



# CIRRELT

Centre interuniversitaire de recherche  
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre  
on Enterprise Networks, Logistics and Transportation

---

## A New Scenario-Tree Generation Approach for Multistage Stochastic Programming Problems Based on a Demerit Criterion

Julien Keutchayan  
David Munger  
Michel Gendreau  
Fabian Bastin

December 2017

CIRRELT-2017-74

Bureaux de Montréal :  
Université de Montréal  
Pavillon André-Aisenstadt  
C.P. 6128, succursale Centre-ville  
Montréal (Québec)  
Canada H3C 3J7  
Téléphone : 514 343-7575  
Télécopie : 514 343-7121

Bureaux de Québec :  
Université Laval  
Pavillon Palasis-Prince  
2325, de la Terrasse, bureau 2642  
Québec (Québec)  
Canada G1V 0A6  
Téléphone : 418 656-2073  
Télécopie : 418 656-2624

[www.cirrelt.ca](http://www.cirrelt.ca)

# A New Scenario-Tree Generation Approach for Multistage Stochastic Programming Problems Based on a Demerit Criterion

Julien Keutchayan<sup>1,2,\*</sup>, David Munger<sup>3,†</sup>, Michel Gendreau<sup>1,2</sup>, Fabian Bastin<sup>1,4</sup>

<sup>1</sup> Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

<sup>2</sup> Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

<sup>3</sup> Kronos, AD OPT Division, 3535 chemin Queen-Mary, Montréal, Canada, H3V 1H8

<sup>4</sup> Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

**Abstract.** An important step in solving a stochastic optimization problem is the search for an efficient method for discretizing the distribution of the random parameters. This step becomes even more critical when one works with *multistage* problems for which the discretization of the underlying stochastic process leads to a scenario tree of potentially gigantic size. Finding smartly designed scenario trees is therefore essential to broaden the class of solvable multistage problems. In this paper, we introduce a new approach for generating scenario trees based on a quality criterion called the *figure of demerit*, which takes into account the structure of the optimization problem to design suitable scenario trees. This approach is versatile as it can be applied to essentially any problem, regardless of linearity, convexity, etc., and combined with a great deal of discretization methods used in numerical integration. The way it is implemented, however, depends on the properties of the problem and the stochastic process, and is explained through examples and algorithms.

**Keywords.** Stochastic optimization, multistage stochastic programming, scenario-tree generation.

**Acknowledgements.** The authors would like to thank for their support Hydro-Québec and the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC/Hydro-Québec Industrial Research Chair on the Stochastic Optimization of Electricity Generation, Polytechnique Montréal, and the NSERC Discovery Grants Program.

† Part of this research was carried out while David Munger was research professional at CIRRELT and Polytechnique Montréal.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

\* Corresponding author: Julien.Keutchayan@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec  
Bibliothèque et Archives Canada, 2017

© Keutchayan, Munger, Gendreau, Bastin and CIRRELT, 2017

# 1 Preliminaries

Multistage stochastic programming provides a mathematical framework for modeling and solving multistage decision-making problems that involve uncertain parameters. Sources of uncertainty are numerous in real-world optimization problems: the prices of assets in a portfolio optimization problem; see, e.g., Ziemba [44] and Yu et al. [43]; the energy demand and price, and the natural inflows in reservoirs in a hydroelectricity production problem; see, e.g., Wallace and Fleten [42] and Kovacevic et al. [21]; the number of customers on each route of a transportation network in a network design problem; see, e.g., Louveaux [24] and Powell and Topaloglu [34]; etc. Provided a probability distribution is available for the underlying stochastic process –most often inferred from available data– a *multistage stochastic programming problem* (MSPP) can be formulated and addressed using different approaches; see, e.g., Ruszczyński and Shapiro [36], Birge and Louveaux [1], and King and Wallace [20].

In this paper, we consider a MSPP with discrete stages  $t = 0, 1, \dots, T$ , with  $T \in \mathbb{N}_+$ , and for which the objective function is given as the expectation of a revenue function. The revenue function depends on the uncertain parameters, which are modeled at each stage  $t = 1, \dots, T$  by a  $\mathbb{R}^{d_t}$ -valued random vector  $\xi_t$ , and on the decisions  $y_t \in \mathbb{R}^{s_t}$  made at  $t = 0, \dots, T$  with the goal to maximize the objective function while satisfying the constraints; for the sake of clarity and without loss of generality we assume that  $s_t = s$  and  $d_t = d$  for all  $t$ . For notational convenience, we add an artificial random vector  $\xi_0$  at stage 0 that takes only one possible value  $\xi_0$  with probability one. We consider, therefore, the following dynamic of actions from stage 0 to  $T$ :

$$y_0 = x_0(\xi_0) \xrightarrow{\xi_1} y_1 = x_1(y_0; \xi_0, \xi_1) \xrightarrow{\xi_2} \dots \xrightarrow{\xi_T} y_T = x_T(y_{..T-1}; \xi_{..T}). \quad (1)$$

This dynamic means that the decision  $y_0$  is made when only the non-random information  $\xi_0$  is available, then a realization  $\xi_1$  of  $\xi_1$  is revealed and based on it a decision  $y_1$  is made at stage 1. The decision  $y_1$  is therefore a function of  $y_0$  and  $(\xi_0, \xi_1)$ , which we denote by  $y_1 = x_1(y_0; \xi_0, \xi_1)$ . Note that we keep the notation with  $y$ 's for decision *vectors* and the notation with  $x$ 's for decision *functions*; the latter are functions of the  $y$ 's and the random parameters  $\xi$ . At each stage  $t \geq 2$ , the decision  $y_t = x_t(y_{..t-1}; \xi_{..t})$  is a function of the previous decisions  $y_{..t-1} := (y_0, y_1, \dots, y_{t-1})$  and the random information  $\xi_{..t} := (\xi_0, \xi_1, \dots, \xi_t)$  available at this stage.

The MSPP considered in this paper can be formulated recursively by introducing the stage- $t$  *recourse function*  $Q_t^*$  and the stage- $t$  *expected recourse function*  $Q_t$ , which are linked by the following set of stochastic dynamic programming equations:

$$Q_t^*(y_{..t-1}; \xi_{..t}) := \sup_{y_t \in Y_t(y_{..t-1}; \xi_{..t}) \subseteq \mathbb{R}^s} Q_t(y_{..t-1}, y_t; \xi_{..t}), \quad t = 0, \dots, T, \quad (2)$$

$$Q_t(y_{..t}; \xi_{..t}) := \mathbb{E}[Q_{t+1}^*(y_{..t}; \xi_{..t+1}) | \xi_{..t} = \xi_{..t}], \quad t = 0, \dots, T-1, \quad (3)$$

where the set-valued mapping  $(y_{..t-1}, \xi_{..t}) \rightrightarrows Y_t(y_{..t-1}; \xi_{..t}) \subseteq \mathbb{R}^s$  models the *feasible set* at stage  $t$  and, to complete the definition at the first and last stage, the equation (2) is initialized at  $t = T$  by setting

$$Q_T(y_{..T}; \xi_{..T}) := q(y_{..T}; \xi_{..T}), \quad (4)$$

where  $q : \mathbb{R}^{(T+1)s} \times \mathbb{R}^{(T+1)d} \rightarrow \mathbb{R}$  is the *revenue function*, and at  $t = 0$  we remove the  $y_{..t-1}$ -argument of the feasible set and recourse function, i.e.,

$$Y_0(y_{..0-1}; \xi_{..0}) = Y_0(\xi_0) \quad \text{and} \quad Q_0^*(y_{..0-1}; \xi_{..0}) = Q_0^*(\xi_0). \quad (5)$$

The quantity  $Q_0^*(\xi_0)$ , which is no longer a function as  $\xi_0$  takes only one value, is the *optimal-value* of the MSPP. The *optimal decision policy* of the MSPP is the sequence  $(x_0^*, x_1^*, \dots, x_T^*)$  where  $x_0^*(\xi_0)$

is a maximizer of  $Q_0(\cdot; \xi_0)$  and  $x_t^*(y_{..t-1}; \xi_{..t})$  is a maximizer of  $Q_t(y_{..t-1}, \cdot; \xi_{..t})$  in (2). We refer to Keutchayan et al. [19], which provides the ground for our approach, for a complete and rigorous presentation of the MSPP.

**Example 1.1.** An important particular case of the above general problem is the *semi-linear* MSPP (linear in the constraints, non-linear in the revenue function) for which the following simplifications occur:

- $q(y_{..T}; \xi_{..T}) = \sum_{t=0}^T q_t(y_t; \xi_t)$ ;
- $Y_0(\xi_0) = \{y_0 \in \mathbb{R}^s : A_0(\xi_0)y_0 = b_0(\xi_0), y_0 \geq 0\}$ ;
- $Y_t(y_{..t-1}; \xi_{..t}) = \{y_t \in \mathbb{R}^s : A_t(\xi_t)y_t + B_t(\xi_t)y_{t-1} = b_t(\xi_t), y_t \geq 0\}$  for every  $t = 1, \dots, T$ ;

where  $q_0(\cdot; \xi_0)$ ,  $b_0(\xi_0)$ ,  $A_0(\xi_0)$  are *deterministic* function, vector and matrix, respectively, and  $q_t(\cdot; \xi_t)$ ,  $b_t(\xi_t)$ ,  $A_t(\xi_t)$ ,  $B_t(\xi_t)$ , for  $t = 1, \dots, T$ , depend on  $\xi_t$  and hence are realizations of *random* quantities. In a *linear* MSPP the revenue function is further simplified as  $q_t(y_t; \xi_t) = c_t(\xi_t)^\top y_t$ .

In this paper, we consider the MSPP in its general form (2)-(3), along with the set of conditions detailed in Keutchayan et al. [19] ensuring that it is well-defined, as the mathematical developments below allows to consider essentially any MSPP.

Problems in the form (2)-(3) are generally hard to solve. First, the exact computation of the expectation (3) cannot be carried out in most applications involving continuous distributions. Second, even when the distribution sits on finitely many points, the size of the resulting problem (measured in terms of the number of decision variables and constraints) is generally too large to be solved exactly by optimization tools. The approach that we follow in this paper to solve *approximately* the MSPP consists in building a *scenario tree*, i.e., a discretized representation of the stochastic process that involves only a limited number of its scenarios; see, e.g., Dupačová et al. [6] and Defourny et al. [3].

More explicitly, we call *scenario tree* a triplet  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  where

- (D1)  $\mathcal{T} = (\mathcal{N}, \mathcal{E}, n_0)$  is the *rooted tree structure* with  $\mathcal{N}$  the finite node set,  $\mathcal{E}$  the edge set and  $n_0$  the root node;
- (D2)  $\mathcal{P} = \{\zeta^n \in \mathbb{R}^d : n \in \mathcal{N}\}$  is the set of *discretization points* (one point  $\zeta^n$  for each node and at the root  $\zeta^{n_0} = \xi_0$  by definition);
- (D3)  $\mathcal{W} = \{w^n \in (0, \infty) : n \in \mathcal{N}\}$  is the set of *discretization weights* (one weight  $w^n$  for each node and at the root  $w^{n_0} = 1$  by definition).

