

# Project Evaluation and Selection with Task Failures

Wenhui Zhao

Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China, zhaowenhui@sjtu.edu.cn

Nicholas G. Hall\*

Fisher College of Business, The Ohio State University, Columbus, Ohio 43210, USA, hall.33@osu.edu

Zhixin Liu

School of Management, University of Michigan - Dearborn, Dearborn, Michigan 48128, USA, zhixin@umich.edu

We consider a company that schedules the tasks of its projects to maximize their expected net present value (ENPV) when tasks may fail. The failure of any task terminates the project immediately. We show that for projects with certain decreasing failure rates, the ENPV optimization problem can be solved using a linear program. The main focus of our work is on how constant task failure rates contribute to decreasing project risk as tasks are completed. Under constant task failure rate, earlier completion of a task improves its probability of success and the risk profile of the project. However, it may also accelerate costs, which worsen discounted cash flow. We show the equivalence of cash flow discount rate and failure rate. Further, if task failures are independent, then the failure rates are additive. We develop a model that (a) recognizes the reduction in project risk when a task is completed, (b) implements this risk reduction into the ENPV calculation, and (c) permits optimization of the ENPV through sequencing and timing decisions for the tasks. We design an algorithm to solve the problem optimally. This enables us to validate the contributions of our work using two computational studies. The first study demonstrates a significant increase in maximum project ENPV from improved project scheduling. The second study demonstrates a significant increase in total project portfolio value as a result of better informed project selection. Our work motivates companies to develop more precise information about the failure risks of their project tasks.

*Key words:* project management; risk of failure; expected net present value; project evaluation and selection

*History:* Received: August 2018; Accepted: August 2019 by Chelliah Sriskandarajah, after 2 revisions

## 1. Introduction

Project management is a highly important, global business process. Various estimates for the global impact of project management within the world's economic activity range from 20% (Project Management Institute 2008) to 30% (Hu et al. 2015), in the latter case implying an annual value of about \$27 trillion. Furthermore, most companies have more available projects than they have the resources to undertake. Substantial evidence suggests that doing the right projects is a big factor in doing projects right. Indeed, well chosen projects are typically easy to manage, whereas poorly selected projects are often dysfunctional and absorb resources from other projects (Cooper et al. 2001). Hence, companies face two problems of central importance to their competitiveness. The first problem is how to *evaluate* their available projects individually. The closely related second problem is, given their limited resources, how to *select* their available projects to run. Project evaluation and selection decisions are typically made by a Project Management Office (Kerzner 2013). Hall (2016) provides an overview of open research problems within project management.

This paper studies a problem that arises generically in the evaluation and selection of a project for which overall success is uncertain. We consider a project that is successful if and only if *each of its component tasks* is successful. Each task is subject to failure at a known constant rate. Many examples arise in new product development, research and development, contract manufacturing, and pharmaceutical development projects. Should any task fail in this environment, the project is immediately terminated, and its projected future revenues and costs are never earned or incurred. We provide detailed motivating examples of such projects in section 2.

As discussed in section 3.1, the most common quantitative measure of a project's return is its net present value (NPV). This performance measure incorporates the anticipated cash flows, both positive and negative, of the project, and also an appropriate discount rate. One of the reasons for the frequent use of the NPV measure is its simplicity, since it discounts all the anticipated cash flows from completion of the project's tasks using the same rate. Many companies use the NPV measure to make decisions about whether to run an individual project. However, when individual

tasks, and therefore the project as a whole, are subject to failure with known probability rates, it is appropriate to use expected net present value (ENPV) in place of NPV.

Thus, consistent with widespread decision making practice, we consider projects that are evaluated based on their ENPV. The consideration of project failure for such projects is a distinguishing feature of our work. The earlier a task is completed, the sooner its risk of failure is eliminated, which may improve its ENPV. However, earlier task completion may also accelerate costs, which worsens discounted cash flow and ENPV. Due to this tradeoff, and because of resulting changes to the risk profile of the project, effective scheduling of the tasks becomes both critical and complex. To address this issue, we model and solve the problem of maximizing the ENPV of a project that is subject to failure. Our work enables significantly more accurate maximization of project ENPV through improved scheduling, and thus selection of a better portfolio of projects.

Chapman and Ward (2002) provide an overview of project risk. Project risk typically originates from uncertainty about the technical and commercial success of the project (MacMillan and McGrath 2002). Technical uncertainty arises, for example, from uncertain outcomes in research and development, prototype testing and regulatory approval. Commercial uncertainty arises, for example, from randomness in time to market, the introduction of competitors' products and general economic factors. Mishra et al. (2016) identify three types of risks in federal technology projects, including *complexity risk* and *contracting risk* in the planning process, and *execution risk* in the execution process. They find that each type of risk has a potentially significant negative effect on project performance. Also, as a project proceeds, its risk level declines. An important example here arises in the development of new pharmaceuticals. As each stage of testing, animal trials, clinical trials, FDA approval, and marketing, is passed, the risk level of the project declines significantly. For example, Myers and Shyam-Sunder (1996) document that risk is higher in early stage pharmaceutical development projects than in mature ones. We model declining risk by removing the risk of failure that is attributable to an individual task when that task is completed.

Our work contributes to the extensive project management literature by modeling and solving the problem of maximizing the ENPV of a project that is subject to failure. We formulate a mathematical model that (a) recognizes the reduction in project risk each time a task is completed, (b) implements this risk reduction into the ENPV evaluation of the project, and (c) permits optimization of the ENPV through sequencing and timing decisions for the tasks within

the project. Our work is designed for a project management environment where the company can estimate the failure rate associated with each of the project's tasks.

We perform two computational studies to validate the contribution of our work. The first computational study finds, for a typical project, a significant improvement in maximum ENPV values that result from improved scheduling decisions identified by our model and algorithm. An interesting outcome of these results is that the improvement in maximum ENPV is much greater for some projects than for others. This changes the value of funding some projects relative to others. Motivated by this observation, the second computational study estimates the increase in overall project portfolio value that results when the improved maximum ENPV values are used to guide selection decisions.

The remainder of this study is organized as follows. Section 2 provides a description of two specific applications. In section 3, we review the relevant literature. In section 4, we describe our notation, formally define the problem, and solve two problems with specific decreasing failure rate functions. Heuristic solutions and upper bounds on problem value are discussed in section 5. An algorithm that solves the problem optimally is described in section 6. Section 7 describes our computational studies. Finally, section 8 contains managerial insights and directions for future research. All proofs appear in an Appendix.

## 2. Motivating Applications

In this section, we describe two widely used applications of project management as motivating examples for our work. We also comment on the mechanism by which task and project failure occur in these applications.

First, we consider a typical software development process managed as a project. This process can be divided into the following six stages (www.synapseindia.com 2019): planning, analysis, design, development and implementation, testing, and maintenance. Each of these stages is composed of multiple tasks, some of which may fail and cause failure of the project. Based on the related literature, we discuss the timing for failure to occur. El Emam and Koru (2008) conduct a survey of companies about the failure of their software projects. They find that the combined rate of canceled and unsuccessful software projects is significant, at between 26% and 34%. They also investigate the causes of project failure. The survey respondents provide 41 responses to this question. The most frequently cited causes are insufficient involvement of senior management, changes to project requirements, and a lack of necessary management skills. Overall, 29 of the 41

responses identify causes of project failure that occur *continuously over time*, another 11 may occur over time or alternatively at project completion, and only one specifically occurs at project completion. Following this time profile of the causes of project failure, we model the task-related causes of project failure as occurring at any point in time up to task completion.

As a second example, consider a typical drug development and commercialization process that is managed as a project (Blau et al. 2004). This process is complex and involves many activities that may fail. First, in a Phase I clinical trial, the company needs to test the new drug on animals and also healthy human volunteers. An unacceptable result will terminate the process and cause the entire project to fail. Second, in a Phase II clinical trial, the company administers the new drug to human patients with a targeted disease. If the treatment is unsatisfactory, for example, producing either inferior results to competitive products or adverse reactions, then the company may end the project. Third, in a Phase III clinical trial, the company conducts large-scale clinical studies on humans patients to confirm the efficacy, and to identify other effects such as drug-to-drug interactions and side effects, of the new drug. This trial is expensive and again failure may terminate the drug development project. Fourth, the information about the new drug from the earlier three trials needs FDA approval. We note that failure in a variety of drug testing trials cannot be determined until the completion of the trial. However, we also observe that project failure is not an objectively defined event. Rather, it is defined by senior management's decision to cancel the project. Such decisions evolve over time as evidence of project performance accumulates and senior management opinion evolves, and need not be finalized only at the completion time of tasks. For example, Zipfel (2003) mentions that "it may be acceptable to consider constant failure rates within each development stage" in drug development. Therefore, following the literature from pharmaceutical industry practice, we believe that it is reasonable to model task failure as occurring at any point in time up to task completion, as in the previous example.

For either of the above applications, failure rates of different tasks can be estimated by investors and financial institutions, which will adjust the interest rate accordingly when they loan money to the project company. Therefore, when the company determines its schedule of the project, it must consider interest rates that change with project progress in scheduling the project tasks, including those with known failure rates, to estimate more accurately and to optimize the expected net present value of the project.

### 3. Literature Review

Section 3.1 reviews the literature that discusses the use of NPV and ENPV for project evaluation. Section 3.2 reviews the literature that considers the problem of scheduling to maximize project NPV. Section 3.3 discusses work on projects that fail because their component tasks fail. Section 3.4 discusses the availability to management of the detailed information about project failure that is needed for our model.

#### 3.1. NPV and ENPV for Project Evaluation

We first review the use of NPV for projects without the risk of failure. As discussed by Remer and Nieto (1995), NPV is among 25 prominent techniques that are used by project companies for evaluating projects. Beaves (1993) discusses the limitations of conventional NPV formulas, and the need for a generalized NPV formula. Haley and Goldberg (1995) discuss the issue of whether emphasizing NPV in the analysis and selection of new product research projects hinders innovation, due to short-term biases. Their empirical results lend support to these concerns. Hodder and Riggs (1995) describe several pitfalls that arise in the overinterpretation of NPV analysis for project evaluation. Archer and Ghasemzadeh (1999) describe an integrated framework for project portfolio selection using NPV. Poh et al. (2002) present a comparative study of several evaluation methods for research and development projects, based on the Analytic Hierarchy Process. Kettunen and Salo (2017) show that project portfolio selection using NPV can be biased in the presence of severe downside risks, and propose a calibration framework to overcome this bias. Several authors use NPV analysis for evaluating and selecting projects for specific applications. These include Cooper (1985) for new product selection, Nelson (1986) for manufacturing modernization projects, Oral et al. (2001) for project selection problems with competing interests among multiple stakeholders, and Kolisch and Meyer (2006) for pharmaceutical development projects. In comparing projects with different makespans using ENPV, their makespans should not be dramatically different because of the opportunity cost of money tied up in the projects. This concern has not prevented NPV being one of the most prominent measures used for project evaluation (Remer and Nieto 1995). While it is necessary for completeness to include ENPV calculations for projects with large failure probability, we do not recommend the use of ENPV as the primary measure of comparison between projects with very different probabilities of success.

Wiesemann and Kuhn (2015) provide an extensive review of the literature of ENPV maximization under

uncertainty about cash flows and durations. Part of the literature assumes that activity durations follow independent exponential distributions and finds optimal solutions, whereas other work allows more general distributions but typically provides suboptimal solutions. An example of the first type of work is by Sobel et al. (2009), who maximize ENPV in a situation with uncertain task durations, costs and revenues. They model the problem as a Markov decision process. They show that discount rates can be absorbed into transition probabilities, and they extend their work to allow for the project to be abandoned during execution. For exponential task durations, they are able to find optimal solutions for projects with up to 25 tasks. An example of the second type of work is by Chen and Zhang (2012), who maximize ENPV in a resource-constrained version of the problem, for which they provide heuristic solutions using ant colony optimization and Monte Carlo simulation. Wiesemann et al. (2010) consider the maximization of ENPV when task durations and cash flows are described by a discrete set of scenarios with known probabilities. They describe a branch and bound algorithm that finds optimal solutions for projects with up to 50 tasks.

