UNIVERSITÉ DE MONTRÉAL

APPROXIMATION D'ESPÉRANCES CONDITIONNELLES GUIDÉE PAR LE PROBLÈME EN OPTIMISATION STOCHASTIQUE MULTI-ÉTAPES

JULIEN KEUTCHAYAN
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(MATHÉMATIQUES DE L'INGÉNIEUR)
DÉCEMBRE 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

APPROXIMATION D'ESPÉRANCES CONDITIONNELLES GUIDÉE PAR LE PROBLÈME EN OPTIMISATION STOCHASTIQUE MULTI-ÉTAPES

présentée par : KEUTCHAYAN Julien
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. SOUMIS François, Ph. D., président
M. GENDREAU Michel, Ph. D., membre et directeur de recherche
M. BASTIN Fabian, Doct., membre et codirecteur de recherche
M. REI Walter, Ph. D., membre
M. PFLUG Georg, Ph. D., membre externe

DÉDICACE

À mes parents et mon petit frère
À la mémoire de mes grands-parents

REMERCIEMENTS

Je tiens en premier lieu à remercier mon directeur de thèse, Michel Gendreau, avec qui j'ai commencé à travailler dans le cadre du projet de maîtrise et qui m'a fait confiance pour poursuivre ce travail en doctorat. Je remercie également mon codirecteur, Fabian Bastin. Leur expertise scientifique et leurs encouragements m'ont permis de mener à bien cette thèse. Je remercie aussi mes coauteurs : David Munger, sans qui ce travail n'aurait pas été possible, et Antoine Saucier pour son aide précieuse dans la rédaction de mon premier article.

À mon arrivée à Montréal en 2013, il y a cinq ans, je n'aurais jamais pu m'imaginer à quel point ce séjour –initialement prévu pour durer deux ans– deviendrait une expérience aussi enrichissante culturellement. Pendant ces cinq années passées au CIRRELT, j'ai eu la chance de côtoyer quotidiennement des étudiants du monde entier. Tous les citer serait impossible et n'en citer que quelques-uns serait injuste. Donc à tous ceux qui ont rendu cette expérience Montréalaise aussi riche : Merci !

Je me dois aussi de remercier mes amis de Polytechnique Paris (l'X) qui ont joué un rôle prépondérant dans ma décision de m'orienter vers la recherche en mathématiques appliquées plutôt qu'en physique. Ces derniers se reconnaîtront et savent à quel point leur rencontre, un certain été de l'année 2010, a été déterminante.

Je n'oublie pas non plus mes amis de l'UFR de physique de l'Université Joseph Fourier. J'ai vécu à leur côtés trois magnifiques années de Licence à étudier intensément les théories de la mécanique quantique, relativité restreinte, physique statistique, etc., lesquelles vous font voir le monde différemment. Ces années ont eu une importance considérable car elles ont marqué le début d'un apprentissage synonyme d'épanouissement personnel. Ce fut le point de départ de ce parcours qui m'a finalement amené à compléter ce doctorat.

Si je devais remonter encore plus loin (avant l'université), ce serait pour remercier ce professeur de mathématiques qui en classe de 5ème nous avait parlé du paradoxe des jumeaux de Langevin. Sans le savoir, cet enseignant venait de bouleverser l'esprit d'un jeune écolier, qui plusieurs années plus tard s'inscrira à l'université avec pour but de comprendre enfin ce paradoxe qui l'avait émerveillé plus jeune.

Pour finir je remercie aussi mes parents. Outre le fait que je leur dois évidemment tout, ils ont toujours fait en sorte de satisfaire ma curiosité en mettant à ma disposition tout le matériel nécessaire à mon épanouissement.

RÉSUMÉ

Dans cette thèse, nous considérons d'une façon générale la résolution de problèmes d'optimisation stochastique multi-étapes. Ces derniers apparaissent dans de nombreux domaines d'application tels que la finance, l'énergie, la logistique, le transport, la santé, etc. Ils sont généralement insolubles de façon exacte car ils contiennent des espérances mathématiques qui ne peuvent pas être calculées analytiquement. Il est donc nécessaire de considérer pour cela des méthodes numériques. Nous nous intéressons particulièrement aux méthodes de génération d'arbres de scénarios. Ceux-ci remplacent le processus stochastique sous-jacent au problème afin de ramener ce dernier à une taille raisonnable permettant sa résolution pratique. Numériquement, cela permet de remplacer les opérateurs d'espérance qui apparaissent dans la formulation originale du problème (et qui tiennent compte de toutes les scénarios possibles en les pondérant avec une certaine densité de probabilité), par des sommes finies qui, pour leur part, ne prennent en compte qu'un sous-ensemble de scénarios seulement. Cette approximation permet ensuite à un ordinateur de résoudre le problème discrétisé à l'aide de solveurs classiques d'optimisation.

L'arbre de scénarios doit satisfaire un compromis entre la qualité d'approximation, qui voudrait que l'arbre soit le plus grand possible, et la complexité de résolution du problème discrétisé qui, à l'inverse, voudrait qu'il soit le plus petit possible. Alors que ce compromis est relativement facile à satisfaire pour les problèmes à deux étapes, il l'est beaucoup moins pour les problèmes multi-étapes (c.-à-d. à partir de trois étapes). Ceci est dû à la nécessité de considérer des structures d'arbres dont la taille (le nombre de nœuds) croît exponentiellement avec le nombre d'étapes. Dans ce contexte multi-étapes, la recherche d'un compromis satisfaisant entre qualité et complexité a mené la communauté d'optimisation stochastique à développer de nombreuses approches de génération d'arbres de scénarios basées sur des justifications théoriques ou pratiques différentes. Ces justifications portent essentiellement sur la qualité d'approximation du processus stochastique par l'arbre de scénarios. Pour cette raison, ces approches sont dites *guidées par la distribution*, étant donné qu'elles souhaitent reproduire le mieux possible –suivant leur propre critère de qualité– la distribution du processus stochastique (ou certaines propriétés de celle-ci).

