Computers & Operations Research 143 (2022) 105737

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Data-driven project portfolio selection: Decision-dependent stochastic programming formulations with reliability and time to market requirements

Janne Kettunen *,[1], Miguel A. Lejeune[1,2]

Department of Decision Sciences, George Washington University, 2201 G Street NW, Washington, DC, 20052, USA

ARTICLE INFO

Keywords:
Stochastic programming
Dynamic chance constraint
Project portfolio selection
New product development
Ideal plan

ABSTRACT

We develop stochastic programming models for project portfolio selection under the plan-driven *waterfall* approach and the more flexible *agile* approach. The models account for the requirement to earn return fast and to generate a certain return with high probability. The models take the form of static (waterfall) and dynamic (agile) disjunctive integer nonconvex chance-constrained problems. To make the models computationally tractable, we devise model strengthening approaches and decomposition methods. We also develop an algorithm to obtain an ideal investment plan that provides the targeted probabilistic return as quickly as possible whilst maximizing the excess return. Using a representative US-based software company data, we show the significance of the benefits given by the ideal plan. Our results show that the probabilistic return can be reached faster under the agile, as compared to the waterfall, approach. This can partially explain why agile approaches are popular in new product development. Counterintuitively, the results show that the agile approach, which includes more stochasticity sources than the waterfall approach, leads to less uncertainty regarding the time to reach a certain return than the waterfall approach. The reason for this outcome is the dynamic abandoning and re-starting of new projects protecting from downside risks, and hence, from outcomes that would result in longer time to reach the required return level. Furthermore, we introduce a visualization tool to guide a venture capitalist's investment. The visualization tool highlights the company's performance regions derived with the proposed models. The numerical tests show that the developed models are robust and computationally tractable and can be used for larger problems, with more projects, time periods, and uncertainties.

1. Introduction

We focus on one of the most important problems in project management, namely the selection of the projects to work on Cooper et al. (2000) and Hall et al. (2015). This problem is prominent in the new product development (NPD) context (Kavadias and Chao, 2008) where about $2 trillion are spent on new development projects each year (Industrial Research Institute, 2016). The selection of the portfolio of NPD projects occurs periodically (Federal Aviation Administration, 2012). The selections are made from the pool of available projects that include the ongoing development projects and new ones (Cooper et al., 2000).

A common periodic project portfolio selection approach, where the selection decisions are made concurrently for all projects, is the stage gate approach with mass gates for all projects (Cooper et al., 2000). Cooper and Edgett (2012) found that about 90% of the best performing companies employ an NPD process, such as the concurrent portfolio review stage gate approach. The benefits of the periodic concurrent portfolio selection approach include accounting for diversification via the correlations among project returns and accounting for the flexibility to reallocate resources if better development projects become available (Chien, 2002; Kavadias and Chao, 2008; Hall et al., 2015). Therefore, the periodic concurrent portfolio selection allows for a more efficient use of resources than the one that could be achieved by evaluating and deciding about each project independently and by making the selection decision only once for each project, without periodic review.

In this study, we consider both the *waterfall* and *agile* project management processes and account for their impact on the project selection. The waterfall process relies on the traditional plan-driven approach (Hass, 2007; Larson and Gray, 2014). Under this approach,

* Corresponding author.
E-mail addresses: jkettune@gwu.edu (J. Kettunen), mlejeune@gwu.edu (M.A. Lejeune).

[1] Both of the authors contributed equally for this work.

[2] M. Lejeune acknowledges the partial support from the Office of Naval Research through Grant #N000141712420 and from the National Science Foundation ECCS-2114100 and ICER2022505.

https://doi.org/10.1016/j.cor.2022.105737
Received 8 April 2021; Received in revised form 3 December 2021; Accepted 28 January 2022
Available online 18 February 2022
0305-0548/

the projects are selected here-and-now for the entire planning horizon. The agile process is dynamic, employing the wait-and-see project selection approach (Blank, 2013; Yoo et al., 2017). Under this approach, projects are selected only for one period at a time after which recourse decisions are made accounting for how the projects' development has succeeded. Consequently, it may be beneficial to abandon some of the ongoing projects whose development has not progressed well and to initiate the development of new projects. In fact, the best performing companies are reported to abandon about 25% of their ongoing development projects during the periodic project portfolio selection (Edgett, 2010).

1.1. Literature review

Our study is related to the project management, especially in project portfolio selection, and stochastic programming fields. Next, we review the relevant literature in these two fields.

The literature on project portfolio selection emphasizes the challenges caused by uncertainties in projects' returns (Kavadias and Chao, 2008). Kleywegt and Papastavrou (1998) model the project selection with uncertainty in their return. They consider that the projects arrive individually according to a Poisson process and that the decision to invest or not has to be made for each project at the time of its arrival. They formulate the problem as a stochastic knapsack problem and propose recursive algorithms to solve it. Lu et al. (1999) consider a similar kind of problem where individual project proposals arrive at random time intervals and the problem is to decide upon arrival of the project whether to invest or not, given a fixed available budget over the planning horizon. They solve the problem using dynamic programming and determine time intervals when projects should be invested in to maximize the expected return obtained within a certain time horizon. Hall et al. (2015) develop a project portfolio selection approach where the portfolio of projects is selected concurrently under uncertainty in projects' return characterized by the moments of their distribution. The goal is to minimize the risk of not attaining a specific target return.

Our review of the stochastic programming literature (Birge and Louveaux, 2011) is focused on chance-constrained problems (see reviews in Prékopa, 2003; Lejeune and Prékopa, 2021). In particular, we review chance-constrained problems with joint probabilistic constraints and random technology matrix where random variables follow a distribution with finite support. The review focuses on these types of problems, since the models proposed in this study share the same characteristics. To solve these types of problems, Prékopa (1990) introduces the p-efficiency concept. In addition, there are two main families of reformulation and solution methods that have been employed, which we refer to as scenario-based and Boolean-based methods. To our knowledge, the first scenario-based approach is due to Ruszczyński (2002) who constructs a partial order on the set of scenarios and proposes a family of cutting planes used within a branch-and-cut algorithm. Tanner and Ntaimo (2010) derive a mixed-integer linear programming formulation in which they incorporate irreducibly infeasible (II) optimality cuts. An extension of this approach was recently proposed by Canessa et al. (2019) for pure binary chance-constrained problems, in which decision variables are binary and joint chance constraints can be handled. The II cuts reduce the number of nodes visited and accelerate convergence to the optimal solution. Within the Boolean-based approach, Kogan and Lejeune (2014) extend the Boolean framework initially proposed for chance constraints with random right-hand sides (Lejeune, 2012) and reformulate or inner-approximate problems with joint probabilistic constraints in which the elements of a multi-row random technology matrix follow a joint probability distribution. Lejeune and Margot (2016) extend the Boolean reformulation framework to joint probabilistic constraints in which quadratic, nonconvex stochastic inequalities are required to hold jointly with some large probability. They design spatial branch-and-bound algorithms and identify conditions under which the reformulated problem is equivalent to the original formulation. The Boolean framework has recently been applied to a forestry-motivated binary stochastic programming problem including both classical and integrated joint chance constraints (Lejeune and Kettunen, 2017). We refer the reader to recent and in-depth reviews of the literature about chance-constrained models under finite (Ahmed and Xie, 2018) or continuous (Lejeune and Prékopa, 2021) distributions.

The proposed models differ in two fundamental ways from the above models. First, they take the form of *disjunctive* stochastic optimization models. Second, one of the two models includes a *dynamic* joint chance constraint while the above studies consider a static decision context, in which decisions are made once at the beginning of the horizon and are not updated as uncertainty is revealed. We now review the even scanter literature on this very complex and specific type of dynamic chance-constrained problem. Lulli and Sen (2004) consider a probabilistic batch-sizing problem under a finite discrete demand distribution. In their model, nonanticipativity of decisions is enforced only for the scenarios that meet the desired service constraint. Andrieu et al. (2010) study chance constraints with dynamic (multi-stage) decisions that appear in hydro power reservoir management. The authors assume a continuous probability distribution and give a finite dimensional approximation of the infinite-dimensional chance constraint by discretizing the continuous decision variables. Relying upon its mixing and continuous mixing substructures, Zhang et al. (2014) derive valid inequalities for the dynamic chance-constrained model and illustrate the computational benefits on an inventory management problem. Ono et al. (2015) propose an algorithmic method for joint chance-constrained dynamic programming (control) problems that restrict the probability of violating state constraints. They solve conservative inner approximations derived with the Boole–Bonferroni bounding scheme and test their method on several optimal control problems.

In comparison to the extant literature, the proposed models have an additional major source of complexity as they belong to the family of disjunctive programming problems (Balas, 1979, 2018), and takes the form of a disjunctive chance programming problems. The disjunctive form of the models is due to the presence of decision-dependent uncertainty of Type 2 in which decisions affect the time at which information is revealed and uncertainty gets resolved (Jonsbraten et al., 1998). Stochastic problems with Type 2 decision-dependent uncertainty are usually formulated as risk-neutral multi-stage stochastic programming problems (see, e.g., Apap and Grossmann, 2017; Colvin and Maravelias, 2010; Goel and Grossmann, 2006; Jonsbraten et al., 1998; Tarhan et al., 2009) in which the non-anticipativity conditions are decision-dependent. In contrast to these earlier studies, the models proposed here are risk-averse chance-constrained stochastic programming problems with decision-dependent uncertainty. We refer the interested reader to Hellemo et al. (2018) for a recent review and taxonomy of stochastic programming problems with decision-dependent uncertainty.

