\mathbf{s})$ subject to the constraints that $\mathbf{s} \in \mathcal{S}(E)$ and $s_n \leq \delta$. If we implicitly or explicitly enumerate all feasible extensions E of A, we are guaranteed to find an optimal schedule for RDPSP, since for each feasible schedule $\mathbf{s} \in \mathcal{S}(A)$ it holds that $\mathbf{s} \in \mathcal{S}(R(\mathbf{s}))$, and $R(\mathbf{s})$ extends A; a corresponding relation E is called an *optimal feasible extension*. This enumeration process is embedded into a branching procedure, which, in combination with upper bounds on the best objective-function value reachable from a given node in the resulting search tree, leads to a branch-and-bound (B&B) procedure that allows us to find optimal solutions.

The second phase (optimization after substitution of the values y_i), to be examined for each feasible extension E, amounts to project scheduling with NPV objective without resource constraints (see [7]). In this case, the scheduling problem is easily solved because all intermediate cash flows are non-positive: each activity can be scheduled to end at the earliest of the starting times of its

successors in E. Depending on whether the corresponding expected NPV is positive or negative, we set $s_0 = 0$ or $s_n = \delta$, respectively.

In our presentation we will report on computational results for the B&B-algorithm on a number of sets of test instances generated by RanGen [5].

References

[1] M. Bartusch, R.H. Möhring, F.J. Radermacher (1988), Scheduling project networks with resource constraints and time windows, *Annals of Operations Research* **16**, 201 – 240.

[2] R. Butterworth (1972), Some reliability fault-testing models, *Operations Research* **20**, 335 – 343.

[3] S.Y. Chiu, L.A. Cox Jr., X. Sun (1999), Optimal sequential inspections of reliability systems subject to parallel-chain precedence constraints, *Discrete Applied Mathematics* **96–97**, 327 – 336.

[4] P. Crama, B. De Reyck, Z. Degraeve, R. Leus (2005), Managing technology risk in R&D project planning: Optimal timing and parallelization of R&D activities, Research Report 0501, Department of Applied Economics, KULeuven.

[5] E. Demeulemeester, M. Vanhoucke, W. Herroelen (2003), A random generator for activity-on-the-node networks, *Journal of Scheduling* **6**, 13 – 34.

[6] O. Gassmann, G. Reepmeyer, M. von Zedtwitz (2004), *Leading Pharmaceutical Innovation, Trends and Drivers for Growth in the Pharmaceutical Industry,* Springer-Verlag, Berlin Heidelberg New York.

[7] W.S. Herroelen, P. Van Dommelen, E.L. Demeulemeester (1997), Project network models with discounted cash flows: a guided tour through recent developments, *European Journal of Operational Research* **100**, 97 – 121.

[8] L.G. Mitten (1960), An analytic solution to the least cost testing sequence problem, *Journal of Industrial Engineering* **11**, 17 – 17.

[9] C.L. Monma, J.B. Sidney (1979), Sequencing with series-parallel precedence constraints. *Mathematics of Operations Research* **4**, 215 – 224.

[10] K. Neumann, C. Schwindt, J. Zimmermann (2003), *Project Scheduling with Time Windows and Scarce Resources: Temporal and Resource-Constrained Project Scheduling with Regular and Nonregular Objective Functions.* Springer-Verlag, 2nd edition.

[11] T. Ünlüyurt (2004), Sequential testing of complex systems: a review, *Discrete Applied Mathematics* **142**, 189 – 205.