### 3.2. Maximization of NPV

Herroelen et al. (1997) survey the literature of project management with discounted cash flows, including both deterministic and stochastic models. They provide a taxonomy of this literature, and critically review the major contributions. Several papers study the problem of sequencing and timing the tasks of a project with a constant discount rate, to maximize NPV. Russell (1970) models this problem as a nonlinear program with linear constraints and a non-concave objective. However, Grinold (1972) shows that this problem can be modeled as a linear program which allows an efficient algorithm based on tree networks. Russell (1986) provides a computational comparison of the performance of six heuristic scheduling rules for a more general problem that considers resource constraints. Elmaghraby and Herroelen (1990) provide a critical review of the research literature on maximizing the NPV of a project. Their main criticism concerns the typical assumption that the cash flow at the end of a task is independent of its completion time, which is inconsistent with penalty clauses in many project contracts. They also provide a solution procedure that is apparently simpler than those of Russell (1970) and Grinold (1972). Computational experience with this procedure is reported by Herroelen and Gallens (1993).

Doersch and Patterson (1977) use a zero-one integer programming model to solve the problem of maximizing NPV, subject to capital rationing constraints. Yang et al. (1992) develop a similar model to

maximize NPV, subject to resource limitations that vary over time. Icmeli and Erenguc (1996) and Vanhoucke et al. (2001b), study the problem of maximizing NPV subject to resource constraints, and develop branch and bound algorithms for small projects. Schwindt and Zimmermann (2001) consider the maximization of project NPV subject to general constraints, and describe a steepest ascent procedure.

Etgar et al. (1996) address the criticism of Elmaghraby and Herroelen (1990) by allowing the cash flow of a task to depend on its completion time. They use simulated annealing to solve this problem heuristically for projects with up to 45 tasks. Etgar and Shtub (1999) consider a special case of the previous model, where a task's cash flow is a linear function of the completion time of the task. They provide a simple, optimal algorithm to maximize NPV, but no computational results. For the same problem, Vanhoucke et al. (2001a) provide a more complex procedure that includes dominance rules and other computational enhancements, and use it to find optimal solutions for projects with up to 120 tasks.

### 3.3. Risk of Project Failure

Herroelen (2005) recognizes risk analysis and proactive scheduling as important factors in closing the gap between theory and practice in project scheduling. De Reyck et al. (2007) provide an extensive survey of the literature of project scheduling with task failures. Browning and Ramasesh (2007) recognize the importance of modeling in cases of uncertainty, ambiguity, and risk for managing product development projects. Wu et al. (2014) examine project risk caused by individuals' cost salience, that is, the perception that the cost of immediate effort is greater than the cost of future effort. Ellinas (2019) shows that, with higher than anticipated probability, task failures can trigger failures of succeeding tasks and lead to systemic failures of a project.

Bard (1985) studies the parallel development of alternative technologies, which provide redundancy in a situation where some technologies may fail. Several structural results and exact algorithms are provided in this environment by Ranjbar and Davari (2013), Coolen et al. (2014) and Creemers et al. (2015). However, our work does not consider alternative technologies. For new product development projects, Schmidt and Grossmann (1996) and Jain and Grossmann (1999) develop optimization models for the scheduling of screening tests in chemical engineering applications.

De Reyck and Leus (2008) describe a generic model for the optimal scheduling of projects with general precedence structure, where all tasks must succeed in order for the project to succeed. They motivate the problem as a pharmaceutical development project,



and show that it is *NP*-hard. A real example from a U.K. biotech company provides a case study. They develop a branch and bound algorithm that is capable of finding optimal solutions for projects with up to 40 tasks. A difference from our work arises in the task failure probability. They model this as a fixed and known probability; whereas, we model task failure using a constant and known failure rate that continues to threaten the task, regardless of when it is started, up to the time when it is completed. Further, their work allows for only one positive cash flow at the end of the project, whereas we more generally allow milestone payments by the project owner.

### 3.4. Information for Modeling Failure

Our modeling of ENPV maximization for projects subject to failure uses predominantly standard data that would typically be available and used in any project evaluation. These standard data include (a) for the project, a common failure rate, and (b) for each task, its deterministic duration and either cost or revenue. Our only additional data requirement is, for each task, its constant failure rate from the start of the project until the task is completed. A constant failure rate is commonly used in studies of reliability and survivability (Barlow and Proschan 1965, Elandt-Johnson and Johnson 1999). The use of a constant failure rate follows immediately from an assumption that the time to failure for a system component is exponentially distributed. This assumption has been applied to a wide variety of systems, ranging from parts in service (Walker 1997) to recidivistic behavior (Stollmack and Harris 1974). Given available data, there are various tests that can be applied to test the validity of the constant failure rate assumption (Fercho and Ringer 1972). In a project management context, such data can be obtained from experiences with the same or similar tasks. Khanfor et al. (2017) empirically investigate failure prediction in crowdsourced software development, including estimation of task failure.

## 4. Problem Definition and Model

We first define our problem and model. In section 4.1, assumptions and limitations of our model are discussed. Sections 4.2 and 4.3 address the maximization of ENPV without and with task-specific risks, respectively.

We consider how a company that is investing in a risky and complex project can evaluate the ENPV of that project. At the end of each task  $i$ , there will be either a positive or negative net cash flow. Positive cash flows represent either milestone payments for partial completion of the project, or a final payment on overall completion, from the project owner.

Negative cash flows represent costs, e.g., labor or material costs that are incurred to perform the task.

Let  $n$  denote the number of tasks in the project. The tasks are indexed  $1, \dots, n$ . In addition, we add two dummy tasks: task 0 at the start, and task  $n + 1$  at the end, where task 0 precedes, and task  $n + 1$  succeeds, all of tasks  $1, 2, \dots, n$ , respectively. Task  $i$  has a cash flow  $F_i$  at its completion time  $C_i$ , where  $F_i > 0$  for cash inflows and  $F_i < 0$  for cash outflows, for  $i = 1, \dots, n$ , and we let  $F_0 = F_{n+1} = 0$  unless otherwise defined. In case a milestone payment is not associated with a specific real task, but received once a set of real tasks are completed, we can model it using a dummy task with appropriate precedence constraints. We can also use dummy tasks to model the case when negative cash flows are not incurred at the completion time of a task, as will be discussed in Remark 2 in section 4.3.

Let  $C_0 = 0$ , that is, task 0 starts and finishes at time 0. Let  $\Delta$  denote the deadline of the project, after which the project is worthless. Let  $D_i$  denote the duration of task  $i$ , where  $D_0 = D_{n+1} = 0$ , and  $D_i \geq 0$  for  $i = 1, \dots, n$ . Let  $S_i$  denote the set of immediate successors of task  $i$  under the given precedence constraints, which are

$$C_k - C_i \geq D_k, \quad k \in S_i, \quad i = 0, \dots, n. \quad (1)$$

Note that  $C_{n+1}$  is the project makespan, and we require that  $C_{n+1} \leq \Delta$ , to ensure that the project finishes before the given deadline.

Let  $r_f$  denote the exogenous *failure rate* of the common risk to the project, which applies to *all* the tasks of the project. Besides this common risk, we also consider each task  $i$  to have a known unique risk that is *independent of the common risk*. We denote the task-specific failure rate of task  $i$  by  $r_i$ . If task  $i$  has no failure probability, we set  $r_i = 0$ . If risk is realized before the completion of task  $i$ , then task  $i$  fails. That is, at any time  $t < C_i$ , there is  $r_i dt$  probability of the failure of task  $i$ , *irrespective of whether task  $i$  has been started or not*.

### 4.1. Assumptions and Limitations

We review the assumptions and limitations of our modeling approach. First, we assume that project cash flows occur at the completion of each task. In practice, cash flows occur at various points in time. The literature assumes that cash flow payment occurs either at the start of each task, or at the end of each task, or at one or several time points between the start and end of each task. For each of these three possibilities, since task durations are fixed, our model works with small technical adjustments, as discussed in Remark 2 in section 4.3. We note that most of the literature, including almost all the work we cite (e.g., Doersch and Patterson 1977, Elmaghraby and Herroelen 1990, Grinold 1972, Herroelen and Gallens 1993, Icmeli and Erenguc 1996, Russell 1970, 1986, Vanhoucke et al. 2001b, Yang

et al. 1992), assumes payment is made at the end of each task. Hence, we adopt this assumption for consistency with the literature.

Second, we assume deterministic activity times. This assumption is made for two reasons: analytical tractability, and the fact that in practice many project time estimates are point estimates. Alternative estimates using probability distributions, where available, would typically perform better than single point estimates.

Third, we assume an exogenous failure rate to model the common risk to the project. A typical example of such a common risk is a competitor releasing a similar product that positions the project company's product out of the market. This is a very frequent risk in projects, especially for new product development in competitive markets (for example, computer chips, consumer electronics, games, and toys). Mitigating this risk of competition is an important issue in supply chain management; see Niu et al. (2019a) for some representative recent work.

Fourth, we assume each task to have a known unique risk independent of the common risk. We specifically assume task risk starts from time zero. This assumption is motivated by practical considerations. For example, consider a pharmaceutical development project, where task  $i$  requires conducting a specific clinical trial. There is a probability that this type of trial will be prohibited by regulators. In this case, whether task  $i$  has been started or not, it will fail immediately. As another example, suppose task  $i$  requires a specific component from a unique supplier. There is a probability that the supplier's factory becomes incapacitated, for example, due to closure, or a disaster such as a fire, flood or earthquake. This causes the failure of task  $i$ , whether it has been started or not. Note that dual sourcing can mitigate such supply risks, though it may introduce competition among suppliers; see Niu et al. (2019b) for some important new developments in this area. A third example is the loss of an environmental or planning license related to the task, as may occur due to a change in local government policy. Further and more generally, assuming that task risk starts from time 0 is consistent with a financier's perception of risk. To a financier, an unfinished task is risky even before it starts, and its cash flow is subject to an interest rate reflecting the risk until task completion.

For consistency with the literature, we assume that the risk of task  $i$  ends at  $C_i$ . In the context of a pharmaceutical project, De Reyck and Leus (2008) pp. 370 write, "Activity success or failure is revealed at the end of each activity." Several practical examples also support this assumption, as we now discuss. Once a clinical trial is completed, the risk vanishes, unless that type of clinical trial is prohibited retroactively,

which is unlikely and punitive. For another example, if the task-specific risk is financial failure or poor technical performance by a subcontractor, these also vanish at task completion.

We assume that the failure caused by the common risk or any task  $i$ 's individual risk results in the failure of the whole project, as a result of which all the future cash flows that have not been collected or paid are lost.

We note that the causes of *general* failure in our model, for example, cancellation of the project or bankruptcy on the part of the project owner, are *external* to the project company. Some causes of *task-specific* failure, for example, poor technical performance by a subcontractor, are also *external* events that specifically impact the task in question. Risks that are associated intrinsically with the performance of the task itself start only when the activity starts. Our work does not model those task-specific risks that arise only within the execution time or performance of the task.

Fifth, we assume that the task-specific risk failure rate  $r_i$  is constant over time, from time 0 until the completion of task  $i$ . We observe that this is a standard way to model a constant risk over time using survival models (Barlow and Proschan 1965, Elandt-Johnson and Johnson 1999).