Prendre en compte la distribution permet sous certaines conditions assez faibles d'assurer la consistance de la méthode de résolution. Pour cette raison, ces méthodes sont utilisées avec succès dans de nombreux problèmes. Cependant, cette stratégie ne permet pas de tirer profit de la structure même du problème d'optimisation, par exemple la variabilité de sa fonction

objectif ou l'influence de ses contraintes, qui joue aussi un rôle important dans la qualité d'approximation. La prise en compte de ces caractéristiques permettrait de construire des arbres de scénarios plus adaptés aux problèmes et ainsi de satisfaire un meilleur compromis entre qualité et complexité. En pratique, cela permettrait de pouvoir résoudre des problèmes avec un plus grand nombre d'étapes.

Dans cette thèse, nous développons une nouvelle approche de génération d'arbres de scénarios *guidée par le problème*. Celle-ci tient compte de toutes les caractéristiques d'un problème d'optimisation stochastique multi-étapes, à savoir, son processus stochastique, sa fonction de revenus (ou de coûts) et ses ensembles de contraintes. Le développement est fait dans un cadre théorique général qui couvre une grande partie des problèmes d'intérêt, mais sans se restreindre à une application ou un domaine d'applications particulier. Les conditions que nous imposons sur la forme de la fonction de revenus, des contraintes ou de la distribution de probabilité visent essentiellement à assurer que le problème d'optimisation multi-étapes est bien défini mathématiquement.

Le travail réalisé dans cette thèse se décompose en quatre contributions principales (une par article). Dans le premier article, nous ne considérons pas la question de la génération d'arbres de scénarios à proprement parler mais plutôt celle de la mise en pratique des décisions obtenues par arbres de scénarios. Cette question est d'une importance majeure car les décisions de l'arbre ne sont données que pour certains scénarios. Par conséquent, elles ne fournissent pas une politique de décision pour le problème original. Pour répondre à cela, nous développons un cadre mathématique qui permet d'étendre les décisions de l'arbre sur l'ensemble des réalisations des paramètres aléatoires afin d'obtenir une politique de décision candidate à une utilisation pratique. Nous introduisons plusieurs critères qui permettent de comparer la qualité de ces politiques afin de choisir la méthode de génération d'arbres de scénarios qui produit les meilleures décisions pour un problème donné. Nous montrons ensuite sur deux problèmes pratiques que ces procédures d'extension peuvent être utilisées avec succès pour trouver la meilleure méthode et pour construire une politique proche de l'optimalité.

La deuxième contribution fournit la base de notre nouvelle méthode de génération d'arbres de scénarios. Nous développons deux résultats théoriques portant sur l'analyse de l'erreur de valeur optimale. Le premier décompose celle-ci en une somme d'erreurs de discrétisation et d'optimisation commises à chaque nœud de l'arbre. Le deuxième fournit un résultat similaire à la différence que l'erreur est cette fois-ci bornée supérieurement par une somme d'erreurs de discrétisation *pire-cas*. L'erreur d'optimisation n'apparait plus explicitement dans la borne, mais elle est implicite dans la définition du pire cas. Ce deuxième résultat est extrêmement informatif au sujet de la qualité de l'approximation par arbres de scénarios car il permet de

relier l'erreur de valeur optimale aux erreurs d'intégration des fonctions de recours à chaque nœud de l'arbre. Ces dernières étant elles-mêmes reliées à la variabilité des fonctions, nous déduisons qu'une caractéristique essentielle pour guider la génération des arbres de scénarios est la variabilité des fonctions de recours à travers les étapes et les scénarios. En se basant sur cette constatation, nous développons un premier exemple de procédure algorithmique qui tire profit de la donnée de variabilité pour construire des arbres aux branchements hétérogènes (non-symétriques) potentiellement plus adaptés aux problèmes.

La troisième contribution se base sur les résultats de la deuxième pour construire le cadre mathématique formel de la nouvelle approche. Nous introduisons deux nouvelles notions en optimisation stochastique multi-étapes. La première, notée Γ, permet de caractériser les propriétés pertinentes du problème liées à la distribution de variabilité de ses fonctions de recours. La deuxième, $\mathcal{M}$, est complémentaire de la première, elle permet de mesurer l'adéquation entre un arbre de scénarios et un problème multi-étapes. L'arbre de scénarios est caractérisé par sa structure et ses ensembles de points et de poids. Nous montrons que la mesure $\mathcal{M}$ est une borne supérieure sur l'erreur de valeur optimale et, par conséquent, celle-ci doit être minimisée pour produire le meilleur arbre. Nous développons trois procédures de minimisation pour des types de Γ différents. Dans les trois cas, les arbres de scénarios obtenus (qui peuvent être *recombinants*) ont de branchements hétérogènes à travers les étapes et les scénarios de sorte à tirer profit des différences de variabilité représentées par Γ.

