Tijdschrift voor Economie en Management
Vol. LII, 3, 2007

A New Challenge in Project Scheduling: The Incorporation of Activity Failures

by B. De Reyck, Y. Grushka-Cockayne and R. Leus

Bert De Reyck
Department of Management Science & Innovation, University College London and Department of Management Science & Operations,
London Business School

Roel Leus
Department of Decision Sciences and Information Management,
K.U.Leuven

Yael Grushka-Cockayne
Department of Management Science & Innovation,
London Business School

ABSTRACT

The goal of this article is to survey the relevant literature on project scheduling with possible activity failures from a number of different disciplines, and to distill from these sources the formulation of a general optimization problem, the further study of which we would like to foster among the scheduling community. The model has been formulated with R&D projects in mind, but its study may be useful also for developing scheduling methods in other contexts. We discuss a number of different aspects of this task selection and scheduling model with task failures by means of a number of illustrative examples.

Keywords: project management; scheduling; risk; research and development; activity failures.

I. INTRODUCTION

Research and Development (R&D) companies, whose core business relies on innovation, face a daily challenge of planning and surviving in an uncertain environment. Not only do uncertainties exist regarding the benefits and costs that their business entails, but they must also bear the risk of technical failure of their R&D activities.

In this paper, we focus on a single firm facing an R&D project with many possible development patterns. The project is composed of a collection of R&D activities with well-defined precedence relationships, where each activity is characterized by a cost, a duration and a probability of success. The successful completion of an activity corresponds to a technological discovery or a scientific breakthrough. Additionally, for obtaining certain results, more than one alternative may exist, and these alternatives can be pursued either in parallel or sequentially; the decision maker might also have discretion over the selection of the alternatives to pursue. The objective 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 goal of this article is to survey the relevant literature on project scheduling with possible activity failures from a number of different disciplines, and to distill from these sources the formulation of a general optimization problem, the further study of which we would like to foster among the scheduling community. We discuss a number of different aspects of this task selection and scheduling model for R&D projects with task failures by means of a number of illustrative examples. Although the model has been formulated with R&D projects in mind, its study may be useful also for developing scheduling methods in other contexts. The remainder of this article is organized as follows. Related work is discussed in Section II. A problem formulation and discussion of some properties is given in Section III, after which four example projects are briefly discussed in Section IV, demonstrating some of the open challenges in this area. Finally, a summary and outlook on further research are given in Section V.

II. RELATED WORK

In this section, we provide references to a number of different sources in the literature related to this paper, subdivided into four categories: technology management, project scheduling, discrete optimization and sequential testing, and chemical engineering.

1. *Technology management*

Parallel development of alternative technologies, where the decision to be made is whether to fund one or more tasks with the same objective in each stage of the project, is studied in Abernathy and Rosenbloom (1969), Bard (1985) and Krishnan and Bhattacharya (2002), and a generic representation of multi-stage R&D problems is provided in Lockett and Gear (1973). In these sources, the project outcome is modeled as a continuous random variable (for instance production cost per unit of the product that is designed); in this article, we focus only on dichotomous outcomes (success or failure). The issue of parallel versus sequential scheduling of project activities has been addressed, among others, by Eppinger et al. (1994), Krishnan et al. (1997) and Roemer and Ahmadi (2004). This topic is also closely related to concurrent engineering, a systematic approach to the integrated, concurrent design of products and their related processes (Hill, 2003). In these latter sources, however, the set of activities to be performed is fixed, and activity failure is not an issue. In all the aforementioned sources, the precedence structure of R&D projects is also limited to sequential R&D stages only.

Teunter and Flapper (2006) investigate issues similar to ours but in a flow production environment. They state

> *"So the relevant question, especially for the lengthy tests with uncertain results that occur in the process industries, is not how, when or which fraction to test, but at what stage of the testing to start further processing/distribution."*

Teunter and Flapper identify all possible alternatives based on the set of tests and their durations, for one particular case in a pharmaceutical company, and evaluate each of the alternatives on a number of performance criteria. Somewhat similar considerations were made by Ronen and Trietsch (1988) regarding the timing and cost of purchasing orders for materials and components for large projects, but their focus is on timeliness and costs, and not on uncertainty in activity outcomes. Similarly, Zemel et al. (2001) focus on the optimal timing of support activities for R&D tasks of variable length: a 'double expenditure' policy is compared with the conservative 'delayed investment' policy, under which all the routine engineering activities are delayed until the risky R&D efforts culminate in a breakthrough.

Dahan (1998) investigates the situation where product or process developers may conduct prototyping experiments to test the technical feasibility of design alternatives. Individual tests are Bernoulli experiments with known rewards, costs and success probabilities, and the author investigates parallel, sequential and so-called 'hybrid' policies (the latter allowing for partial overlap). Dahan (1998) shows that for high discount rates the optimal policy is a hybrid sequential/parallel policy, whereas with low discount rates, when time-to-market is not important, there is no economic advantage to building prototypes in parallel and therefore a purely sequential policy will be optimal. Dahan and Mendelson (2001) examine a similar framework and develop a closed-form solution for the optimal number of tests under three extreme-value distributions for the expected reward. They model the R&D process of drug development and focus on the testing of different concepts in order to determine which to develop further. This leads to the finding that the optimal number of parallel tests under uncertainty depends not only on the cost of testing and the scale of uncertainty, but also on the upper-tail shape of that uncertainty.

Nelson (1961) focuses on R&D efficiency, i.e. achieving a given objective at minimum cost. He motivates a parallel path strategy by the learning of the characteristics of the activities, i.e. cost, development duration and performance, for which we get better estimates as development advances. Nelson characterizes the optimal number of parallel development activities and observes that parallelism seems most beneficial when the cost of executing several alternatives is relatively small.

Loch et al. (2001) also look into the testing of a number of design alternatives, among which the 'most preferred' solution needs to be identified based on a number of tests. Loch et al. (2001) find that parallel processing dominates when the cost of delay increases relative to the cost of testing and development. However, they also highlight that as the uncertainty in the information provided by the tests increases, the value of parallel processing decreases.