To our knowledge, disjunctive chance constraints have been studied for the first time in the recent study by Kettunen and Lejeune (2020) which we extend here in the following way. As detailed in Section 3, the proposed chance-constrained problems involve a number of periods at which investment decisions are taken. The number of periods to reach the pursued return objective and at which decisions are taken is unknown ex-ante. The chance constraint is a joint one in which the number of stochastic inequalities that must hold jointly with a prescribed probability level is equal to the number of periods. At each period, one will check if the targeted return level is attained. If yes, the investment plan is successful and interrupted. If not attained, one must consider an additional period. This gives the disjunctive nature to the proposed formulations – hence the name disjunctive stochastic programming – since one must reach the targeted return level at (any) one of the periods (the earlier the better) in the planning horizon. We design a modeling and algorithmic framework that (i) defines the fastest time

τ^* by which the targeted return can be attained (as done in Kettunen and Lejeune (2020)) and (ii) determines the *ideal plan* that provides the *highest* excess return that can be reached by τ^* (not considered in Kettunen and Lejeune (2020)). In addition to using a decomposition method similar to the one proposed by Kettunen and Lejeune (2020) for (i), we (1) propose new valid inequalities and tightening procedures to supplement the decomposition method and improve the solution process; (2) develop a new bisection-type algorithm to elicit the ideal plan; (3) carry out a computational study to evaluate the efficiency of the proposed modeling and algorithmic developments; (4) assess the robustness of the models and their results with respect to the number of scenarios used to represent uncertainty; and (5) employ the framework to derive insights for a US-based software start-up company in a project portfolio selection problem.

1.2. Contributions and paper structure

We contribute to the stochastic programming and project management literature in the following two main ways:

1. We make methodological and algorithmic contributions. Our key methodological contributions are to develop optimization models for the static (waterfall) and dynamic (agile) project portfolio selection approaches of the software start-up company's project selection problem. Our algorithmic contributions involve two major extensions to the solution framework proposed by Kettunen and Lejeune (2020). First, we develop model tightening procedures to support the decomposition method and to accelerate the solution process. Second, we formulate a bisection algorithm to obtain an ideal investment plan that provides the targeted probabilistic return as quickly as possible whilst maximizing the excess return (over the targeted one). We show that the developed algorithms allow for the quick and optimal solution of realistic-size problem instances. Furthermore, our computational experiments show that the developed models are robust and can be conveniently used to solve even larger problems (i.e., with more projects, time periods, and uncertainties) than those considered in this study, since the decisions are stable and converge to the same outcome with different sizes of the scenario set.

2. Our contributions to practice relate to data-driven insights based on a US-based software company and a possible approach to visualize and benchmark venture's profitability over time in order to support venture capitalists' investment decisions. First, our results show that the *ideal* investment plan can lead to significantly higher probabilistic return than an approach that solely focuses on the time at which the targeted return is reached, such as the one developed by Kettunen and Lejeune (2020). In particular, our results show that the ideal investment plan can provide a 13% excess return. Second, our results show that there is less uncertainty to attain a pre-specified return under the agile development approach than under the waterfall one. This additional benefit of the agile approach can make a difference on receiving funding from a venture capitalist with limited time window for the investment. Third, we propose and demonstrate how to visualize the venture's profitability over time vis-á-vis three regions that characterize the company's performance. This type of visualization can be used to guide a venture capitalist's investment on a company over time. Specifically, depending on the region corresponding to the company's actual performance, the guidelines suggest the venture capitalist to either (i) intervene on company's operations, (ii) support the operations, or (iii) retain the company and let it run independently. The regions are derived so that they are company-specific and account for the type of projects a company is expected to have in the future and for the project management approach used, i.e., waterfall or agile.

The paper is structured as follows. Section 2 describes the project selection problem in the multi-period NPD context and its main characteristics. In Section 3, we develop mathematical formulations for the waterfall and agile product development approaches. In Section 4, we propose several model strengthening approaches that facilitate solving the models, develop an algorithmic method that solves exactly and efficiently the proposed optimization problems, and design a bisection-type algorithm to determine the ideal investment plan that provides the largest probabilistic excess return. Section 5 describes the data, analyzes the scalability and efficiency of the algorithmic method, and presents managerial implications. In Section 6, we provide conclusions.

2. Problem description

We consider one of the essential operational decisions of companies with research and development capabilities, namely which new products to develop. The development of new products is a multi-period problem. In other words, the progress of the products under development is evaluated periodically (e.g., quarterly or annually) when also new development projects are proposed. At each period, companies face the *project portfolio selection problem* (Linton et al., 2002; Heidenberger and Stummer, 2003; Hassanzadeh et al., 2014; Hall, 2016), i.e., each proposed project has to be either selected, and its development fully funded for the following period, or abandoned. The selection decisions are constrained so that only a certain number of projects can be developed concurrently.

The multi-periodicity in the NPD selection process incorporates several key characteristics. First, the returns of the projects, which were chosen for development in the last review period, evolve stochastically depending on the progress of the development efforts (Sommer et al., 2007). Second, the returns and risks of the new development projects are not known until at the period when they are proposed and evaluated (Kleywegt and Papastavrou, 1998; Lu et al., 1999). Third, during each period the company occurs development costs, regardless whether a project is completed. These costs are fixed and consist mainly from the salaries of the developers as is exemplified in IT, electronics, and software industries (Sommerville, 2016; OECD, 2017).

In our study, we focus on the NPD selection decisions made by start-up companies. One of the most important goals of start-up companies is the speed to get the products on market and earn return fast to the company and its investors (Kamuriwo and Baden-Fuller, 2013; Blank, 2013; Yoo et al., 2017). This goal is especially relevant to keep the company in business and to secure additional funding for growth (Bodily, 2016). The second goal relates to the requirement to generate a certain return with some guarantee or probability level, which represents the goals of decision makers (DMs) in the NPD and several other contexts (Hall et al., 2015). Therefore, the objective of DMs can be viewed as minimizing the time to attain a return target with a specified probabilistic guarantee or reliability level (Ibrahim, 2008; Kamuriwo and Baden-Fuller, 2013).

The same problem can be also considered from the venture capitalists' point of view, who must assess whether the start-up company is worth investing in, i.e., will the start-up company attain early enough the required return with a certain reliability level (Zider, 1998). Consequently, the models developed for this problem serve as decision support tools for both start-up's DMs and venture capitalists.

We consider two distinct product development processes, namely (i) *waterfall* and (ii) *agile*. The waterfall approach employs the traditional plan-driven product development process (Hass, 2007; Larson and Gray, 2014). Under this process, the projects' returns are uncertain and the project selection decisions are irreversible. The agile approach accounts for the dynamic decision making (Blank, 2013; Yoo et al., 2017). The dynamic decision making implies that DMs can during each evaluation period abandon previously initiated projects, if their development seems to fail or better projects become available, and start new projects. The dynamic decision making implies also that the overall development path is uncertain. As a result, the type of projects that will become available in the future periods are uncertain as well as their duration. Thereby, under the agile development process, the projects' returns and durations are uncertain and the project selection decisions are made dynamically.

Table 1
Notations.

Parameters	
a	Earliest time at which a project can be completed
d_i	Duration of project i
s_i	Earliest starting time for project i
f_i	Earliest completion time for project i
e	Vector of ones
F	Fixed development cost per period
U	Upper bound on number of projects that can be developed at any period
R	Targeted return level
Sets	
I	Set of projects
T	Set of time periods
$T_i = \{s_i, \dots, f_i\}$	Set of periods during which project i can be developed
$C_t = \{i \in I : f_i = t\}$	Set of projects that can be completed at period t
$I_t = \{i \in I : f_i = t\}$	Set of projects that can be developed at period t
$\bar{T} = \{a, \dots, \|T\|\}$	Set of periods when a project can be developed
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}_+$	Set of nonnegative integer numbers
$\mathbb{B} = \{0, 1\}$	Set of binary variables
Decision variables	
$x_{i,t} \in \mathbb{B}$	Binary variable equal to 1 if project i is developed in period t and to 0 otherwise
$\tau \in \mathbb{Z}_+$	Time at which targeted return level is attained
Uncertainty related notations	
$\xi_{i,t}$	Random revenue generated at period t by project i
$\tilde{\xi}^k_{i,t}$	Realization of revenue for project i in period t in scenario k
K	Scenario index set
$\Omega = \{\tilde{\xi}^k : k \in K\}$	Set of joint scenarios for random revenue vector ξ
q^k	Probability of scenario k
p	Prescribed reliability level

3. Models and reformulations

This section consists of two subsections. In Section 3.1, we focus on the waterfall model and derive its stochastic and deterministic reformulations. Section 3.2 focuses on the stochastic agile model and its reformulation. These models are similar to those presented by Kettunen and Lejeune (2020) in terms of defining the fastest time by which the targeted return can be attained.

3.1. Stochastic waterfall project selection model and deterministic reformulation

We present now the formulation of the chance-constrained waterfall project selection problem that accounts for the key features of the waterfall selection approach and for uncertainty in the generated revenue. The uncertainty in the projects' revenue is modeled with a finite set of joint, multi-dimensional scenarios $\tilde{\xi}^k \in \mathbb{R}^{|I|\times|T|}, k \in K$ with each component $\tilde{\xi}^k_{i,t}$ of $\tilde{\xi}^k$ representing the revenue generated by project i at time t in scenario k. The sum of the probabilities of the scenarios is equal to 1: $\sum_{k\in K} q^k = 1$.Table 1 summarizes the notations used in the formulation of the waterfall model.

The waterfall project selection model is the following disjunctive chance-constrained nonconvex integer programming problem:

$$\textbf{W-M} : \min \ \tau \tag{1a}$$

$$\text{s.to}(x,\tau) \in \bigvee_{t\in\{a,\dots,\tau\}} \mathcal{H}_t \tag{1b}$$

$$x \in \mathcal{X} \tag{1c}$$

$$\tau \in \bar{T} \subset \mathbb{Z}_+ \tag{1d}$$

where the deterministic integer linear feasible set $\mathcal{X}$ is given by

$$\mathcal{X} := \left\{ x \in \{0,1\}^h : e'x_t \le U, t \in T;\ x_{i,t} = x_{i,t+1}, i \in I, t \in T_i \setminus \{f_i\} \right\} \tag{2}$$

with $h = \sum_{i\in I} |T_i|$ denoting the number of binary variables. Each $\mathcal{H}_t$ corresponds to the feasible set of the chance constraint $\mathbb{P}\left(\sum_{t'=a}^{t} \sum_{i\in C_{t'}} x_{i,t'}\xi_{i,t'} - tF \ge R\right) \ge p$ at time $t \in \{a, \dots, \tau\} \subseteq \bar{T}$:

$$\mathcal{H}_t := \left\{ x \in \{0,1\}^{|C_t|} : \mathbb{P}\left(\sum_{t'=a}^{t} \sum_{i\in C_{t'}} x_{i,t'}\xi_{i,t'} - tF \ge R \right) \ge p \right\}. \tag{3}$$

The expression $(x,\tau) \in \bigvee_{t\in\{a,\dots,\tau\}} \mathcal{H}_t$ in (1c) is a disjunction and requires the finding of a feasible solution (x,τ) that provides the targeted return level R in at least one of the periods in the planning horizon. In Boolean mathematics terms (Crama and Hammer, 2011), the expression $\bigvee_{t\in\{a,\dots,\tau\}} \mathcal{H}_t$ is called a *disjunctive normal form* over a set of *literals* $\mathcal{H}_t$ with each literal defining the set of conditions required to reach the targeted return at period t with a reliability level at least equal to p. The number $(\tau - a)$ of literals $\mathcal{H}_t$ in the disjunction is unknown ex-ante and itself a decision variable since τ is a decision variable determined via the solution of problem **W-M**. The constraint included in $\mathcal{H}_t$ is a chance constraint with random technology matrix (1b) that requires to generate a return R with probability at least p at t. The term tF is the cumulative cost until period t.