La quatrième contribution met en pratique la nouvelle méthode dans le cas d'un problème d'évaluation d'options asiatiques. Une caractéristique importante de ces options est le fait que leur revenu dépend de tout l'historique du prix de l'actif sous-jacent à travers sa valeur moyenne (arithmétique). A mesure que l'option s'approche de la date de maturité, le nombre de termes dans la moyenne augmente. Ainsi, celle-ci est moins sujette à variations, ce qui a pour conséquence de réduire la variabilité de son prix. Cette décroissance de variabilité en fonction du temps est exploitée par notre méthode qui construit des arbres de scénarios avec des facteurs de branchement décroissant à travers les étapes de sorte à réduire l'erreur d'estimation. Considérer des branchements décroissants n'est pas nouveau en soit, mais jusqu'à présent cela était fait essentiellement de façon empirique. Dans notre approche, ils sont précisément déterminés par les propriétés de la fonction de revenus du problème. De plus, dans notre cas ils varient aussi à travers les nœuds d'une même étape (en fonction du prix de l'actif sous-jacent à un instant donné), ce qui permet de considérer des facteurs de branchement *fractionnaires*, par opposition aux facteurs de branchement restreints à des valeurs entières. Les expériences numériques démontrent la validité de notre approche à différents niveaux. Tout d'abord, nous montrons que la mesure $\mathcal{M}$ est corrélée de façon positive à l'erreur d'estimation avec un facteur de corrélation extrêmement proche de un. Cela confirme expé-

rimentalement l'idée générale de minimiser $\mathcal{M}$ pour trouver le meilleur arbre. Ensuite, nous observons que les arbres de scénarios obtenus en minimisant $\mathcal{M}$ ont des taux de convergence de l'erreur significativement supérieurs aux arbres à structures symétriques. Ces derniers approximent convenablement le processus mais ignorent la propriété de décroissance de la variabilité des revenus, ce qui confirme l'intérêt de considérer une approche guidée par la problème plutôt que par la distribution du processus. Dernièrement, l'analyse des taux de convergence démontre que l'avantage de nos arbres de scénarios a tendance à augmenter avec le nombre d'étapes. En particulier, pour les instances à 13 étapes, les arbres minimisant $\mathcal{M}$ exhibent une propriété de réduction du nombre d'étapes effectif du problème. Cette réduction, de l'ordre de 2 à 3 étapes, n'est pas observée pour les arbres symétriques.

ABSTRACT

In this thesis, we consider solution methods for general multistage stochastic optimization problems. Such problems arise in many fields of application, including finance, energy, logistic, transportation, health care, etc. They generally do not have closed-form solutions since they feature mathematical expectations, which cannot be computed exactly in most applications. For this reason, it is necessary to consider solutions through numerical methods. One of them, which is the focus of this thesis, is the scenario-tree generation approach. Its aim is to substitute the underlying stochastic process with a finite subset of scenarios so as to replace the conditional expectations with their finite sum estimators. This reduces the size of the problem, which is then solved using some generic optimization solvers.

The generation of scenario trees is subject to a trade-off between the approximation accuracy and the complexity of the resulting discretized problem. The former tends to increase the number of scenarios, whereas the latter tends to decrease it. This trade-off turns out to be fairly easy to satisfy when dealing with two-stage problems. However, it becomes much more difficult when problems are multistage, that is, when they have 3 stages of more. This stems from the fact that multistage problems require specific tree structures whose size (the number of nodes) grow exponentially as the number of stages increases. For this reason, a lot of attention has been drawn on generating scenario trees in the multistage setting. Many methods have been developed based on different theoretical or practical grounds. Most of them can be described as *distribution-driven*, as they aim at approximating the distribution of the stochastic process (or some features of it), according to their own idea of what a good approximation is.

The distribution-driven strategy allows to have consistent scenario-tree estimators under some weak conditions. For this reason, these methods have been successfully applied to many problems. However, it does not allow to capitalize on some specific features of the multistage problem (e.g., the variability of its revenue function or the influence of its constraints), although they play an important role in the scenario-tree approximation quality as well. Taking them into account would lead to more suitable scenario trees that may satisfy a better trade-off between accuracy and complexity. This, in turn, may allow to consider problems with more stages.

In this thesis, we introduce a new *problem-driven* scenario-tree generation approach. It takes into account the whole structure of the optimization problem through its stochastic process, revenue (or cost) function and sets of constraints. This approach is developed in a general

setting of multistage problems, hence it is not tied to a particular application or field of applications. The conditions that are introduced along the lines of this thesis about the revenue function, constraints, or probability distribution essentially aims at making sure that the problems is mathematically well-defined.

Our work can be decomposed into four main contributions (one for each article). In the first article, we do not yet consider the question of how to generate scenario trees, but rather of how to implement the decisions computed from the scenario tree. This question is of prime importance because these decisions are given for a subset of scenarios only. Thus, they do not provide a decision policy for the original problem as such. To circumvent this issue, we introduce a mathematical framework based on the concept of extending the tree's decisions over the whole set of realizations of the random parameters in order to obtain a candidate policy for the original problem. We introduce three criteria to compare the quality of these policies and to choose the best scenario-tree generation method for a given problem. We show on two numerical examples that these so-called extension procedures allow to single out the best method, which leads to an implementable policy close to optimality.

The second contribution provides the cornerstone of our scenario-tree generation method. We develop two theoretical results about the optimal-value error. The first one is an exact decomposition of the error as a sum of discretization and optimization errors made at each node of the tree. In the second one, the optimal-value error is bounded from above by a sum of *worst-case* discretization errors. The optimization errors no longer appear explicitly in the bound but only implicitly through the definition of worst case. The second result is highly insightful to guide the scenario-tree generation as it connects the optimal-value error to the discretization error of the recourse functions at each node. Since the latter error are connected to the variability of the recourse functions, we deduce that an important feature of the problem (as far as scenario-tree generation is concerned) is the variability of the recourse functions across the stages and scenarios. Based on this observation, we develop a first algorithmic procedure that leverages on the distribution of variability to build non-symmetrical scenario trees better suited to problems.