On top of the above definition, we introduce the following additional notations and terminology that will help us describe conveniently a scenario tree:

- (N1)  $[n, m]$  is the (unique) sequence of nodes from  $n$  to  $m$  where  $m$  is a descendant of  $n$ ; we write  $(n, m]$  when  $n$  is excluding from the sequence;
- (N2)  $t(n)$  is the *stage* (or *depth*) of  $n$ , i.e., the number of edges between  $n_0$  and  $n$ , and  $\mathcal{N}_t := \{n \in \mathcal{N} : t(n) = t\}$ ;
- (N3)  $\zeta^{\cdot n} := (\zeta^m)_{m \in [n_0, n]}$  is the *discretization sequence* leading to  $n$  and  $W^n := \prod_{m \in [n_0, n]} w^m$  is the *product weight* of  $n$ ;
- (N4)  $C(n)$  is the set of *children* nodes of  $n \in \mathcal{N} \setminus \mathcal{N}_T$ , i.e., those connected to  $n$  at stage  $t(n) + 1$ , and  $p(n)$  is the *parent* node of  $n \in \mathcal{N} \setminus \{n_0\}$ , i.e., that connected to  $n$  at stage  $t(n) - 1$ ;

- (N5)  $\mathcal{P}^n$  (resp.  $\mathcal{W}^n$ ) is the set of discretization points (resp. weights) at  $C(n)$ ;
- (N6)  $b := (b_0, \dots, b_{T-1})$  is the *bushiness* of  $\mathcal{T}$  where  $b_{t(n)} := |C(n)|$  is the *branching coefficient* at stage  $t(n)$ ; the concept of bushiness is defined only for *symmetrical* tree structures, i.e., those for which  $|C(n)| = |C(m)|$  whenever  $t(n) = t(m)$ ;
- (N7)  $\mathcal{W}$  is said to be *normalized* if  $\sum_{m \in C(n)} w^m = 1$  for all  $n \in \mathcal{N} \setminus \mathcal{N}_T$  (this also implies that  $\sum_{n \in \mathcal{N}_t} W^n = 1$  for all  $t = 0, \dots, T-1$ ), and  $\mathcal{W}$  is said to be *standardized* if it is normalized and satisfies  $w^n = w^m$  whenever  $p(n) = p(m)$  (standardized weights have a unique form given by  $w^n = |C(p(n))|^{-1}$  for all  $n \in \mathcal{N} \setminus \{n_0\}$ );
- (N8)  $\mathcal{N}(n)$  and  $\mathcal{E}(n)$  denote the node set and edge set of the *sub-scenario tree* rooted at  $n$ .

To give sense to the scenario tree as a discrete approximation of the stochastic process, we require that the triplet  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  satisfies the additional two properties:

- (D4) each scenario-tree leaf is at depth  $T$  (a *leaf* is any node  $n \neq n_0$  incident to only one edge);
- (D5) each discretization sequence  $\zeta^{\cdot n}$  for  $n \in \mathcal{N}_T$  is a possible realization (or *scenario*) of the stochastic process.

Given a scenario tree, the discrete approximation of the MSPP, called the *scenario-tree approximate problem* (STAP), is defined as follows:

$$\widehat{Q}_t^*(y_{..t-1}; \zeta^{\cdot n}) := \sup_{y_t \in Y_t(y_{..t-1}; \zeta^{\cdot n}) \subseteq \mathbb{R}^s} \widehat{Q}_t(y_{..t-1}, y_t; \zeta^{\cdot n}), \quad t = 0, \dots, T, \quad n \in \mathcal{N}_t, \quad (6)$$

$$\widehat{Q}_t(y_{..t}; \zeta^{\cdot n}) := \sum_{m \in C(n)} w^m \widehat{Q}_{t+1}^*(y_{..t}; \zeta^{\cdot n}, \zeta^m), \quad t = 0, \dots, T-1, \quad n \in \mathcal{N}_t, \quad (7)$$

where  $\widehat{Q}_t^*$  and  $\widehat{Q}_t$  are the scenario-tree estimators of  $Q_t^*$  and  $Q_t$ , respectively, and, similarly to (4) and (5), at  $t = T$  the equation (6) is initialized by setting

$$\widehat{Q}_T(y_{..T}; \zeta^{\cdot n}) := q(y_{..T}; \zeta^{\cdot n}), \quad \text{for every } n \in \mathcal{N}_T, \quad (8)$$

and at  $t = 0$  we remove the  $y_{..t-1}$ -argument, i.e.,

$$Y_0(y_{..0-1}; \zeta^{\cdot n_0}) = Y_0(\zeta^{n_0}) \quad \text{and} \quad \widehat{Q}_0^*(y_{..0-1}; \zeta^{\cdot n_0}) = \widehat{Q}_0^*(\zeta^{n_0}). \quad (9)$$

The quantity  $\widehat{Q}_0^*(\zeta^{n_0})$  is therefore the *optimal-value* of the STAP, it is the scenario-tree estimator of  $Q_0^*(\xi_0)$  and the difference  $|\widehat{Q}_0^* - Q_0^*$  is called the *optimal-value error* (we drop the arguments for conciseness). The *optimal decision policy* of the STAP is the sequence  $(\widehat{x}_0^*, \widehat{x}_1^*, \dots, \widehat{x}_T^*)$  where  $\widehat{x}_0^*(\zeta^{n_0})$  is a maximizer of  $\widehat{Q}_0(\cdot; \zeta^{n_0})$  and  $\widehat{x}_t^*(y_{..t-1}; \zeta^{\cdot n})$  is a maximizer of  $\widehat{Q}_t(y_{..t-1}, \cdot; \zeta^{\cdot n})$  in (6).

When dealing with a real-world application, the scenario tree must be carefully chosen in order to provide a good approximation of the original problem. What constitutes a “good approximation” is an open question and will generally depend on the considered problem; see Kaut and Wallace [17] and Keutchan et al. [18]. In this paper, we follow the common guideline that consists in constructing the scenario tree with the goal to keep the optimal-value error as small as possible for a given finite computational cost. We emphasize that by “scenario tree” we mean the whole triplet  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  described above, and not only the set of discretization points and weights  $(\mathcal{P}, \mathcal{W})$ . In particular, we want to improve the efficiency of the scenario-tree approach by searching for appropriate tree structures  $\mathcal{T}$ , which means searching also among structures that are *not* symmetrical. We will see that using non-symmetrical structures allows to capitalize on some knowledge that the

decision-maker may have about the revenue function and the constraints of the problem, which is not permitted by purely distribution-based scenario-tree generation methods, which focus only on the stochastic process and are blind to the specific structure of the problem.

Our approach differs from those based on scenario reduction (see Dupačová et al. [7] and Heitsch and Römisch [14]) also by the fact that we do not reduce a given large set of scenarios so that it reaches the required size. We rather build the scenario tree directly with the required size; by “size” we mean the number of scenarios, i.e.,  $|\mathcal{N}_T|$ . The way we generate the scenario tree relies to some extent on pure discretization methods used in numerical integration, such as quasi-Monte Carlo methods, optimal quantization methods, numerical integration rules, etc.; see, e.g., Sobol [41], Pagès et al. [30], and Davis and Rabinowitz [2]. These methods have already been applied to scenario-tree generation, for instance in Pflug [31] and Hilli and Pennanen [16], however, they generally require a predetermined tree structure. The novelty of our work is to provide a systematic approach for finding suitable tree structures to be used alongside any discretization method to improve the overall efficiency of the scenario-tree solution approach.

To develop our approach, we use the optimal-value error bound derived in Keutchayan et al. [19] and derive from it a *figure of demerit* for scenario trees. The decision-maker shall use this figure as a quality criterion to find an appropriate scenario tree to solve a given MSPP or a given *class* of MSPPs (a *class* is a set of problems having similar structure).

The remainder of this paper is organized as follows: In Section 2, we introduce the figure of demerit and explain the general procedure to find scenario trees. In Section 3, we show how the framework can be simplified and implemented in different settings; in particular, Section 3.1 deals with the case of stagewise independent stochastic processes, while Sections 3.2 and 3.3 deal with stochastic processes with dependence across stages, the former is specifically on short-range dependence and the latter is on long-range dependence. In Section 4, we conclude the paper and discuss the outlook. Along the lines of our development, we illustrate the approach using several examples; all corresponding figures are gathered in Appendix A.

## 2 Definition of the figure of demerit

We recall below the optimal-value error bound that we use to derive the figure of demerit:

**Proposition 2.1** (Keutchayan et al. [19]). *The optimal-value error between the MSPP and the STAP for any scenario tree  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  is bounded as follows:*

$$|Q_0^* - \widehat{Q}_0^*| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \sup_{f \in \mathcal{G}^n} |\mathcal{E}^n(f)|, \quad (10)$$

where, for every  $t = 0, \dots, T-1$  and  $n \in \mathcal{N}_t$ ,  $\mathcal{E}^n(f)$  is the integration error at node  $n$  defined as

$$\mathcal{E}^n(f) = \mathbb{E}[f(\boldsymbol{\xi}_{t+1}) | \boldsymbol{\xi}_{..t} = \zeta^{..n}] - \sum_{m \in \mathcal{C}(n)} w^m f(\zeta^m), \quad (11)$$

and  $\mathcal{G}^n$  is any set of functions integrable with respect to the conditional distribution of  $\boldsymbol{\xi}_{t+1}$  given  $\boldsymbol{\xi}_{..t} = \zeta^{..n}$  that contains the set  $\mathcal{Q}^n$  of recourse functions defined as

$$\mathcal{Q}^n = \{Q_{t+1}^*(y_{..t}; \zeta^{..n}, \cdot) : y_{..t} = x_{..t}(\zeta^{..n}) \text{ with } x_{..t} \in \prod_{i=0}^t \{x_i^*, \widehat{x}_i^*\}\}. \quad (12)$$

In (12),  $\prod_{i=0}^t$  denotes the  $(t+1)$ -fold Cartesian product, hence  $x_{..t}$  is any decision policy having each of its components equal to either  $x_i^*$  or  $\widehat{x}_i^*$ , and  $x_{..t}(\zeta^{..n})$  denotes the decisions made with  $x_{..t}$  when the realization is  $\zeta^{..n}$  in the dynamic of actions (1).

The above result shows that the optimal-value error can be kept small if the discretization at each node  $n \in \mathcal{N} \setminus \mathcal{N}_T$  is adequate to approximate the expectation of any recourse function in  $\mathcal{Q}^n$ . These functions are not known exactly but we may know some of their properties, such as the magnitude of their variation with respect to the random parameters, that are useful to forecast the magnitude of the integration error (11) at node  $n$ . Thus, the general idea to adapt the tree structure to the MSPP is the following: for a given node  $n \in \mathcal{N} \setminus \mathcal{N}_T$ , it may occur that the functions in  $\mathcal{G}^n$  are easy to integrate numerically, in the sense that  $\mathcal{E}^n(f)$  tends to be small for all  $f \in \mathcal{G}^n$  regardless of the number of nodes in  $C(n)$ ; in this case it makes sense to have only few nodes in  $C(n)$  and to use the extra nodes in  $C(m)$  where  $m$  is such that  $\mathcal{E}^m(f)$  tends to be larger.

A core step in the development of our approach is therefore the search of relevant measures for assessing the difficulty of numerically integrating the functions  $f \in \mathcal{G}^n$ , in order to identify when  $\mathcal{E}^n(f)$  tends to be large or small. To this end, we introduce the set  $\mathcal{G}_t(\xi_{..t})$  of recourse functions defined as

$$\mathcal{G}_t(\xi_{..t}) = \{Q_{t+1}^*(y_{..t}; \xi_{..t}, \cdot) : y_{..t} \in Z_t(\xi_{..t})\}, \quad (13)$$

for every  $t = 0, \dots, T-1$  and  $\xi_{..t}$ , where  $Z_t(\xi_{..t})$  is the set of all feasible decision sequences up to stage  $t$  for the realization  $\xi_{..t}$  defined recursively as  $Z_0(\xi_0) = Y_0(\xi_0)$  and

$$Z_t(\xi_{..t}) = \{y_{..t} \in \mathbb{R}^{s(t+1)} : y_{..t-1} \in Z_{t-1}(\xi_{..t-1}), y_t \in Y_t(y_{..t-1}; \xi_{..t})\}. \quad (14)$$

We make the two following assumptions regarding  $\mathcal{G}_t(\xi_{..t})$ :

**Condition 1.** For any scenario tree  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  and node  $n \in \mathcal{N} \setminus \mathcal{N}_T$ , the set  $\mathcal{G}_{t(n)}(\zeta^{..n})$  is included in a function space  $\mathcal{F}$  for which it holds that

$$|\mathcal{E}^n(f)| \leq \mathcal{V}_{\mathcal{F}}(f) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n), \quad \text{for every } f \in \mathcal{F}, \quad (15)$$

where  $0 \leq \mathcal{V}_{\mathcal{F}}(\cdot) < \infty$  depends only on the integrand  $f$  and  $0 \leq \mathcal{D}_{\mathcal{F}}(\cdot, \cdot) < \infty$  depends only on the discretization points and weights  $(\mathcal{P}^n, \mathcal{W}^n)$ .

It is also possible to assume a weaker form of Condition 1, where the bound (15) does not hold for *any* scenario tree but only for points and weights  $(\mathcal{P}, \mathcal{W})$  that are restricted to some specific form: normalized and standardized weights (cf. (N7), Section 1), points forming lattices, etc. If the weaker form is assumed, then the development below holds only for the corresponding subset of scenario trees. However, for the sake of conciseness, we do not treat this case separately and we simply assume that Condition 1 holds in its strong form.