The objective function (1a) minimizes the time τ needed to provide the targeted revenue level R with the set reliability level p. The knapsack constraints in $\mathcal{X}$ do not allow for more than U projects to be concurrently developed at each period and result from the tightness of the available resources. The precedence constraints $x_{i,t} = x_{i,t+1}$ in $\mathcal{X}$ ensure first that the initiated projects must not be abandoned and must be completed, and second that projects not initiated at the period they became available cannot be started later. This means that the benefits of a project can only be collected if the project is supported from inception to completion and without interruption: $x_{i,t} = 1, t = s_i, \dots, f_i$.

The above deceptively simple-looking nonconvex optimization problem is extremely challenging to solve and includes several sources of nonconvexity stemming from the integrality restrictions on the decision variables and the fact that the number of literals $\mathcal{H}_t$ in the disjunction is itself a decision variable. Problem **W-M** is not, in this form, amenable to an analytical solution, nor to a numerical solution by any off-the-shelf optimization solvers. Therefore, we need to first derive a deterministic equivalent reformulation of problem **W-M** (see

Theorem 1) and then to devise an algorithmic method allowing for its numerical solution of practice-sized problem instances (see Section 4).

We shall now derive in Theorem 1 a deterministic, equivalent, and computationally tractable reformulation **W-RM** of the waterfall project selection model **W-M**. We denote by $\gamma_t, t \in \tilde{T}$ a binary variable taking value 1 if the return level R can be attained at period t and taking value 0 otherwise. Additionally, the binary variable $\beta_t^k, k \in K, t \in \tilde{T}$ is defined in such a way that it can take value 1 if all conditions defined by scenario $\tilde{\xi}^k$ in time t hold jointly and is forced to be 0 otherwise. The notation $\bar{\xi}_{i,t}^k$ refers to the series of realized revenues sorted in increasing order while $M \in \mathbb{R}^{t\in|\tilde{T}|\times|K|}$ is a vector of fixed parameters. The components of the vector M are set up ex-ante to the smallest possible value that does not eliminate any feasible solution of the problem.

Theorem 1. *The disjunctive integer nonlinear chance-constrained problem* **W-M** *can be equivalently reformulated as the following deterministic integer linear programming problem* **W-RM**:

$$\textbf{W-RM}: \ \min \sum_{t\in\tilde{T}} t\gamma_t \tag{4a}$$

$$s.to \quad \sum_{t'=a}^{t}\sum_{i\in C_{t'}} x_{i,t'}\bar{\xi}_{i,t'}^k - tF \geq R\beta_t^k$$

$$+ (1-\beta_t^k)\sum_{t'=1}^{t} M_{t'}^k \qquad k \in K, t \in \tilde{T} \tag{4b}$$

$$q'\beta_t \geq p\gamma_t \qquad t \in \tilde{T} \tag{4c}$$

$$e'\gamma = 1 \tag{4d}$$

$$\beta \in \{0,1\}^{|\tilde{T}|\times|K|} \tag{4e}$$

$$\gamma \in \{0,1\}^{|\tilde{T}|} \tag{4f}$$

$$x \in \mathcal{X}.$$

Any feasible solution of the above problem guarantees that the targeted return R can be attained in admissible time and with probability p. Any optimal solution of the above problem indicates the minimum time needed to reach R. Constraint (1b) is a disjunctive constraint and defines a nonconvex feasible area. The problem is further compounded here as the number τ of terms or literals in the disjunction of (1b) is itself a (general integer) decision variable. Besides providing a deterministic equivalent reformulation for the disjunctive chance-constrained problem **W-M**, the decisive contribution from Theorem 1 is to remove altogether the variable τ (and the associated nonconvexities) and to derive an exact linearization of the nonlinear expression involving τ. The proofs for theorems are given in Appendix A.1 in the supplement document.

3.2. Stochastic agile project selection model and deterministic reformulation

We propose in this section the formulation of a chance-constrained problem that is representative of the dynamic and more flexible of agile project selection approach and accounts for the uncertainty in the generated revenue and in the time needed to complete the projects. The dynamic nature of the agile approach and the resulting additional layer of stochasticity translates into a setting where: (i) DMs can abandon projects previously initiated and (ii) both projects' returns and durations are stochastic. The notations presented in Table 2 are used in the formulation of the agile model and are explained in details in the next paragraphs.

The random duration of projects has multiple modeling consequences. First, it implies that the completion date of a project is itself a stochastic variable. We respectively denote by $\zeta \in \mathbb{R}_+^{|I|}$ and $\psi \in \mathbb{R}_+^{|I|}$ the random project duration and completion time vectors, which connect as follows:

$$\psi_i = s_i + \zeta_i - 1 \ , i \in I \ . \tag{5}$$

Their respective realizations in scenario $k \in K$ are referred to as $\bar{\zeta}^k$ and $\bar{\psi}^k, k \in K$. The vector $\omega^k = [\bar{\xi}^k, \bar{\zeta}^k, \bar{\psi}^k] \in \Omega$ concatenates the realizations in scenario k of the three sources of uncertainties ξ, ζ, and ψ. Second, the random project duration implies the construction of *random sets* (Cressie and Laslett, 1986; Nguyen, 2006) since the elements included in the sets $I_t, C_t, t \in T$, and $T_i, i \in I$ defined above are unknown ex-ante and stochastic. This has in turn a major effect on the chance-constrained problem since we will have to use *dynamic chance constraints* (Lulli and Sen, 2004) in the formulation instead of using the traditional static ones. Investment decisions in the waterfall model **W-M** are taken ex-ante, before the realization of the random variables, for the whole planning horizon and are not subsequently modified. It ensues that the decision to invest in project i in t applies in all scenarios. This contrasts with the agile model since, under the corresponding random project duration assumption, investing in a certain project i at t may not be implementable in all scenarios provided that in some of them project i might be completed at an earlier period $t' < t$. In order to accurately model the higher flexibility and the increased stochasticity level induced by the agile mode, we adjust our modeling approach and propose a multistage model with dynamic chance constraints in which decisions are adaptive to the realizations of the random variables.

In this setting, the decisional space is lifted and the decision vector is now given by

$$y = (y_1, y_2(\omega_1), y_3(\omega_2), \ldots, y_{|T|}(\omega_{|T|-1}))$$

in which each component y_t, $1 < t < |T|$, is a function of previously observed values $\omega_1, \ldots, \omega_{t-1}$ of the random process for a given time t. The first-stage decisions y_1 are taken at the start of the first period before observing the realization of the random events taking place in period 1 and then taking the recourse second-stage decisions y_2 on the basis of the observed outcomes ω_1. The same process carries on until reaching the end of the planning horizon. Besides introducing the parameters $V_i = \max_{k\in K} \bar{\zeta}_i^k, i \in I$ equal to the longest possible duration of project i, the dynamic feature of the agile project selection model requires to adjust the definition of the sets $T_i, i \in I$, I_t, and $C_t, t \in T$ whose random counterparts $T_i(\zeta) = \{s_i, \ldots, V_i\} \subseteq T, i \in I$, $I_t(\zeta) = \{i \in I : t \in T_i(\zeta)\} \subseteq I, t \in T$, and $C_t(\zeta) = \{i \in I : V_i = t\} \subseteq I, t \in T$ to represent that the composition of those sets is stochastic and varies across scenarios. Similarly, the notations for the binary decisions variables $y_{i,t}(\zeta)$ reflect that they are scenario-dependent and that the periods at which the DM can invest in project i depends on its random duration ζ_i: $y_{i,t}(\zeta), i \in I, t \in T$ is equal to 1 if the DM supports project i in t and is 0 otherwise.

The disjunctive integer nonlinear and dynamic chance-constrained problem for the agile project selection mode reads:

$$\mathbf{A-M}: \ \min \tau$$

$$\text{s.to } (y,\tau) \in \bigvee_{t\in\{a,\ldots,\tau\}} \mathcal{G}_t \tag{6a}$$

$$y \in \mathcal{Y} \tag{6b}$$

$$\tau \in \tilde{T} \subset \mathbb{Z}_+ \ . \tag{6c}$$

where the deterministic integer linear feasible set $\mathcal{Y}$ is given by

$$\mathcal{Y} := \{y \in \{0,1\}^{h_\zeta} : e' y_t(\zeta) \leq U, t \in T; \ y_{i,t}(\zeta) = y_{i,t+1}(\zeta), i \in I, t \in T_i(\zeta) \setminus \{|T_i(\zeta)|\}\} \tag{7}$$

with $h_\zeta = \sum_{i\in I}|T_i(\zeta)|$ denoting the uncertain number of binary variables that varies across scenarios. The integer linear feasible set $\mathcal{Y}$ is the counterpart of $\mathcal{X}$ in the dynamic setting and is defined in a lifted decision space due to the dynamic nature of the agile mode. The constraints in $\mathcal{Y}$ are stochastic inequalities since they must hold for each component of the random sets $T_i(\zeta)$ whose composition is stochastic. The constraints $y_{i,t}(\zeta) = y_{i,t+1}(\zeta), i \in I, t \in T_i(\zeta) \setminus \{|T_i(\zeta)|\}$ are the non-anticipativity constraints which force the decision variables $y_{i,t}^k$ to take identical values for each scenario sharing the same history

Table 2
Additional notations for agile model.