Ding and Eliashberg (2002) examine the so-called 'pipeline problem': since New Product Development (NPD) projects may fail in each stage, multiple projects are started simultaneously in order to increase the likelihood of having at least one successful product. The motivation underlying this model is rather similar to the foregoing, in that in both cases the success probability of the overall undertaking (either one project or the pipeline) is to be maximized.

2. *Project scheduling*

To the best of our knowledge, the only treatment of scheduling with activity failures in the area of project scheduling is given in De Reyck and Leus (2007), where project success is achieved only if all individual activities succeed. The literature on *deterministic* project scheduling, however, is vast and contains numerous methods and algorithms for producing project schedules; for recent comprehensive overviews of the literature we refer to Demeulemeester and Herroelen (2002) and Neumann et al. (2003). For specific surveys of scheduling with NPV objective, we refer to Herroelen et al. (1997) and Padman et al. (1997). The incorporation of *uncertainty* into project planning and scheduling has also resulted in numerous research efforts, particularly focusing on uncertainty in the activities' duration or cost; for a recent survey, see Herroelen and Leus (2005). None of these literature reviews, however, discuss models that incorporate technological uncertainty in the form of stochastic-success activities.

3. *Discrete optimization and sequential testing*

Closely related to the setting of this paper is that of Weitzmann (1979), who describes an optimal search procedure for obtaining maximum reward from a number of independent testing efforts; only sequential testing is considered. Granot and Zuckerman (1991) also examine sequencing for R&D projects with success or failure in individual activities but only consider sequential stages, where each stage is interrupted as soon as one activity in that phase is successful; they also look into learning effects. Denardo et al. (2004) consider R&D projects that are successful if a successful path of edges from stem to leaf in a forest is found.

The 'least-cost fault-detection problem' has been treated under multiple similar names by, among others, Boothroyd (1960), Mitten (1960), and Monma and Sidney (1979); a variant of this problem was tackled by integer programming in Wagner and Davis (2001). The goal is to sequentially perform (testing) activities until one fails (i.e., the system is 'defective') or until all components pass their test (i.e., the system successfully completes inspection); associated with each activity/component test is a testing cost and a probability of passing the test. Extensions toward so-called 'k-out-of-n' reliability systems, where the system works if and only if k or more of its components work, are studied by Butterworth (1970), Ben-Dov (1981), Boros and Ünlüyurt (1997) and Chiu et al. (1999); a comprehensive review is provided in Ünlüyurt (2004). In the literature on reliability systems, the term *series system* is often used to refer to an n-out-of-n system, while a *parallel system* is a 1-out-of-n system. The setting is rather similar to ours, apart from the fact that (1) overall success or failure is always determined only by the *number* of successful activities while it may be a more complex function of the individual activity outcomes in this paper (see Section III); (2) only sequential testing is allowed; and (3) no discounting is considered (it turns out that (2) and (3) are closely connected).

4. *Chemical engineering*

Another interesting source of relevant literature stems from the discipline of chemical engineering, most notably the work by Grossmann and his colleagues. Schmidt and Grossmann (1996) initiated the work on scheduling failure-prone NPD testing tasks when also non-sequential testing is admitted. They point out that in many industries, including the chemical and pharmaceutical sectors, a number of the tasks involved in producing a new product are regulatory requirements such as environmental and safety tests. The failure of a single required test may prevent a potential product from reaching the marketplace. Extensions of this basic model towards the incorporation of resource constraints (Jain and Grossmann, 1999), correlation between activities' parameters (Choi et al., 2004) and investment in new resources (Maravelias and Grossmann, 2004) have also been investigated. Rogers et al. (2002) study portfolio selection for projects with sequential stages, where the outcome of each stage is a discrete random variable whose value evolves in discrete time. Apart from Choi et al. (2004), the solution methodology is mathematical programming; in order to be able to produce adequate solutions for medium-sized projects, however, the cited sources resort to solving approximations of the original problem formulation.

An informal overview of the importance of including the possibility of technical failure into planning is given in Blau et al. (2000), who focus especially on the pharmaceutical industry. DiMasi (2001) also refers to economic, efficacy, safety and 'other' reasons for cutting projects. In this paper, we will mainly refer to 'technical' success of products. More information on success probabilities in the pharma sector can be found in Zipfel (2003). Finally, a broader overview of key issues and strategies for optimization in pharmaceutical supply chains is provided by Shah (2004), and Papageorgiou et al. (2001) present an optimization model for the pharmaceutical industry that pursues also more strategic objectives.

III. PROBLEM FORMULATION AND PROPERTIES

In the following paragraphs, we provide a number of definitions, a formal problem statement and a brief discussion of two special cases, namely single-activity-module projects and single-module projects.

1. *Definitions*

We consider the planning of one project in isolation, and we wish to maximize the expected net present value (expected NPV, eNPV) of this project by constructing a project schedule specifying when to execute each activity. A project consists of a set of *modules* $M = \{0, 1, \ldots, m\}$ containing one or more *activities,* each of which should be executed without interruption; a module $i \in M$ is a set of activities N_i that pursue a similar target. We denote the set of all activities by $N = \bigcup_{i \in M} N_i$; $n = |N| - 1$. A is a (strict) partial order on M, i.e. an irreflexive and transitive relation, which represents technological precedence constraints. (Dummy) modules 0 and m represent the start and the end of the project, respectively; they are the (unique) least and greatest element of the partially ordered set (M, A), and are assumed to contain only one (dummy) activity, indexed by 0 and n, respectively. We associate the directed acyclic graph $G(M, A)$ with the partially ordered set (M, A). On the activities within each module i, we also impose a partial order B_i.

Activity $i \in N$ has duration $d_i \in \mathbb{N}$; we assume that $d_i = 0$ for $i = 0$, n and $d_i > 0$ otherwise. Each activity $i \in N \backslash N_m$ has a probability of technical success (PTS) p_i; we assume that $p_0 = 1$. We make abstraction of resource constraints and duration uncertainty, and consider the outcomes of the different tasks to be independent. Quantity c_i represents the cost (cash outflow) of activity $i \in N \backslash N_m$, which is a non-positive integer; this cost is incurred at the start of the activity. Overall project success generates an end-of-project payoff $C \geq 0$, which (if the project succeeds) is received at the start of activity n. The final project payoff is only achieved when all modules are successful. Module $i \in M$ is said to be successful if at least one of its constituent activities succeeds. If we define u_i to be the probability of success of module i, then

$$u_i = 1 - \prod_{j \in N_i} \left(1 - p_j\right).$$

The probability π of project success equals $\prod_{i \in M \backslash \{m\}} u_i$.