Sixth, we assume that task failures are independent. That is, the event that there is a task-specific failure of task  $i$  is assumed to be independent of the event that there is a task-specific failure of event  $j$  for every pair  $(i, j)$  of distinct tasks. Moreover, each such event is also independent of the event that there is a general failure that impacts all the tasks. Although the events are probabilistically independent, the occurrence of any failure, either internal or external, immediately results in the failure of the project. We note that independence of task failures is a key assumption that may be potentially violated in practice. For example, tasks typically use shared resources, and such resource sharing directly creates dependencies, e.g., positive correlation, among task times and among failures.

## 4.2. ENPV Maximization without Task-Specific Risk

In classical maximization of project NPV, a project is assumed to have an overall and constant discount rate to reflect the risk level of the project and the cost of capital. We model this using a common failure rate  $r_f$ . Given a constant failure rate  $r_f \geq 0$ , the ENPV of a project is

$$\text{ENPV} = \sum_{i=1}^n F_i \exp(-r_f C_i). \quad (2)$$

Equation (2) is maximized in classical ENPV analysis (for example, Wiesemann et al. 2010). Using

$C_1, C_2, \dots, C_{n+1}$  as the decision variables, the decision problem is to schedule the tasks so as to maximize the ENPV, subject to Constraints (1). The scheduling decisions include timing the completion of all the tasks, which also implies a sequencing decision for tasks that are not preordered by (1). Thus, the ENPV maximization problem is as follows:

$$\max_{C_1, C_2, \dots, C_n} \sum_{i=1}^n F_i \exp(-r_f C_i) \tag{3}$$

s.t. Constraints(1),

$$C_0 = 0, \tag{4}$$

$$C_{n+1} \leq \Delta. \tag{5}$$

Following Grinold (1972), we transform problem (3)–(5) into a linear program. Let  $v_i = \exp(-r_f C_i)$ . Then, problem (3)–(5) is linearized as follows:

$$\max_{v_1, v_2, \dots, v_n} \sum_{i=1}^n F_i v_i$$

$$\begin{aligned} \text{s.t. } \exp(r_f D_k) v_k - v_i &\leq 0, & k \in S_i, \quad i = 0, \dots, n, \\ v_0 &= 1, \\ v_{n+1} &\geq \exp(-r_f \Delta). \end{aligned}$$

Now, we consider overall project failure rates that decrease with project progress. First, consider a mixture of  $h$  exponential distributions, each with exogenously given failure rate of  $r'_j$  and weight of  $w_j$  for  $j = 1, \dots, h$ , where  $\sum_{j=1}^h w_j = 1$ . For such a distribution, the cumulative distribution function at time  $t$  is  $\sum_{j=1}^h w_j (1 - \exp(-r'_j t))$ , and accordingly the survival function at time  $t$  is  $\sum_{j=1}^h w_j \exp(-r'_j t)$ . Thus, the ENPV of a project with these task failure rates is:

$$\text{ENPV} = \sum_{i=1}^n F_i \sum_{j=1}^h w_j \exp(-r'_j C_i).$$

Let  $v_{ij} = \exp(-r'_j C_i)$ . Then, the problem is linearized as:

$$\max_{v_{11}, v_{12}, \dots, v_{nh}} \sum_{i=1}^n F_i \sum_{j=1}^h w_j v_{ij}$$

$$\begin{aligned} \text{s.t. } \exp(r'_j D_k) v_{kj} - v_{ij} &\leq 0, & k \in S_i, \\ & i = 0, \dots, n; \quad j = 1, \dots, h, \\ v_0 &= 1, \\ v_{n+1, j} &\geq \exp(-r'_j \Delta), & j = 1, \dots, h. \end{aligned}$$

Another commonly used decreasing task failure rate is defined by the Weibull distribution with

exogenously given scale parameter  $1/r$  and shape parameter  $k$ . For such a distribution, the cumulative distribution function at time  $t$  is  $1 - \exp(-r^k t^k)$ , and accordingly the survival function at time  $t$  is  $\exp(-r^k t^k)$ . When  $k = 1$ , the Weibull distribution becomes the exponential distribution with constant failure rate  $r$ . When  $0 < k < 1$ , the Weibull distribution has a decreasing failure rate. With the project failure rate defined by the Weibull distribution, the ENPV of a project is

$$\text{ENPV} = \sum_{i=1}^n F_i \exp(-r^k C_i^k), 0 < k < 1.$$

Let  $v_i = \exp(-r^k C_i^k)$ . Then, the problem is linearized as:

$$\max_{v_1, v_2, \dots, v_n} \sum_{i=1}^n F_i v_i$$

$$\begin{aligned} \text{s.t. } \exp(r^k D_k) v_k - v_i &\leq 0, & k \in S_i, \quad i = 0, \dots, n, \\ v_0 &= 1, \\ v_{n+1} &\geq \exp(-r^k \Delta). \end{aligned}$$

REMARK 1. For a project with decreasing task failure rates characterized by either a mixture of exponential distributions or a Weibull distribution, the ENPV maximization problem can be solved via linear programming after appropriate logarithmic transformations.

Remark 1 shows that ENPV maximization can be easily achieved for projects with certain decreasing failure rates. However, these models assume exogenously given decreasing project failure rates, which do not always apply. Moreover, this analysis does not explain why, in practice, project risk decreases as tasks are completed. Our work, described below, addresses this issue.

### 4.3. ENPV Maximization with Task-Specific Risk

In the above ENPV analysis, a key assumption is that only a common failure rate  $r_f$  is used to model the overall risk of the project, which ignores the task-specific risk factors. We now propose an alternative model that includes both common and task-specific risk factors. This model is consistent with the reality that the risk level of a project typically decreases as the project proceeds and more tasks are finished.

We first consider a given task completion time sequence. For notational convenience, suppose that the tasks are indexed based on their completion time sequence, that is, we have  $C_0 = 0 \leq C_1 \leq C_2 \leq \dots \leq C_n$

and  $C_{n+1} = C_n$ . The problem defined by a *given sequence* of task completion times is a subproblem of the more general problem we are solving. Our optimization model, more generally, is over all feasible schedules of task completion times.

We now discuss the discounting of the cash flows due to the risks. Note that cash flow  $F_1$  occurs at time  $C_1$ . At time  $t \leq C_1$ , whether or not tasks  $1, \dots, n$  have been started, each of them may fail. We divide the time  $[0, C_1]$  into  $m$  equal-length intervals, with the lengths of each interval satisfying  $\delta = \frac{C_1}{m} \rightarrow 0$  as  $m \rightarrow \infty$ . We denote the  $m$  intervals by  $[0, \delta], [\delta, 2\delta], \dots, [k\delta, (k+1)\delta], \dots, [(m-1)\delta, C_1]$ . For interval  $[0, \delta]$ , the probability for tasks  $1, \dots, n$  to succeed is  $(1 - r_f\delta)\prod_{i=1}^n(1 - r_i\delta)$ , since the common risk and unique risks of tasks  $1, \dots, n$  are all independent by assumption.

Given the success of all tasks in the first time interval  $[0, \delta]$ , the probability for no task to fail in the second time interval is  $(1 - r_f\delta)\prod_{i=1}^n(1 - r_i\delta)$ . Therefore, the probability for the project not to fail by the end of the second time interval is  $(1 - r_f\delta)^2\prod_{i=1}^n(1 - r_i\delta)^2$ . Continuing similarly, we obtain the probability for the project not to fail by time  $C_1$  as the following:

$$\begin{aligned} & \lim_{m \rightarrow \infty} (1 - r_f\delta)^m \prod_{i=1}^n (1 - r_i\delta)^m \\ &= \lim_{m \rightarrow \infty} \left(1 - \frac{r_f C_1}{m}\right)^m \left(1 - \frac{r_1 C_1}{m}\right)^m \left(1 - \frac{r_2 C_1}{m}\right)^m \\ & \dots \left(1 - \frac{r_n C_1}{m}\right)^m \\ &= \exp(-r_f C_1) \exp(-r_1 C_1) \exp(-r_2 C_1) \dots \exp(-r_n C_1) \\ &= \exp(-(r_f + r_1 + r_2 + \dots + r_n)C_1) = \exp(-R_1 C_1), \end{aligned}$$

where  $R_i$  is defined as

$$R_i \equiv r_f + \sum_{j=i}^n r_j, \quad i = 1, \dots, n, \quad (6)$$

for notational convenience. Therefore, the ENPV of cash flow  $F_1$  at time  $C_1$  is defined as

$$\text{ENPV}_1 = F_1 \cdot \exp(-R_1(C_1 - C_0)),$$

where  $C_0 = 0$ .

We next consider cash flow  $F_2$ . If any failure happens before  $C_1$ , including the failure of tasks  $1, \dots, n$ , then the project company will not pay or receive  $F_2$ , since the whole project fails. Given that tasks  $1, \dots, n$  have not failed by  $C_1$ , the probability for tasks  $2, \dots, n$  not to fail during the interval  $[C_1, C_2]$  is  $\exp(-(r_f + r_2 + \dots + r_n)(C_2 - C_1)) = \exp(-R_2(C_2 - C_1))$ . Thus, the probability for the project company to pay or receive  $F_2$  is  $\exp(-R_1(C_1 - C_0))$

$\exp(-R_2(C_2 - C_1)) = \exp[-R_1(C_1 - C_0) - R_2(C_2 - C_1)]$ . Then, the ENPV for  $F_2$  is

$$\text{ENPV}_2 = F_2 \cdot \exp[-R_1(C_1 - C_0) - R_2(C_2 - C_1)].$$

Continuing this process for  $i=1, \dots, n$ , the expected net present value of cash flow  $F_i$  of task  $i$  can be written as

$$\begin{aligned} \text{ENPV}_i &= F_i \exp \\ & \quad [-R_1(C_1 - C_0) - R_2(C_2 - C_1) - \dots - R_i(C_i - C_{i-1})] \\ &= F_i \exp \left[ -\sum_{j=1}^i R_j(C_j - C_{j-1}) \right]. \end{aligned} \quad (7)$$

REMARK 2. Consider the perspective that  $F_i < 0$  occurring at  $C_i$  essentially implies that it is free to start task  $i$ . Other timings of cash flows can alternatively be modeled as follows. Let  $F_i$  occur at time  $C_i - \alpha_i D_i$ , where  $0 < \alpha_i \leq 1$  is an exogenous parameter. Note that this is the start time of task  $i$  if  $\alpha_i = 1$ . Further, add a dummy task  $i'$  with  $F_{i'} = F_i$ ,  $D_{i'} = 0$ ,  $r_{i'} = 0$ , and precedence constraint task  $i'$  partly preceding task  $i$ , that is,  $C_i - C_{i'} \geq \alpha_i D_i$ , and reset  $F_i = 0$ . Since task  $i'$  has a negative cash flow and no risk or processing time, under maximization of ENPV it will be postponed to start at  $C_i - \alpha_i D_i$ .

Note that when each task has a constant failure rate, the expected net present value of each cash flow can be written as an exponential function where the exponent is a linear function of task completion times, as in Equation (7). However, for tasks with decreasing failure rate, for example, modeled by either a mixture of exponential distributions or a Weibull distribution, the expected net present value of a cash flow is characterized by more complicated nonlinear functions of task completion times, which are hard to linearize.

We now interpret Equation (6). For a project that starts at time 0,  $R_1$  is the failure rate of the whole project during the period  $[0, C_1]$ , since no tasks have been finished and all tasks contribute to the overall risk of the project. After time  $C_1$ ,  $R_2$  becomes the failure rate of the whole project during the period  $(C_1, C_2]$ , since task 1 has completed at time  $C_1$  but all the remaining  $n-1$  tasks contribute to the overall risk of the project. Continuing thus, finally  $R_n$  becomes the failure rate of the whole project during the period  $(C_{n-1}, C_n]$ .