Function spaces for which (15) holds (in strong or weak form) are for instance:

- (S1)  $\mathcal{F} = W_{2,\gamma,\text{mix}}^{(1,\dots,1)}([0,1]^d)$  is the *weighted tensor product* Sobolev space:  $\mathcal{V}_{\mathcal{F}}(f)$  is the norm of  $f$  in that space and  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  is the so-called *weighted discrepancy*; see, e.g., Sloan and Woźniakowski [40], Sloan et al. [39], and Dick and Pillichshammer [5], Chapter 3.
- (S2)  $\mathcal{F}$  is a *reproducing kernel* Hilbert space of integrable functions for which the functional  $f \mapsto \mathcal{E}^n(f)$  is continuous:  $\mathcal{V}_{\mathcal{F}}(f)$  is the norm of  $f$  and  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  is the norm of the representer of  $\mathcal{E}^n(\cdot)$  given by Riesz representation theorem; see, e.g., Dick and Pillichshammer [5], Chapter 2, and Dick et al. [4].

- (S3)  $\mathcal{F}_q$  is the space of integrable Lipschitz continuous functions of order  $q \in [1, \infty)$ :  $\mathcal{V}_{\mathcal{F}_q}(f)$  is the order- $q$  Lipschitz constant of  $f$  and  $\mathcal{D}_{\mathcal{F}_q}(\mathcal{P}^n, \mathcal{W}^n)$  is the order- $q$  Fortet-Mourier distance between the conditional distribution of  $\xi_{t(n)+1}$  given  $\xi_{..t(n)} = \zeta^{..n}$  and the distribution  $\sum_{m \in C(n)} w^m \delta_{\zeta^m}$ , where  $\delta_{\zeta^m}$  is the Dirac measure at  $\zeta^m$  and  $\mathcal{W}^n$  is normalized; see, e.g., Pflug and Pichler [32].
- (S4)  $\mathcal{F}$  is a Banach space of integrable functions for which the functional  $f \mapsto \mathcal{E}^n(f)$  is continuous:  $\mathcal{V}_{\mathcal{F}}(f)$  is the norm of  $f$  and  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  is the worst-case error of  $(\mathcal{P}^n, \mathcal{W}^n)$  in  $\mathcal{F}$ ; see, e.g., Hickernell et al. [15], Kuo et al. [22], and Novak and Wozniakowski [28].

The setting (S1) is a particular case of (S2), and the settings (S2) and (S3) are particular cases of (S4). In their recent work on two-stage linear problems, Heitsch et al. [13] showed that the functions  $f \in \mathcal{Q}^{n_0}$  almost belong to the function space of setting (S1) if the effective dimension of  $f$  is small; see also Griebel et al. [11].

What makes Condition 1 useful is the fact that when it holds  $\mathcal{V}_{\mathcal{F}}(f)$  can be interpreted as a measure of difficulty for numerically integrating  $f \in \mathcal{F}$ . Indeed, if  $\mathcal{V}_{\mathcal{F}}(f)$  almost equals zero, then the integration error  $\mathcal{E}^n(f)$  is small regardless of the number of discretization points and weights. Conversely, if the bound is tight, a large value of  $\mathcal{V}_{\mathcal{F}}(f)$  means that  $f$  is difficult to integrate numerically. In the latter case, it makes sense to add more nodes in  $C(n)$  and to spend more computational time searching for  $(\mathcal{P}^n, \mathcal{W}^n)$  minimizing  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$ . We call  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  the *demerit of node  $n$*  and  $\mathcal{V}_{\mathcal{F}}(f)$  the  $\mathcal{V}_{\mathcal{F}}$ -variation of  $f$ , as the latter is typically linked with the intuitive sense of how much the values of  $f$  vary over its definition domain.

The second condition ensures that  $\mathcal{V}_{\mathcal{F}}(f)$  does not go to infinity as  $f$  varies in  $\mathcal{G}_{t(n)}(\zeta^{..n})$ :

**Condition 2.** For each  $t = 0, \dots, T-1$  and realization  $\xi_{..t}$ , the recourse functions  $Q_{t+1}^*(y_{..t}; \xi_{..t}, \cdot)$  have uniformly bounded  $\mathcal{V}_{\mathcal{F}}$ -variation with respect to  $y_{..t} \in Z_t(\xi_{..t})$ .

When the set  $Z_t(\xi_{..t})$  is a compact subset of  $\mathbb{R}^{s(t+1)}$  (as holds under the conditions given in Keutchan et al. [19]), Condition 2 is satisfied in particular if the mapping  $Z_t(\xi_{..t}) \ni y_{..t} \mapsto \mathcal{V}_{\mathcal{F}}(Q_{t+1}^*(y_{..t}; \xi_{..t}, \cdot))$  is upper semi-continuous.

We define now two new concepts of prime importance in the development of our scenario-tree generation approach:

**Definition 2.2.** We call set of guidance functions the set  $\Gamma := \{\gamma_t(\cdot) : t = 0, \dots, T-1\}$ , where each  $\gamma_t(\cdot)$  is a  $\mathbb{R}_{\geq 0}$ -valued function defined on the set of all realizations  $\xi_{..t}$ . We call figure of demerit of the scenario tree  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  for the set of guidance functions  $\Gamma$  the quantity

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma) := \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\zeta^{..n}) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n). \quad (16)$$

The guidance functions  $\gamma_0(\cdot), \gamma_1(\cdot), \dots, \gamma_{T-1}(\cdot)$  are specified externally to the problem by the decision-maker with the goal to match the  $\mathcal{V}_{\mathcal{F}}$ -variability of the recourse functions at each stage and for each value of the random realizations. Thus, these guidance functions carry information about the structure of the MSPP related to its stochastic process, revenue function and constraints. For a suitable choice of  $\Gamma$ , the figure of demerit plays the role of a measure of quality for the scenario tree, as shown by the following theorem:

**Theorem 2.3.** Under Conditions 1 and 2, there exists a set of guidance functions  $\tilde{\Gamma} = \{\tilde{\gamma}_t(\cdot) : t = 0, \dots, T-1\}$  such that for every scenario tree  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  the figure of demerit  $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \tilde{\Gamma})$  is an upper bound on the optimal-value error, i.e.,

$$|Q_0^* - \hat{Q}_0^*| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \tilde{\gamma}_{t(n)}(\zeta^{..n}) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n). \quad (17)$$

*Proof.* We show that the guidance functions defined as

$$\tilde{\gamma}_t(\xi_{..t}) := \sup_{f \in \mathcal{G}_t(\xi_{..t})} \mathcal{V}_{\mathcal{F}}(f), \quad \text{for every } t = 0, \dots, T-1 \text{ and } \xi_{..t}, \quad (18)$$

are a possible choice for (17) to hold. Such  $\tilde{\gamma}_0(\cdot), \dots, \tilde{\gamma}_{T-1}(\cdot)$  are well-defined since the supremum is finite by Condition 2. Let  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  be any scenario tree. Taking  $\mathcal{G}^n := \mathcal{G}_{t(n)}(\zeta_{..n})$  for every  $n \in \mathcal{N} \setminus \mathcal{N}_T$  in Proposition 2.1 ensures that  $\mathcal{Q}^n \subseteq \mathcal{G}^n$  and allows to write by Condition 1:

$$|Q_0^* - \hat{Q}_0^*| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \sup_{f \in \mathcal{G}^n} |\mathcal{E}^n(f)| \quad (19)$$

$$\leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n) \sup_{f \in \mathcal{G}^n} \mathcal{V}_{\mathcal{F}}(f). \quad (20)$$

The last term is precisely the right-hand side of (17) for this choice of guidance functions.  $\square$

The above theorem provides the theoretical justification of our approach to generate scenario trees. This approach is in two steps: first, a set of guidance functions  $\Gamma$  is specified by the decision-maker with the goal to match the  $\mathcal{V}_{\mathcal{F}}$ -variability of the recourse functions with respect to the random parameters; second, the scenario tree with the lowest figure of demerit  $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$  is chosen among all scenario trees  $(\mathcal{T}, \mathcal{P}, \mathcal{W})$  of a given size. This last step amounts to finding a suitable tree structure along with the corresponding discretization points and weights.

We make now several remarks about the first step of the approach: The choice of guidance functions is not unique. Not only any functions larger than the ones used in the proof will keep the inequality valid, but more importantly, more suitable functions can be chosen (at least theoretically) by considering functions sets that are strictly smaller than  $\mathcal{G}_t(\xi_{..t})$  (in the sense of inclusion) but still include  $\mathcal{Q}^n$  for each possible scenario tree. Finding such sets is however difficult since we face the problem that we do not know  $x^*$  and that  $\hat{x}^*$  depends on the scenario tree. On the practical side, though, the choice of guidance functions is much more flexible that it can appear at first in the theorem. Indeed, even if the figure of demerit is not a valid bound on the optimal-value error for the chosen functions, we argue that minimizing the figure of demerit still provides a suitable scenario tree if the guidance functions single out the “easy” scenarios (those with low variability) from the “difficult” ones (those with high variability). Therefore, *in practice*, the guidance functions are scale-free, in the sense that multiplying each function by the same constant essentially scales up or down the figure of demerit but does not change the minimizing scenario tree.

We also make several remarks about the second step of the approach: Minimizing the figure of demerit in the general form above turns out to be a difficult task in itself. It requires to optimize simultaneously over the Cartesian product of the discrete set of all tree structures  $\mathcal{T}$  and the continuous set of all discretization points and weights  $(\mathcal{P}, \mathcal{W})$ . One way to tackle this issue is to decouple the minimization, i.e., iterate over the structures and minimize in  $(\mathcal{P}, \mathcal{W})$  at each iteration. However, the minimization in  $(\mathcal{P}, \mathcal{W})$  at fixed  $\mathcal{T}$  remains difficult as it is highly non-linear. A way to go around this difficulty is to restrict our attention to points and weights  $(\mathcal{P}^n, \mathcal{W}^n)$  of some specific forms that are designed to keep the node- $n$  demerit as small as possible. This is where discretization methods used in numerical integration are useful, because they provide the theory and the algorithmic tools to find  $(\mathcal{P}^n, \mathcal{W}^n)$  that minimizes  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  for each node  $n \in \mathcal{N} \setminus \mathcal{N}_T$ ; see, e.g., L’Ecuyer and Munger [23]. Thus, by employing the previous simplification, we essentially reduce the minimization of  $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$  in  $(\mathcal{P}, \mathcal{W})$  at fixed  $\mathcal{T}$  to the following two steps: (i) the minimization of  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^n, \mathcal{W}^n)$  at each node, and (ii) the right assignment of the discretization points and weights to the appropriate nodes. The importance of the latter step will become clear in the next section.

We end this section by showing in the example below, which we purposely simplify, how a tree structure can be found by minimizing the figure of demerit:

**Example 2.1.** Consider that we want to build a scenario tree for the following problem: At the beginning of the year (stage 0) a humanitarian organization chooses the different locations of its facilities in some part of the world that faces natural disasters on a yearly basis. In spring (stage 1), the organization observes the risk forecast for the year and, based on it, assigns the equipment to the different facilities accordingly. The risk forecast is represented by a random variable  $\xi_1$  taking four different values:  $\xi_1^1$  (low risk) to  $\xi_1^4$  (extreme risk) with probability  $p^1$  to  $p^4$  inferred from historical data. In summer and fall (stage 2), the natural disasters may occur, their intensities and locations are represented by a random vector  $\xi_2$  having continuous distribution correlated to  $\xi_1$ , and the organization responds to them in a way to maximize the rescue efficiency.

We consider a scenario tree with four stage-1 nodes  $\mathcal{N}_1 = \{n_1, n_2, n_3, n_4\}$  having discretization points  $\zeta^{n_i} = \xi_1^i$  and weights  $w^{n_i} = p^i$  and we want to find the number of children nodes  $M_i := |C(n_i)|$  for each stage-1 node  $n_i$  that minimizes the figure of demerit among all tree structures having less than  $N$  scenarios. Since the scenario-tree discretization at stage 1 is exact, the integration error  $\mathcal{E}^{n_0}(f)$  is zero regardless of  $f$  and hence we can set  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{n_0}, \mathcal{W}^{n_0}) = 0$ . The figure of demerit of the scenario tree is therefore

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma) = \sum_{i=1}^4 p^i \gamma_1(\xi_1^i) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{n_i}, \mathcal{W}^{n_i}), \quad (21)$$

where the guidance function  $\gamma_1(\xi_1^i)$  has the following interpretation: *it measures the variability of the rescue efficiency at stage 2 given the realization  $\xi_1^i$ .* For more clarity, let us assume that  $\gamma_1(\xi_1^1) \leq \gamma_1(\xi_1^2) \leq \gamma_1(\xi_1^3) \leq \gamma_1(\xi_1^4)$ . Such a difference in the variability may come from the fact that the conditional variance of  $\xi_2$  given  $\xi_1^i$  increases with  $i$ , i.e., years of extreme risk coincide with natural disasters of highly variable intensities and locations.