Parameters	
V_i	Longest possible duration of project i
Uncertainty related notations	
ζ_i	Random project duration of project i
ψ_i	Random completion time of project i
$\bar{\zeta}_i^k$	Realization of project duration for project i in scenario k
$\bar{\psi}_i^k$	Realization of completion time for project i in scenario k
$\omega^k = [\bar{\xi}^k, \bar{\zeta}^k, \bar{\psi}^k]$	Joint realization vector for scenario k
$T_i(\zeta)$	Random set of periods during which project i can be developed
$C_t(\zeta)$	Random set of projects that can be completed at period t
$I_t(\zeta)$	Random set of projects that can be developed at period t
S_t^k	Set of scenarios with identical history with scenario k until period t
Decision variables	
$y_1 \in \mathbb{B}$	First-stage decisions taken before any uncertainty is revealed
$y_{i,t}^k(\omega_{t-1}) \in \mathbb{B}$	Decision to invest in project i at time t in scenario k
$t \in \{2, \ldots, \lvert T \rvert\}$	taken after observing the outcomes $\omega_{t'}$ in previous periods $t' < t$

until $t-1, t = 1, \ldots, |T|$. The notation $\mathcal{G}_t$ refers to the tth literal of the disjunctive normal form $\bigvee_{t \in \{a,\ldots,\tau\}} \mathcal{G}_t$. The literal $\mathcal{G}_t$

$$\mathcal{G}_t := \left\{ y_{it}(\zeta) \in \{0,1\}^{|C_{t'}(\zeta)|} : \mathbb{P}\left(\sum_{t'=a}^{t} \sum_{i \in C_{t'}(\zeta)} y_{i,t'}(\zeta)\xi_{i,t'} - tF \geq R \right) \geq p \right\} \tag{8}$$

is said to be *true* if the dynamic chance constraint $\mathbb{P}\left(\sum_{t'=a}^{t} \sum_{i \in C_{t'}(\zeta)} y_{i,t'}(\zeta)\xi_{i,t'} - tF \geq R\right) \geq p$ at time $t \in \{a, \ldots, \tau\}$ holds. As in the waterfall model, the number of literals $\mathcal{G}_t$ is a decision variable.

We shall now propose in Theorem 2 a deterministic and equivalent reformulation for the dynamic chance-constrained problem **A-M** corresponding to the agile project selection mode. We denote by $y_{i,t'}^k, k \in K, i \in I, t \in T$ the binary decision variable taking value 1 if the DM invests in project i in period t and scenario ω^k ans taking value 0 otherwise. The size of the vectors $y_t^k, k \in K$ fluctuates since the realized duration $\bar{\zeta}_i^k$ of project i can differ across scenarios, which in turn implies that we must define a scenario-specific version of the sets T_i, I_t, and C_t (see Table 2). In particular, $C_t^k = \{i \in I : \bar{\psi}_i^k = t\}, t \in T, k \in K$ refers to the set C_t specific to scenario ω^k. We also introduce the new sets S_t^k that include the scenarios that have identical history with scenario $\bar{\xi}^k$ until period t and that are needed to define the non-anticipativity constraints. To ease the notations, we drop (ζ) in $y_{i,t}(\zeta)$.

Theorem 2. *The disjunctive integer nonlinear dynamic chance-constrained problem* **A-M** *can be equivalently reformulated as the following deterministic integer linear programming problem* **A-RM**:

$$\textbf{A-RM}: \ \min \sum_{t \in \tilde{T}} t\gamma_t \tag{9a}$$

$$s.to \ \sum_{t'=a}^{t} \sum_{i \in C_{t'}^k} y_{i,t'}^k \bar{\xi}_{i,t'}^k - tF \geq R\beta_t^k + (1-\beta_t^k) \sum_{t'=1}^{t} M_{t'}^k \qquad k \in K, t \in \tilde{T} \tag{9b}$$

$$q'\beta_t \geq p\gamma_t \qquad t \in \tilde{T} \tag{9c}$$

$$e'\gamma = 1 \tag{9d}$$

$$\beta \in \{0,1\}^{|\tilde{T}| \times |K|} \tag{9e}$$

$$\gamma \in \{0,1\}^{|\tilde{T}|} \tag{9f}$$

$$y \in \mathcal{Y}. \tag{9g}$$

The dynamic nature of the formulation **A-M** and the resulting non-anticipativity constraints lead to a significant increase in the number of decision variables and constraints, which should further complexify the solution of the reformulation **A-RM**.

4. Solution methods

This section is decomposed into three main subsections. Section 4.1 presents specific valid inequalities and strengthening techniques that tighten the deterministic reformulations **W-RM** and **A-RM**. Section 4.2 proposes a decomposition method to solve the reformulation problems **W-RM** and **A-RM**. Section 4.3 devises a bisection algorithm that determines – among the typically many project selection plans that provides the targeted return level as quickly as possible – the *ideal plan* that generates the *highest* excess return in the minimal time needed to obtain a return at least equal to R.

4.1. Model strengthening

We derive several specific valid inequalities and fixing approaches that strengthen the continuous relaxations of the integer linear problems **W-RM** and **A-RM**. The computational benefits of the tightening techniques proposed in this section will be evaluated numerically in Section 5.2.2. The valid inequalities will tighten the continuous relaxation of the problem solved at each node of the branch-and-bound tree. They will eliminate solutions feasible for the continuous relaxations but infeasible for the true integer problem, provide a tighter lower bound on the optimal value of the problem, reduce the number of nodes in the branch-and-bound tree, which results into faster solution times.

Theorem 3. *The linear inequalities*

$$I \quad \begin{cases} x_{i,t''} \leq 1 - \sum_{t=a}^{t'} \gamma_t \ , \ t' = a, \ldots, |T|-1, i \in I, \\ \quad t'' \in \{T_i : f_i > t'\} \qquad for\ \textbf{W-RM} \qquad (a) \\ y_{i,t''+1}^k \leq 1 - \sum_{t=1}^{t'} \gamma_t \ , \ t' = 1, \ldots, |T|-1, i \in I, k \in K, \\ \quad t'' \in \{T_i^k : \bar{\zeta}_i^k > t'\} \qquad for\ \textbf{A-RM} \qquad (b) \end{cases} \tag{10}$$

$$II \quad \begin{cases} 1 - \frac{\sum_{i \in I_t} x_{i,t}}{U} \geq \beta_{t'}^k \ , \ t = a+1, \ldots, |T|, \\ \quad t' = a, \ldots, t-1, \qquad for\ \textbf{W-RM} \qquad (a) \\ 1 - \frac{\sum_{k \in K} \sum_{i \in I_t^k} y_{i,t}^k}{|K| \ U} \geq \beta_{t'}^{k'} \ , \ t = 1+1, \ldots, |T|, \\ \quad t' = 1, \ldots, t-1, k' \in K \qquad for\ \textbf{A-RM} \qquad (b) \end{cases} \tag{11}$$

$$III \quad \begin{cases} \gamma_t \geq \beta_t^k \ , \ k \in K, \qquad t \in \tilde{T} \ for\ \textbf{W-RM}\ and \\ \quad t \in T \ for\ \textbf{A-RM} \end{cases} \tag{12}$$

are valid.

The next set of inequalities apply to the waterfall approach exclusively.

Theorem 4. *I The linear inequalities*

$$\sum_{t\in T_i} x_{i,t} = d_i \; x_{i,f_i} \ , \ i \in I \tag{13}$$

are valid for **W-RM**.

II If for any arbitrary tuple $(t,k,k') \in \bar{T} \times K \times \{K \setminus \{k\}\}$,

$$\tilde{\xi}^{k}_{i,t'} \leq \tilde{\xi}^{k'}_{i,t'} \ , \ i \in C_{t'}, t' = a,\dots,t \ , \tag{14}$$

then the following inequalities

$$\beta^{k}_{t'} \leq \beta^{k'}_{t'} \ , \ t' = a,\dots,t \tag{15}$$

are valid for **W-RM**.

Theorems 3 and 4 tighten the reformulations **W-RM** and **A-RM** which should make them easier and quicker to solve.

4.2. Decomposition method

The reformulated problems **W-RM** and **A-RM** are large-scale integer problems and pose computational challenges. We use a decomposition method similar to the one proposed by Kettunen and Lejeune (2020). The decomposition method does not deal directly with the large-size integer problems **W-RM** and **A-RM**, but converges finitely to find the provably optimal solution of **W-RM** and **A-RM**. The certificate of optimality is obtained in a simple manner that iteratively checks whether a finite set of integer linear inequalities admits a feasible solution. In what follows, we illustrate the method for the reformulation **A-RM** that is transferable to **W-RM** with minor modifications.

Central to the method and to its finite convergence property is the system of integer linear inequalities presented in Theorem 5. This system of inequalities is used to deliver certificates of feasibility and optimality and is obtained by dropping the binary variables γ_t and adjusting the constraints in the feasible set of **A-RM** in which the variables γ_t appear. Let $T^k_i = \{s_i,\dots,\bar{\zeta}^k_i\}, i \in I, k \in K$ and $I^k_t = \{i \in I : t \in T^k_i\}, t \in T, k \in K$ be the scenario-specific versions of the sets T_i and I_t.

Theorem 5. *Consider* $r \in \{a,\dots,|T|\}$. *The system of integer linear inequalities*

$$\mathbf{FA_r}: \begin{cases} \sum_{t=a}^{r} \sum_{i\in C^k_t} y^k_{i,t}\tilde{\xi}^k_{i,t} - rF \\ \quad \geq R\beta^k_{t'} + (1-\beta^k_t)\sum_{t'=1}^{t} M^k_{t'} & k \in K \quad (a)\\ \sum_{i\in I^k_t} y^k_{i,t} \leq U & t = 1,\dots,r, k\in K \quad (b)\\ y^k_{i,t} \geq y^k_{i,t+1} & k\in K, i \in I^k_t, \\ & t \in T^k_i \bigcap \{1,\dots,r-1\} \quad (c)\\ y^k_{i,t} \in \{0,1\} & k\in K, i\in I^k_t, \\ & t\in T^k_i \bigcap\{1,\dots,r\} \quad (d)\\ y^k_{i,t} = y^{k'}_{i,t} & i\in I^k_t, k\in K, \\ & k' \in S^k_t \setminus \{k\}, \\ & t = s_i,\dots,r \quad (e)\\ \sum_{k\in K} q^k \beta^k_r \geq p & \quad (f)\\ \beta^k_r \in \{0,1\} & k \in K \quad (g) \end{cases} \tag{16}$$

admits a feasible solution if and only if the return level R *in* **A-M** *can be reached by time* r.