The third contribution builds upon the second one to develop a complete mathematical framework for this new approach. It introduces two new concepts in stochastic optimization. The first one, represented by Γ, gathers all relevant information about the problem linked to the distribution of variability of its recourse functions. The second one, $\mathcal{M}$, measures the suitability between a scenario tree and a multistage problem. The scenario tree is characterized by its structure and its set of points and weights. We show that the measure $\mathcal{M}$ is an upper bound on the optimal-value error, thus it must be minimized to produce a suitable

scenario tree. We develop three minimization procedures for different types of Γ. In all three cases, the output scenario trees (which can be *recombining*) have heterogeneous branching across the stages and scenarios in order to take advantage of the heterogeneous variability described by Γ.

The fourth contribution illustrates the new approach in a problem of pricing Asian options. An important feature of these options is the fact that their payoff depends on the whole price history of the underlying asset through its arithmetic average. As the option contract gets closer to the maturity date, the number of terms in the average increases. This make the average less volatile, and hence the option price becomes less volatile too. This sharp decrease of variability over the time is exploited by our method, which generate scenario trees with decreasing branching factors accordingly. Unlike the literature, where branching factors are essentially chosen empirically, our approach determines them precisely from the properties of the revenue function. Furthermore, in our case they also vary across the node at any given stage (based on the asset price at that stage), which allows to consider a *fractional* bushiness, as opposed to the integer bushinesses where the branching factors are restricted to integers. Numerical experiments demonstrate the validity of our approach on different levels. First, we show that the measure $\mathcal{M}$ is positively correlated to the estimation error with a correlation coefficient extremely close to one. This confirms the general idea of minimizing $\mathcal{M}$ in order to compute the best tree. Second, we observe that the scenario trees obtained by minimizing $\mathcal{M}$ have significantly higher convergence rates than symmetrical scenario trees. The latter trees approximate the stochastic process but ignore the sharp decrease of variability of the Asian option pricing problem. This demonstrates the relevance of considering a problem-driven approach rather than an approach purely driven by the distribution. Lastly, we observe that the rates of convergence typically increase with the number of stages. In particular, for the instances with 13 stages, the scenario trees minimizing $\mathcal{M}$ induce a reduction of the effective number of stages of the problem. This reduction, of about 2 to 3 stages, do not occur with symmetrical trees.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

D-OSM	Formulation dynamique de l'optimisation stochastique multi-étapes
D-PDM	Formulation dynamique des processus de décision Markoviens
EP	Extension procedure
EPS	Échantillonnage préférentiel séquentiel
EVPI	Expected value of perfect information
F-OSM	Formulation fonctionnelle de l'optimisation stochastique multi-étapes
F-PDM	Formulation fonctionnelle des processus de décision Markoviens
GAP	Gestion actif-passif
MC	Monte Carlo
MSOP	Multistage stochastic optimization problem
OSM	Optimisation stochastique multi-étapes
OQ	Optimal quantization
PDM	Processus de décision Markovien
QMC	Quasi-Monte Carlo
RQMC	Randomized quasi-Monte Carlo
STGM	Scenario-tree generation method
STAP	Scenario-tree approximate problem

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Définitions et concepts de base

L'optimisation sous incertitude (aussi appelée optimisation *stochastique*) est un cadre mathématique qui permet la modélisation et la résolution de problèmes où un agent cherche à prendre des décisions optimales dans un environnement incertain. Contrairement à l'optimisation *déterministe*, où tous les paramètres du problème sont connus et de valeurs fixes, l'optimisation sous incertitude considère que certains paramètres sont incertains du point de vue de l'agent décideur. Ceux-ci sont alors modélisés par des variables aléatoires. Ainsi, le caractère stochastique rajoute une difficulté supplémentaire dans la prise de décisions puisque l'effet des décisions ne peut pas être connu avec certitude (avec une probabilité égale à un). Cette augmentation de la difficulté du problème est cependant nécessaire puisque la plupart des situations réelles incluent naturellement de l'incertitude.

En pratique, l'incertitude peut avoir plusieurs origines : elle peut être inhérente au problème, comme l'évolution des valeurs mobilières en finance ou des séries de précipitations et de températures en météorologie dont il est impossible de prévoir le comportement à court ou moyen terme ; elle peut être due à un accès limité à l'information, comme c'est le cas par exemple si l'information est payante ; ou simplement due à des erreurs de mesure sur des paramètres a priori déterministes. Quelle que soit l'origine, il est établi que de nombreux problèmes d'optimisation doivent prendre en compte l'incertitude car celle-ci modélise le *risque* auquel l'agent décideur fait face. Ne pas prendre en compte ce risque peut mener à des décisions aux conséquences humaines et économiques désastreuses. Le risque est d'autant plus grand que les distributions de probabilité des paramètres aléatoires ont une grande dispersion autour de leur moyenne, ou s'ils incluent des événements rares et extrêmes (les *événements signes noirs*[1]).

L'optimisation stochastique a des applications dans de nombreux domaines. Par exemple, en finance, une banque veut optimiser sa stratégie d'investissement dans un environnement où les prix des marchés financiers sont aléatoires ; une compagnie d'assurance fait aussi face à l'incertitude des accidents ou sinistres qu'elle couvre. Dans le domaine de l'énergie, un producteur d'électricité veut optimiser sa production et sa distribution dans un environnement où la demande de ses clients est incertaine ; si l'électricité est produite à partir de panneaux solaires, d'éoliennes ou de barrages, l'incertitude porte aussi sur les conditions naturelles et

[1] Taleb (2007)

météorologiques qui influent la production. Dans le domaine de la logistique, un fournisseur de biens veut optimiser sa gestion d'inventaire dans un environnement où l'approvisionnement et la demande sont aléatoires. De nombreuses autres applications existent en finance, énergie et logistique, ainsi que dans d'autres domaines tels que le transport, la télécommunication, l'agriculture, l'environnement, la santé, etc.