If we assume that the optimal points and weights  $(\mathcal{P}^{n_i}, \mathcal{W}^{n_i})$  minimizing the node- $n_i$  demerit satisfy  $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{n_i}, \mathcal{W}^{n_i}) = A/|C(n_i)|^\alpha = A/M_i^\alpha$  for each  $i = 1, \dots, 4$ , with  $\alpha$  and  $A$  two positive constants, then the distribution of children nodes at stage 2 that minimizes the figure of demerit is given by the optimal solution of

$$\min_{M=(M_1, \dots, M_4) \in \mathbb{N}_+^4} \sum_{i=1}^4 \frac{p^i \gamma_1(\xi_1^i)}{M_i^\alpha} \quad \text{subject to} \quad \sum_{i=1}^4 M_i \leq N. \quad (22)$$

As a result, the optimal  $M_i$  tends to be large if the product  $p^i \gamma_1(\xi_1^i)$  is large. Conversely, if  $p^i \gamma_1(\xi_1^i)$  is close to zero (i.e., if  $\xi_1^i$  almost never occurs or if the conditional future given  $\xi_1^i$  is almost not uncertain), then the optimal  $M_i$  equals one, which means that only one node is necessary in  $C(n_i)$  to reduce the discretization error, the remaining nodes being more useful elsewhere. We show in Figure 1, Appendix A, the optimal tree structures obtained for  $\alpha = 1$  and different choices of  $p^i$  and  $\gamma_1(\xi_1^i)$ .

### 3 Simplifications and implementation

The computation of the node- $n$  demerit is a core step in the application of our approach. A general formula for it that would cover all possible types of scenario trees does not exist, it can only be derived in some particular cases after having specified the function space  $\mathcal{F}$  and the discretization points and weights  $(\mathcal{P}^n, \mathcal{W}^n)$  (cf. the settings (S1) to (S4) in Section 2). It is however possible to work with a simplified form, while keeping some generality, by merely assuming the following:

**Condition 3.** We know a discretization method that generates points and weights, denoted specifically by  $(\mathcal{P}_*, \mathcal{W}_*)$  throughout this section, that satisfy

$$\mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^n, \mathcal{W}_*^n) = f_{t(n)}(|C(n)|), \quad \text{for every } n \in \mathcal{N} \setminus \mathcal{N}_T, \quad (23)$$

where  $f_0, \dots, f_{T-1} : \mathbb{N}_+ \rightarrow [0, \infty)$  are monotonically decreasing functions.

This condition allows to consider a great deal of discretization methods, each having its specific speed of convergence toward zero for the node demerit given by  $f_0, \dots, f_{T-1}$ , while providing a more practical formula. The interest of a decision-maker should obviously go toward methods such that each  $f_t$  decreases to zero sufficiently fast. A typical family of decreasing function is given by  $f_t(x) = A/x^\alpha$  with  $\alpha > 0$  the *rate of convergence* of the discretization method and  $A > 0$  a constant, both  $\alpha$  and  $A$  may depend on  $t$  and  $d$ .

We show now how Condition 3 allows us to find the scenario tree of lowest figure of demerit given some decreasing functions  $f_0, \dots, f_{T-1}$  and a set of guidance functions  $\Gamma$ . The way this is done depends on the range of dependence of the guidance functions.

### 3.1 Constant guidance functions

We consider the *semi-linear* MSPP described in Example 1.1 with the additional assumption that the stochastic process  $(\xi_0, \dots, \xi_T)$  is *stagewise independent*, i.e., for each  $t = 1, \dots, T$  the random vector  $\xi_t$  is independent of  $(\xi_0, \dots, \xi_{t-1})$ . In this setting, the variability with respect to  $\xi_t$  of the stage- $t$  recourse function does not depend on the realization  $(\xi_0, \dots, \xi_{t-1})$ , hence it is relevant to consider guidance functions  $\Gamma_{\text{ind}}$  that do not depend on past realizations:

$$\Gamma_{\text{ind}} := \{\gamma_0, \gamma_1, \dots, \gamma_{T-1}\}. \quad (24)$$

We restrict our attention to scenario trees with symmetrical tree structures, which are characterized by the *bushiness*  $b = (b_0, \dots, b_{T-1})$ ; cf. (N6), Section 1. As for the discretization points and weights  $(\mathcal{P}, \mathcal{W})$ , it is fruitful to differentiate two ways of generating them: in the first way, for any two nodes  $n, m$  such that  $t(n) = t(m)$ , the points and weights  $(\mathcal{P}^n, \mathcal{W}^n)$  and  $(\mathcal{P}^m, \mathcal{W}^m)$  need *not* be equal. This leads to a scenario tree with a number of scenarios that typically grows exponentially with increase of the number of stages; specifically the number of scenarios equals  $\prod_{t=0}^{T-1} b_t$ . In the second way,  $(\mathcal{P}^n, \mathcal{W}^n)$  and  $(\mathcal{P}^m, \mathcal{W}^m)$  are *equal* whenever  $t(n) = t(m)$ . This leads to a scenario tree with many redundant nodes, i.e., nodes being the roots of identical sub-scenario trees, and hence the tree structure can be recombined without loss of information so that its number of nodes only grows linearly with increase of the number of stages; specifically the number of nodes equals  $1 + \sum_{t=0}^{T-1} b_t$ . Scenario trees of the second kind generally provide a less accurate optimal-value  $\widehat{Q}_0^*$  (see Shapiro et al. [38], Section 5.8.1, where this conclusion is reached for scenario trees filled with Monte Carlo sampling points), but they improve the computational efficiency of the *nested cutting plane* method since they allow *cut sharing* (see Ruszczyński [35], Section 5.3). The latter property is also key in the *stochastic dual dynamic programming* (SDDP) method, which has proved to be very efficient for solving linear MSPP with long time horizon using recombined scenario trees; see, e.g., Shapiro [37]. Whichever the type of scenario tree considered, standard or recombined, the tree structures minimizing the figure of demerit are given by the following proposition:

**Proposition 3.1.** *Consider a semi-linear MSPP with a stagewise independent stochastic process and a set of guidance functions  $\Gamma_{\text{ind}}$ . If  $\mathcal{W}_*$  is normalized, then:*

(a) the symmetrical tree structure of lowest figure of demerit having at most  $N$  scenarios is the one with the bushiness given by the optimal solution of

$$\min_{b=(b_0, \dots, b_{T-1}) \in \mathbb{N}_+^T} \sum_{t=0}^{T-1} f_t(b_t) \gamma_t \quad \text{subject to} \quad \prod_{t=0}^{T-1} b_t \leq N. \quad (25)$$

(b) the recombined tree structure of lowest figure of demerit having at most  $N$  nodes is the one with the bushiness given by the optimal solution of

$$\min_{b=(b_0, \dots, b_{T-1}) \in \mathbb{N}_+^T} \sum_{t=0}^{T-1} f_t(b_t) \gamma_t \quad \text{subject to} \quad \sum_{t=0}^{T-1} b_t \leq N - 1. \quad (26)$$

*Proof.* It follows directly from the equality (23) and the fact that the structure  $\mathcal{T}$  is symmetrical that

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{ind}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)} f_{t(n)} (|C(n)|) \quad (27)$$

$$= \sum_{t=0}^{T-1} f_t(b_t) \gamma_t \sum_{n \in \mathcal{N}_t} W^n \quad (28)$$

$$= \sum_{t=0}^{T-1} f_t(b_t) \gamma_t, \quad (29)$$

where at the last equality we use the fact that  $\mathcal{W}_*$  is normalized, i.e.,  $\sum_{n \in \mathcal{N}_t} W^n = 1$  for any  $t = 0, \dots, T-1$ . This proves the objective function in (a) and (b). The two constraints follow from the discussion above about the number of scenarios and nodes of standard and recombined scenario trees.  $\square$

We show in Figures 2 and 3, Appendix A, the tree structures of lowest figure of demerit in their standard and recombined forms for different choices of  $f_t$  and  $\Gamma_{\text{ind}}$  (because of the exponential growth of the number of scenarios, standard tree structures are displayed with less stages than recombined structures).

### 3.2 Current-stage dependent guidance functions

We consider the *semi-linear* MSPP described in Example 1.1 with the additional assumption that the stochastic process  $(\xi_0, \dots, \xi_T)$  is modeled as

$$\xi_0 = \epsilon_0 \quad \text{and} \quad \xi_t = \varphi_t(\epsilon_{t-1}, \epsilon_t), \quad \text{for every } t = 1, \dots, T, \quad (30)$$

where  $\epsilon_0$  is deterministic and  $\epsilon_1, \dots, \epsilon_T$  are independent random vectors of arbitrary dimensions. It is convenient to work with the stochastic process  $(\epsilon_0, \dots, \epsilon_T)$  instead of  $(\xi_0, \dots, \xi_T)$  as the former is stagewise independent. Note, however, that the MSPP modeled with  $(\epsilon_0, \dots, \epsilon_T)$  is no longer written in the form of a semi-linear problem, hence the results of the previous section do not apply here. In particular, the recourse function at stage  $t$  is now composed with  $\varphi_t$ , and hence the variability with respect to  $\epsilon_t$  of the composite function depends on the realization  $\epsilon_{t-1}$ . Thus, it is relevant to consider guidance functions  $\Gamma_{\text{cs}}$  that depend only on the current-stage realization:

$$\Gamma_{\text{cs}} := \{\gamma_0(\epsilon_0), \gamma_1(\epsilon_1), \dots, \gamma_T(\epsilon_T)\}. \quad (31)$$

The figure of demerit of the scenario tree takes the simplified form

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\epsilon^n) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^n, \mathcal{W}_*^n), \quad (32)$$

where  $\mathcal{P}_* = \{\epsilon^n : n \in \mathcal{N}\}$  and  $\mathcal{W}_* = \{w^n : n \in \mathcal{N}\}$  are the sets of discretization points and weights of  $(\epsilon_0, \dots, \epsilon_T)$  generated by the discretization method of Condition 3. Unlike the setting of the previous section, it is not straightforward to find the tree structure minimizing (32). An algorithmic approach to do so is to enumerate tree structures, minimize the figure of demerit with respect to  $(\mathcal{P}_*, \mathcal{W}_*)$  for each one and retain the one with the lowest figure. The following result provides a necessary and sufficient condition for  $(\mathcal{P}_*, \mathcal{W}_*)$  to be a minimizer of (32) at fixed  $\mathcal{T}$ :

**Proposition 3.2.** *Consider a semi-linear MSPP with a stochastic process modeled as (30) and a set of guidance functions  $\Gamma_{\text{cs}}$ . If  $\mathcal{W}_*$  is standardized, then  $(\mathcal{P}_*, \mathcal{W}_*)$  minimize the figure of demerit (32) at fixed structure  $\mathcal{T}$  if and only if:*

$$|C(m)| \leq |C(n)| \iff \gamma_{t(m)}(\epsilon^m) \leq \gamma_{t(n)}(\epsilon^n), \quad \text{whenever } p(n) = p(m) \notin \mathcal{N}_{T-1}. \quad (33)$$

*Proof.* Under Condition 3, the figure of demerit is simplified as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\epsilon^n) f_{t(n)}(|C(n)|). \quad (34)$$