As explained next in Corollaries 6 and 7, the system of inequalities (16)(a)–(g) can be conveniently used to check the feasibility of **A-RM**, to bound the optimal value of **A-RM** (Corollary 6), and to deliver a certificate of optimality (Corollary 7).

Corollary 6. *If* $(\bar{y},\bar{\beta}) \in \mathbf{FA}_r$, *then*

- $(\bar{y},\bar{\beta},\bar{\gamma})$ *is feasible for* **A-RM** *and* $(\bar{y}, \tau = r)$ *is feasible for* **A-M**.
- $(\bar{y},\bar{\beta},\bar{\gamma})$ *with* $\bar{\gamma}_t = \begin{cases} 1 & \text{if } t = r \\ 0 & \text{otherwise} \end{cases}$ *is feasible for* **A-RM** *and* $(\bar{y}, \tau = r)$ *is feasible for* **A-M**.

Corollary 7. *Let* $r \in \{a,\dots,|T|-1\}$ *and* τ^* *be the optimal value of* **A-M** *and* **A-RM**.

- *If* $\mathbf{FA}_r = \emptyset$, *then* $\tau^* > r$.
- *If* $\mathbf{FA}_r \neq \emptyset$, $\tau^* \leq r$.
- *If* $\mathbf{FA}_r = \emptyset$ *and* $\mathbf{FA}_{r+1} \neq \emptyset$, *the optimal value of* **A-M** *is* $\tau^* = r + 1$.

The finite convergence of the decomposition method follows.

Corollary 8. *The algorithm converges finitely in at most* $|T| - a + 1$ *iterations.*

We present the pseudo-code of the decomposition method in Appendix A.2 in the supplementary document.

4.3. Algorithmic method for ideal investment plan with maximal excess return

As the objective is to uncover the earliest period at which a specified return level can be achieved, problems **W-M** and **A-M** admit multiple optimal solutions prescribing possibly quite different investment policies. It is therefore of interest to discriminate among those optimal solutions. In this section, we introduce the concept of *ideal* investment plan and design an algorithm that permits to identify it.

Definition 9 (*Ideal Investment Plan*)**.** Let R be any targeted return level and τ^* be the earliest period at which it can be obtained. The ideal solution or investment plan is the optimal plan that maximizes the probabilistic excess return over R obtained in period τ^* with probability p.

The ideal investment plan is the optimal plan (i.e., providing the targeted return R as quickly as possible with probability p) that maximizes the excess return over R.

The proposed method relies on the decomposition approach presented in Section 3 and is supplemented by the incorporation of a bisection algorithm. The method is exact and converges finitely to the optimal solution. A bisection algorithm is an iterative method that splits the search interval on the value of the objective function into two (usually equal) parts at each iteration. Consider a maximization problem with an objective function taking value in the interval $[l^0, u^0]$. A bisection algorithm divides at each iteration v the incumbent interval $[l^v, u^v]$ in two equal parts and calculates the objective function value at the midpoint $R^v = (l^v + u^v)/2$ of the interval. If the problem is feasible for $R = R^v$, we discard the sub-interval $[l^v, R^v)$ and the search goes on $[R^v, u^v]$. If the problem is infeasible for $R = R^v$, we discard the sub-interval $[R^v, u^v]$ and the search is then focused on $[l^v, R^v)$. The search continues until the interval becomes smaller than the required precision level.

We now design a bisection algorithm with the objective of finding the ideal investment plan for any given targeted return level R. We present a detailed description of the bisection algorithm for model **A-M**. The algorithm can be used in the same way for model **W-M**.

The first step is to determine the (smallest possible) interval $[l^0, u^0]$ of the objective function on which the search is carried out. While it is obvious that the lower bound l^0 is R, we need to find out a valid value for the upper bound. To do so, we solve the expected value problem $\mathbf{EAM}_{\tau^*}$

$$\mathbf{EAM}_{\tau^*} : \max \sum_{i\in C^k_{\tau^*}} \sum_{k\in K} q^k y^k_{i,\tau^*} \tilde{\xi}^k_{i,\tau^*} - \tau^* F \tag{17a}$$

$$\text{s. to} \sum_{i \in I_t^k} y_{i,t}^k \le U \qquad t = 1, \dots, \tau^*, k \in K \tag{17b}$$

$$y_{i,t}^k \ge y_{i,t+1}^k \qquad k \in K, i \in I_t^k, t \in T_i^k \bigcap \{1, \dots, \tau^* - 1\} \tag{17c}$$

$$y_{i,t}^k = y_{i,t}^{k'} \qquad i \in I_t^k, k \in K, k' \in S_t^k \setminus \{k\}, t = s_i, \dots, r^* \tag{17d}$$

$$y_{i,t}^k \in \{0,1\} \qquad k \in K, i \in I_t, t \in T_i^k \bigcap \{1, \dots, \tau^*\} . \tag{17e}$$

that maximizes the expected return level R_E^* that can be obtained by period τ^*. Clearly, the optimal value of $\mathbf{EAM}_{\tau^*}$ is at least equal to the optimal value of the problem maximizing the return level that can be obtained with a high probability level p by τ^*. Hence, R_E^* is a valid upper bound and we set $u^0 = R_E^*$.

The algorithm proceeds as follows. For any given R, we determine τ^* using our decomposition method, calculate R_E^* by solving $\mathbf{EAM}_{\tau^*}$, and construct the search interval $[l^0, u^0]$ by setting $l^0 = R$ and $u^0 = R_E^*$. At each iteration v, we check whether the set of inequalities

$$\mathbf{FW}_{\tau^*}^{v} : \begin{cases} \text{(17b)–(17e)} & \\ \sum_{i \in C_{\tau^*}^k} y_{i,\tau^*}^k \xi_{i,\tau^*}^k - \tau^* F \ge R^v \beta_{\tau^*}^k & \\ \quad +(1-\beta_{\tau^*}^k) \sum_{t=1}^{\tau^*} M_t^k & k \in K \quad \text{(a)} \\ \sum_{k \in K} q^k \beta_{\tau}^k \ge p & \text{(b)} \\ \beta_{\tau^*}^k \in \{0,1\} & k \in K \quad \text{(c)} . \end{cases} \tag{18}$$

admits a feasible solution by solving a feasibility optimization problem (see Bauschke and Borwein (1996)) in which R is fixed to the midpoint R^v of the incumbent interval $[l^v, u^v]$. The only objective is to check whether $\mathbf{FW}_{\tau^*}^{v}$ admits a feasible solution. The tolerance level for the stopping criterion is ϵ. The iterative procedure stops if $u^t - l^t \le \epsilon$.

Note that the size of the interval $[l^t, u^t]$ shrinks at each iteration by a factor of $(u^t - l^t)/2$, and the algorithm therefore converges to the optimal value as formalized in Proposition 10.

Proposition 10. $R^* \in [l^0, u^0]$. *The standard bisection algorithm finds the optimal solution R^* in finitely many (i.e., at most $\log_2((u^0 - l^0)/\epsilon)$) iterations with precision level ϵ.*

We give the pseudo-code of the algorithm in Appendix A.2 of the supplement document.

Instead of using the developed bisection algorithm we could have formulated an integer problem to define the optimal investment plan. Preliminary tests however have shown that solving a few feasibility problems as requested by the proposed bisection algorithm is quicker than solving to optimality such an (much larger) integer programming problem.

5. Modeling and managerial insights

5.1. Experimental setup

We conduct experiments using illustrative data that is obtained from a US-based software start-up company. The company has three software development teams, which each work on one project at a time. The development decisions, concerning which projects are worked on, are made every three month. The development costs for each three month period per project amount to \$0.3 million and consist mainly of the salaries of the software developers.

Expert elicitation is used to assess the required project parameters. Table 3 lists the seven initially proposed projects with their (i) time to complete the development, (ii) estimated expected revenues, and (iii) revenue standard deviations. The development team expects to receive four proposals for new projects at each period. The durations of the new projects depend on whether the waterfall or agile project management approach is employed. Under the waterfall approach, two of the projects are expected to take five periods to develop, one is expected to take three periods, and one is expected to take two periods. Under the agile approach, the duration of the development projects is stochastic with development lasting $[5, 4, 3]$ periods with probabilities $[0.5, 0.25, 0.25]$.

The revenues of projects proposed in the future periods account for two specific characteristics that are present in the analyzed setting. In particular, the longer lasting projects are expected to (i) yield higher revenues, such that the expected revenues in \$ millions are λd_i with $\lambda = 0.65$ as assessed by the DMs and d_i referring to the number of development periods of the project $i \in I$, and (ii) embed higher risk, which is captured through the standard deviation of revenues denoted αd_i, $i \in I$ with $\alpha = 0.4$ as assessed by the DMs. The DMs estimate projects' revenues to be independent and normally distributed. The time available to reach a specified probabilistic return is 5 years and the planning horizon is hence split into 20 periods during which 87 projects are available to invest $(7 + 4 \times 20)$. We represent the uncertainties in the available projects by simulating scenarios. Under the waterfall project management approach, each scenario is a 87-dimensional vector representing the randomness of the projects' revenues. Under the agile project management approach each scenario is a 174-dimensional (2×87) vector because both projects' revenues and their duration are defined as random variables. The data and codes for the developed algorithms and optimization models are publicly available at an online repository (Kettunen and Lejeune, 2021).