Hence, Equation (6) indicates that, when tasks  $1, \dots, i-1$  have been completed but tasks  $i, \dots, n$  have not, the risk of the project is the *sum of the total risks of the unfinished tasks*. In this sense, risk is *additive*. As a



result, the risk level of the project declines during its execution, that is,  $R_1 \geq R_2 \geq \dots \geq R_n \geq R_{n+1} = r_f$ .

REMARK 3. For ENPV maximization, our model includes the following four features:

1. The failure rates,  $r_f, r_1, r_2, \dots, r_n$ , additively form the failure rate of the whole project at different time periods as in Equation (6), even though the risks can arise from different sources.
2. For each time interval  $(C_{i-1}, C_i]$ , for  $i = 1, \dots, n$ , the failure rate  $R_i$  of the whole project is equivalent to the commonly used discount rate compounded continuously, see (7).
3. The overall discount rate of the project decreases with project execution.
4. Each cash flow has its own unique discount rate, depending on the project schedule.

We note that the third and fourth features in Remark 3 are consistent with those recommended by Damodaran (2007) for risk-adjusted discount rates, that is, to apply a changing discount rate over time and to use different discount rates for different cash flows.

Using  $C_1, C_2, \dots, C_n$  as the decision variables, the decision problem is to maximize the total ENPV,

$$\text{ENPV} = \sum_{i=1}^n \text{ENPV}_i. \tag{8}$$

The implicit assumption in Equation (7) is that the task completion time sequence is given, and hence the coefficients  $R_j$  are given parameters. However, this sequence is decision dependent. Therefore, to formulate an overall optimization model, we define the binary variables

$$x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is scheduled as the } j\text{th task to complete,} \\ 0, & \text{otherwise,} \end{cases}$$

for  $i, j = 1, \dots, n$ . Then, the task completion time sequence can be represented by the variables  $x_{ij}$ , and we have

$$R_j = r_f + \sum_{k=j}^n \sum_{l=1}^n r_l x_{lk}, \quad j = 1, \dots, n. \tag{9}$$

Given the binary variables  $x_{ij}$  and the discount rates  $R_j$  defined above, we now formulate the following model to maximize the ENPV of the project.

(MIP)

$$\max_{C_i, x_{ij}} \sum_{i=1}^n \left\{ \left[ \sum_{k=1}^n F_k x_{ki} \right] \exp \left( - \sum_{j=1}^i \left[ r_f + \sum_{k=j}^n \sum_{l=1}^n r_l x_{lk} \right] (C_j - C_{j-1}) \right) \right\} \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^n x_{ij} = 1, \quad 1 \leq j \leq n, \tag{11}$$

$$\sum_{j=1}^n x_{ij} = 1, \quad 1 \leq i \leq n, \tag{12}$$

$$\sum_{j=1}^n x_{kj} C_j - \sum_{j=1}^n x_{ij} C_j \geq D_k, \quad 0 \leq i \leq n, k \in S_i, \tag{13}$$

$$x_{ij} \in \{0, 1\}, \quad 1 \leq i, j \leq n, \tag{14}$$

(4) and (5).

In problem MIP, Constraints (11) and (12) ensure that each task is scheduled in exactly one position in the completion time sequence, Constraints (13) enforce the precedence constraints, and Constraint (5) enforces the deadline constraint. Observe that the risk profile of the project at any point in time is dependent on scheduling decisions. This results in the following difficulties from an optimization perspective:

1. the problem contains many binary variables  $x_{ij}$ ,
2. the objective function is neither convex nor concave, due to the coexistence of both positive and negative  $F_i$  values,
3. the exponential term in the objective function contains products of  $x_{ij}$  variables with task completion time decisions, and
4. Constraints (13) are not linear, due to products of  $x_{ij}$  variables with task completion time decisions  $C_j$ .

The above features make it mathematically challenging to solve problem MIP optimally. We approach this problem by considering a fully or partially specified task completion time sequence. Then, based on structural results and bounds obtained for fully or partially specified sequences, we develop a branch and bound algorithm to solve the overall problem MIP.

## 5. Approximating the Maximum ENPV

In section 5.1, we establish upper and lower bounds on the maximum value of ENPV, as defined by (7) and (8), for a fully specified task completion time sequence. Similar bounds for a partially specified sequence are established in section 5.2.

### 5.1. Bounds for a Full Sequence

We now consider a given completion time sequence of the  $n$  tasks. Under this given sequence, we reindex the tasks such that  $C_0 = 0 \leq C_1 \leq C_2 \leq \dots \leq C_n$  and  $C_{n+1} = C_n$ . Then, problem MIP can be simplified to:

$$\begin{aligned}
 \text{(SSP)} \quad & \max_{C_1, \dots, C_n} \sum_{i=1}^n F_i \exp\left(-\sum_{j=1}^i R_j(C_j - C_{j-1})\right) \\
 \text{s.t. constraints(1), (4), (5),} \\
 & C_{i+1} - C_i \geq 0, \quad 0 \leq i \leq n,
 \end{aligned} \tag{15}$$

where  $R_j$  is sequence dependent, and Constraints (15) ensure that the tasks are completed following the specified sequence.

**5.1.1. Transformation of Problem (SSP).** We note that the objective function of problem SSP contains an exponential term, and is in general neither convex nor concave. We next develop methods to find lower and upper bounds on the optimal objective function value of SSP. Let

$$\begin{aligned}
 y_i &= \exp\left(-\sum_{j=1}^i R_j(C_j - C_{j-1})\right) \Rightarrow \ln(y_i) \\
 &= -\sum_{j=1}^i R_j(C_j - C_{j-1}),
 \end{aligned} \tag{16}$$

and  $y_0 = 1$  so that  $\ln(y_0) = 0$ . Using transformations described in the Appendix, we reformulate problem SSP as the following problem SSP0 with decision variables  $y_0, y_1, \dots, y_{n+1}$ :

$$\begin{aligned}
 \text{(SSP0)} \quad & \max_{y_0, y_1, \dots, y_{n+1}} \sum_{i=1}^n F_i y_i \\
 \text{s.t.} \quad & \sum_{j=i+1}^k \frac{\ln(y_j) - \ln(y_{j-1})}{R_j} \leq -D_k, \quad \text{for } 0 \leq i \leq n, k \in S_i,
 \end{aligned} \tag{17}$$

$$\sum_{j=1}^{n+1} \frac{\ln(y_j) - \ln(y_{j-1})}{R_j} \geq -\Delta, \tag{18}$$

$$y_{i+1} - y_i \leq 0, \quad 0 \leq i \leq n, \tag{19}$$

$$y_0 = 1 \text{ and } y_{n+1} \geq 0. \tag{20}$$

The objective function of problem SSP0 is linear. However, SSP0 is hard to solve, since Constraints (17) and (18) are nonlinear. Therefore, we next linearize Constraints (17) and (18) approximately, to obtain lower and upper bounds on the optimal objective function value, as described in the Appendix.

**5.1.2. Upper and Lower Bounds.** First, we formulate the following linear program to find an upper bound for the original problem formulated as SSP0:

$$\begin{aligned}
 \text{(SSP1)} \quad & \max_{0 \leq y_i \leq 1} \sum_{i=1}^n F_i y_i \\
 \text{s.t.} \quad & \sum_{j=m+1}^{k-1} \frac{R_k}{R_j} (y_{j-1} - y_j) + y_{k-1} - y_k \exp(R_k D_k) \geq 0, \\
 & \text{for } 0 \leq m \leq n, k \in S_m,
 \end{aligned} \tag{21}$$

$$\begin{aligned}
 & \left[1 - \frac{R_{n+1}}{R_1} - \exp(-R_{n+1}\Delta)\right] y_0 \\
 & - \sum_{j=1}^n \left(\frac{R_{n+1}}{R_{j+1}} - \frac{R_{n+1}}{R_j}\right) y_j + y_{n+1} \geq 0,
 \end{aligned} \tag{22}$$

(19) and (20).

Observe that in the linear program SSP1, Constraints (21) linearize the original Constraints (17), and Constraint (22) linearizes Constraint (18). We have the following result.

**THEOREM 1.** *The optimal objective function value of problem SSP1 is an upper bound on the optimal objective function value of problem SSP0.*

Next, we formulate the following linear program to find a lower bound for the original problem formulated as SSP0:

$$\begin{aligned}
 \text{(SSP2)} \quad & \max_{0 \leq y_i \leq 1} \sum_{i=1}^n F_i y_i \\
 \text{s.t.} \quad & \left[1 - \frac{R_k}{R_{m+1}} - \exp(-R_k D_k)\right] y_m \\
 & - \sum_{j=m+1}^{k-1} \left(\frac{R_k}{R_{j+1}} - \frac{R_k}{R_j}\right) y_j + y_k \leq 0, \quad \text{for } 0 \leq m \leq n, k \in S_m,
 \end{aligned} \tag{23}$$

$$\sum_{j=1}^n \frac{R_{n+1}}{R_j} (y_{j-1} - y_j) + y_n - y_{n+1} \exp(R_{n+1}\Delta) \leq 0, \tag{24}$$

(19) and (20).

In problem SSP2, Constraints (23) linearize the original Constraints (17), and Constraint (24) linearizes the deadline Constraint (18). Then, we have the following result.

**THEOREM 2.** *The optimal objective function value of problem SSP2 is a lower bound on the optimal objective function value of problem SSP0.*

It is possible that the project schedule found by problem SSP1 is not feasible, due to the relaxation of the constraints in the linearization process. On the other hand, the schedule found by problem SSP2 is always

feasible, since the linearized constraints are tighter than the initial ones. However, problem SSP2 may not be able to find a feasible schedule when one exists, due to its more stringent constraint on the deadline. We find from our computational study that when the deadline  $\Delta$  is 20% or more than the minimum possible project makespan, problem SSP2 typically does find a feasible schedule.

To estimate the quality of the bounds found by problems SSP1 and SSP2, we perform a computational study. We use the 330 instances generated as described in section 7.1 below. We report our results in Table 1, where  $n_c$  denotes the number of cash flows in each instance. For each instance, the sequence of tasks follows the topological order of the tasks generated by RanGen (Demeulemeester et al. 2003). Table 1 reports the average relative percentage gap, “Gap%,” between the upper and lower bounds for projects with different numbers of cash flows, that is,  $100(UB-LB)/UB$ . These results show that our bounds for the maximum ENPV of a project with fixed task completion time sequence are very accurate. The success of these bounding techniques is an important factor in the efficient maximization of ENPV, as discussed in section 7.1.

**5.2. Bounds for a Partial Sequence**

During branch and bound enumeration, it is often necessary to consider a partially specified task completion time sequence. In that situation, the bounds established in section 5.1 cannot be applied. Hence, in this section, we develop upper and lower bounds on the optimal ENPV for a partial sequence.

We consider a partial sequence of a set of tasks indexed by  $0, 1, \dots, l$  such that  $C_0 = 0 \leq C_1 \leq C_2 \leq \dots \leq C_l$ . Here,  $l$  is the last task completed under the partially specified sequence. For tasks  $i = l + 1, l + 2, \dots, n + 1$ , the task completion times are unknown, but we require  $C_i \geq C_l$ . Let  $\sigma' = \{0, 1, \dots, l\}$ ,  $\sigma'' = \{l + 1, l + 2, \dots, n, n + 1\}$ , and  $\sigma = \sigma' \cup \sigma''$ .