The equality (23) holds regardless of the way we assign the points  $\mathcal{P}_*^n = \{\epsilon^m : m \in C(n)\}$  to the nodes in  $C(n)$ , hence (34) is minimized by finding at each node  $n \in \mathcal{N} \setminus (\mathcal{N}_T \cup \mathcal{N}_{T-1})$  the optimal permutation of the points  $\{\epsilon^{\sigma(m)} : m \in C(n)\}$  with  $\sigma : C(n) \rightarrow C(n)$ . Using the decomposition

$$\mathcal{N} \setminus \mathcal{N}_T = \{n_0\} \cup \left( \bigcup_{t=0}^{T-2} \bigcup_{n \in \mathcal{N}_t} \bigcup_{m \in C(n)} \{m\} \right), \quad (35)$$

and the equality  $W^m = W^n / |C(n)|$  for all  $m \in C(n)$  (cf. (N3) and (N7), Section 1), we can write the above sum as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \gamma_0(\epsilon^{n_0}) f_0(|C(n_0)|) \quad (36)$$

$$+ \sum_{t=0}^{T-2} \sum_{n \in \mathcal{N}_t} \frac{W^n}{|C(n)|} \sum_{m \in C(n)} \gamma_{t(m)}(\epsilon^{\sigma(m)}) f_{t(m)}(|C(m)|). \quad (37)$$

Thus, for any  $n \in \mathcal{N} \setminus (\mathcal{N}_T \cup \mathcal{N}_{T-1})$ , the optimal permutation  $\sigma^* : C(n) \rightarrow C(n)$  is the one that minimizes

$$\sum_{m \in C(n)} \gamma_{t(m)}(\epsilon^{\sigma(m)}) f_{t(m)}(|C(m)|). \quad (38)$$

It is easy to show that for any  $x_i, y_i \geq 0$ ,  $i = 1, \dots, N$ , the sum  $\sum_{i=1}^N x_{\sigma(i)} y_i$  is minimized by the permutation  $\sigma^* : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$  such that  $x_{\sigma^*(i)} \leq x_{\sigma^*(j)} \iff y_i \geq y_j$  for every  $i, j = 1, \dots, N$ . The assertion (33) follows directly from this and the fact that  $f_{t(m)}(\cdot)$  is monotonically decreasing.  $\square$

The necessary and sufficient condition (33) is quite intuitive: a node  $n$  that has a larger value of  $\gamma_{t(n)}(\epsilon^n)$  leads to a larger variability of the recourse function at stage  $t(n) + 1$ , and therefore should have more children nodes to reduce the integration error at this stage.

The implementation of Proposition 3.2 to find the scenario tree of lowest figure of demerit is done as follows :

**Inputs.**

- Discretization method such that (23) holds.
- Guidance functions  $\Gamma_{cs}$  in the form (31).

**Step 1.** Pick a tree structure  $\mathcal{T}$  and set  $\epsilon^{n_0} := \epsilon_0$  and  $w^{n_0} := 1$ .

**Step 2.** For each stage  $t = 0, \dots, T - 2$  and node  $n \in \mathcal{N}_t$ :

**Step 2.1.** Set  $N := |C(n)|$  and index the nodes  $C(n) = \{m_1, \dots, m_N\}$  such that

$$|C(m_1)| \leq |C(m_2)| \leq \dots \leq |C(m_N)|. \quad (39)$$

**Step 2.2.** Generate  $N$  discretization points  $\{\epsilon^{m_i} : i = 1, \dots, N\}$  of  $\epsilon_{t+1}$  and index the elements such that:

$$\gamma_{t+1}(\epsilon^{m_1}) \leq \gamma_{t+1}(\epsilon^{m_2}) \leq \dots \leq \gamma_{t+1}(\epsilon^{m_N}). \quad (40)$$

**Step 2.3.** Compute (38) for the optimal permutation:

$$v^n := \frac{1}{N} \sum_{i=1}^N \gamma_{t+1}(\epsilon^{m_i}) f_{t+1}(|C(m_i)|). \quad (41)$$

**Step 3.** Compute the figure of demerit using (36):

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{cs}) = \gamma_0(\epsilon^{n_0}) f_0(|C(n_0)|) + \sum_{t=0}^{T-2} \sum_{n \in \mathcal{N}_t} W^n v^n. \quad (42)$$

**Step 4.** If some stopping criteria is fulfilled: go to Step 5; otherwise: go to Step 1.

**Step 5.** Set  $\zeta^{n_0} := \epsilon^{n_0}$  and for each node  $n \in \mathcal{N} \setminus \mathcal{N}_T$ :

- if  $t(n) \leq T - 2$ : set  $\zeta^m := \varphi_{t(n)+1}(\epsilon^n, \epsilon^m)$  for each  $m \in C(n)$ ;
- otherwise: generate  $|C(n)|$  discretization points  $\{\epsilon^m : m \in C(n)\}$  of  $\epsilon_T$  and set  $\zeta^m := \varphi_T(\epsilon^n, \epsilon^m)$  for each  $m \in C(n)$ .

The algorithm starts in Step 1 by picking a tree structure. An exhaustive iteration over all structures is computationally impossible unless the MSPP has a small number of stages and the structure is restricted to a fairly small number of scenarios. A more reasonable approach may consist in exploring the space of tree structures using an heuristic method such as the *variable neighborhood search*; see Mladenović and Hansen [26] and Gendreau and Potvin [8]. In Step 2, the algorithm assigns the discretization points to the appropriate nodes in accordance with Proposition 3.2 to minimize the figure of demerit at fixed structure. The figure of demerit is computed in Step 3 and a stopping criteria is met in Step 4. The algorithm may stop if the improvement of the figure of demerit over the  $k$  previous iterations is smaller than a certain threshold or if a maximum number of iterations is reached. In step 5, the discretization points of the original stochastic process  $(\xi_0, \dots, \xi_T)$  are computed from those of  $(\epsilon_0, \dots, \epsilon_T)$  using (30).

**Example 3.1.** The stochastic modeling (30) can be seen as a first step in generalizing the stagewise independent setting of the previous section. This generalization allows the realization at the previous stage (and only this one) to influence the realization at the current stage. A simple example of such influence is when the previous-stage realization alters the degree of uncertainty of the current-stage random vector, i.e., when it alters the dispersion of the distribution (e.g., the variance) while keeping the location of the distribution the same (e.g., the mean). Consider for example the following stochastic process:

$$\boldsymbol{\xi}_0 = \boldsymbol{\epsilon}_0 \quad \text{and} \quad \boldsymbol{\xi}_t = \begin{cases} \boldsymbol{\epsilon}_t & \text{if } \boldsymbol{\epsilon}_{t-1} \in \Theta_{t-1}, \\ \lambda_t(\boldsymbol{\epsilon}_t) & \text{otherwise,} \end{cases} \quad \text{for every } t = 1, \dots, T, \quad (43)$$

where  $\Theta_{t-1}$  is a subset of possible realizations of  $\boldsymbol{\epsilon}_{t-1}$  and the function  $\lambda_t(\cdot)$  is such that

$$\mathbb{E}[\lambda_t(\boldsymbol{\epsilon}_t)] = \mathbb{E}[\boldsymbol{\epsilon}_t] \quad \text{and} \quad \frac{\text{Var}[\lambda_t(\boldsymbol{\epsilon}_t)]}{\text{Var}[\boldsymbol{\epsilon}_t]} = \beta^2. \quad (44)$$

Let us derive the form of guidance functions that is relevant for such stochastic modeling. Suppose that we want to build a scenario tree that provides a suitable discretization of  $(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_T)$  for the approximation of  $\mathbb{E}[\sum_{t=0}^T \boldsymbol{\xi}_t]$  by a finite sum. By the law of total expectation, this means that the scenario tree must approximate at each stage  $t = 1, \dots, T$  the conditional expectations defined recursively as

$$g_{t-1}(\xi_0, \dots, \xi_{t-1}) := \mathbb{E}[g_t(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_t) | \xi_0, \dots, \xi_{t-1}], \quad (45)$$

where  $g_T(\xi_0, \dots, \xi_T) := \sum_{t=0}^T \xi_t$ . The functions  $g_0, g_1, \dots, g_{T-1}$  have a closed-form given by

$$g_t(\xi_0, \dots, \xi_t) = \sum_{i=0}^t \xi_i + \mathbb{E}[\boldsymbol{\xi}_{t+1} | \xi_t] + \sum_{i=t+2}^T \mathbb{E}[\boldsymbol{\xi}_i], \quad (46)$$

which shows that the difficulty in approximating the right-hand side of (45) lies in the conditional variability of  $\boldsymbol{\xi}_t + \mathbb{E}[\boldsymbol{\xi}_{t+1} | \boldsymbol{\xi}_t]$  given  $\xi_{t-1}$ . A simple measure of variability is given by the standard deviation  $\text{Var}(\cdot)^{1/2}$ , hence we may define the guidance functions as

$$\gamma_{t-1}(\boldsymbol{\epsilon}_{t-1}) = \text{Var}(\boldsymbol{\xi}_t + \mathbb{E}[\boldsymbol{\xi}_{t+1} | \boldsymbol{\xi}_t] | \boldsymbol{\epsilon}_{t-1})^{1/2} \quad (47)$$

$$= \begin{cases} \text{Var}(\boldsymbol{\epsilon}_t)^{1/2} & \text{if } \boldsymbol{\epsilon}_{t-1} \in \Theta_{t-1}, \\ \beta \text{Var}(\boldsymbol{\epsilon}_t)^{1/2} & \text{otherwise.} \end{cases} \quad (48)$$

### 3.3 Long-range dependent guidance functions

We consider the general MSPP with a stochastic process  $(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_T)$  modeled as

$$\boldsymbol{\xi}_0 = \boldsymbol{\epsilon}_0 \quad \text{and} \quad \boldsymbol{\xi}_t = \boldsymbol{\psi}_t(\boldsymbol{\epsilon}_0, \dots, \boldsymbol{\epsilon}_t), \quad \text{for every } t = 1, \dots, T, \quad (49)$$

where  $\boldsymbol{\epsilon}_0$  is deterministic and  $\boldsymbol{\epsilon}_1, \dots, \boldsymbol{\epsilon}_T$  are independent random vectors of arbitrary dimensions. As in the previous section, it is convenient to work with the stochastic process  $(\boldsymbol{\epsilon}_0, \dots, \boldsymbol{\epsilon}_T)$  instead of  $(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_T)$  since the former is stagewise independent. Under the above stochastic modeling, the recourse function at stage  $t$ , which in the general MSPP depends on  $(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_t)$ , is now composed with  $(\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_t)$ , and hence the variability with respect to  $\boldsymbol{\epsilon}_t$  of the composite function depends on all the realizations  $\boldsymbol{\epsilon}_0, \dots, \boldsymbol{\epsilon}_{t-1}$ . Thus, the guidance functions  $\Gamma$  must take into account all previous realizations and the figure of demerit has the general form:

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\boldsymbol{\epsilon}^{\cdot n}) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*, \mathcal{W}_*^n), \quad (50)$$

where  $\mathcal{P}_* = \{\epsilon^n : n \in \mathcal{N}\}$  and  $\mathcal{W}_* = \{w^n : n \in \mathcal{N}\}$  are the points and weights of  $(\epsilon_0, \dots, \epsilon_T)$  generated by the discretization method of Condition 3.