Quel que soit le domaine d'application, l'optimisation stochastique considère que l'aléatoire de l'environnement est décrit par un modèle probabiliste où les distributions de probabilité des paramètres aléatoires sont connues. Ces distributions sont généralement estimées par des outils statistiques à partir de données historiques. En cela, l'optimisation stochastique se distingue de l'optimisation *robuste*, qui considère que les paramètres incertains appartiennent à certains ensembles d'incertitude sans supposer de distribution sous-jacente. Elle se distingue aussi d'une approche comme la programmation dynamique *approchée* (ou apprentissage *par renforcement*) qui considère que les paramètres aléatoires suivent un modèle probabiliste mais sans chercher à l'estimer explicitement. Toutes ces approches sont pertinentes et complémentaires pour la prise de décisions sous incertitude. Le choix d'une approche particulière dépend des caractéristiques du problème.

Dans cette thèse, nous considérons qu'il est possible de déduire de façon suffisamment précise un modèle probabiliste de l'environnement. Celui-ci sert d'entrée au problème d'optimisation stochastique et l'agent décideur tire avantage de cette connaissance pour prendre les meilleures décisions. Ces dernières sont définies comme celles qui optimisent (maximisent ou minimisent) une certaine fonction probabiliste, tout en satisfaisant certaines contraintes dans un sens aussi probabiliste. La fonction à optimiser ou les contraintes à satisfaire peuvent être représentées sous forme d'espérance ($\mathbb{E}$) de variables aléatoires ou de probabilité ($\mathbb{P}$) d'évènements aléatoires. Les variables et évènements aléatoires, disons X_y et A_y respectivement, sont alors dépendants des décisions, notées y, de l'agent qui peut par exemple chercher à minimiser $\mathbb{E}[X_y]$ sous la contrainte $\mathbb{P}[A_y] \geq 0.95$, ou maximiser $\mathbb{P}[A_y]$ sous la contrainte $\mathbb{E}[X_y] \leq 2$. Il existe une grande variété de formulations en optimisation stochastique et nous décrirons les principales dans le chapitre de revue de littérature.

La difficulté principale dans la résolution de ces problèmes provient du fait que les opérateurs probabilistes (espérance, probabilité, ou autres) ne peuvent généralement pas être calculés analytiquement. Il est donc nécessaire d'utiliser des méthodes d'estimation numérique. Celles-ci vont discrétiser la distribution des paramètres aléatoires en un nombre fini de points (appelés *scénarios* dans ce contexte) et de poids afin de remplacer les opérateurs par leurs estimateurs statistiques. Une fois discrétisé, le problème est résolu comme un programme mathématique déterministe exprimé sur un ensemble fini de scénarios pondérés par

des poids. Cette résolution numérique s'accompagne inévitablement d'erreurs (sur la valeur et les décisions optimales) et celles-ci sont d'autant plus grande que les problèmes ont un grand nombre de paramètres aléatoires. Cet effet est connu sous le nom de *malédiction de la dimension*[2]. La nécessité de procéder à une étape de *discrétisation* préalablement à l'étape de résolution différencie l'approche stochastique de son équivalent déterministe.

Une autre différence fondamentale réside dans le concept d'*étapes*. Contrairement à l'idée d'étape (ou de période) en optimisation déterministe, qui réfère simplement à un instant temporel donné, en optimisation stochastique celle-ci fait référence à une *quantité d'information disponible* à un instant de prise de décisions donné. En effet, dans le cas stochastique l'information se révèle progressivement aux yeux de l'agent, qui se base sur celle-ci pour adapter ses décisions en cours de route. A l'inverse, dans le cas déterministe l'information n'évolue pas, les décisions au début et à la fin de l'horizon d'optimisation sont basées sur la même information. Une conséquence de cela est le fait que deux instants différents, mais avec une information disponible identique, correspondent à une seule et même étape en optimisation sous incertitude. Cette caractéristique offre plus de flexibilité dans la prise de décisions, qui est vue comme un processus temporel, d'où le nom de *processus de décision*. La contrepartie de cette flexibilité est une augmentation importante de la complexité des problèmes, à la fois mathématique et numérique.

Le passage d'un problème à une étape vers un problème à deux étapes voit un saut de complexité mathématique. En effet, à une étape les décisions sont prises avant la révélation des paramètres aléatoires, et donc l'optimisation se fait sur un espace de *dimension fini*, aussi large soit-il. A l'inverse, à partir de deux étapes, une partie des décisions (celles de *recours*) sont données pour chaque réalisation des paramètres aléatoires. Si leurs distributions sont continues, alors le nombre de réalisations est infinie, et donc l'optimisation se fait sur un espace de *dimension infinie* (un espace de fonctions). Cependant, la complexité numérique des problèmes à une ou deux étapes est similaire, car les deux requièrent la même structure de discrétisation. Le saut de complexité numérique se produit en réalité au passage de deux à trois étapes. A partir de trois, le processus de décision possède certaines étapes où l'agent décideur a accès à une information partielle, strictement comprise entre l'absence d'information de la première étape et l'accès complet de la dernière étape. Cette situation a une incidence sur les structures de discrétisation, qui doivent parvenir à recréer un accès partiel à l'information pour ces étapes intermédiaires. Ce saut de complexité numérique explique pourquoi la dénomination historique de *problèmes multi-étapes* inclut les problèmes à partir de trois étapes.

[2] Bellman (1957a)

1.2 Éléments de la problématique

Les structures utilisées pour discrétiser les problèmes d'optimisation stochastique multi-étapes sont connues sous le nom d'*arbres de scénarios*. Des exemples de structures d'arbres pour des problèmes à trois étapes sont représentés en Figure 1.1.