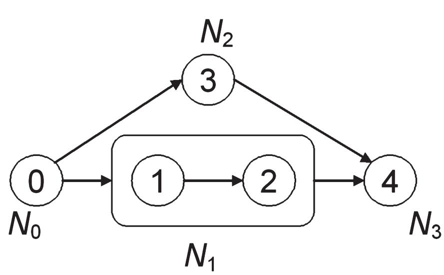
For an illustration of these definitions, we refer the reader to Figure 1. The project consists of five activities, $N = \{0,1,2,3,4\}$, where 0 and $n = 4$ are dummies. There are four modules, so $m = 3$: $N_0 = \{0\}$, $N_1 = \{1,2\}$, $N_2 = \{3\}$ and $N_3 = \{4\}$, two of which are dummies. The modules that consist of a single activity are not separately distinguished in the figure. The module order is $A = \{(0,1), (0,2), (0,3), (1,3), (2,3)\}$; the activity order for module 1 is $B_1 = \{(1,2)\}$ (Figure 1 actually shows the transitive reduction of A, and also the single element of B_1). The success probabilities for the non-dummy modules are

$$u_1 = 1 - (1 - p_1)(1 - p_2) \quad \text{and} \quad u_2 = p_3.$$

Consequently, the project payoff is achieved with probability

$$\pi = u_1 u_2 = p_3(1 - (1 - p_1)(1 - p_2)).$$

FIGURE 1
A first example project



We define a *state vector* as an $(n-1)$-component binary vector $\mathbf{x} = (x_1,...,x_{n-1})$, with one component associated with each non-dummy activity in N. The *system success function* is the boolean mapping $\sigma : \mathbb{B}^{n-1} \rightarrow \mathbb{B}$ characterizing the states in which the project is successful, i.e.

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{in case of project success, and} \\ 0 & \text{otherwise} \end{cases}$$

As mentioned above, $\sigma(\mathbf{x}) = 1$ if and only if each module contains at least one successful activity, so

$$\sigma(\mathbf{x}) = \left(\bigvee_{i \in N_1} x_i\right) \wedge \ldots \wedge \left(\bigvee_{i \in N_{m-1}} x_i\right).$$

Quantity $s_i \geq 0$ represents the starting time of activity i; we call $(n+1)$-vector $\mathbf{s} = (s_0, s_1, ..., s_{n-1}, s_n)$ a schedule. For convenience, we associate an artificial completion time $e_i(\mathbf{s})$ with each module i, as follows (here and later, we omit the arguments if no misinterpretation is possible): $e_i = \min\left\{+\infty; \min_{j \in N_i | x_j = 1}\left\{s_j + d_j\right\}\right\}$. Clearly, if the second min-operator optimizes over the empty set then e_i takes the value $+\infty$. For a given state vector $\mathbf{x}$, we say that a schedule $\mathbf{s}$ is *feasible* if the following conditions are fulfilled, with δ a deadline on the project length:

$$e_i \leq s_j \qquad \forall (i,k) \in A,\ \forall j \in k \tag{1}$$

$$s_i + d_i \leq s_j \qquad \forall k \in M,\ \forall (i,j) \in B_k \tag{2}$$

$$s_n \leq \delta \quad \text{or} \quad s_n = +\infty \tag{3}$$

An activity's starting time equal to infinity corresponds with not executing the activity (and therefore not incurring any related expenses, or in case of activity n, not receiving the project payoff).

We compute the NPV for schedule $\mathbf{s}$ and state vector $\mathbf{x}$ as

$$f(\mathbf{s}, \mathbf{x}) = \sigma(\mathbf{x}) C e^{-rs_n} + \sum_{i=1}^{n-1} c_i e^{-rs_i}$$

with r a continuous discount rate, applied to take into account the time value of money. In the context of this article, we assume $e^{-0 \cdot \infty} = 0$.

If we let X_i represent the Bernouilli random variable (r.v.) with parameter p_i of success ($p_i = 1$) or failure ($p_i = 0$) of activity *i,* then $\sigma(\mathbf{X})$, with $\mathbf{X} = (X_1,...,X_{n-1})$, is also a Bernouilli r.v. with success probability π. The realization of X_i is known at the end of activity *i.*

In the setting of this article, and in line with Igelmund and Radermacher (1983), Möhring (2000) and Stork (2001), who study project scheduling with resource constraints and stochastic activity durations, the execution of a project can best be seen as a dynamic decision process. A solution is a *policy* Π, which defines *actions* at *decision times.* Decision times are typically $t = 0$ (the start of the project) and the completion times of activities; a tentative next decision time can also be specified by the decision maker. An action can entail the start of a set of activities that is precedence feasible (no constraints in (1) or (2) are violated). A schedule is thus constructed gradually through time. Next to the input data of the problem instance, a decision at time t may only use information on realizations of components of $\mathbf{X}$ that has become available before or at time t; this requirement is often referred to as the *non-anticipativity constraint.* Project completion occurs at time s_n; note that not all activities need to be completed by time s_n, nor that the realization of all X_i needs to be known.

A scheduling policy Π may alternatively be interpreted as a function $\mathbb{R}_{\geq}^{n-1} \rightarrow \mathbb{R}_{\geq}^{n+1}$ that maps given samples $\mathbf{x}$ of activity success or failure to vectors $\mathbf{s}(\mathbf{x}; \Pi)$ of feasible activity starting times (schedules). Our objective is to select a policy Π^* within a specific class that maximizes $E[f(\mathbf{s}(\mathbf{X}; \Pi), \mathbf{X})]$, with $E[\cdot]$ the expectation operator with respect to $\mathbf{X}$; we write $E[f(\Pi)]$, for short.

The generality of this problem statement suggests that optimization over the class of all policies will often be computationally intractable; a resulting policy is referred to as a globally optimal policy. One therefore usually restricts the attention to subclasses that have a simple combinatorial representation and where decision points are limited in number. Remark also that in deterministic scheduling, NPV is a non-regular measure of performance: starting activities as early as possible is not necessarily optimal.