Finding the scenario tree that minimizes the figure of demerit (50) for general  $\Gamma$  is computationally cumbersome, even under the simplification (23), since each  $(\epsilon^n, w^n)$  for  $t(n) \leq T - 2$  appears in exponentially many terms of the sum. However, a systematic and computationally less costly approach to minimizing (16) exists if we consider guidance functions  $\Gamma_{\text{prod}}$  of the *product* form:

$$\gamma_0(\epsilon_0) = 1 \quad \text{and} \quad \gamma_t(\epsilon_0, \dots, \epsilon_t) = \prod_{i=1}^t \rho_i(\epsilon_i), \quad \text{for every } t = 1, \dots, T - 1, \quad (51)$$

for some functions  $\rho_1, \dots, \rho_{T-1} : \mathbb{R}^d \rightarrow [0, \infty)$ . With the above guidance functions, the figure of demerit takes the form

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \left( \prod_{m \in (n_0, n]} \rho_{t(m)}(\epsilon^m) \right) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^n, \mathcal{W}_*^n). \quad (52)$$

This form can be minimized recursively from the tree leaves to the root node, as shown by the following proposition:

**Proposition 3.3.** *Consider a general MSPP with a stochastic process modeled as (49) and a set of guidance functions  $\Gamma_{\text{prod}}$ . The discretization points and weights  $(\mathcal{P}_*, \mathcal{W}_*)$  minimize the figure of demerit (52) at fixed structure  $\mathcal{T}$  if and only if:*

$$\begin{aligned} \mathcal{M}^m(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) &\geq \mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) \\ \iff w^m \rho_{t(m)}(\epsilon^m) &\leq w^n \rho_{t(n)}(\epsilon^n), \quad \text{whenever } p(n) = p(m) \notin \mathcal{N}_{T-1}, \end{aligned} \quad (53)$$

where  $\mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}})$  is the figure of demerit of the sub-scenario tree rooted at node  $n \in \mathcal{N} \setminus \mathcal{N}_T$  defined as

$$\mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) = \frac{1}{W^n} \sum_{m \in \mathcal{N}(n) \setminus \mathcal{N}_T} W^m \left( \prod_{o \in (n, m]} \rho_{t(o)}(\epsilon^o) \right) f_{t(m)}(|C(m)|). \quad (54)$$

*Proof.* It follows from Lemma 3.4(a) in Keutchan et al. [19] that the figures of demerit of the sub-scenario trees can be written recursively (from the tree leaves to the root node) as

$$\mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) = f_{t(n)}(|C(n)|), \quad \text{for all } n \in \mathcal{N}_{T-1}, \quad (55)$$

and, for all  $n \in \mathcal{N} \setminus (\mathcal{N}_{T-1} \cup \mathcal{N}_T)$ :

$$\mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) = f_{t(n)}(|C(n)|) \quad (56)$$

$$+ \sum_{m \in C(n)} w^m \rho_{t(m)}(\epsilon^m) \mathcal{M}^m(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}). \quad (57)$$

Thus, the figure of demerit of the whole scenario tree, which corresponds to  $\mathcal{M}^{n_0}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}})$ , is minimized by finding for each  $n \in \mathcal{N} \setminus (\mathcal{N}_T \cup \mathcal{N}_{T-1})$  the permutation of the points and weights  $\{(\epsilon^{\sigma(m)}, w^{\sigma(m)}) : m \in C(n)\}$ , with  $\sigma : C(n) \rightarrow C(n)$ , that minimizes

$$\sum_{m \in C(n)} w^{\sigma(m)} \rho_{t(m)}(\epsilon^{\sigma(m)}) \mathcal{M}^m(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}). \quad (58)$$

Using the same argument as in the proof of Proposition 3.2, we conclude that the optimal rearrangement of the nodes is the one that satisfies (53).  $\square$

The necessary and sufficient condition (53) has an intuitive meaning similar to that of (33): a node  $n$  that has a larger value of  $w^n \rho_{t(n)}(\epsilon^n)$  leads to a larger variability of the recourse functions at stage  $t(n) + 1$  and at all stages afterwards, as the term  $\rho_{t(n)}(\epsilon^n)$  appears also in the guidance function  $\gamma_{t(m)}(\epsilon^m)$  for every node  $m$  in the sub-scenario tree rooted at  $n$ . Therefore, the node  $n$  should be associated with the sub-scenario tree of lower figure of demerit to reduce the integration error at all subsequent stages.

The implementation of Proposition 3.3 to find the scenario tree of lowest figure of demerit is done as follows (for conciseness we write  $\mathcal{M}^n$  as a shorthand of  $\mathcal{M}^n(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}})$  in the algorithm below):

**Inputs.**

- Discretization method such that (23) holds.
- Guidance functions  $\Gamma_{\text{prod}}$  in the form (51).

**Step 1.** Pick a tree structure  $\mathcal{T}$  and set  $\epsilon^{n_0} := \epsilon_0$  and  $w^{n_0} := 1$ .

**Step 2.** For each node  $n \in \mathcal{N}_{T-1}$ :

**Step 2.1.** Generate  $|C(n)|$  discretization points and weights  $\{(\epsilon^m, w^m) : m \in C(n)\}$  of  $\epsilon_T$ .

**Step 2.2.** Set  $\mathcal{M}^n := f_{t(n)}(|C(n)|)$ .

**Step 3.** For each stage  $t = T - 2, \dots, 0$  (backward iteration) and node  $n \in \mathcal{N}_t$ :

**Step 3.1.** Set  $N := |C(n)|$  and index the nodes  $C(n) = \{m_1, \dots, m_N\}$  such that

$$\mathcal{M}^{m_1} \geq \mathcal{M}^{m_2} \geq \dots \geq \mathcal{M}^{m_N}. \quad (59)$$

**Step 3.2.** Generate  $N$  discretization points and weights  $\{(\epsilon^{m_i}, w^{m_i}) : i = 1, \dots, N\}$  of  $\epsilon_{t+1}$  and index the elements such that:

$$w^{m_1} \rho_{t+1}(\epsilon^{m_1}) \leq w^{m_2} \rho_{t+1}(\epsilon^{m_2}) \leq \dots \leq w^{m_N} \rho_{t+1}(\epsilon^{m_N}). \quad (60)$$

**Step 3.3.** Compute the figure of demerit  $\mathcal{M}^n$  for the optimal permutation using (56):

$$\mathcal{M}^n := f_{t(n)}(|C(n)|) + \sum_{i=1}^N w^{m_i} \rho_{t+1}(\epsilon^{m_i}) \mathcal{M}^{m_i}. \quad (61)$$

**Step 4.** If some stopping criteria is fulfilled: go to Step 5; otherwise: go to Step 1.

**Step 5.** Set  $\zeta^{n_0} := \epsilon^{n_0}$  and  $\zeta^n := \psi_{t(n)}(\epsilon^{\cdot n})$  for each node  $n \in \mathcal{N} \setminus \{n_0\}$ .

Steps 1, 4, and 5 are identical to the corresponding steps in the algorithm of Section 3.2. In Step 2, the algorithm initializes the figures of demerit of the sub-scenario trees rooted at node  $n \in \mathcal{N}_{T-1}$  using the relation (55). In Step 3, it assigns the discretization points and weights to the appropriate nodes according to Proposition 3.3 and compute recursively the figure of demerit of each sub-scenario tree. The last figure computed is  $\mathcal{M}^{n_0}$ , which is the figure of demerit of the whole scenario tree.

**Example 3.2.** Consider a stochastic process  $(\xi_0, \dots, \xi_T)$  that models a *discrete-time geometric Brownian motion*  $\text{GBM}(\mu, \sigma, T)$ :

$$\xi_0 = 1 \quad \text{and} \quad \xi_{t+1} = \xi_t e^{\mu - \frac{\sigma^2}{2} + \sigma(\mathbf{W}_{t+1} - \mathbf{W}_t)}, \quad \text{for every } t = 0, \dots, T-1, \quad (62)$$

where  $(\mathbf{W}_t)_{t \geq 0}$  is a Wiener process, and  $\mu$  (drift) and  $\sigma \geq 0$  (volatility) are constant; see, e.g., Glasserman [9]. This process can be written in the form (49):

$$\xi_0 = \epsilon_0 \quad \text{and} \quad \xi_t = \prod_{i=0}^{t-1} \epsilon_i, \quad \text{with} \quad \epsilon_0 = 1 \quad \text{and} \quad \epsilon_i = e^{\mu - \frac{\sigma^2}{2} + \sigma(\mathbf{W}_i - \mathbf{W}_{i-1})}. \quad (63)$$

Let us show that guidance functions of the product form naturally arise for such stochastic process. Although to specify the set of guidance functions we would need to specify the MSPP we are dealing with, in this example we simply assume that the variability of the recourse functions exactly matches the variability of the stochastic process, and we consider that the latter is measured by the standard deviation. We denote by  $h_t(\xi_t) := \mathbb{E}[\xi_T | \xi_t]$  the function whose conditional expectation given  $(\epsilon_0, \dots, \epsilon_{t-1})$  needs to be computed at each stage  $t = 0, \dots, T-1$ . The difficulty in approximating  $\mathbb{E}[h_t(\xi_t) | \epsilon_0, \dots, \epsilon_{t-1}]$  by a finite sum is directly given by the conditional standard deviation of  $h_t(\xi_t)$  given  $(\epsilon_0, \dots, \epsilon_{t-1})$ , hence we define the guidance functions accordingly:

$$\gamma_{t-1}(\epsilon_0, \dots, \epsilon_{t-1}) := \text{Var}(h_t(\xi_t) | \epsilon_0, \dots, \epsilon_{t-1})^{1/2}, \quad \text{for every } t = 1, \dots, T. \quad (64)$$

Using the fact that for a GBM we have that  $\mathbb{E}[\xi_T | \xi_t] = \xi_t e^{(T-t)\mu}$  and  $\text{Var}(\epsilon_t) = e^{2\mu}(e^{\sigma^2} - 1)$  for all  $t = 1, \dots, T$ , we deduce the following closed-form formula:

$$\gamma_{t-1}(\epsilon_0, \dots, \epsilon_{t-1}) = e^{(T-t+1)\mu} (e^{\sigma^2} - 1)^{1/2} \prod_{i=0}^{t-1} \epsilon_i, \quad \text{for every } t = 1, \dots, T. \quad (65)$$

The product form (51) requires that the guidance function at stage 0 equals one. Since, in practice, the guidance functions are scale-free (cf. discussion after Theorem 2.3), this requirement can be fulfilled by dividing each  $\gamma_t(\cdot)$  by  $\gamma_0(\epsilon_0) = e^{T\mu}(e^{\sigma^2} - 1)^{1/2}\epsilon_0$ . This finally gives the required product form:

$$\gamma_0(\epsilon_0) = 1 \quad \text{and} \quad \gamma_t(\epsilon_0, \dots, \epsilon_t) = \prod_{i=1}^t \frac{\epsilon_i}{e^\mu}, \quad \text{for every } t = 1, \dots, T-1. \quad (66)$$

We show in Figures 4 and 5, Appendix A, two scenario trees of low figure of demerit obtained by employing a *rank-1 lattice rule* and an *optimal quantization method*, respectively; see, e.g., L'Ecuyer and Munger [23] for the former and Pflug and Pichler [33] for the latter. Each scenario tree is computed by the algorithmic procedure described above where the iteration over the tree structures is done heuristically by variable neighborhood search (for this reason the scenario tree obtained may not be the one of *lowest* figure of demerit) and where the neighborhood of a tree structure  $\mathcal{T}$  is defined as all the structures that can be obtained from  $\mathcal{T}$  by splitting a node or merging two nodes. The overall computation takes about two minutes on a Linux machine (Intel Xeon X5472 @ 3.00GHz) with the algorithm implemented in Python 3.6.2. For both discretization methods, we see that the scenario trees have denser branching at nodes where the geometric Brownian motion takes higher values, as it is where the future of the process has more variability.

## 4 Conclusion

Building on the results derived in Keutchayan et al. [19], we have developed a new framework for generating scenario trees that takes into account the structure of the multistage stochastic problem, specifically, the variability of the recourse functions with respect to the random parameters. Variability of the recourse functions is indeed what should drive the generation of efficient scenario trees, as it is where the recourse functions have larger variability that the approximation error tends to be larger. To measure this variability, we have introduced the concept of *guidance functions*, which summarize the relevant information about the problem for the generation of suitable scenario trees. Problems with similar guidance functions are therefore essentially identical as far as scenario-tree generation is concerned. Based on these functions, we have introduced the *figure of demerit* as a general quality criterion for scenario trees. The versatility of this criterion is demonstrated by applying it to different settings of problems. Firstly, in the setting of stagewise independent stochastic processes, where it leads to symmetrical tree structures of suitable bushinesses. Secondly, in the setting of stagewise dependent stochastic processes, where it leads to *non*-symmetrical structures having denser branching where the conditional variability of the recourse functions is higher. The latter was illustrated by the discretization of a geometric Brownian motion.

Future works building on the present paper may address, for instance, the following topics: (i) The introduction of *randomization* techniques, i.e., the use of random discretization points instead of deterministic ones. Besides other benefits related to the improvement of the convergence rate of the discretization method, randomization in scenario-tree generation allows to derive a confidence interval for the optimality gap; see, e.g., Mak et al. [25] and Shapiro et al. [38], Section 5.6. (ii) The search for appropriate function spaces  $\mathcal{F}$  used in Condition 1. Quasi-Monte Carlo methods consider mainly spaces of smooth functions defined on the hypercube  $[0, 1]^d$ , but we know that recourse functions usually have “kinks” and are often defined over the whole space  $\mathbb{R}^d$ . Recent works are going toward the generalization of quasi-Monte Carlo methods to non-smooth functions over general domains; see, e.g., Owen [29], Hartinger and Kainhofer [12], Griebel et al. [11], Griebel et al. [10], and Novak et al. [27]. (iii) The development of exact or heuristic techniques to explore efficiently the space of tree structures in the two algorithms described in Sections 3.2 and 3.3. (iv) The application of the method on real-world problems and the comparison with other scenario-tree generation methods.