For any task  $i \in \sigma''$ , let  $A_i$  be the set of all predecessor tasks in  $\sigma''$  that must finish no later than the start time of task  $i$ , and  $B_i$  be the set of all successor tasks in  $\sigma''$  that must start no earlier than the completion time of task  $i$ , as specified by the precedence constraints. To remove the sequence dependence on the unsequenced tasks in  $\sigma''$  and develop upper and lower bounds on optimal ENPV, for each task  $i \in \sigma''$ , we let

1.  $R_{i,\min} = r_f + \sum_{j \in B_i} r_j + r_i$ , and
2.  $R_{i,\max} = r_f + \sum_{j \in (\sigma'' \setminus A_i)} r_j$ .

We have the following inequality regarding the relationship of  $R_i, R_{i,\min}$ , and  $R_{i,\max}$ .

$$R_{i,\min} \leq R_i \leq R_{i,\max} \leq R_{l+1} = r_f + \sum_{j \in \sigma''} r_j. \tag{25}$$

Then, to find an upper bound on the maximum ENPV for a partially specified sequence, we solve the following program:

$$\begin{aligned} \text{(PSSP1)} \quad & \max_{0 \leq y_i \leq 1} \sum_{i=1}^n F_i y_i \\ \text{s.t.} \quad & \sum_{j=i+1}^{k-1} \frac{R_k}{R_j} (y_{j-1} - y_j) + y_{k-1} - y_k \exp(R_k D_k) \geq 0, \\ & i \in \sigma', k \in S_i \cap \sigma', \end{aligned} \tag{26}$$

$$\begin{aligned} & \sum_{j=i+1}^l \frac{R_{k,\max}}{R_j} (y_{j-1} - y_j) + y_l - y_k \exp(R_{k,\min} D_k) \geq 0, \\ & i \in \sigma', k \in S_i \cap \sigma'', \end{aligned} \tag{27}$$

$$y_i - y_k \exp(R_{k,\min} D_k) \geq 0, \quad i \in \sigma'', k \in S_i \cap \sigma'', \tag{28}$$

$$\begin{aligned} & \left[ 1 - \frac{R_{n+1}}{R_1} - \exp(-R_{n+1} \Delta) \right] y_0 - \sum_{j=1}^l \left( \frac{R_{n+1}}{R_{j+1}} - \frac{R_{n+1}}{R_j} \right) y_j \\ & + \frac{R_{n+1}}{R_{i,\min}} y_i \geq 0, \quad i \in \sigma'', \end{aligned} \tag{29}$$

$$y_{i+1} - y_i \leq 0, \quad 0 \leq i \leq l - 1, \tag{30}$$

$$y_i - y_l \leq 0, \quad i \in \sigma'', \tag{31}$$

$$y_0 = 1, \quad y_i \geq 0, \quad i \in \sigma''.$$

The next result shows that problem PSSP1 establishes an upper bound on the maximum ENPV.

**THEOREM 3.** *Given  $C_0 = 0 \leq C_1 \leq C_2 \leq \dots \leq C_l$  and  $C_l \leq C_{l+1}, C_{l+2}, \dots, C_{n+1}$ , the optimal objective function value of problem PSSP1 is an upper bound on the maximum ENPV of the project scheduling problem.*

We now develop a model for finding a lower bound on the maximum ENPV. First, we assume the following condition:

$$D_i R_1 \leq 1, \tag{32}$$

**Table 1 Accuracy of Bounds for a Full Sequence**

$n_c$	10	12	14	16	18	20	22	24	26	28	30
Gap%	0.005	0.004	0.005	0.003	0.005	0.005	0.005	0.008	0.011	0.013	0.015

for all  $0 \leq i \leq n + 1$ . We note that condition (32) is not restrictive, else the failure rate is unreasonably high and it is hard to accept the project. Under condition (32), we develop the following model to find a lower bound on the maximum ENPV for a partially specified task completion time sequence:

$$\begin{aligned}
 & \text{(PSSP2)} \quad \max_{0 \leq y_i \leq 1} \sum_{i=1}^n F_i y_i \\
 \text{s.t.} \quad & \left[ 1 - \frac{R_k}{R_{i+1}} - \exp(-R_k D_k) \right] y_i - \sum_{j=i+1}^{k-1} \left( \frac{R_k}{R_{j+1}} - \frac{R_k}{R_j} \right) y_j + y_k \leq 0, \quad (33) \\
 & i \in \sigma', k \in S_i \cap \sigma',
 \end{aligned}$$

$$\begin{aligned}
 & \left[ 1 - \frac{R_{k,\min}}{R_{i+1}} - \exp(-R_{k,\min} D_k) \right] y_i \\
 & - \sum_{j=i+1}^l \left( \frac{R_{k,\min}}{R_{j+1}} - \frac{R_{k,\min}}{R_j} \right) y_j + y_k \leq 0, \quad (34) \\
 & i \in \sigma', k \in S_i \cap \sigma'',
 \end{aligned}$$

$$\begin{aligned}
 & \left[ 1 - \frac{R_{k,\min}}{R_{i,\max} - r_i} - \exp(-R_{k,\min} D_k) \right] y_i + y_k \leq 0, \quad (35) \\
 & i \in \sigma'', k \in S_i \cap \sigma'',
 \end{aligned}$$

$$\sum_{j=1}^l \frac{R_{n+1}}{R_j} (y_{j-1} - y_j) + y_l - y_{n+1} \exp(R_{n+1} \Delta) \leq 0, \quad (36)$$

$$y_{i+1} - y_i \leq 0, \quad 0 \leq i \leq l - 1, \quad (37)$$

$$\begin{aligned}
 & y_i - y_l \leq 0, \quad i \in \sigma'', \quad (38) \\
 & y_0 = 1, \quad y_i \geq 0, \quad i \in \sigma''.
 \end{aligned}$$

The next result shows that problem PSSP2 finds a lower bound on the maximum ENPV.

**THEOREM 4.** *Given  $C_0 = 0 \leq C_1 \leq C_2 \leq \dots \leq C_l$  and  $C_l \leq C_{l+1}, C_{l+2}, \dots, C_{n+1}$ , the optimal objective function value of problem PSSP2 is a lower bound on the maximum ENPV of the project scheduling problem.*

Similar to the above discussion for a fully specified sequence, it is possible that the project schedule found by problem PSSP1 is not feasible, whereas if problem PSSP2 finds a schedule it is always feasible. Problem PSSP2 can fail to find a feasible solution even if the original problem is feasible, due to the stronger deadline and other constraints. However, problem PSSP2 typically finds a feasible schedule when the deadline  $\Delta$  is not very close to the minimum possible project makespan.

## 6. Branch and Bound Algorithm

In this section, we incorporate the bounding techniques presented in section 5 into a branch and bound

algorithm to maximize the ENPV of a project with risk of task failure. In section 6.1, we introduce an elimination rule to restrict the candidate task sequences. In section 6.2, we introduce two heuristic rules that simplify the problem by assuming a constant failure rate. One provides a benchmark, and the other provides a lower bound on optimal value. A heuristic based on sequence generation is also described in section 6.2. In section 6.3, we describe our branch and bound algorithm.

### 6.1. An Elimination Rule

For a given completion sequence  $\sigma$ , let  $\mathbf{C}^*(\sigma)$  be the optimal completion time vector, and  $\text{ENPV}^*(\sigma)$  be the corresponding net present value. Let  $l < m$  be two tasks in  $\sigma$ , where  $l$  has no successors that are not shared with task  $m$ , and task  $m$  has no predecessors that are not shared with task  $l$ . Let  $\sigma'$  be the sequence obtained from  $\sigma$  after interchanging  $l$  and  $m$ . We apply the following elimination rule.

**THEOREM 5.** *If  $r_l = r_m$ ,  $D_l \geq D_m$ , and  $F_l \leq F_m$ , then  $\text{ENPV}^*(\sigma') \geq \text{ENPV}^*(\sigma)$ .*

Theorem 5 is useful in that, if the project manager does not have precise information about the exact risk value of each task, we may simply classify all the tasks into a few risk categories, e.g., high risk tasks, medium risk tasks, and low risk tasks. Then, the same risk value can be assigned to all the tasks within each category. In this case, since all the tasks within a category have the same risk, we can apply Theorem 5 if the durations and cash flows of two tasks satisfy the conditions of the lemma.

### 6.2. Heuristics without Task-Specific Risks

A straightforward way to maximize the ENPV heuristically is to use an estimation method that maximizes ENPV for a fixed failure rate. We introduce two heuristics that apply this idea. In section 6.2.1, we introduce a simple method for ENPV estimation, assuming an average failure rate for both scheduling and ENPV computation. In section 6.2.2, we describe a heuristic that uses a constant failure rate to schedule tasks, and then computes the ENPV using both common and task-specific failure rates.

**6.2.1. ENPV Maximization with Only Common Failure Rate.** Grinold (1972) formulates the NPV maximization problem with a constant discount rate as a simple linear program. Given a constant failure rate  $r_f$ , or equivalently a constant discount rate, the ENPV maximization problem can be formulated as a linear program, as shown in section 4.2.

In situations where there is a lack of detailed understanding of how project risk changes as tasks are



completed, it is natural to assume a constant failure rate for the whole project. There are various ways to determine the constant failure rate to use. Given the task-specific failure rate  $r_i$  of each task  $i$ , one possible constant failure rate is the average rate during project execution as the risk declines from  $\sum_{i=1}^n r_i$  to 0. Note that during the execution of a project, the minimum and maximum task-specific failure rates are 0 and  $\sum_{i=1}^n r_i$ , respectively, and thus we use the average of the two values:

$$\bar{r} = \sum_{i=1}^n r_i / 2.$$

Using this rate, the maximum project ENPV can be estimated as follows.

#### Grinold estimation with averaged failure rate (GE)

Use Grinold's method to estimate the project ENPV, where a constant failure rate  $r_f + \bar{r}$  is used both for scheduling and for the ENPV computation of the schedule found.

Note that since the value of ENPV obtained by procedure GE uses a midrange estimate of the failure rate, it is neither a lower bound nor an upper bound on the maximum ENPV. For this reason, GE is not implemented within our branch and bound algorithm. However, it is used as a benchmark in our computational studies in section 7.

**6.2.2. Grinold Heuristic.** The estimation procedure GE can be improved with more detailed analysis of the risks affecting the project. A feasible project schedule is found by Grinold's method which assumes a constant failure rate. However, task-specific failure rates can be used to calculate the ENPV of this schedule. Further, other values of the common failure rate can be used. We propose the following heuristic for task scheduling and ENPV computation. Let  $K \geq 1$  denote a constant integer and  $\varepsilon = (\sum_{i=1}^n r_i) / K$ .

#### Grinold Heuristic (GH) for ENPV computation

1. Use Grinold's algorithm to find a feasible schedule for each constant failure rate  $r_f + \varepsilon, r_f + 2\varepsilon, \dots, r_f + K\varepsilon$ .
2. Compute the ENPV of each schedule found in Step 1 using Equations (7) and (8) with task-specific failure rates, and choose a schedule with the largest ENPV.

**REMARK 4.** Since each schedule found by Grinold's algorithm in Step 1 is feasible, the schedule selected in Step 2 is also feasible. Moreover, the ENPV of that schedule is evaluated using task-specific failure rates, consistent with Equation (7). Hence, Heuristic GH finds a lower bound on the maximum ENPV.

**6.2.3. Sequence-Based Heuristic.** Note that for any sequence of task completion times satisfying precedence constraints, by solving problem SSP2, we can find a feasible schedule with an ENPV that is a lower bound on the maximum ENPV. Consequently, we propose the following sequence generation process with ENPV computation, which can be run multiple times to deliver multiple lower bounds.