For the example that was introduced in Section 1, assuming that δ is non-restrictive and that $d_3 \leq d_1$, one possible policy Π_1 is the following: start the project at time 0 ($s_0 = 0$), and immediately initiate both activities 1 and 3 ($s_1 = s_3 = 0$). If $X_3 = 0$ then abandon the project:

set $s_2 = s_4 = +\infty$. Otherwise, observe the outcome of activity 1 and either complete the project at time $s_4 = d_1$ if $X_1 = 1$, or continue with activity 2 at time $s_2 = d_1$. Subsequently, let $s_4 = d_1 + d_2$ if $X_2 = 1$, otherwise $s_4 = +\infty$. Represented as a function, Π_1 can be written as follows:

$$\Pi_1 : (x_1, x_2, x_3) \rightarrow (0, 0,\ d_1 + \infty \cdot x_1 + \infty \cdot (1 - x_3), 0,$$
$$\min\{d_1 + \infty \cdot (1 - x_1 x_3);\ d_1 + d_2 + \infty \cdot (1 - x_2 x_3)\}).$$

In the context of this article, we let $0 \cdot \infty = 0$.

In Section 3 we look into the special case where each module contains only one activity, while Section 4 deals with the case where the entire project consists of a single module.

3. *Single-activity-module projects*

In this section, we investigate the special case where each module contains only one activity, which is equivalent to a 'series' reliability system; De Reyck and Leus (2007) referred to this setting as the 'R&D-Project Scheduling Problem' (RDPSP), and this section is largely based on that article. In a series system, all activities need to succeed in order to achieve the project payoff. Failure of one of the project's tasks results in overall project termination, which is actually not required (one can also continue the payments for remaining tasks), but it is clearly a dominant decision. A series system is of particular interest in modeling 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. As stated by Gassmann et al. (2004):

> *"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."*

Define an *elementary policy* for a series system as a policy $\Pi(\mathbf{s})$ that takes a deterministic schedule $\mathbf{s}$ as input and adheres to (imitates) $\mathbf{s}$ as long as all completed activities are successful, but abandons the project (does not execute the remaining activities) as soon as it is known that one or more activities have failed. The following result was implicitly assumed by De Reyck and Leus (2007), but never explicitly stated:

Lemma 1. *An optimal elementary scheduling policy is globally optimal for series systems.*

The proofs appear in the appendix.

It is shown in De Reyck and Leus (2007) that the series problem is NP-hard in the ordinary sense, even if $r = 0$, $\forall i \in N \backslash \{0,n\} : d_i = 1$, and $\delta \geq \sum_{i \in N \backslash \{n\}} d_i$. In light of this complexity status, an exact algorithm with better than exponential time complexity is unlikely to exist, and De Reyck and Leus (2007) devise a branch-and-bound (B&B) algorithm to implicitly enumerate the solution space.

De Reyck and Leus construct a schedule $\mathbf{s}°$ for an optimal elementary policy $\Pi°$ in two phases. For an arbitrary relation R on M, if $A \subseteq R$ and $G(M, R)$ is acyclic, we say that R is a feasible extension of A. In the first phase, a feasible extension R of A is produced, which yields values

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

for modules $i \in M \backslash \{0,m\}$, which represents the probability that activity i is executed, and thus needs to be paid for. Note that the second equality assumes that modules and activities are indexed similarly, and that $u_k = p_{k°}$ for each module k, with $N_k = \{k°\}$, and normally $k = k°$. In the second phase, expected cash outflows $y_i(R)c_i$ are associated with each activity i, and expected payoff πC with activity n, and a deterministic schedule $\mathbf{s}$ is sought that maximizes

$$\pi C e^{-rs_n} + \sum_{i=1}^{n-1} y_i(R) c_i e^{-rs_i}$$

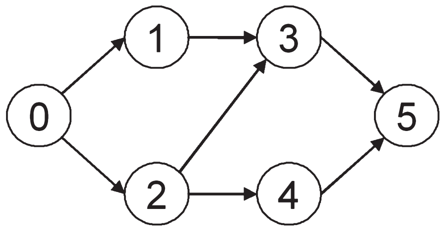
subject to

$$s_i + d_i \leq s_j \qquad \forall (k,l) \in R,\ \{i\} = N_k,\ \{j\} = N_l$$
$$s_n \leq \delta$$

If all feasible extensions R of A are implicitly or explicitly enumerated, it can be shown that we are guaranteed to find a schedule defining an optimal elementary policy; a corresponding relation R is called an optimal feasible extension. This enumeration can be embedded in a B&B procedure. The second phase (deterministic scheduling with expected cash flows) amounts to project scheduling with NPV

FIGURE 2

Precedence graph for an example of a series system



objective without resource constraints (see Herroelen et al., 1997). 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 *R*. Depending on whether the corresponding expected NPV is positive or negative, we set $s_0 = 0$ or $s_n = \delta$, respectively.

We illustrate these computations for series systems by means of an example project. In Figure 2, the precedence network $G(M, A)$ of the modules in the project is represented; we associate one activity with each module, with the same indices, so $n = m = 5$. We use a discount rate $r = 0.01$ and a project payoff $C = 200$; the remaining data are provided in Table 1. A feasible schedule is represented in Figure 3, the eNPV of the (optimal) elementary policy based on this schedule is 37.197983.

4. *Single-module projects*

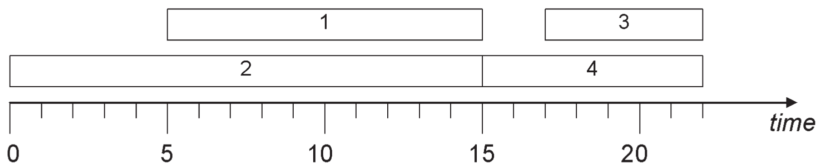
The 'parallel' reliability problem examines the situation where project payoff *C* is obtained as soon as a single activity finishes successfully.

TABLE 1

Costs, durations and PTS for the non-dummy activities in the example project

i	p_i	c_i	d_i
1	0.8	–20	10
2	0.9	–15	15
3	0.85	–10	5
4	0.85	–10	7

FIGURE 3
A feasible schedule



This is an appropriate description of one module in a larger NPD project, where alternative technologies or trials are brought together for reaching the same (module) result. Parallel systems also allow to model so-called *fallback options*: alternative plans devised by management in the event the primary option falters.