## A Figures

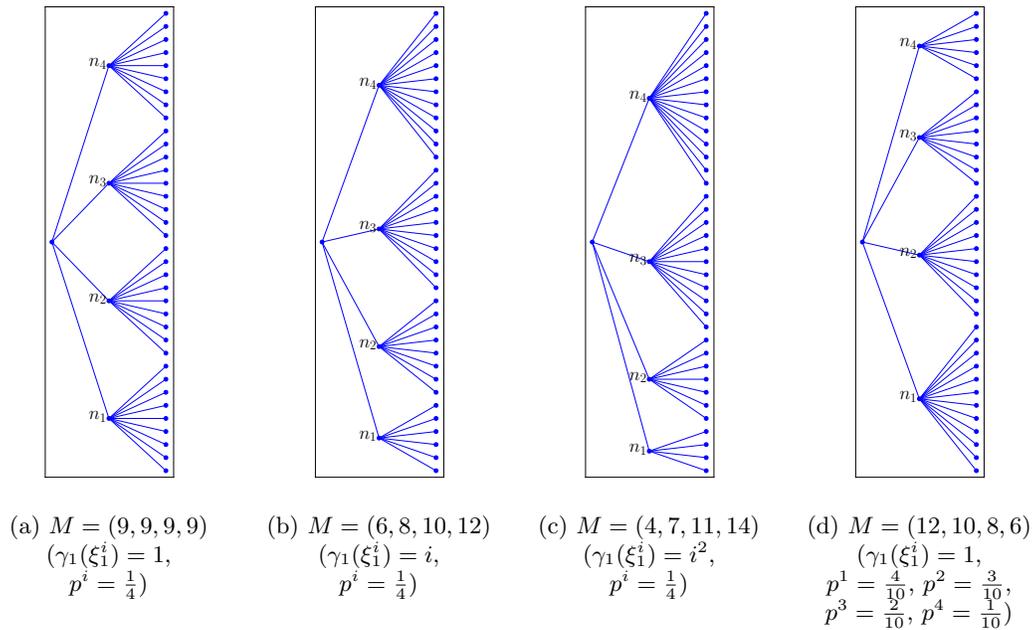


Figure 1: Tree structures given by (22) for  $N = 36$  and  $\alpha = 1$  (cf. Example 2.1).

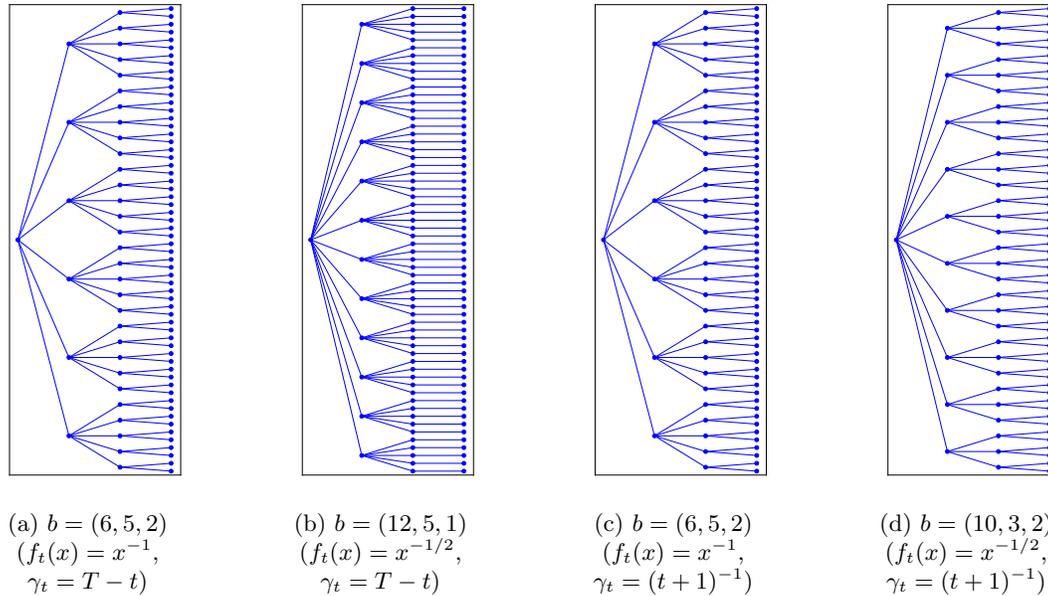
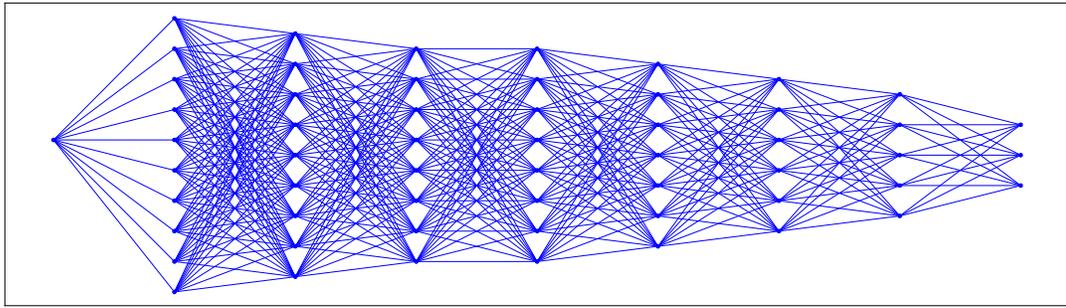
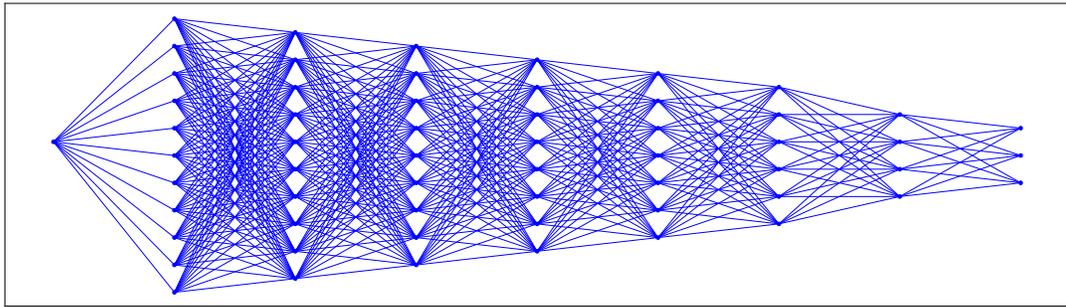


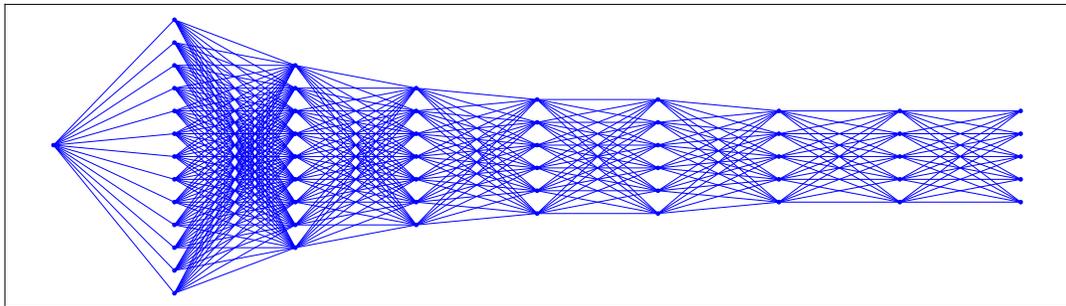
Figure 2: Tree structures given by (25) for  $N = 60$  and  $T = 3$  (cf. the stagewise independent setting of Section 3.1).



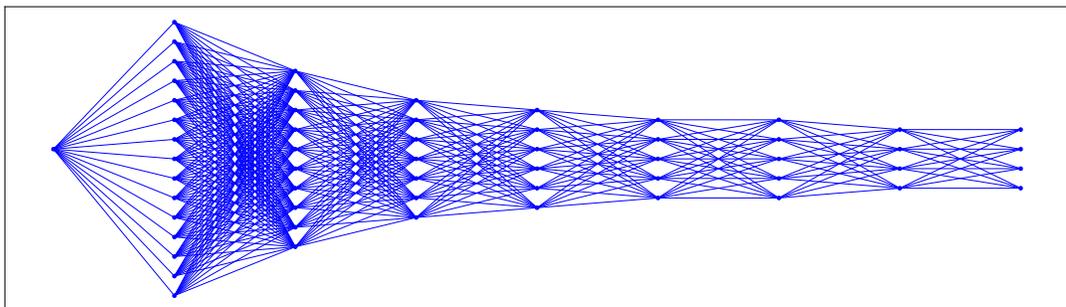
(a)  $b = (10, 9, 8, 8, 7, 6, 5, 3)$  ( $f_t(x) = x^{-1}$  and  $\gamma_t = T - t$ )



(b)  $b = (11, 10, 9, 8, 7, 6, 4, 3)$  ( $f_t(x) = x^{-1/2}$  and  $\gamma_t = T - t$ )

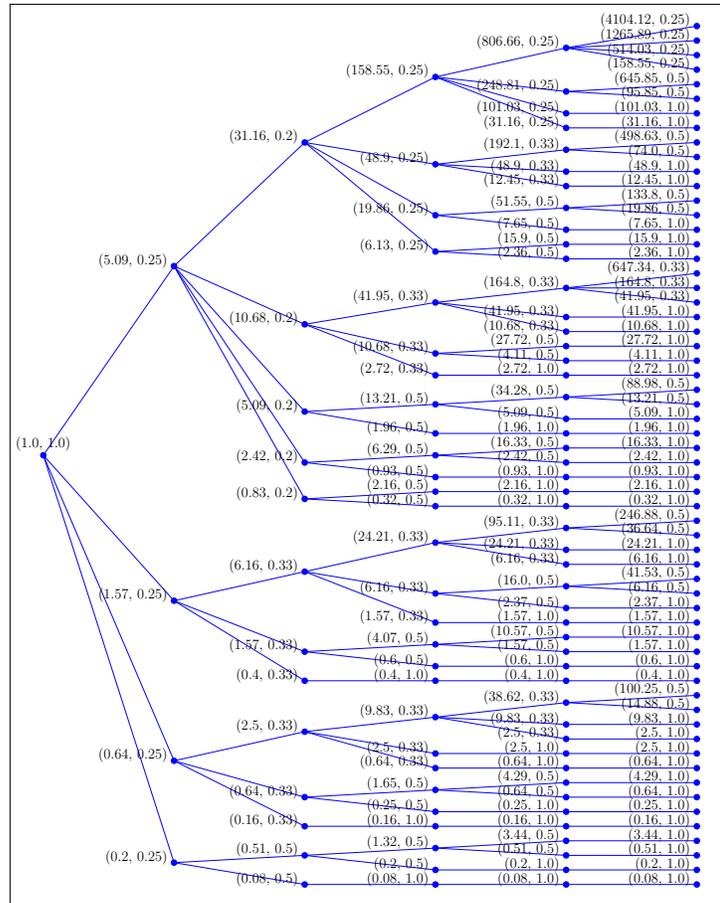


(c)  $b = (13, 9, 7, 6, 6, 5, 5, 5)$  ( $f_t(x) = x^{-1}$  and  $\gamma_t = (t + 1)^{-1}$ )

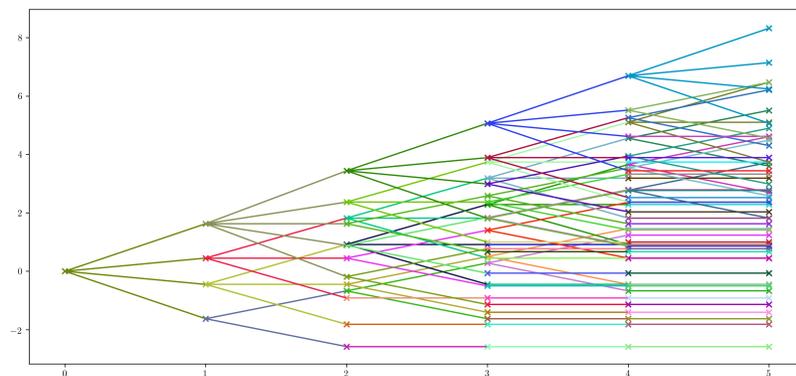


(d)  $b = (15, 10, 7, 6, 5, 5, 4, 4)$  ( $f_t(x) = x^{-1/2}$  and  $\gamma_t = (t + 1)^{-1}$ )

Figure 3: Recombined tree structures given by (26) for  $N = 57$  and  $T = 8$  (cf. the stagewise independent setting of Section 3.1).