#### Sequence based heuristic (SH) for ENPV computation

1. Let  $i = 1$  and the initial sequence be empty.
2. Find all the tasks that can be feasibly placed as the  $i$ th task of the current sequence, and from those tasks, with equal probability randomly choose one and place it to be the  $i$ th task to complete in the current sequence.
3. If  $i = n$ , stop; else, let  $i = i + 1$  and go to Step 1.
4. Solve problem SSP2 for the full sequence to find a lower bound on the maximum ENPV.

#### 6.3. Branch and Bound

Using Heuristics GH and SH described in section 6.2 as lower bound methods, and the lower and upper bounds established in sections 5.1 and 5.2, we design a branch and bound algorithm to find a schedule of the tasks in a project with maximum ENPV. Our branch and bound algorithm starts with only the dummy task 0 scheduled, and constructs partial sequences, one task at a time, while maintaining feasibility with respect to precedence constraints, and using the elimination rule established by Theorem 5. From Theorems 3 and 4, we repeatedly fix longer partial sequences, and thus obtain smaller upper bounds and larger lower bounds. If the upper bound of a branch is no larger than the lower bound of another branch, then we eliminate the first branch. The details of our branch and bound algorithm are as follows:

#### Branch and Bound Heuristic (BH)

1. **Initialization.** Specify an allowable tolerance  $\alpha$ ; a current node is deleted if the maximum lower bound found so far is no less than  $(1-\alpha)$  times the upper bound of the current node. Index all the tasks in topological order based on the precedence constraints. Let  $\sigma'$  contain only the dummy task 0, and let  $\sigma''$  contain tasks  $1, \dots, n + 1$ . Run Heuristics GH and SH to find two feasible schedules. Let the global lower bound,  $LB$ , be the maximum of the ENPV values of the schedules found. Let the global upper bound,  $UB$ , be the upper bound found by problem PSSP1 with  $\sigma'$  and  $\sigma''$  as defined above.
2. **Root node.** If  $UB - LB \leq (1-\alpha)UB$ , then stop; otherwise, for each task  $j$  in  $\sigma''$  that feasibly

completes earliest among all the tasks in  $\sigma''$ , create a subnode with updated subsequences  $\sigma'$  and  $\sigma''$  as follows: task  $j$  is scheduled to complete no earlier than the last completed task in  $\sigma'$ , and is removed from  $\sigma''$ . If no subnode exists, then stop. Each subnode defines a lower and an upper bound, as specified in Step 3. Let  $LB_{\max}$  and  $UB_{\max}$  denote the maximum of the lower and upper bounds of all the subnodes, respectively. Update  $LB = \max\{LB, LB_{\max}\}$  and  $UB = \min\{UB, UB_{\max}\}$ .

3. **Nonroot node.** At initialization of a node,  $\sigma'$  and  $\sigma''$  are defined by its parent node. The lower bound of the current node,  $LB'$ , is found by solving problem SSP2, using a full sequence where tasks in  $\sigma'$  are in the sequence specified by  $\sigma'$ , and tasks in  $\sigma''$  are in increasing index sequence. The upper bound of the current node,  $UB'$ , is found by solving problem PSSP1, using partial sequence specified by  $\sigma'$  defined above. Let  $LB = \max\{LB, LB'\}$ . If  $UB' - LB \leq (1 - \alpha)UB'$ , then discontinue branching and return to the parent node; otherwise, evolve the current node with  $LB'$  and  $UB'$  updated when necessary, as for the root node.

## 7. Computational Studies

In this section, we test the performance of our models and solution procedures for project evaluation and selection. In section 7.1, we study the performance of Heuristic BH for finding the maximum ENPV, and demonstrate its improvement over the benchmark GE and Heuristic GH. Observe that the schedule found in GE uses methodology proposed by Grinold (1972), whereas Heuristic GH extends the literature by comparing multiple schedules. In section 7.2, we study how Heuristic BH improves project selection, relative to procedure GE and Heuristic GH. Throughout this section, the granularity parameter  $K$  in Heuristic GH is fixed at 100, and we run Heuristic SH 30,000 times to find a lower bound at the root node. All algorithms are coded in Microsoft Visual C++ 2010 Express with IBM ILOG CPLEX Optimization Studio 12.3, and run on a 4.0 GHz Intel Core i7-6700K computer with 32GB of memory. Section 7.3 provides some insights derived from our results.

### 7.1. Project Evaluation

Our experimental design varies only parameters that affect the performance of the procedures being tested (Hall and Posner 2001). We control parameters including the number of cash flows, depth of project network (i.e., the maximum number of tasks in series in the network), tightness of project deadline, size of failure rates, and pattern of cash flows. For each

parameter combination specified below, we evaluate 30 random instances using networks generated from RanGen (Demeulemeester et al. 2003). RanGen generates networks with controlled *depth indicator*,  $I = (m-1)/(n-1)$ , where  $m$  is depth of the network, and  $n$  is the number of tasks without including any dummy task. However, for greater consistency with the project management literature (Dawson and Dawson 1998, Perry and Greig 1975), we discard the discrete uniformly distributed task times generated by RanGen, and instead generate task times from the Beta(2,2) distribution, and then rescale them to be between 1 and 5 (by multiplying by 4 and then adding 1). In all our studies, we use a common failure rate  $r_f = 0.004$ .

Our test instances are generated as follows. We assign a positive cash flow to the end-of-project dummy task that has a zero processing time and is preceded by all the other tasks. Thus, the number of cash flows is  $n_c = n + 1$ , and task  $n_c$  is the end of project dummy task. For other parameters, we set the depth indicator to 0.5, the ratio of the deadline to the minimum makespan to 2.0, the task-specific failure rate to be generated with value 0 with probability 0.7, and values  $r$ ,  $2r$  and  $3r$  each with probability 0.1, where  $r = 0.001$ , and the number of positive cash flows to 2 (see the Appendix for details on cash flow generation).

We now study how the total number of cash flows, the depth indicator of the project network, the ratio of the deadline to the minimum makespan, the magnitude of the task-specific failure rate, and the number of positive cash flows, affect the performance of the benchmark procedure GE and Heuristics GH and BH, respectively.

First, we investigate the effect of the number of cash flows. We let the number of cash flows be  $n_c \in \{10, 12, \dots, 30\}$ , and generate 30 instances for each  $n_c$  value. In total, we generate  $11 \times 30 = 330$  instances to test the effects of  $n_c$ . Recall that  $n_c = n + 1$ . When  $n_c \leq 20$ , every instance can be solved by Heuristic BH within one CPU hour with an allowable gap of  $\alpha = 0$ . However, when  $n_c \geq 22$ , we need to specify a nonzero tolerance between the lower and upper bounds for several instances to be solved within one CPU hour. Specifically, we increase  $\alpha$  by 0.01 for each additional CPU hour used, until the corresponding instance is solved. For  $n_c = 22, 24, 26, 28, 30$ , the numbers of instances solved with  $\alpha > 0$  are 1, 5, 13, 16, and 22, out of 30, respectively.

Let  $z_{GE}$ ,  $z_{GH}$ ,  $z_{LB}$  and  $z_{UB}$  denote the ENPV found by GE, the same by Heuristic GH, and the global lower and upper bounds from Heuristic BH, respectively. Our computational results for the effect of  $n_c$  appear in Table 2, where each row represents a mean or median result over the corresponding 30 instances. In

Table 2, column “EstGap%” is the mean relative percentage gap between the ENPV found by GE and the upper bound from Heuristic BH (i.e.,  $100(z_{GE}-z_{UB})/z_{UB}$ ); column “Imp%” is the mean percentage improvement from the ENPV found by Heuristic GH over that found by Heuristic BH (i.e.,  $100(z_{LB}-z_{GH})/z_{GH}$ ); column “GHgap%” is the mean relative percentage gap between the ENPV found by Heuristic GH and the upper bound from Heuristic BH (i.e.,  $100(z_{UB}-z_{GH})/z_{UB}$ ); column “BHgap%” is the mean relative percentage gap between the lower and upper bounds found by Heuristic BH (i.e.,  $100(z_{UB}-z_{LB})/z_{UB}$ ); column “Time” is the median running time of Heuristic BH in CPU seconds; and finally, column “Nodes” is the median number of branch nodes used by Heuristic BH.

From Table 2, we observe that procedure GE typically overestimates the maximum ENPV of projects, especially those with a moderate number of cash flows. It appears that the overestimation is less severe when the number of cash flows exceeds 26, which may be because upper bounds for those instances are loose. Heuristic BH significantly improves the schedule found by Heuristic GH, especially for larger projects, providing an overall average improvement of 5.45%, and 13.05% when  $n_c = 30$ . The gap between the ENPV found by Heuristic GH and the upper bound from Heuristic BH increases as the number of cash flows increases. The gap between the lower and upper bounds from Heuristic BH also increases with the number of cash flows, with an overall average of 0.79%.

Next, we set  $n_c = 16$  and investigate the effect of the four other parameters together. First, we study the effect of the depth indicator,  $I \in \{0.3, 0.5, 0.7\}$ . Second, we consider the ratio of the deadline to the minimum makespan,  $d \in \{1.4, 1.7, 2.0\}$ . Third, we set the task-specific failure rate to be 0 with probability 0.7, and values  $r$ ,  $2r$  and  $3r$  each with probability 0.1, with  $r \in \{0.0005, 0.0010, 0.0015\}$ . Fourth, we study the

number of positive cash flows  $n_p \in \{1, 2, 3\}$ ; see the Appendix for details about cash flow generation. Thus, we generate 90 networks from RanGen for  $n_c = 16$  and  $I = 0.3, 0.5, 0.7$ , and using these networks we generate  $3 \times 3 \times 3 \times 30 = 2430$  instances in total.

The effects of depth indicator, deadline, failure rate, and positive cash flows on the various procedures are summarized in Table 3, using the same column interpretations as for Table 2. The overestimation by GE increases with the depth of the project network, the tightness of the deadline where a tighter deadline is represented by a smaller  $d$  value, the task-specific failure rate, and the number of positive cash flows. The improvement from Heuristic BH over Heuristic GH increases with a greater depth indicator, a larger task-specific failure rate, and fewer positive cash flows. The gap between the ENPV from Heuristic GH and the upper bound from Heuristic BH is not significantly affected by the tightness of the deadline, but increases with the depth indicator and the task-specific failure rate, and decreases with the number of positive cash flows. For Heuristic BH, a smaller depth indicator, a larger task-specific failure rate, and a smaller number of positive cash flows make instances harder to solve.

## 7.2. Project Selection

In this section, we study how Heuristic BH improves project selection, relative to procedure GE and Heuristic GH. Recall that in our study of the effects of various parameters in section 7.1, we have in total  $11 + 3 \times 3 \times 3 \times 3 = 92$  parameter combinations, with 30 instances generated for each of them. We assume that the 30 projects for each parameter combination are candidates for project selection, and define them to form a *group*.