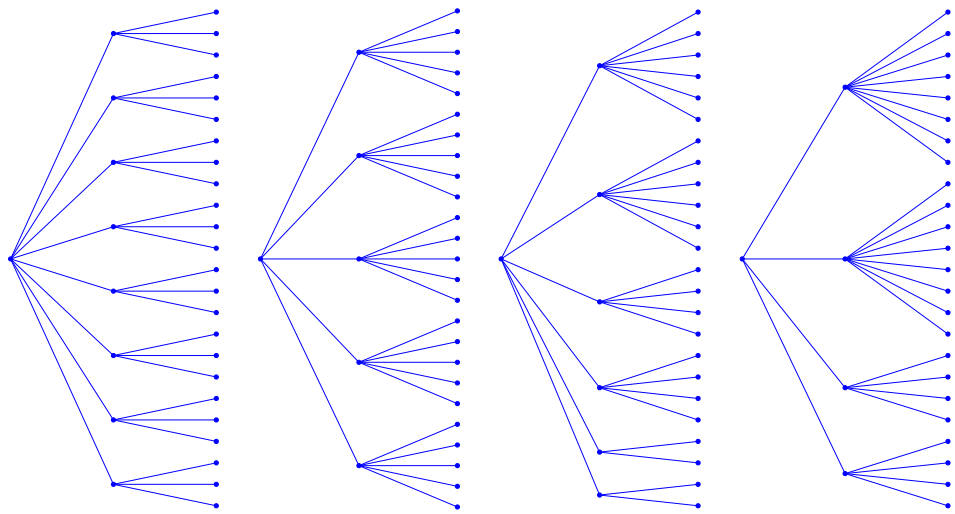


Figure 1.1 Structures d'arbre à trois étapes. Le nombre de scénarios (de gauche à droite) est 24, 25, 24, 24.

Le nœud le plus à gauche, appelé *racine*, correspond à la première étape du problème ; par convention il s'agit de l'étape 0. Les nœuds les plus à droite sont les *feuilles*, elles correspondent à la dernière étape ; par convention l'étape T. Le nombre de feuilles correspond au nombre de scénarios de l'arbre. Les deux structures de gauche vérifient la propriété que tous les nœuds d'une même étape ont le même facteur de branchement (égal au nombre de nœuds fils), pour cette raison elles sont dites *symétriques*. La deuxième structure vérifie la propriété supplémentaire que son facteur de branchement est constant à travers les étapes (ici égal à 5). Les deux structures de droite ne sont pas symétriques, car elles discrétisent plus finement les nœuds du haut que ceux du bas. A partir d'une structure d'arbre, on construit un arbre de scénarios en ajoutant un point (multidimensionnel) et un poids (positif) de discrétisation à chaque nœud et arête, respectivement.

Le processus de décision associe un vecteur de décision à chaque nœud de l'arbre. De part la structure de l'arbre, les décisions sont dépendantes de l'intégralité du chemin menant de la racine au nœud en question, donc de toute l'information révélée depuis le début. Cette caractéristique permet aux arbres de scénarios de pouvoir être utilisés quel que soit le processus stochastique sous-jacent au problème. En particulier, ils ont l'avantage de pouvoir

naturellement représenter des dépendances à longue portée entre les étapes. A l'inverse, les approches Markoviennes, basées sur la notion d'*état*, voient leur efficacité dépendre fortement de la possibilité (ou non) d'ignorer le chemin suivi par le processus depuis l'étape 0 étant donnée la connaissance de l'état courant. Notons que des fortes dépendances entre les étapes du problème peuvent apparaitre pour différentes raisons : si le processus stochastique exhibe de fortes corrélations entre les étapes (par exemple, pour les processus à *longue mémoire*), ou si les revenus ou les contraintes à chaque étape dépendent du chemin suivi par le processus.

L'arbre de scénarios, du fait de tenir compte de l'intégralité du chemin à chaque étape, a un inconvénient majeur : sa taille croît généralement exponentiellement en fonction du nombre d'étapes. Plus précisément, dans le cas des problèmes d'optimisation stochastique à $T+1$ étapes avec contraintes linéaires et une fonction objectif définie en espérance, un calcul de complexité montre que, sous certains conditions, le nombre de scénarios d'un arbre symétrique à branchement constant, construit par un échantillonnage Monte Carlo (MC), croît en ε^{-2T} pour une précision requise $\varepsilon > 0$ décroissant vers zéro[3].

Ce résultat peut être interprété de façon informelle comme suit : pour un problème à deux étapes ($T = 1$), on retrouve la complexité en ε^{-2} de l'échantillonnage MC pour l'estimation d'espérances (de fonctions de carré intégrable). Cette complexité découle elle-même du fait que l'erreur MC décroit en $b^{-1/2}$ où b est le nombre de points d'échantillonnage. Quant à la puissance en T, elle est inhérente à la structure symétrique à branchement constant de l'arbre de scénarios. En effet, si toutes les espérances conditionnelles aux nœuds de l'arbre sont approximées à une erreur $b^{-1/2}$ près avec b le facteur de branchement (le nombre de nœuds fils), alors pour un arbre symétrique à $N = b^T$ scénarios l'erreur est en $N^{-1/(2T)}$. Cela montre qu'avec un arbre symétrique il est extrêmement coûteux de garantir une faible erreur pour chaque espérance conditionnelle.