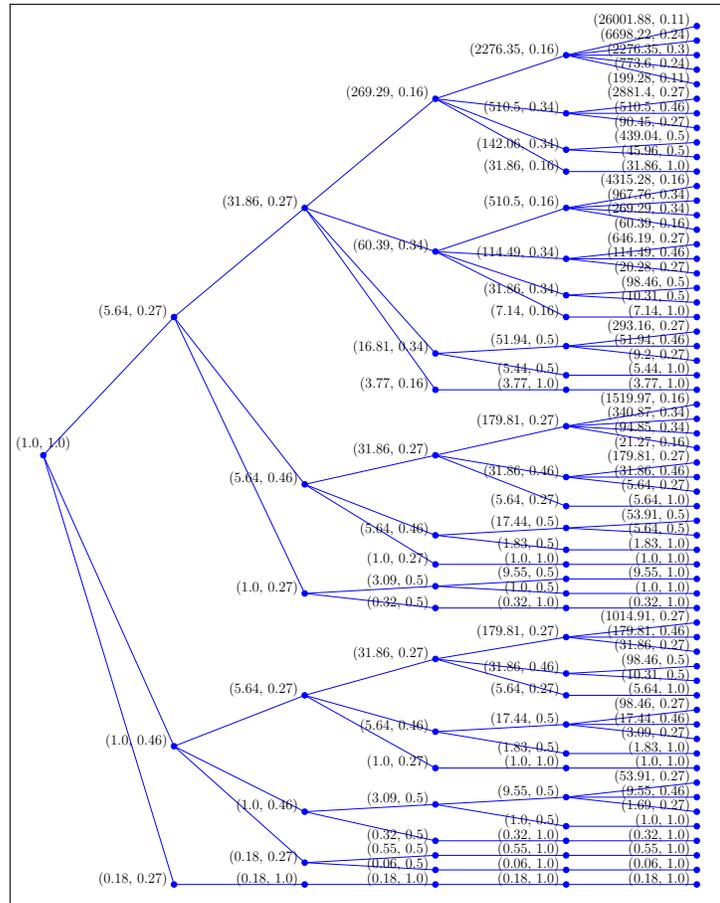
(a) Scenario tree of low figure of demerit for 60 scenarios ( $f_t(x) = x^{-1}$ )



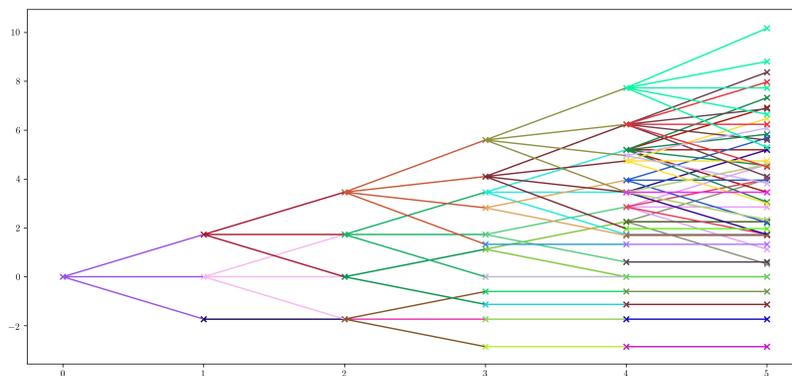
(b) Trajectories of the above scenario tree ( $y$ -axis is in log-scale)

Figure 4: Scenario tree discretization of  $GBM(\mu = 1, \sigma = \sqrt{2}, T = 5)$  (cf. Example 3.2) by a *quasi-Monte Carlo* method, specifically, a *rank-1 lattice rule* in one dimension. Each couple  $(\zeta^n, w^n)$  is displayed next to the corresponding node. The discretization of the standard normal distribution is done by transforming the low discrepancy set of  $N$  points

$$\left\{ \frac{i+0.5}{N} : i = 0, \dots, N-1 \right\} \subseteq [0, 1] \text{ through the inverse normal cumulative } \phi^{-1}: \\ \left\{ \phi^{-1}\left(\frac{i+0.5}{N}\right) : i = 0, \dots, N-1 \right\} \subseteq \mathbb{R}. \text{ The weights are } \textit{standardized} \text{ (cf. (N7), Section 1) as it is} \\ \text{customary in quasi-Monte Carlo methods.}$$



(a) Scenario tree of low figure of demerit for 60 scenarios ( $f_t(x) = x^{-1}$ )



(b) Trajectories of the above scenario tree ( $y$ -axis is in log-scale)

Figure 5: Scenario tree discretization of  $GBM(\mu = 1, \sigma = \sqrt{2}, T = 5)$  (cf. Example 3.2) by an *optimal quantization* method. Each couple  $(\zeta^n, w^n)$  is displayed next to the corresponding node.

The discretization of the standard normal distribution is done by minimizing the Wasserstein distance of order 2 between the standard normal distribution and its approximation sitting on finitely many points; see Algorithm 2 in Pflug and Pichler [33]. This provides the points as well as the *normalized weights* (cf. (N7), Section 1).

## References

- [1] Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media, second edition.
- [2] Davis, P. J. and Rabinowitz, P. (2007). *Methods of numerical integration*. Dover Publications, Inc. Mineola, New York.
- [3] Defourny, B., Ernst, D., and Wehenkel, L. (2011). Multistage stochastic programming: A scenario tree based approach. In Sucar, L. E., Morales, E. F., and Hoey, J., editors, *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions: Concepts and Solutions*, pages 97–143. IGI Global.
- [4] Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288.
- [5] Dick, J. and Pillichshammer, F. (2010). *Digital nets and sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press.
- [6] Dupačová, J., Consigli, G., and Wallace, S. W. (2000). Scenarios for multistage stochastic programs. *Annals of operations research*, 100(1-4):25–53.
- [7] Dupačová, J., Gröwe-Kuska, N., and Römisch, W. (2003). Scenario reduction in stochastic programming. *Mathematical programming*, 95(3):493–511.
- [8] Gendreau, M. and Potvin, J.-Y., editors (2010). *Handbook of metaheuristics*, volume 2. Springer.
- [9] Glasserman, P. (2013). *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media.
- [10] Griebel, M., Kuo, F., and Sloan, I. (2013). The smoothing effect of integration in  $\mathbb{R}^d$  and the ANOVA decomposition. *Mathematics of Computation*, 82(281):383–400.
- [11] Griebel, M., Kuo, F. Y., and Sloan, I. H. (2010). The smoothing effect of the ANOVA decomposition. *Journal of Complexity*, 26(5):523–551.
- [12] Hartinger, J. and Kainhofer, R. (2006). Non-uniform low-discrepancy sequence generation and integration of singular integrands. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 163–179. Springer.
- [13] Heitsch, H., Leövey, H., and Römisch, W. (2016). Are Quasi-Monte Carlo algorithms efficient for two-stage stochastic programs? *Computational Optimization and Applications*, 65(3):567–603.
- [14] Heitsch, H. and Römisch, W. (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406.
- [15] Hickernell, F. J., Sloan, I. H., and Wasilkowski, G. W. (2004). On tractability of weighted integration for certain banach spaces of functions. In Niederreiter, H., editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 51–71. Springer, Heidelberg.
- [16] Hilli, P. and Pennanen, T. (2008). Numerical study of discretizations of multistage stochastic programs. *Kybernetika*, 44(2):185–204.

- [17] Kaut, M. and Wallace, S. W. (2007). Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271.
- [18] Keutchayan, J., Gendreau, M., and Saucier, A. (2017a). Quality evaluation of scenario-tree generation methods for solving stochastic programming problems. *Computational Management Science*, 14(3):333–365.
- [19] Keutchayan, J., Munger, D., and Gendreau, M. (2017b). On the scenario-tree optimal-value error for stochastic programming problems. *CIRRELT preprint: CIRRELT-2017-73*, <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2017-73.pdf>.
- [20] King, A. J. and Wallace, S. W. (2012). *Modeling with stochastic programming*. Springer Science & Business Media.
- [21] Kovacevic, R. M., Pflug, G. C., and Vespucci, M. T., editors (2013). *Handbook of risk management in energy production and trading*. Springer.
- [22] Kuo, F. Y., Schwab, C., and Sloan, I. H. (2011). Quasi-Monte Carlo methods for high-dimensional integration: the standard (weighted Hilbert space) setting and beyond. *The ANZIAM Journal*, 53(1):1–37.
- [23] L’Ecuyer, P. and Munger, D. (2016). Algorithm 958: Lattice builder: A general software tool for constructing rank-1 lattice rules. *ACM Transactions on Mathematical Software*, 42(2):15:1–15:30.
- [24] Louveaux, F. (1998). An introduction to stochastic transportation models. In Labbé, M., Laporte, G., Tanczos, K., and Toint, P., editors, *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, pages 244–263. Springer.
- [25] Mak, W.-K., Morton, D. P., and Wood, R. K. (1999). Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1):47–56.
- [26] Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100.
- [27] Novak, E., Ullrich, M., Woźniakowski, H., and Zhang, S. (2017). Reproducing kernels of Sobolev spaces on  $\mathbb{R}^d$  and applications to embedding constants and tractability. *arXiv preprint arXiv:1709.02568*.
- [28] Novak, E. and Wozniakowski, H. (2010). Tractability of multivariate problems. *Volume II: Standard Information for Functionals*, EMS, Zürich.
- [29] Owen, A. B. (2006). Quasi-Monte Carlo for integrands with point singularities at unknown locations. In Niederreiter, H. and Talay, D., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 403–417. Springer.
- [30] Pagès, G., Pham, H., and Printems, J. (2004). Optimal quantization methods and applications to numerical problems in finance. In Rachev, S. T., editor, *Handbook of Computational and Numerical Methods in Finance*, pages 253–297. Birkhäuser Boston, Boston, MA.
- [31] Pflug, G. C. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical programming*, 89(2):251–271.

- [32] Pflug, G. C. and Pichler, A. (2011). Approximations for probability distributions and stochastic optimization problems. In Bertocchi, M., Consigli, G., and Dempster, M. A. H., editors, *Stochastic optimization methods in finance and energy: New financial products and energy market strategies*, pages 343–387. Springer.
- [33] Pflug, G. C. and Pichler, A. (2015). Dynamic generation of scenario trees. *Computational Optimization and Applications*, 62(3):641–668.
- [34] Powell, W. B. and Topaloglu, H. (2003). Stochastic programming in transportation and logistics. In Ruszczyński, A. and Shapiro, A., editors, *Handbooks in operations research and management science: Stochastic Programming*, volume 10, pages 555–635. Elsevier.
- [35] Ruszczyński, A. (2003). Decomposition methods. In Ruszczyński, A. and Shapiro, A., editors, *Handbooks in operations research and management science*, volume 10, pages 141–211. Elsevier.
- [36] Ruszczyński, A. and Shapiro, A., editors (2003). *Handbooks in operations research and management science*, volume 10. Elsevier.
- [37] Shapiro, A. (2011). Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72.
- [38] Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2014). *Lectures on stochastic programming: modeling and theory*, volume 16. SIAM.
- [39] Sloan, I. H., Wang, X., and Woźniakowski, H. (2004). Finite-order weights imply tractability of multivariate integration. *Journal of Complexity*, 20(1):46–74.
- [40] Sloan, I. H. and Woźniakowski, H. (1998). When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *Journal of Complexity*, 14(1):1–33.
- [41] Sobol, I. M. (1990). Quasi-Monte Carlo methods. *Progress in Nuclear Energy*, 24(1-3):55–61.
- [42] Wallace, S. W. and Fleten, S.-E. (2003). Stochastic programming models in energy. In Ruszczyński, A. and Shapiro, A., editors, *Handbooks in operations research and management science: Stochastic Programming*, volume 10, pages 637–677. Elsevier.
- [43] Yu, L.-Y., Ji, X.-D., and Wang, S.-Y. (2003). Stochastic programming models in financial optimization: A survey. *Advanced Modeling and Optimization*, 5(1):1–26.
- [44] Ziemba, W. T. (2003). *The stochastic programming approach to asset, liability, and wealth management*. AIMR, Charlottesville, VA.