In the setting outlined in Section 1, a parallel system actually corresponds with three modules: a dummy start and end, and one non-dummy module containing all non-dummy activities. The following observation is again straightforward: it is a dominant decision to terminate the project as soon as the payoff is obtained. We define an elementary policy $\Pi°(\mathbf{s}°)$ for parallel systems as a policy that takes a deterministic schedule $\mathbf{s}°$ as input and adheres to $\mathbf{s}°$ until the project payoff is obtained, after which the remaining unstarted activities are abandoned. The equivalent of Lemma 1 is the following:

Lemma 2. *An optimal elementary scheduling policy is globally optimal for parallel systems.*

We also have the following result:

Theorem 1. *The parallel problem is NP-hard in the ordinary sense, even if* $r = 0$, $\forall i \in N\backslash\{0,n\} : d_i = 1$, *and* $\delta \geq \sum_{i \in N\backslash\{n\}} d_i$.

The proof of this theorem describes a reduction of a series system with $r = 0$ to a parallel system with $r = 0$, and actually, these two can be seen to be equivalent (a reduction is also possible in the other direction). Unfortunately, this is no longer the case for $r > 0$. The reason for this is that C is obtained immediately when one activity is successful in a parallel system, whereas all activities have to be executed for payoff in a series system. As a consequence, project payoff can no longer simply be associated with the start of activity n in a deterministic schedule defining an elementary policy; when $r = 0$, however, the exact timing of obtaining the payoff is irrelevant.

Since there is only one non-dummy module (indexed '1'), all the relevant precedence constraints are in order relation B_1 on N_1. If R is a feasible extension of B_1 then we call schedule **s** a *tree schedule* if

$$s_i + d_i \leq s_j \qquad \forall (i,j) \in R$$

and additionally

$$\forall i \in N_1: \quad \left(\exists j \in N_1 \middle| \left((i,j) \in R \wedge s_i + d_i = s_j \right) \vee \left((j,i) \in R \wedge s_j + d_j = s_i \right) \right) \vee \\ \left(s_0 + d_0 = s_i \right) \vee \left(s_i + d_i = s_n \right) \vee \left(s_i = +\infty \right).$$

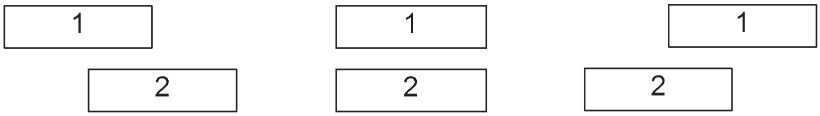
Disregarding unselected activities (with $s_i = \infty$), the set of activity pairs with binding starting-time inequalities for such a tree schedule **s** contains a spanning tree on N, whence the name of this type of schedule. We conjecture that in the search for a deterministic schedule defining an elementary policy for parallel systems, one does not lose all optimal schedules by restricting the search to tree schedules: any arbitrary schedule can be rearranged (locally shifting activities earlier or later in time) to a tree schedule without harmful effects on the objective function.

Under this conjecture, an optimal schedule for parallel systems can be produced in two phases, analogously to series systems. The first step entails computing a feasible extension R of A, leading to the execution probabilities $\prod_{(j,i) \in R}(1 - p_j)$ for each activity i, and then optimizing the elementary policy's eNPV subject to starting-time constraints for all elements of R. Unfortunately, contrary to the serial case, this second phase does not simply reduce to a deterministic scheduling problem with NPV objective, because the probability of obtaining the project payoff at specific activity completion times is not only dependent on R but also on the scheduling decisions for activities in parallel and since receipt of C is not always associated with exactly one activity; this is illustrated by an example described in the next paragraph. Nevertheless, the solution space can at least be reduced to a finite and enumerable set.

Consider the three schedules shown in Figure 4 for the non-dummy activities in $N_1 = \{1,2\}$ of an example parallel system, with $R = B_1 = \varnothing$. The probability of obtaining the project payoff at the completion of

FIGURE 4

An illustration of a parallel system, assuming a Gantt-chart representation as in Figure 3; time is again on the horizontal axis (not shown)



(a) schedule $\mathbf{s}_1$ (b) schedule $\mathbf{s}_2$ (c) schedule $\mathbf{s}_3$

activity 2 for schedule $\mathbf{s}_1$ equals $p_2(1 - p_1)$, while for schedule s_2 this is $1-(1- p_1)(1-p_2)$ and for $\mathbf{s}_3$ we have p_2. Nevertheless, all three schedules correspond with the same schedule-induced strict order (see Neumann et al., 2003, for definitions), namely R, on N_1. This illustrates that the second phase of the sketched solution approach for serial systems is difficult to extend to parallel systems.

IV. FURTHER CHALLENGES

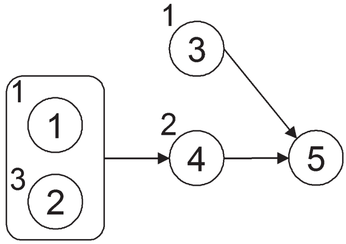
In this section, we provide four illustrations of project scheduling with possible task failures and the solution of which remains an open challenge.

1. *Dynamic decision making*

We give a brief summary of a very simple project in Figure 5. Realization of the project payoff is activity 5, which has two predecessor activities 3 and 4, which both need to be successful for 5 to be able

FIGURE 5

Project network for the example of Section 1
Activity durations are indicated next to each node



to occur. Additionally, activity 4 needs a beneficial test result from at least one of two testing activities 1 and 2. The precedence constraints between activities and modules are represented by the network in the figure. As before, success or failure of each activity is uncertain, and is only known at the end of the activity.