Companies often select their projects by ranking them (Goletsis et al. 2003, Green et al. 1996). First, we study how GE, GH, and BH rank projects differently, based on their maximum ENPV values. For each

**Table 2 Performance Sensitivity to Number of Cash Flows**

$n_c$	EstGap% $\frac{100(z_{GE}-z_{UB})}{z_{UB}}$	Imp% $\frac{100(z_{LB}-z_{GH})}{z_{GH}}$	GHgap% $\frac{100(z_{UB}-z_{GH})}{z_{UB}}$	BHgap% $\frac{100(z_{UB}-z_{LB})}{z_{UB}}$	Time CPU seconds median	Nodes Median
10	1.45	0.42	0.42	0.01	6.5	40
12	1.28	1.43	1.39	0.02	7.6	92
14	2.89	0.96	0.96	0.02	10.3	149
16	1.84	2.16	2.06	0.03	14.0	584
18	1.65	3.28	3.13	0.04	19.8	2,565
20	2.54	5.60	5.12	0.04	52.9	52,274
22	2.70	5.04	4.68	0.11	45.1	20,326
24	6.75	6.37	5.90	0.39	327.5	195,615
26	0.23	10.37	9.37	0.88	1,928.8	1,006,923
28	2.17	11.23	11.43	2.28	3,600.4	2,393,527
30	-0.02	13.05	14.86	4.90	10,335.9	6,403,286
All	2.13	5.45	5.39	0.79	32.3	10,724

**Table 3 Sensitivity to Deadline, Failure Rate, and Number of Positive Cash Flows**

Parameter	EstGap% $\frac{100(z_{GE} - z_{UB})}{z_{UB}}$	Imp% $\frac{100(z_{UB} - z_{GH})}{z_{GH}}$	GHgap% $\frac{100(z_{UB} - z_{GH})}{z_{UB}}$	BHgap% $\frac{100(z_{UB} - z_{LB})}{z_{UB}}$	Time CPU seconds median	Nodes Median
$l = 0.3$	0.69	2.33	2.25	0.06	33.0	15,939.5
$l = 0.5$	0.75	2.81	2.62	0.07	14.8	792.0
$l = 0.7$	0.95	2.98	2.75	0.06	12.2	263.5
$d = 1.4$	0.85	2.78	2.57	0.06	16.7	1,062.0
$d = 1.7$	0.80	2.69	2.54	0.08	16.7	1,133.5
$d = 2.0$	0.73	2.65	2.50	0.07	16.7	1,045.0
$r = 0.0005$	0.64	1.12	1.11	0.02	17.0	1,010.0
$r = 0.0010$	0.69	2.79	2.64	0.06	16.6	1,116.0
$r = 0.0015$	1.05	4.21	3.86	0.12	16.6	1,147.5
$n_p = 1$	-1.19	4.15	3.77	0.04	18.9	1,733.0
$n_p = 2$	1.33	2.63	2.46	0.03	15.8	860.0
$n_p = 3$	2.25	1.33	1.38	0.13	15.5	928.0
All	0.79	2.70	2.54	0.07	16.7	1,087.0

group, we compute the *Kendall tau rank correlation coefficient* (Kendall 1938), as defined in the Appendix, between the ENPV estimates from GE and BH, and also between the ENPV estimates found by Heuristics GH and BH. We first consider the Kendall coefficient between the ENPV estimates found by GE and BH. For the eleven groups of instances with  $n_c \in \{10, 12, \dots, 30\}$ , the coefficients are 0.96, 0.94, 0.92, 0.88, 0.86, 0.80, 0.83, 0.82, 0.75, 0.76, and 0.70, respectively, with a mean of 0.84. For the other 81 groups of instances, and for different parameter values of  $l$ ,  $D$ ,  $r$ , and  $n_p$ , results are summarized in the second column of Table 4. The correlation decreases with the depth indicator and the task-specific failure rate, but increases with the number of positive cash flows.

Second, we study the rank correlation between the ENPV's achieved by Heuristics GH and BH. For the eleven groups of instances with  $n_c \in \{10, 12, \dots, 30\}$ , the coefficients are 0.99, 0.96, 0.97, 0.89, 0.92, 0.85, 0.83, 0.86, 0.77, 0.79, and 0.66, respectively, with a mean of 0.86. From Table 4, we observe that the correlation has a similar pattern to that between the ENPV estimates found by GE and BH. Overall, compared with

GE, Heuristic GH provides project rankings that are more consistent with those from Heuristic BH.

Third, we study the effect on total project portfolio value that results from Heuristics GH and BH selecting projects differently, when selection is limited by a budget constraint. To set the budget constraint, we first specify the cost of each individual project. We define the cost of a project as the sum of all the negative cash flows of tasks that are not successors, as specified by the precedence constraints, of the smallest indexed task with a positive cash flow. Then, the portfolio budget limit is set to half the total cost of the 30 projects that are available for selection. Projects are selected to maximize the sum of their estimated ENPVs, subject to the budget constraint on cost. This project selection problem is a classical 0–1 knapsack problem (Kellerer et al. 2004), which we solve optimally using dynamic programming.

We compare the total values of the project portfolios selected by Heuristics GH and BH. For the eleven groups of projects with  $n_c \in \{10, 12, \dots, 30\}$ , Heuristic BH selects project portfolios with 0.2%, 1.0%, 0.8%, 2.9%, 4.2%, 4.5%, 6.7%, 5.1%, 11.6%, 8.5%, and 13.3% greater value, respectively, for an overall mean of 5.4%. We observe that the gain in project portfolio value is particularly significant for larger project sizes. For the other 81 groups of instances, and for different parameter values of  $l$ ,  $D$ ,  $r$ , and  $n_p$ , the fourth column of Table 4 shows the % improvement in project portfolio value delivered by Heuristic BH relative to Heuristic GH. This improvement increases slightly with the depth indicator and decreases slightly with the tightness of the deadline, but increases strongly with the task-specific failure rate and decreases strongly with the number of positive cash flows.

### 7.3. Insights

Our main insights, supported by our computational results, are (a) how project risk declines as tasks are completed, even with constant task failure rate, and

**Table 4 Project Ranking and Portfolio Selection**

Parameter	Coef (GE & BH)	Coef (GH & BH)	Imp%
$l = 0.3$	0.92	0.92	1.5
$l = 0.5$	0.87	0.90	1.7
$l = 0.7$	0.86	0.90	1.7
$d = 1.4$	0.88	0.91	1.6
$d = 1.7$	0.89	0.92	1.6
$d = 2.0$	0.88	0.91	1.8
$r = 0.0005$	0.93	0.95	0.7
$r = 0.0010$	0.88	0.90	1.7
$r = 0.0015$	0.84	0.89	2.5
$n_p = 1$	0.85	0.88	2.6
$n_p = 2$	0.88	0.91	1.6
$n_p = 3$	0.91	0.94	0.7
All	0.88	0.91	1.7



(b) how this decreasing project risk affects the scheduling, evaluation and selection of projects. The quantification of the effect of risk on the project is technically challenging, and we provide an approach to achieve it. Consistent with intuition, our work first offers the following general insights:

1. Tasks with larger cash flow, that is, larger cash inflow or smaller cash outflow, should be processed earlier, holding other factors constant.
2. For a typical project with a positive ENPV, tasks with larger risk should be processed earlier, holding other factors constant.
3. For a typical project with a positive ENPV, tasks with shorter processing time should be processed earlier, holding other factors constant.

However, the three factors of cash flow, risk, and processing time interact with each other in complex ways. For example, in the following instance, it is better to process a task with smaller cash inflow earlier. The instance contains two tasks with  $r_f = 0$ ,  $F_1 = 10$ ,  $F_2 = 100$ ,  $r_1 = r_2 = 0.004$ ,  $D_1 = 10$ , and  $D_2 = 100$ , and no precedence constraints. Processing the two tasks consecutively in the sequence 1→2 results in  $ENPV = 71.10950$ , while sequence 2→1 results in  $ENPV = 49.25000$ . Further, in the following instance, it is better to process a task with smaller risk earlier. The instance contains two tasks with  $r_f = 0$ ,  $F_1 = 10$ ,  $F_2 = 100$ ,  $r_1 = 0.008$ ,  $r_2 = 0.004$ ,  $D_1 = 100$ , and  $D_2 = 10$ . Processing the two tasks consecutively in the sequence 1→2 results in  $ENPV = 31.95036$ , while sequence 2→1 results in  $ENPV = 92.67723$ .<sup>1</sup>

The above examples only consider two sequences 1→2 and 2→1, under which two tasks are processed consecutively without inserted idle time; whereas, we use a continuous optimization model that allows a task to start at any time if feasible, which is more complicated than a pure sequencing decision. These resulting complications support the value of our optimization approach.

## 8. Concluding Remarks

This paper studies the maximization of ENPV for projects under risk, and the selection of projects based on their maximum ENPV. Both the academic and business literatures extensively discuss the issue that project risk declines as tasks are completed. However, our work is apparently the first that explicitly models the reality that the project failure rate at any point in time is dependent on which tasks have been completed. Our optimization model demonstrates that the resulting improvements in scheduling decisions for a project yield a significant improvement in ENPV.

Further, this more precise maximization of ENPV for individual projects enables improved project selection that yields a significant increase in project portfolio value.

Our work contains several insights that inform the project scheduling and selection decisions of managers of project companies. First, it is important to recognize, and incorporate into decision making, the reduction in project risk that occurs when a task is completed, and how this affects project evaluation. Second, in order to achieve this, it is important to quantify the risk associated with individual project tasks and to use that information to schedule projects for maximum ENPV. Third, tasks with higher failure rate and without large cost should be prioritized, since this more quickly reduces the failure rate of a project, which increases its ENPV. Fourth, ENPV values that are adjusted for risk of failure may differ significantly from those that are not, and these differences may be greater for some projects than others, which affects relative project value. Consequently, a significant improvement in the value of a portfolio of projects can be achieved by using our model, and this should be useful to companies that run many projects affected by risk. Finally, our sensitivity analysis results identify project characteristics that are associated with larger improvement in ENPV from our modeling approach, and this information may be especially useful to companies that run projects with such characteristics.

Our work also suggests several directions for future research, including for modeling, algorithm design, and generalizability for additional applications. For modeling, an optimization model can be developed to represent decreasing task failure rates such as occur with a mixture of exponential distributions or a Weibull distribution, and/or project failure rate using a more general function of combinations of the failure rates of the unfinished tasks. We consider fatal risks that cause project failure, and it would be useful to study less severe risks that affect task duration or cash flow. Also, it would be valuable to model intrinsic task risks that start only when the task starts. Further, an empirical study of the way in which project failure rate reduces during execution in specific project management applications would be valuable. For algorithm design, we recommend the development and testing of heuristic rules for adjusting failure rates that are simpler than the optimal task-level optimization approach we describe. In order to address more applications, our work can usefully be generalized by the consideration of additional practical constraints, for example to model resource limitations. Also, the scheduling and selection models in our work can be generalized to support the management of multiple concurrent projects with maximization of total ENPV

objective. In conclusion, we hope that our work will encourage further research on the important topic of ENPV maximization for projects subject to failure.

## Acknowledgments

The authors are grateful to the Department Editor Chelliah Sriskandarajah, the Senior Editor, and the two anonymous reviewers for their valuable comments, which have significantly improved the quality of this work. The first author was partially supported by the National Natural Science Foundation of China [Grants 71421002, 71531010, and 71790592]. The second author was supported in part by Grant No. 71732003 (January 2018–December 2022) from the National Natural Science Foundation of China, and in part by the Berry Professorship at the Fisher College of Business, The Ohio State University.

## Note

<sup>1</sup>The above three insights and two instances are recommended by an anonymous reviewer.