5.2. Modeling insights

5.2.1. Generation of reliable return

To validate the importance of explicitly modeling the reliability constraints, we first assess the reliability of the project selection approach where the constraint is set on the expected return instead of the probabilistic return. This corresponds to the *risk-neutral project selection* model. The variant of model **W-M** imposing a lower bound on the expected return is obtained when (1b) is replaced by

$$\bigvee_{\tau \in T} \left\{ \mathbb{E}\left[\sum_{t'=a}^{\tau} \sum_{i \in C_{t'}} x_{i,t'} \xi_{i,t'} - \tau F \right] \ge R \right\} .$$

Similarly, the variant of model **A-M** imposing a minimal expected return threshold is obtained when (6a) is replaced by

$$\bigvee_{\tau \in T} \left\{ \mathbb{E}\left[\sum_{t'=1}^{\tau} \sum_{i \in I_{t'}(\zeta)} y_{i,t'}(\zeta) \left(\xi_{i,t'} - \varsigma_{i,t'} \right) \right] \ge R \right\} .$$

We conduct experiments over five distinct simulated instances in which 2000 scenarios are accounted for using the risk-neutral model variants of models **W-M** and **A-M** that require a minimum expected return level (i.e., $R = 1, 7$, and 13) and derive the optimal project selection strategies. The optimal project selection strategies obtained with the risk-neutral variants of models **W-M** and **A-M** reached the target return levels on average in 55% and 59% of the scenarios, respectively. Apparently, the project selection strategies derived using the risk-neutral project selection models fail to reach the target return with high reliability. By accounting for the reliability requirements, as in the original *risk-averse project selection* models **W-M** and **A-M**, the reliability of the attainable return can be guaranteed with a high (i.e., 95%) reliability level. Therefore, capturing the DMs' reliability requirements necessitates the use of chance constraints.

Table 3
Initially proposed projects.

Project	Development time (quarter of a year periods)	Expected revenues ($M)	Standard deviation (% of revenues)
Data analytics tools 1	5	3.71	30.5%
Customizable reports	5	3.34	27.5%
Compatibility with other software	5	3.17	27.4%
Data analytics tools 2	4	2.91	25.8%
Enhanced administrative tools	4	2.62	23.3%
Enhanced user interface	3	1.95	23.6%
Social media components	3	1.79	21.8%

Table 4
Average computational times.

			Time (s)					Time (s)	
p	$\|K\|$	R	**W-RM**	**A-RM**	p	$\|K\|$	R	**W-RM**	**A-RM**
0.9	500	1	0.0	0.6	0.95	500	1	0.0	0.7
0.9	500	7	1.9	7.7	0.95	500	7	158.2	6.4
0.9	500	13	698.2	123.6	0.95	500	13	1381.0	74.4
0.9	1000	1	0.0	2.0	0.95	1000	1	0.0	2.1
0.9	1000	7	0.4	39.4	0.95	1000	7	141.3	59.3
0.9	1000	13	823.2	384.2	0.95	1000	13	1369.4	302.0
0.9	2000	1	0.1	9.0	0.95	2000	1	0.1	8.0
0.9	2000	7	802.4	245.6	0.95	2000	7	796.4	165.6
0.9	2000	13	1525.4	1103.9	0.95	2000	13	1423.4	1087.3
		Average	427.9	212.9			Average	585.5	189.5

5.2.2. *Computational efficiency*

We first attempt to solve the original formulation of **W-M** and **A-M**. The models are coded with AMPL (Fourer et al., 2003) and solved with the Gurobi 9.1.0 solver on a 64-bit computer with Intel Core i7-6700 CPU 3.40GHZ. We note that none of the 180 problem instances ($p = \{0.9, 0.95\}, |K| = \{500, 1000, 2000\}, R = \{1, 7, 13\}$; each run with 5 different scenario sets) can be solved in three hours of computing time regardless of whether or not we use the valid inequalities presented in Section 4.1, and most resulted in running out of available memory. The reformulated versions of the models, utilizing the decomposition method and incorporating the valid inequalities in Section 4.1, are solved on average in 354 s. In Table 4, we report for each tuple $(p, |K|, R)$ the average (i.e., taken across the five scenarios sets considered for each of the 18 types of problem instances) total computational times (i.e., including time for the decomposition method) needed to solve **W-RM** and **A-RM** with the proposed decomposition algorithm when the valid inequalities are utilized. These results highlight the computational efficiency of the proposed method. The results in Table 4 show that model **A-RM**, which has more decision variables and constraints than model **W-RM**, is faster to solve than model **W-RM** for the instances with high probabilistic return requirement R. For example, **A-RM** is faster to solve for complex problems with larger required return level (i.e., $R = 13$), whereas **A-RM** is slower to solve than model **W-RM** for problems in which the required return level is low (i.e., $R = 1$) and can be achieved quickly. When $R = 7$, then model **A-RM** is slower than model **W-RM** in two of the five cases, i.e., when the reliability level is low, $p = 0.9$ and there are less scenarios, $|K| = 500, 1000$. These correspond to problems that are smaller (less constraints and decision variables).

Surprisingly, Table 4 shows that the average computational time is about invariant with the reliability level p with model **A-RM** and is an increasing function of p with **W-RM**. This is in stark contrast with the results reported in most studies involving scenario-based reformulations of chance constraints. The average computational time increases in both models when the targeted return level R increases, which follows from the fact that a higher R value makes the associated constraints more difficult to satisfy and hence finding a feasible solution more time-consuming.

5.2.3. *Convergence and stability*

We next evaluate the number of scenarios needed to observe convergence and stability in the optimal solution, i.e., the earliest time at which the targeted return can be achieved. To conduct this analysis, we employ two methods to generate and select scenarios and run numerical experiments with each in which the number of selected scenarios varies, ranging from 50 to 2000. The results with the scenario selection methods are shown in Table 5 for both models **W-RM** and **A-RM**.

The first scenario generation approach is a basic Monte Carlo simulation method. The second method employs a scenario reduction algorithm (Dupačová et al., 2003; Heitsch and Römisch, 2007) that selects a representative subset (of given cardinality) of scenarios from a larger scenario set. The construction of the subset of representative scenarios is carried out in such a way that the distance between this scenario subset and the original scenario set is minimal and the distance is expressed in terms of a probability distance metric. We generate initially 5000 scenarios via a Monte Carlo simulation (original scenario set) and construct subsets of scenarios of cardinality 50, 500, 1000, and 2000. To obtain the reduced set of scenarios, we utilize the GAMS software package "scenred2" (GAMS Development Corporation, 2018). This package implements a backward reduction algorithm proposed by Dupačová et al. (2003). The backward algorithm works such that it removes scenarios iteratively (i.e., one at-a-time) by solving a scenario deletion optimization problem, until the desired number of scenarios is reached. The applied algorithm relies on the Wasserstein distance metric of degree 1 to measure the distance between the reduced scenario set and the original one. The intuition behind the Wasserstein distance measure is that it minimizes the transportation cost of the probability mass distribution defined by the original scenario set to the probability distribution defined by the reduced set of scenarios.

Results in Table 5 show the convergence to a stable optimal solution when 500 scenarios are considered. The conclusion applies to both

Table 5
Stability results for $R = 7$, $p = 0.95$.

	Time period when probabilistic return level is reached			
	Monte Carlo simulation of scenarios		Scenarios selected using scenario reduction algorithm	
$\|K\|$	**W-RM**	**A-RM**	**W-RM**	**A-RM**
50	9	6	9	7
500	10	6	10	6
1000	10	6	10	6
2000	10	6	10	6

models and scenario generation methods. In particular, it can be seen that, when 500 scenarios are considered, the earliest period τ^* at which the targeted return level $R = 7$ is reached (i.e., optimal value) with model **W-RM** (resp., **A-RM**) is 10 (resp., 6) periods. The optimal value $\tau^* = 10$ (resp., $\tau^* = 6$) with model **W-RM** (resp., **A-RM**) remains unchanged when the number of scenarios increases and goes to 1000 and 2000. Even when only 50 scenarios are considered, the obtained value τ^* is at most one period away from the stable optimal value (i.e., with 500 or more scenarios). We have confirmed these stability results using four additional sets of scenarios for which similar observations prevail, i.e., results remain stable when using only 50 or 500 scenarios and the earliest period τ^* at which the targeted return level $R = 7$ is reached remains at most one period away from the stable optimal value when the scenarios are reduced from 2000 to 50. Since the models can be solved for a number of scenarios four times larger than the reduced set of 500 scenarios (see Table 4), the models are scalable to deal with more projects and additional uncertainties and the results will still converge.

To confirm the robustness of the derived investment strategies, we carried out an out-of-sample analysis. Considering a training set comprising 2000 scenarios, we have solved the two models **W-RM** and **A-RM** to derive investment strategies that we refer to as the training set optimal investment strategies. Next, we have applied the **W-RM** and **A-RM** training set optimal investment strategies to four 2000-scenario testing tests (i.e., sets of scenarios that were not considered in the training set optimal investment strategies). For each testing set and with each model **W-RM** and **A-RM**, it turns out that the optimal training set optimal investment strategies permits to reach the required return level.

5.3. Managerial insights

5.3.1. Comparison of waterfall and agile development models

Based on the results shown in Table 6 and confirmed across all the 180 studied problem instances, we make the following observation.

Observation 1. *The probabilistic return is reached faster (or equally fast) under the agile development approach than under the waterfall development approach.*

This outcome is due to the capability to dynamically adjust the project portfolio under the agile development approach. In particular, the company employing the agile approach can pivot away from projects, which development has not went well and start developing new projects. This approach limits the risks, as compared to the waterfall approach, where each committed project is completed even if it ends up being a failed or nearly failed project, in terms of providing little revenue. Observation 1 and the intuition behind it help understand why the agile development approach has become popular among NPD projects (Blank, 2013).

5.3.2. Risk-seeking project selection and assessment of NPD company's performance

The developed project selection framework is flexible to represent a *risk-seeking project selection*, where the DM requires to reach the targeted return with a low probability, such as $p = 0.05$. In the stochastic programming and distributionally robust optimization literature, (ambiguous) chance constraints in which the required reliability level is low are referred to as optimistic chance constraints (Hanasusanto et al., 2017). They require the set of stochastic inequalities to hold in the best 5% of the possible scenarios. Models with optimistic chance constraints are enticing for a risk-seeking behavior. They can represent the behavior of a venture capitalist with investments on many start-ups, requiring that the companies pursue projects that can provide a very high return even if with only a low probability.