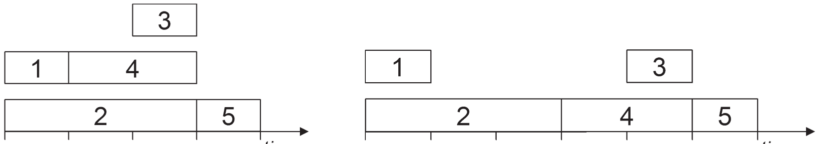
Two possible project schedules are set up for this problem in Figure 6, one for the case in which activity 1 is successful (a), another one for the case where activity 1 turns out to fail (b). We cannot make a definitive choice between these two schedules at time 0 (if this were true, we would not execute activity 2 in (a)). Rather, we start both tests 1 and 2 at project initiation, and at the end of activity 1, dependent on its outcome, the appropriate schedule is selected for the remainder of the schedule. Remark that, although activity 3 can be started at time 2 even when activity 1 fails, this is generally not recommendable as the activity costs money, and so paying for it only at time 4 is preferred. Additionally, if activity 2 is also found to end in failure, then we better interrupt the project altogether and not suffer the cost of activity 3 at all (because the payoff will not be achieved anyway). This example clearly illustrates the need for a dynamic decision process that cannot be cast into an elementary policy anymore. We also see that some activities are started as early as possible, while other are started as late as possible.

2. *Impact of the time value of money*

Our second example uses the project network that was shown in Figure 1 in Section 1, in which payoff (activity 4) only requires success from activity 3 and from the test module consisting of tests labeled as activities 1 and 2, at least one of which needs to be successful for

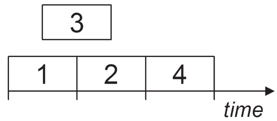
FIGURE 6

Two different schedules for the example of Section 1



(a) schedule $\mathbf{s}_1$ (b) schedule $\mathbf{s}_2$

FIGURE 7
Schedule for Example 2



module success. Additionally, activities 1 and 2 need to be performed in series (2 could be a repeated version (retrial or rework) of 1), and all activities have unit durations. Consider the schedule represented in Figure 7, where activity 3 is scheduled in an 'intermediate' position so that it does not overly delay project completion in case 1 is successful, and at the same time its cost is not incurred excessively early in case of failure for 1. Such a compromise may seem preferable, because contrary to the first example, we cannot wait for uncertainty to unfold and produce an outcome for activity 1 before we decide which (set of) scenario(s) to optimize for. Therefore, in the search for an optimal solution, one might opt for a schedule that performs acceptably well in both cases. The question whether this approach is dominated by either starting 3 at time 0 or at time 1 is open. Note also that if the activities have unequal durations, a higher number of possibilities arises.

3. *Activity selection*

A third small example illustrates another dynamic aspect of the decision process, which actually boils down to the appropriate selection of activities to be implemented. Consider Figure 8, and suppose that test 1 has a cost of 1 and a probability 0.5 of success, test 2 costs 4.5 and is successful with probability 0.5, and project payoff is 4. If we neglect

FIGURE 8
Third example project

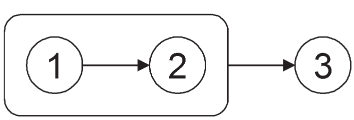
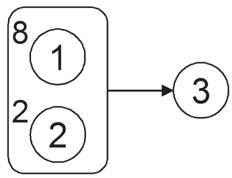


FIGURE 9

Fourth example project. Activity durations are indicated next to each node



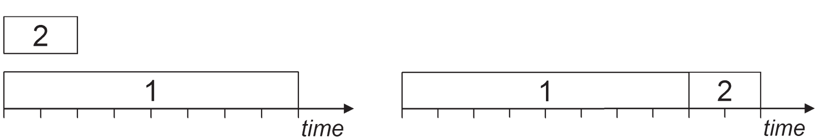
the time value of money ($r = 0$), the optimal way to execute this project (for a risk-neutral investor) is clearly to perform activity 1 and then either collect the payoff if the test is successful, or abandon the project. In other words, activity 2 is never executed.

4. *Risk management*

Our fourth example highlights another trade-off that is inherent in the concept of stochastic activity success. Consider the project in Figure 9, and suppose that activity 1 has a cost of 7 and a probability 0.6 of success and activity 2 costs 34 and is successful with probability 0.4. The project payoff is 240, associated with the start of activity 3. If we assume a discount rate of $r = 0.05$, the two schedules depicted in Figure 10 have approximately the same eNPV of 103.7. The schedules correspond with quite different risk profiles, however, as illustrated in Figure 11. The sequential schedule is more conservative and involves less downside risk, but the expected total project execution time is larger. The parallel schedule leads to a larger downside risk, which is compensated for by the possibility of an earlier launch date, yielding a higher upside potential.

FIGURE 10

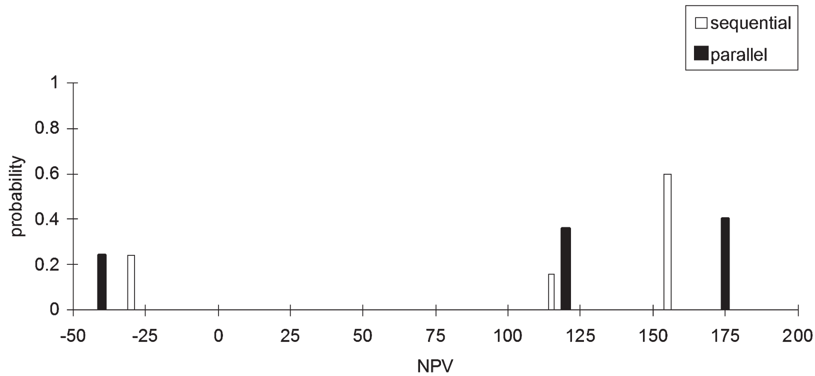
Two different schedules for the example of Section 4



(a) Parallel schedule (b) Sequential schedule

FIGURE 11

Risk profile of two possible schedules



V. SUMMARY AND OUTLOOK ON FURTHER RESEARCH

In this article, we have presented a model for scheduling R&D projects to maximize the expected net present value when the activities have an inherent possibility of failure. The link to overall project success was laid via the intermediate step of activity modules, each of which needs to be successful in order to achieve the project payoff; a module is successful if at least one of its constituent activities leads to success.

The generality of our problem statement suggested that optimization over the class of all policies will often be computationally intractable. The decision maker will therefore tend to prefer to restrict the attention to subclasses that have a simple combinatorial representation. For series and parallel systems, we have looked into elementary policies defined by a deterministic schedule and we have shown that this class contains a globally optimal policy. It would be interesting to investigate to what extent the concept of an elementary policy can be extended towards general scheduling problems, and what is the performance of this class of policies. An alternative line of study might look into the development of an extension of priority policies for deterministic scheduling towards scheduling with possible activity failures.