On peut alors se demander quelle amélioration il est possible d'espérer en considérant des méthodes de discrétisation qui ont des meilleurs taux de convergence que MC[4]. Pour une convergence de l'erreur en $b^{-\alpha}$ avec $\alpha > 0$, on peut espérer pouvoir changer la complexité en $\varepsilon^{-T/\alpha}$. Si, en étant optimiste, il est possible d'obtenir un taux $\alpha = 3/2$ (au lieu du $1/2$ de MC), cela donnerait une complexité en $\varepsilon^{-2T/3}$ au lieu de ε^{-2T}. La puissance est divisée par trois, mais celle-ci dépend toujours du nombre d'étapes. Pour cette raison, il est nécessaire de s'intéresser à des arbres ayant des facteurs de branchement variables et de restreindre l'attention à des sous-classes de problèmes d'optimisation stochastique multi-étapes pour lesquels ces arbres pourraient être efficaces.

[3]Nous référons à Shapiro (2006) pour l'énoncé précis de ce résultat.

[4]Par simplicité de l'argument nous passons sur les conditions qui permettraient d'avoir ces meilleurs taux de convergence.

La première structure de la Figure 1.1 offre un exemple de facteur de branchement qui varie à travers les étapes ($b_0 = 8$ et $b_1 = 3$). Les deux dernières structures ont des facteurs de branchement qui varient à la fois à travers les étapes et à travers les nœuds d'une même étape. Cette flexibilité rend les structures non-symétriques particulièrement intéressantes, car elle permet de limiter la croissance exponentielle du nombre de scénarios. Par exemple, on peut montrer qu'une structure construite par la règle récurrente suivante (illustrée Figure 1.2) :

- 3 nœuds à l'étape 1 ;
- $3 \to (1, 2, 3)$: chaque groupe de 3 nœuds frères donne naissance à 1, 2 et 3 fils ;
- $2 \to (1, 2)$: chaque groupe de 2 nœuds frères donne naissance à 1 et 2 fils ;
- $1 \to (1)$: chaque groupe de 1 nœud donne naissance à 1 fils ;

résulte en un nombre de scénarios égal à[5]

$$\frac{(T-1)^2 + 5(T-1) + 6}{2}, \quad \text{pour } T \geq 1, \tag{1.1}$$

donc asymptotiquement de l'ordre *polynomial* $T^2/2$.

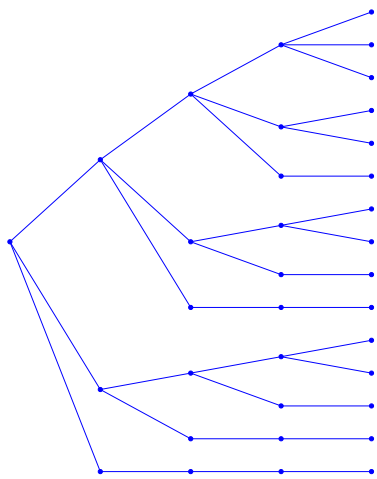


Figure 1.2 Structure d'arbre construite récursivement par la règle [$3 \to (1, 2, 3)$; $2 \to (1, 2)$; $1 \to (1)$].

La généralisation de cette règle à des groupes d'au plus m nœuds frères ($m \geq 4$) (c.-à-d., $[m \to (1, \ldots, m); m-1 \to (1, \ldots, m-1); \ldots; 2 \to (1, 2); 1 \to (1)]$) produit des structures dont le nombre de scénarios est de l'ordre (aussi polynomial[6]) $T^{m-1}/(m-1)!$. La croissance

[5] Ce résultat se démontre en résolvant le système récurent suivant : $\gamma_{t+1} = \gamma_t$, $\beta_{t+1} = \gamma_t + \beta_t$, $\alpha_{t+1} = \gamma_t + \beta_t + \alpha_t$ avec la condition initiale $\gamma_1 = 1$, $\beta_1 = \alpha_1 = 0$, où γ_t (resp. β_t et α_t) est le nombre de groupes de 3 nœuds (resp. 2 et 1) à l'étape $t \geq 1$. La solution est $\gamma_t = 1$, $\beta_t = t - 1$ et $\alpha_t = \frac{(t-1)^2+t-1}{2}$.

[6] On peut montrer que le nombre de groupes de i nœuds frères ($1 \leq i \leq m$) à l'étape t ($t \geq 1$) est égal à $\frac{(t-2+m-i)!}{(t-2)!(m-i)!}$.

seulement polynomial s'explique par le fait que ces structures ont une majorité de scénarios qui ne se diversifie plus (les facteurs de branchement égaux à 1 dominent tous les autres asymptotiquement).

Un exemple où les trois facteurs de branchement sont égaux à chaque étape est la structure construite récursivement suivant la règle (voir Figure 1.3) :
- 3 nœuds à l'étape 1 ;
- $3 \to (1, 2, 3)$;
- $2 \to (1, 3)$;
- $1 \to (2)$.

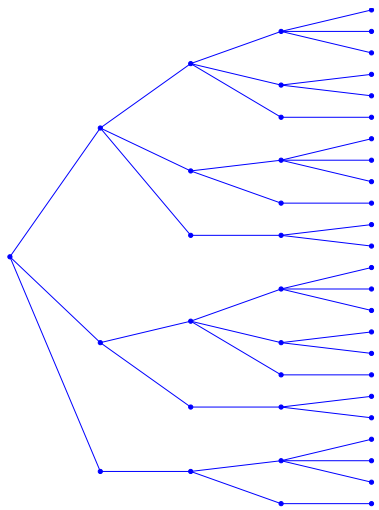


Figure 1.3 Structure d'arbre construite récursivement par la règle $[3 \to (1, 2, 3)\,;\ 2 \to (1, 3)\,;\ 1 \to (2)]$.