Fig. 1 illustrates the return trajectories at the $p = 0.05$ and $p = 0.95$ reliability levels. The figure is beneficial in illustrating three key results. First, it shows that the same return can be achieved earlier under the risk-seeking project selection approach with a 5% reliability level than under the risk-averse project selection approach with a 95% reliability level. This is as expected because the risk-seeking probabilistic return focuses on the best 5% of the returns. These results hold across the randomly generated scenario data sets.

Second, and more importantly, the probabilistic return trajectories at $p = 0.05$ and $p = 0.95$, shown in Fig. 1, can be useful for the investors to assist their decision on how to proceed with their investment on the company. Particularly, the actual returns of the company over time can be compared to these return trajectories to assess whether the company is likely to become a high-performing company with return above the $p = 0.05$ trajectory, an average performing company with return between the $p = 0.05$ and $p = 0.95$ trajectories, or a low-performing company with return below the $p = 0.95$ trajectory. Consequently, depending on whether the company is high-, average-, or low-performing company, the investor can choose to intervene on the company's development operations, provide additional support, or wait longer. We summarize these guidelines in Observation 2.

Observation 2. *The probabilistic return trajectories can be used to guide a venture capitalist as follows (i) intervene on the operations or sell the company, if the return is below the $p = 0.95$ trajectory, (ii) support the company's operations, if the return is between $p = 0.95$ and $p = 0.05$ trajectories, and (iii) retain the company and let it operate independently, if the return is above the $p = 0.05$ trajectory.*

The derived thresholds are company specific and as such they account for the type of projects the company is expected to have and whether the waterfall or the agile development approach is used. Therefore, comparing the company's actual performance over time against the derived trajectories allows to assess how well the company is operated. For example, if the company employs the waterfall development approach and its return at period 10 is $5M, which is below $p = 0.95$ trajectory, this puts the company's performance within the worst 5% of the initially estimated scenarios. Based on this information and using Observation 2 guidelines, the venture capital can request operational changes within the company to improve the performance or let the company go by selling the ownership.

Third, Fig. 1 shows that under the agile development approach the gap between $p = 0.95$ and $p = 0.05$ trajectories is narrower than that under the waterfall development approach. For example, the gap between $p = 0.95$ and $p = 0.05$ trajectories in reaching $7M is under the agile development approach 3 periods $(8 - 5)$ and under the waterfall development approach 5 periods $(10-5)$. This leads us to Observation 3.

Observation 3. *The agile, as compared to the waterfall, development approach results in less uncertainty in time to reach a certain return.*

Table 6
Comparison of waterfall (**W-RM**) and agile (**A-RM**) development models.

Probabilistic return at p=0.95 ($M)	Time period when probabilistic return reached	
	W-RM	**A-RM**
-1	3	3
7	10	8
13	16	11

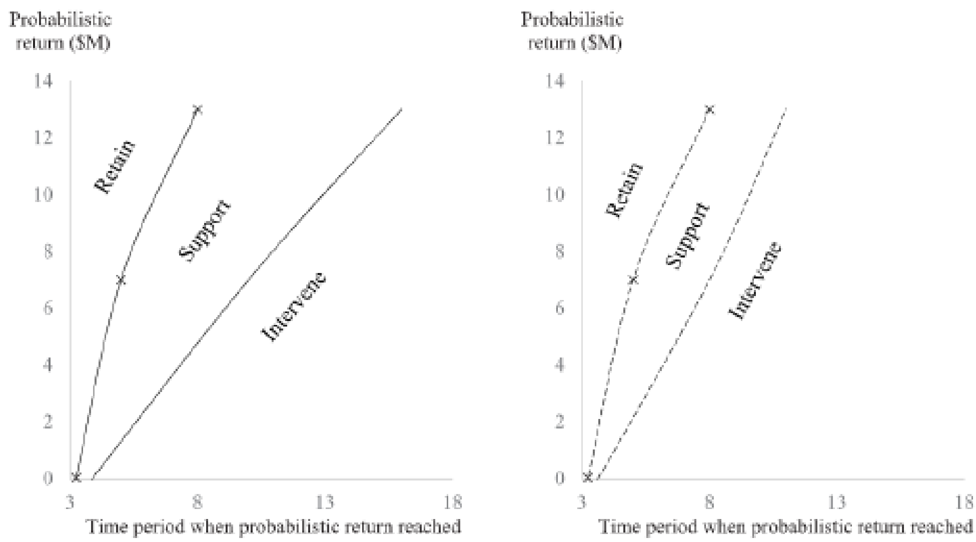

Fig. 1. Time to reach return under waterfall **W-RM** (left) and agile **A-RM** (right) development approaches for $p = 0.95$ (line without markers) and for $p = 0.05$ (line with markers).

This phenomenon is counterintuitive because the agile development approach involves more stochasticity given that projects' durations are random, which is not the case under the waterfall development approach. The reason for the phenomenon is the capability, under the agile approach, to dynamically abandon projects that seem to fail, and thereafter, start new projects. Thereby, the dynamic decision making protects from failed outcomes and allows the company to reach higher probabilistic return levels faster. Consequently, the gap between $p =$ 0.95 and $p = 0.05$ trajectories is narrower under the agile, as compared to the waterfall, approach implying less uncertainty regarding the time when a certain return is reached. This is a further benefit of the agile development approach and can make a difference whether a company obtains funding from a venture capitalist who prefers a more certain time horizon for the investment.

5.3.3. *Maximal excess probabilistic return*

We show in this section that the benefit of taking decisions according to the ideal investment plan (see Section 4.3) can be significant for start-up companies. Fig. 2 displays the largest excess return (dotted line) that can be obtained with the ideal investment plan identified via the proposed bisection algorithm. The results show for example that when the goal is to reach a targeted return of $7 million with a 90% reliability level as fast as possible, we could obtain, within the same time frame and an identical probability level, a return level of $7.9 million, which corresponds to a 13% excess return with respect to the targeted $7 million.

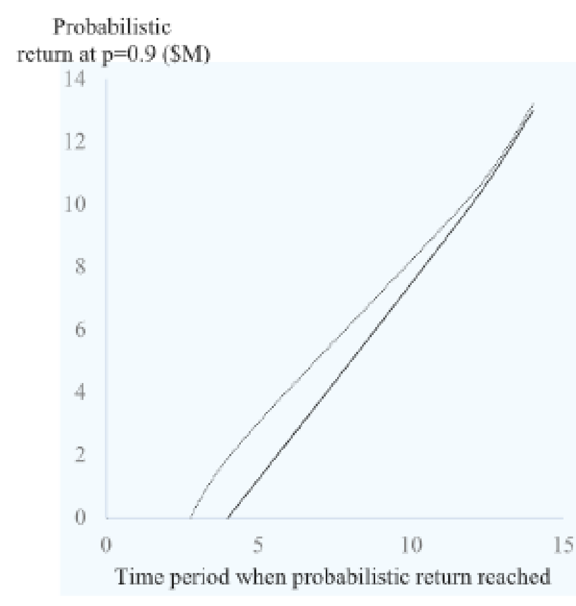

Fig. 2. Time to reach optimal return (solid line) and ideal return (dotted line) with the waterfall development model **W-RM**.

Additionally, the results for the waterfall development model **W-RM** show that when the reliability level increases from $p = 0.9$ to $p = 1$, whilst keeping the ideal return at $4.4 million, the average duration of completed projects increases from 3.5 to 4.3 periods (i.e., a change from 10.5 months to 12.9 months). Consequently, increasing the reliability level may promote investing in longer lasting projects that have in our dataset higher risk and higher expected return. The reason for this result follows from the fact that is takes 11 periods to reach the reliability target when $p = 1$ as compared to only 7 time periods to reach the same reliability goal when $p = 0.9$. These additional four development periods when $p = 1$ allow investments in longer lasting projects, which, regardless of their risks, contribute to reaching the reliability target earlier than if investments were made on projects with shorter development time. Similar results can be observed with the agile development model (**A-RM**).

6. Conclusion

We develop optimization models for project portfolio selection under two different types of project management approaches, namely, the plan-driven *waterfall* approach and the more flexible *agile* approach. The models account for the (i) speed to get the products on market and earn return fast and (ii) requirement to attain a certain return with some probability level. Methodologically, our key contributions include the introduction of the concept of an ideal investment plan and the design of a bisection algorithm to determine the plan that provides the targeted return as quickly as possible whilst maximizing the excess return. We show that the algorithm enjoys finite convergence and that the ideal plan can provide a significantly higher return than an approach that only focuses on the time when a certain probabilistic return is reached (see. e.g., Kettunen and Lejeune (2020)). We also develop model strengthening techniques and valid inequalities that speeds up the solution of the static (waterfall) and dynamic (agile) disjunctive integer nonconvex chance-constrained problems. Our computational results show the efficiency of the proposed modeling and algorithmic developments and the robustness of the models and their results with respect to the number of scenarios used to represent uncertainty.

Employing the developed models, we provide three key managerial insights in the project management domain. First, our results can partially explain why the agile product development approaches are popular in NPD (Hass, 2007; Blank, 2013). In particular, this is due to reaching probabilistic return faster under the agile, as compared to the waterfall, approach. The reason for this, in turn, is the dynamic capability to abandon projects which development has not succeeded and initiate new projects under the agile approach.

Second, we propose a visualization tool to guide a venture capitalist's investment on a company over time. The visualization tool uses the developed models to derive three regions characterizing the company's performance. These regions, along with the associated guidelines, are as follows: (i) A low-performing region where the venture capitalist should either intervene on the company's operations (e.g., by making structural changes or via strongly influencing decision making) or sell the ownership of the company. (ii) An average-performing region where the venture capitalist should support the company's operations, e.g., by providing advice, additional funding, or facilitating creation of social capital. (iii) A high-performing region where the venture capitalist should retain the company and let it operate independently. These three regions are derived in such a way that they are company-specific, and account for the chosen project management approach (i.e., waterfall or agile) and the risk-return profile of the projects the company is expected to consider.