The model we have presented and analyzed is rather stylized, and will not always be of immediate use for decision support. Decision makers faced with planning R&D projects in industry will often be confronted with resource constraints and duration uncertainty, an observation that was also made by Schmidt and Grossmann (1996) and Jain and Grossmann (1999). The easiest setting with respect to resource constraints is the single-machine case, which corresponds with the requirement of *sequential* testing. From the point of view of scheduling theory, the examination of both positive and negative intermediate cash flows also constitutes a non-trivial challenge.

Another practically relevant generalization is to make project payoff a function of the project completion time. The choice for a non-increasing function would be appropriate for most innovative projects: the earlier a new product enters the market, the longer it can benefit from a monopoly position and first-mover advantages, or the longer it can exploit a patent. A further open option for model extension is correlated activity success, an inherent characteristic of many R&D projects. Quantifying correlations may be difficult, however. Finally, decision makers may also desire to take into account that some R&D activities can be performed in different ways, e.g. by allocating more or less money, resulting in different success probabilities associated with these multiple activity execution modes.

APPENDIX: PROOFS

Proof (Lemma 1): Let Π^* be any globally optimal policy. We will construct a schedule $\mathbf{s}^\circ$ and elementary policy $\Pi^\circ(\mathbf{s}^\circ)$ such that $E[f(\Pi^*)] = E[f(\Pi^\circ(\mathbf{s}^\circ))]$. If $E[f(\Pi^*)] = 0$ then let $\mathbf{s}_i^\circ = +\infty$ for all $i \in N$; we see for this case that $E[f(\Pi^\circ(\mathbf{s}^\circ))] = 0$. Otherwise (when $E[f(\Pi^\circ(\mathbf{s}^\circ))] > 0$), let $\mathbf{s}^\circ = \mathbf{s}(\mathbf{l}; \Pi^*)$, with $\mathbf{1}$ an $(n-1)$-vector of values 1: $\mathbf{s}^\circ$ is the schedule that is obtained by the application of Π^* if all activities are successful. Call $D_t(\Pi, \mathbf{x})$ the set of activities that are started by policy Π for scenario $\mathbf{x}$ at time t. At time $t = 0$, $D_0(\Pi^*, \mathbf{x})$ is necessarily the same for all scenarios $\mathbf{x}$, in light of the non-anticipativity constraint, and therefore $D_0(\Pi^*, \mathbf{x}) = D_0(\Pi^*, \mathbf{1}) = D_0(\Pi^\circ, \mathbf{x})$. Consider now an arbitrary scenario $\mathbf{x}$, and call $t_\mathbf{x}$ the time instant at which the first unsuccessful activity ends.

1. For any time instant $t < t_\mathbf{x}$, due to the non-anticipativity requirement, $D_t(\Pi^*, \mathbf{x}) = D_t(\Pi^*, \mathbf{1}) = D_t(\Pi^\circ, \mathbf{x})$: Π° behaves the same as Π^*.
2. At time $t_\mathbf{x}$, Π° abandons the remainder of the project, and so $f(\mathbf{s}(\mathbf{x}; \Pi^\circ), \mathbf{x}) \geq f(\mathbf{s}(\mathbf{x}; \Pi^*), \mathbf{x})$.

3. If the project is successful then Π° behaves the same as Π^* over the entire scheduling horizon.

Consequently, $E[f(\Pi^*)] = E[f(\Pi^\circ(\mathbf{s}^\circ))]$, which concludes the proof. □

Proof (Lemma 2): The proof follows the same reasoning as for Lemma 1, but we now choose $\mathbf{s}^\circ = \mathbf{s}(\mathbf{0}, \Pi^*)$, with $\mathbf{0} = (0, 0, ..., 0)$ an $(n-1)$-vector of values 0: $\mathbf{s}^\circ$ is the schedule that is obtained by the application of Π^* if all activities fail; $t_{\mathbf{x}}$ is the time instant at which the first successful activity ends for scenario $\mathbf{x}$.

1. For any time instant $t < t_{\mathbf{x}}$, due to the non-anticipativity requirement, $D_t(\Pi^*, \mathbf{x}) = D_t(\Pi^*, \mathbf{0}) = D_t(\Pi^\circ, \mathbf{x})$, with D_t as defined before: Π° behaves the same as Π^*.
2. At time $t_{\mathbf{x}}$, Π° initiates activity n, collects payoff C, and abandons the remainder of the project, so $f(\mathbf{s}(\mathbf{x}; \Pi^\circ), \mathbf{x}) \geq f(\mathbf{s}(\mathbf{x}; \Pi^*), \mathbf{x})$.
3. If the project payoff is not obtained then Π° behaves the same as Π^* over the entire scheduling horizon.

Consequently, $E[f(\Pi^*)] = E[f(\Pi^\circ(\mathbf{s}^\circ))]$, which concludes the proof. □

Proof (Theorem 1): We describe a reduction from the series problem with $r = 0$, $\forall i \in N\backslash\{0, n\} : d_i = 1$, and $\delta \geq \sum_{i \in N\backslash\{n\}} d_i$; we assume that C is large enough so that each activity is worth being executed. For an arbitrary such series instance, we construct an instance of the parallel problem where all non-dummy activities are grouped into module 1; the module order B_1 copies the order A on the modules in the series problem. We also let all cash flows coincide between the two problems, but we invert the probabilities: the success probabilities $p_i^{par} = 1 - p_i^{ser}$ for all the non-dummy activities i, with p_i^{par} and p_i^{ser} the probabilities for the parallel and for the series problem, respectively. □

REFERENCES

Abernathy, W.J., R.S. Rosenbloom. 1969. Parallel strategies in development projects. *Management Science* 15, B486-B505.

Bard, J.F. 1985. Parallel funding of R&D tasks with probabilistic outcomes. *Management Science* 31(7), 814-828.

Ben-Dov, Y. 1981. Optimal testing procedures for special structures of coherent systems. *Management Science* 27(12), 1410-1420.