## References

- Archer, N. P., F. Ghasemzadeh. 1999. An integrated framework for project portfolio selection. *Int. J. Project Manage.* **17**: 207–216.
- Bard, J. F. 1985. Parallel funding of R&D tasks with probabilistic outcomes. *Management Sci.* **31**(7): 814–828.
- Barlow, R. E., F. Proschan. 1965. *Mathematical Theory of Reliability*. Wiley, Hoboken, NJ.
- Beaves, R. G. 1993. The case for a generalized net present value formula. *Eng. Econ.* **38**(2): 119–133.
- Blau, G. E., J. F. Pekny, V. A. Varma, P.R. Bunch. 2004. Managing a portfolio of interdependent new product candidates in the pharmaceutical industry. *J. Prod. Innov. Manag.* **21**(4): 227–245.
- Browning, T. R., R. V. Ramasesh. 2007. A survey of activity network-based process models for managing product development projects. *Prod. Oper. Manag.* **16**(2): 217–240.
- Chapman, C., S. Ward. 2002. *Managing Project Risk and Uncertainty: A Constructively Simple Approach to Decision Making*. Wiley, Chichester, U.K.
- Chen, W.-N., J. Zhang. 2012. Scheduling multi-mode projects under uncertainty to optimize cash flows: A Monte Carlo ant colony system approach. *J. Comput. Sci. Technol.* **27**(5): 950–965.
- Coolen, K., W. Wei, F. T. Nabibon, R. Leus. 2014. Scheduling modular projects on a resource bottleneck. *J. Sched.* **17**: 67–85.
- Cooper, R. G. 1985. Selecting winning new product projects: Using the NewProd system. *J. Prod. Innov. Manag.* **2**(1): 34–44.
- Cooper, R. G., S. J. Edgett, E. J. Kleinschmidt. 2001. Portfolio management for new product development: Results of an industry practices study. *R&D Management* **31**(4): 361–380.
- Creemers, S., B. De Reyck R. Leus. 2015. Project planning with alternative technologies in uncertain environments. *Eur. J. Oper. Res.* **242**: 465–476.
- Damodaran, A. 2007. *Strategic risk taking: A framework for risk management*. Chapter 5, Risk adjusted value, 99–144, Pearson Prentice Hall, Upper Saddle River, NJ.
- Dawson, R. J., C. W. Dawson. 1998. Practical proposals for managing uncertainty and risk in project planning. *Int. J. Project Manage.* **16**(5): 299–310.
- De Reyck, B., R. Leus. 2008. R&D project scheduling when activities may fail. *IIE Trans.* **40**: 367–384.
- De Reyck, B., Y. Grushka-Cockayne, R. Leus. 2007. A new challenge in project scheduling: The incorporation of activity failures. *Tijdschrift voor Economie en Management* **LIII**(3) 411–434, Katholieke Universiteit Leuven, Belgium.
- Demeulemeester, E., M. Vanhoucke, W. Herroelen. 2003. RanGen: A random network generator for activity-on-the-node networks. *J. Sched.* **6**: 17–38.
- Doersch, R. H., J. H. Patterson. 1977. Scheduling a project to maximize its net present value: A zero-one programming approach. *Manage. Sci.* **23**(8): 882–889.
- El Emam, K., A. G. Koru. 2008. A replicated survey of IT software project failures. *IEEE Softw.*, September–October 84–90.
- Elandt-Johnson, R. C., N. L. Johnson. 1999. *Survival Models and Data Analysis*. John Wiley & Sons, New York, NY.
- Ellinas, C. 2019. The domino effect: An empirical exposition of systemic risk across project networks. *Prod. Oper. Manag.* **28**(1): 63–81.
- Elmaghraby, S. E., W. S. Herroelen. 1990. The scheduling of activities to maximize the net present value of projects. *Eur. J. Oper. Res.* **49**(1): 35–49.
- Etgar, R., A. Shtub. 1999. Scheduling project activities to maximize the net present value - The case of linear time-dependent cash flows. *Int. J. Prod. Res.* **37**(2): 329–339.
- Etgar, R., A. Shtub, L. J. LeBlanc. 1996. Scheduling projects to maximize net present value - The case of time-dependent, contingent cash flows. *Eur. J. Oper. Res.* **96**(1): 90–96.
- Fercho, W.W., L.J. Ringer. 1972. Small sample power of some tests of the constant failure rate. *Technometrics* **14**(3): 713–724.
- Goletsis, Y., J. Psarras, J. E. Samouilidis. 2003. Project ranking in the Armenian energy sector using a multicriteria method for groups. *Ann. Oper. Res.* **120**(1–4): 135–157.
- Green, R.H., J.R. Doyle, W.D. Cook. 1996. Preference voting and project ranking using DEA and cross-evaluation. *Eur. J. Oper. Res.* **90**(3): 461–472.
- Grinold, R.C. 1972. The payment scheduling problem. *Nav. Res. Logist. Q.* **19**(1): 123–136.
- Haley, G. T., S. M. Goldberg. 1995. Net present value techniques and their effect on new product research. *Ind. Mark. Manage.* **24**(3): 177–190.
- Hall, N. G. 2016. Research and teaching opportunities in project management. *Optimization Challenges in Complex, Networked and Risky Systems*. INFORMS, Catonsville, MD, 329–388.
- Hall, N. G., M. E. Posner. 2001. Generating experimental data for computational testing with machine scheduling applications. *Oper. Res.* **49**: 854–865.
- Herroelen, W. 2005. Project scheduling – Theory and practice. *Prod. Oper. Manag.* **14**(4): 413–432.
- Herroelen, W. S., E. Gallens. 1993. Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value of projects. *Eur. J. Oper. Res.* **65**(2): 274–277.
- Herroelen, W. S., P. Van Dommelen, E. L. Demeulemeester. 1997. Project network models with discounted cash flows – A guided tour through recent developments. *Eur. J. Oper. Res.* **100**(1): 97–121.
- Hodder, J. E., H. E. Riggs. 1995. Pitfalls in evaluating risky projects. *Harv. Bus. Rev.* **63**(1): 128–135.
- Hu, X., N. Cui, E. Demeulemeester. 2015. Effective expediting to improve project due date and cost performance through buffer management. *Int. J. Prod. Res.* **53**(5): 1460–1471.
- Icmeli, O., S. S. Erenguc. 1996. A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows. *Management Sci.* **42**(10): 1395–1408.
- Jain, V., I. E. Grossmann. 1999. Resource-constrained scheduling of tests in new product development. *Ind. Eng. Chem. Res.* **38**: 3013–3026.

- Kellerer, H., U. Pferschy, D. Pisinger. 2004. *Knapsack Problems*. Springer, Berlin, Germany.
- Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika* 30(1–2): 81–93.
- Kerzner, H. 2013. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 11th edition*. Wiley, Hoboken, NJ.
- Kettunen, J., A. Salo. 2017. Estimation of downside risks in project portfolio selection. *Prod. Oper. Manag.* 26(10): 1839–1853.
- Khanfor, A., Y. Yang, G. Vesonder, G. Ruhe, D. Messinger. 2017. Failure prediction in crowdsourced software development, The 24th Asia-Pacific Software Engineering Conference, accepted.
- Kolisch, R., K. Meyer. 2006. Selection and scheduling of pharmaceutical research projects. Perspectives in Modern Project Scheduling. *Int. Ser. Oper. Res. Manag. Sci.* 92: 321–344.
- MacMillan, I. C., R. G. McGrath. 2002. Crafting R&D project portfolios. *Res. Technol. Manag.* 45(5): 48–59.
- Mishra, A., S. R. Das, J. J. Murray. 2016. Risk, process maturity, and project performance: An empirical analysis of US federal government technology projects. *Prod. Oper. Manag.* 25(2): 210–232.
- Myers, S. C., L. Shyam-Sunder. 1996. Measurement of pharmaceutical industry risk and the cost-of-capital. R. B. Helms, ed. *Competitive Strategies in the Pharmaceutical Industry*. American Enterprise Institute, Washington, D.C., 208–237.
- Nelson, C. A. 1986. A scoring model for flexible manufacturing system project selection. *Eur. J. Oper. Res.* 24(3): 346–369.
- Niu, B., K. Chen, X. Fang, X. Yue, X. Wang. 2019a. Technology specifications and production timing in a co-opetitive supply chain. *Prod. Oper. Manag.* 28(8): 1990–2007.
- Niu, B., J. Li, J. Zhang, H. K. Cheng, Y. Tan. 2019b. Strategic analysis of dual sourcing and dual channel with an unreliable alternative supplier. *Prod. Oper. Manag.* 28(3): 570–587.
- Oral, M., O. Kettani, Ü. Cinar. 2001. Project evaluation and selection in a network of collaboration: A consensual disaggregation multi-criterion approach. *Eur. J. Oper. Res.* 130(2): 332–346.
- Perry, C., I. D. Greig. 1975. Estimating the mean and variance of subjective distributions in PERT and decision analysis. *Management Sci.* 21(12): 1477–1480.
- Poh, K. L., B. W. Ang, F. Bai. 2002. A comparative analysis of R&D project evaluation methods. *R&D Management* 31(1): 63–75.
- Project Management Institute. 2008. Should You Be Teaching Project Management. Project Management Institute, Newtown Square, PA. Available at [http://www.displayosaicprojects.com.au/PDF/Why\\_teach\\_PM.pdf](http://www.displayosaicprojects.com.au/PDF/Why_teach_PM.pdf) (accessed date August 23, 2019).
- Quinn, J. B., F. G. Hilmer. 1994. Strategic outsourcing. *Sloan Manag. Rev.* 35: 43–55.
- Ranjbar, M., M. Davari. 2013. An exact method for scheduling of the alternative technologies in R&D projects. *Comput. Oper. Res.* 40: 395–405.
- Remer, D. S., A. P. Nieto. 1995. A compendium and comparison of 25 project evaluation techniques. *Int. J. Prod. Res.* 42: 79–96.
- Russell, A. H. 1970. Cash flows in networks. *Management Sci.* 16(1): 357–373.
- Russell, R. A. 1986. A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Management Sci.* 32(10): 1291–1300.
- Schmidt, C. W., I. E. Grossmann. 1996. Optimization models for the scheduling of testing tasks in new product development. *Ind. Eng. Chem. Res.* 35: 3495–3510.
- Schwindt, C., J. Zimmermann. 2001. A steepest ascent approach to maximizing the net present value of projects. *Math. Method Oper. Res.* 53: 435–450.
- Sobel, M. J., J. G. Szmerekovsky, V. Tilson. 2009. Scheduling projects with stochastic activity duration to maximize expected net present value. *Eur. J. Oper. Res.* 198: 697–705.
- Stollmack, S., C.M. Harris. 1974. Failure-rate analysis applied to recidivism data. *Oper. Res.* 22(6): 1192–1205.
- Vanhoucke, M., E. Demeulemeester, W. Herroelen. 2001a. Maximizing the net present value of a project with linear time-dependent cash flows. *Int. J. Prod. Res.* 39(14): 3159–3181.
- Vanhoucke, M., E. Demeulemeester, W. Herroelen. 2001b. On maximizing the net present value of a project under renewable resource constraints. *Management Sci.* 47(8): 1113–1121.
- Walker, J. 1997. Base stock level determination for “insurance type” spares. *Int. J. Qual. Reliab. Manag.* 14(6): 569–574.
- Wiesemann, W., D. Kuhn. 2015. The stochastic time-constrained net present value problem. Chapter 35 in *Handbook on Project Management, v.2, International Handbooks on Information Systems*, Springer, Berlin, 753–780.
- Wiesemann, W., D. Kuhn, B. Rustem. 2010. Maximizing the net present value of a project under uncertainty. *Eur. J. Oper. Res.* 202(2): 356–367.
- Wu, Y., K. Ramachandran, V. Krishnan. 2014. Managing cost salience and procrastination in projects: Compensation and team composition. *Prod. Oper. Manag.* 23(8): 1299–1311.
- www.synapseindia.com 2019. 6 stages of software development process. Available at <https://www.synapseindia.com/6-stages-of-software-development-process/141> (accessed date August 23, 2019).
- Yang, K.K., F.B. Talbot, J.H. Patterson. 1992. Scheduling a project to maximize its net present value: an integer programming approach. *Eur. J. Oper. Res.* 64(2): 188–198.
- Zipfel, A. 2003. Modeling the probability-cost-profitability architecture of portfolio management in the pharmaceutical industry. *Drug Inf. J.* 37 185–205.

## Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

## Appendix: Technical Details.