Third, our results show that there is less uncertainty to achieve a target return level under the agile approach than under the waterfall approach. This is a counterintuitive result since the agile development approach includes also stochasticity in the projects' duration, which is not the case in the waterfall approach. The reason for this result is the dynamic abandoning and re-starting of new projects, under the agile approach, which protects from downside risks, and thereby, from worse outcomes that would delay reaching the desired return level. This is an additional benefit of the agile approach and can make a difference whether a company obtains funding from a venture capitalist who has a limited time window for the investment.

The developed models can incorporate several extensions. For example, the models can be adjusted to account for the case when different versions of the same development project are available representing different resource loadings or technological approaches with associated expected return and risk levels. In such situations, set packing constraints can be added to make sure that at most one of the different project versions can be chosen for development. These type of mutual exclusivity constraints on a subset of projects can be smoothly included in the current formulation. Finally, whereas in the current formulation the projects have homogeneous resource requirements, this does not have to be the case. The inclusion of heterogeneous resource requirements can be easily dealt with by adjusting the budget knapsack constraint.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cor.2022.105737.

References

Ahmed, S., Xie, W., 2018. Relaxations and approximations of chance constraints under finite distributions. Math. Program. 170 (1), 43–65.
Andrieu, L., Henrion, R., Römisch, W.G., 2010. A model for dynamic chance constraints in hydro power reservoir management. European J. Oper. Res. 2 (207), 579–589.
Apap, R.M., Grossmann, I.E., 2017. Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. Comput. Chem. Eng. 103, 233–274.
Balas, E., 1979. Disjunctive programming. Ann. Discrete Math. (5), 3–51.
Balas, E., 2018. Disjunctive Programming. Springer.
Bauschke, H.H., Borwein, J.M., 1996. On projection algorithms for solving convex feasibility problems. SIAM Rev. 38 (3), 367–426.
Birge, J., Louveaux, F., 2011. Introduction to Stochastic Programming, second ed. In: Springer Series in Operations Research and Financial Engineering, Springer.
Blank, S., 2013. Why the lean start-up changes everything. Harv. Bus. Rev. 91 (5), 64–68.
Bodily, S.E., 2016. Reducing risk and improving incentives in funding entrepreneurs. Decis. Anal. 13 (2), 101–116.
Canessa, G., Gallego, J.A., Ntaimo, L., Pagnoncelli, B.K., 2019. An algorithm for binary chance-constrained problems using IIS. Comput. Optim. Appl. (72), 589–608.
Chien, C.-F., 2002. A portfolio-evaluation framework for selecting R&D projects. R & D Manage. 32 (4), 359–368.
Colvin, M., Maravelias, C.T., 2010. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. European J. Oper. Res. 203 (1), 205–215.
Cooper, R.G., Edgett, S.J., 2012. Best practices in the idea-to-launch process and its governance. Res. Technol. Manage. 55 (2), 43–54.
Cooper, R.G., Edgett, S.J., Kleinschmidt, E.J., 2000. New problems, new solutions: Making portfolio management more effective. Res. Technol. Manage. 43 (2), 18–33.
Crama, Y., Hammer, P.L., 2011. Boolean Functions: Theory, Algorithms, and Applications. Cambridge Press.
Cressie, N., Laslett, G.M., 1986. Random set theory and problems of modeling. SIAM Rev. 29 (4), 557–574.
Dupačová, J., Gröwe-Kuska, N., Römisch, W., 2003. Scenario reduction in stochastic programming an approach using probability metrics. Math. Program. 95 (3), 493–511.
Edgett, S., 2010. Latest Research: New Product Success, Failure and Kill Rates. Stage-Gate International, Available at www.stage-gate.com/resources_stage-gate_latestresearch.php.
Federal Aviation Administration, 2012. FY 2015 research and development (R&D) portfolio development process. Guidance reference document. Available from www.faa.gov.
Fourer, R., Gay, D.M., Kernighan, B.W., 2003. AMPL: A Modeling Language for Mathematical Programming, second ed. Duxbury-Thompson.
GAMS Development Corporation, 2018. Scenred2. Available at https://www.gams.com/24.8/docs/tools/scenred2/index.html.
Goel, V., Grossmann, I.E., 2006. A class of stochastic programs with decision dependent uncertainty. Math. Program. 108, 355–394.
Hall, N.G., 2016. Research and teaching opportunities in project management. Tutor. Oper. Res. 329–388.
Hall, N.G., Long, D.Z., Qi, J., Sim, M., 2015. Managing underperformance risk in project portfolio selection. Oper. Res. 63 (3), 660–675.
Hanasusanto, G.A., Roitch, V., Kuhn, D., Wiesemann, W., 2017. Ambiguous joint chance constraints under mean and dispersion information. Oper. Res. 3 (65), 751–767.
Hass, K.B., 2007. The blending of traditional and agile project management. PM World Today 9 (5), 1–8.
Hassanzadeh, F., Nemati, H., Sun, M., 2014. Robust optimization for interactive multiobjective programming with imprecise information applied to R&D project portfolio selection. European J. Oper. Res. 238 (1), 41–53.
Heidenberger, K., Stummer, C., 2003. Research and development project selection and resource allocation: a review of quantitative modelling approaches. Int. J. Manage. Rev. 1 (2), 197–224.
Heitsch, H., Römisch, W., 2007. A note on scenario reduction for two-stage stochastic programs. Oper. Res. Lett. 35 (6), 731–738.

Hellemo, L., Barton, P.I., Tomasgard, A., 2018. Decision-dependent probabilities in stochastic programs with recourse. Comput. Manag. Sci. 15 (4), 369–395.
Ibrahim, D.M., 2008. The (not so) puzzling behavior of angel investors. Vanderbilt Law Rev. 61 (5), 1405–1452.
Industrial Research Institute, 2016. 2016 Global R&D funding forecast. A supplement to R&D magazine. available from www.iriweb.org.
Jonsbraten, T.W., Wets, R.J., Woodruff, D.L., 1998. A class of stochastic programs with decision dependent random elements. Ann. Oper. Res. 82, 83–106.
Kamuriwo, D.S., Baden-Fuller, C., 2013. Sparrow therapeutics exit strategy. Entrepreneurship Theory Pract. 38 (3), 691–712.
Kavadias, S., Chao, R.O., 2008. Resource allocation and new product development portfolio management. In: Loch, C., Kavadias, S. (Eds.), Handbook of New Product Development Management. Elsevier, Oxford, pp. 135–163.
Kettunen, J., Lejeune, M.A., 2020. Technical note – Waterfall and agile product development approaches: Disjunctive stochastic programming formulations. Oper. Res. 68 (5), 1356–1363.
Kettunen, J., Lejeune, M.A., 2021. Data and code for the developed algorithms and optimization methods. Available at https://github.com/mlejeune15206/Computers-OR---Disjunctive-Stochastic-Programming.
Kleywegt, A.J., Papastavrou, J.D., 1998. The dynamic and stochastic knapsack problem. Oper. Res. 46 (1), 17–35.
Kogan, A., Lejeune, M.A., 2014. Threshold boolean form for joint probabilistic constraints with random technology matrix. Math. Program. 147 (1), 391–427.
Larson, E.W., Gray, C.F., 2014. Project Management: The Managerial Process, sixth ed. McGraw-Hill/Irwin.
Lejeune, M.A., 2012. Pattern-based modeling and solution of probabilistically constrained optimization problems. Oper. Res. 60 (6), 1356–1372.
Lejeune, M.A., Kettunen, J., 2017. Managing reliability and stability risks in forest harvesting. Manuf. Serv. Oper. Manage. 4 (19), 620–638.
Lejeune, M.A., Margot, F., 2016. Solving chance constrained problems with random technology matrix and stochastic quadratic inequalities. Oper. Res. 64 (4), 939–957.
Lejeune, M.A., Prékopa, A., 2021. Relaxations for probabilistically constrained stochastic programming problems: Review and extensions. Ann. Oper. Res. http://dx.doi.org/10.1007/s10479-018-2934-8.
Linton, J.D., Walsh, S.T., Morabito, J., 2002. Analysis, ranking and selection of R&D projects in a portfolio. R&D Manage. 32 (2), 139–148.
Lu, L.L., Chiu, S.Y., Cox, Jr., L.A., 1999. Optimal project selection: Stochastic knapsack with finite time horizon. J. Oper. Res. Soc. 50 (6), 645–650.
Lulli, G., Sen, S., 2004. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. Manage. Sci. 6 (50), 786–796.
Nguyen, H.T., 2006. An Introduction to Random Sets. Chapman and Hall/CRC Press.
OECD, 2017. Business enterprise R-D by industry and by type of cost. Available at www.stats.oecd.org/.
Ono, M., Pavone, M., Kuwata, Y., Balaram, J.B., 2015. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. Auton. Robots (39), 555–571.
Prékopa, A., 1990. Dual method for a one-stage stochastic programming with random rhs obeying a discrete probability distribution. Z. Oper. Res. 34, 441–461.
Prékopa, A., 2003. Probabilistic programming models. In: Ruszczyński, A., Shapiro, A. (Eds.), Stochastic Programming: Handbook in Operations Research and Management Science, Vol. 10. Elsevier, pp. 267–351.
Ruszczyński, A., 2002. Probabilistic programming with discrete distribution and precedence constrained knapsack polyhedra. Math. Program. 93 (2), 195–215.
Sommer, S., Loch, C., Pich, M., 2007. Project risk management in new product development. In: Loch, C., Kavadias, S. (Eds.), Handbook of New Product Development Management. Elsevier, pp. 439–465.
Sommerville, I., 2016. Software Engineering. Pearson.
Tanner, M.W., Ntaimo, L., 2010. IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. European J. Oper. Res. 207 (1), 290–296.
Tarhan, B., Grossmann, I.E., Goel, V., 2009. Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision-dependent uncertainty. Ind. Eng. Chem. Res. 48 (6), 3078–3097.
Yoo, O.S., Huang, T., Arifoglu, K., 2017. A theoretical analysis of the lean start-up's agile product development process. Available at SSRN: https://ssrn.com/abstract=3070613 or http://dx.doi.org/10.2139/ssrn.3070613.
Zhang, M., Küçükyavuz, S., Goel, S., 2014. A branch-and-cut method for dynamic decision making under joint chance constraints. Manage. Sci. 5 (60), 1317–1333.
Zider, B., 1998. How venture capital works. Harv. Bus. Rev. 76 (6), 131–139.