Blau, G., B. Mehta, S. Bose, J. Pekny, G. Sinclair, K. Keunker, P. Bunch. 2000. Risk management in the development of new products in highly regulated industries. *Computers and Chemical Engineering* 24, 659-664.

Boothroyd, H. 1960. Least-cost testing sequence. *Operational Research Quarterly* 11(3), 137-138.

Boros, E., T. Ünlüyurt. 1999. Diagnosing double regular systems. *Annals of Mathematics and Artificial Intelligence* 26, 171-191.

Butterworth, R. 1972. Some reliability fault-testing models. *Operations Research* 20, 335-343.

Chiu, S.Y., 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.

Choi, J., M.J. Realff, J.H. Lee. 2004. Dynamic programming in a heuristically confined state space: A stochastic resource-constrained project scheduling application. *Computers and Chemical Engineering* 28, 1039-1058.

Dahan, E. 1998. Reducing technical uncertainty in product and process development through parallel design of prototypes. *Working Paper, Graduate School of Business, Stanford University.*

Dahan, E., H. Mendelson. 2001. An extreme-value model of concept design. *Management Science* 47(1), 102-116.

De Reyck, B., R. Leus. 2007. R&D-project scheduling when activities may fail. *IIE Transactions,* forthcoming.

Demeulemeester, E., W. Herroelen. 2002. *Project Scheduling – A Research Handbook.* Kluwer Academic Publishers, Boston.

Denardo, E.V., U.G. Rothblum, L. Van der Heyden. 2004. Index policies for stochastic search in a forest with an application to R&D project management. *Mathematics of Operations Research* 29, 162-181.

DiMasi, J.A. 2001. Risks in new drug development: Approval success rates for investigational drugs. *Clinical Pharmacology and Therapeutics* 69, 297-307.

Ding, M., J. Eliashberg. 2002. Structuring the new product development pipeline. *Management Science* 48, 343-363.

Eppinger, S.D., D.E. Whitney, R.P. Smith, D.A. Gebala. 1994. A model-based method for organizing tasks in product development. *Research in Engineering Design* 6, 1-13.

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

Granot, D., D. Zuckerman. 1991. Optimal sequencing and resource allocation in research and development projects. *Management Science* 37, 140-156.

Herroelen, W.S., 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.

Herroelen, W.S., R. Leus. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* 165(2), 289-306.

Hill, A.V. 2003. *The Encyclopedia of Operations Management Terms.* Available on-line as: http://www.poms.org/EducationResources/omencyclopedia.pdf.

Igelmund, G., F.J. Radermacher. 1983. Preselective strategies for the optimization of stochastic project networks under resource constraints. *Networks* 13, 1-28.

Jain, V., I.E. Grossmann. 1999. Resource-constrained scheduling of tests in new product development. *Industrial and Engineering Chemistry Research* 38, 3013-3026.

Krishnan, V., S. Bhattacharya. 2002. Technology selection and commitment in new product development: The role of uncertainty and design flexibility. *Management Science* 48, 313-327.

Krishnan, V., S.D. Eppinger, D.E. Whitney. 1997. A model-based framework to overlap product development activities. *Management Science* 43(4) 437-451.

Loch, C.H., C. Terwiesch, S. Thomke. 2001. Parallel and sequential testing of design alternatives. *Management Science* 45(5), 663-678.

Lockett, A.G., A.E. Gear. 1973. Representation and analysis of multi-stage problems in R&D. *Management Science* 19(8), 947-960.

Maravelias, C.T., I.E. Grossmann. 2004. Optimal resource investment and scheduling of tests for new product development. *Computers and Chemical Engineering* 28, 1021-1038.

Mitten, L.G. 1960. An analytic solution to the least cost testing sequence problem. *Journal of Industrial Engineering* 11, 17-17.

Möhring, R.H. 2000. Scheduling under uncertainty: Optimizing against a randomizing adversary. *Lecture Notes in Computer Science* 1913, 15-26.

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

Nelson, R.R. 1961. Uncertainty, learning, and the economics of parallel research and development efforts. *The Review of Economics and Statistics* 43(4), 351-364.

Neumann, K., 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.

Padman, R., D.E. Smith-Daniels, V.L. Smith-Daniels. 1997. Heuristic scheduling of resource-constrained projects with cash flows. *Naval Research Logistics* 44(4) 365-381.

Papageorgiou, L.G., G.E. Rotstein, N. Shah. 2001. Strategic supply chain optimization for the pharmaceutical industries. *Industrial and Engineering Chemistry Research* 40, 275-286.

Roemer, T.A., R. Ahmadi. 2004. Concurrent crashing and overlapping in product development. *Operations Research* 52(4), 606-622.

Ronen, B., D. Trietsch. 1988. A decision support system for purchasing management of large projects. *Operations Research* 36(6), 882-890.

Rogers, M.J., A. Gupta, C.D. Maranas. 2002. Real options based analysis of optimal pharmaceutical research and development portfolios. *Industrial and Engineering Chemistry Research* 41, 6607-6620.

Schmidt, C.W., I.E. Grossmann. 1996. Optimization models for the scheduling of testing tasks in new product development. *Industrial and Engineering Chemistry Research* 35, 3498-3510.

Shah, N. 2004. Pharmaceutical supply chains: Key issues and strategies for optimisation. *Computers and Chemical Engineering* 28, 929-941.

Stork, F. 2001. Stochastic resource-constrained project scheduling. *Ph.D. Thesis, Technische Universität Berlin.*

Teunter, R.H., S.D.P. Flapper. 2006. A comparison of bottling alternatives in the pharmaceutical industry. *Journal of Operations Management* 24, 215-234.

Unlüyurt, T. 2004. Sequential testing of complex systems: A review. *Discrete Applied Mathematics* 142, 189-205.

Wagner, B.J., D. J. Davis. 2001. Discrete sequential search with group activities. *Decision Sciences* 32(4), 557-573.

Weitzman, M.L. 1979. Optimal search for the best alternative. *Econometrica* 47 641-654.

Zemel, D., I. David, A. Mehrez. 2001. On conducting simultaneous versus sequential engineering activities in risky R&D. *International Transactions in Operational Research* 8, 585-601.

Zipfel, A. 2003. Modeling the probability-cost-profitability architecture of portfolio management in the pharmaceutical industry. *Drug Information Journal* 37, 185-205.