Cette structure produit 2^{t-2} groupes de 1, 2 et 3 nœuds frères[7] à chaque étape $t \geq 2$, donc elle possède un nombre de scénarios égal à

$$(3 + 2 + 1)2^{T-2} = 1.5 \times 2^T, \quad \text{pour } T \geq 2. \tag{1.2}$$

La croissance est exponentielle en 2^T, donc inférieure à celle en 3^T qui serait obtenue en utilisant un facteur de branchement constant égal à 3. En généralisant la règle précédente à des groupes d'au plus m nœuds ($m \geq 4$), c.-à-d.,
- m nœuds à l'étape 1 ;
- $m \to (1, 2, \ldots, m)$;
- $m - i \to (1, 2, \ldots, m - i - 1, m - i + 1)$ pour $i = 1, \ldots, m - 3$;

[7] Dans ce cas le système à résoudre est : $\gamma_{t+1} = \gamma_t + \beta_t$, $\beta_{t+1} = \gamma_t + \alpha_t$, $\alpha_{t+1} = \gamma_t + \beta_t$ avec la condition initiale $\gamma_1 = 1$, $\beta_1 = \alpha_1 = 0$. La solution est $\gamma_t = \beta_t = \alpha_t = 2^{t-1}$.

– $2 \to (1,3)$;
– $1 \to (2)$;
on observe (numériquement) que la croissance peut être grandement réduite par rapport à un facteur de branchement constant égal à m (voir Tableau 1.1).

Tableau 1.1 Croissance du nombre de scénarios pour une structure construite récursivement par la règle $[m \to (1,2,\ldots,m)\,;\, m-i \to (1,2,\ldots,m-i-1,m-i+1)\,;\, 2 \to (1,3)\,;\, 1 \to (2)]$.

	$m=5$	$m=10$	$m=15$	$m=20$
nombre de scénarios	2.0×2.25^T	11.8×2.91^T	31.6×3.3^T	62.3×3.68^T

Une question importante est de savoir si les structures non-symétriques peuvent bénéficier à certains problèmes d'optimisation stochastique multi-étapes.

1.3 Objectifs de recherche

L'objectif général de cette recherche est d'améliorer l'efficacité de l'approche par arbres de scénarios, afin de pouvoir résoudre des problèmes d'optimisation stochastique à plus grand nombre d'étapes que ce qui est fait généralement dans la littérature (où dans la grande majorité des cas $T \leq 10$).

Pour cela, nous souhaitons répondre aux deux questions suivantes :

(1) Pour quel type de problèmes les arbres de scénarios sont-ils efficaces ?
(2) Comment générer des arbres de scénarios adaptés à ces problèmes ?

La question (1) peut se diviser en deux sous-questions :

(1.1) Quelles caractéristiques du problème influencent la qualité de l'approximation par arbres de scénarios ?
(1.2) Comment représenter mathématiquement ces caractéristiques de sorte à indiquer la difficulté du problème ?

Pour une classe P de problèmes d'optimisation stochastique multi-étapes suffisamment large (c.-à-d., qui contient la plupart des problèmes d'intérêt), on veut définir pour chaque problème $p \in P$ un représentant $\Gamma(p)$ qui contient quantitativement toute l'information nécessaire à

la génération d'arbres de scénarios pour p. Ce représentant doit permettre de caractériser la difficulté de résoudre p dans la classe P, c'est-à-dire, on veut pouvoir dire que :

(i) si $\Gamma(p_1) \prec \Gamma(p_2)$, pour une certaine relation d'ordre $\prec$, alors p_2 est plus difficile à résoudre que p_1 ;

(ii) si $\Gamma(p_1) \sim \Gamma(p_2)$, pour une certaine relation d'équivalence $\sim$, alors les arbres de scénarios adaptés à p_1 sont aussi adaptés à p_2.

La question (2) se divise aussi en deux sous-questions :

(2.1) Comment exploiter les caractéristiques du problème dans la construction de l'arbre ?

(2.2) Comment faire en sorte de pouvoir générer la structure la plus adaptée au problème ?

On veut définir une mesure $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma(p))$ qui indique l'adéquation entre un problème $p \in P$ et un arbre de scénarios de structure $\mathcal{T}$, points $\mathcal{P}$ et poids $\mathcal{W}$ arbitraires. Cette mesure doit pouvoir être utilisée pour construire l'arbre de scénarios $(\mathcal{T}, \mathcal{P}, \mathcal{W})^*$ le plus adapté à p en imposant le moins de contraintes possibles sur la forme de l'arbre.

1.4 Plan de la thèse

Nous présentons maintenant le plan de la thèse. Le Chapitre 2 est une revue de littérature. La première section, 2.1, décrit l'approche d'optimisation stochastique multi-étapes que nous considérons, ainsi que d'autres approches alternatives pour la prise de décisions sous incertitude. Les deux sections suivantes, 2.2 et 2.3, décrivent les méthodes et les applications de l'optimisation stochastique multi-étapes. Le Chapitre 3 explique brièvement les résultats obtenus dans les quatre articles intégrés à la thèse. Les Chapitres 4, 5, 6 et 7 contiennent les articles par ordre chronologique. Nous discutons des résultat de la thèse dans le Chapitre 8 et nous concluons celle-ci par le Chapitre 9 qui fait la synthèse de travaux, présente les limitations et les améliorations futures.

CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Les approches de prise de décisions multi-étapes sous incertitude

Les différentes approches de prise de décisions multi-étapes sous incertitude se distinguent par la façon dont elles modélisent le processus de décision, par la définition de la fonction objectif à optimiser, et par la famille d'algorithmes utilisés pour parvenir à calculer exactement ou approximativement les décisions optimales. Dans cette section, nous présentons brièvement les modélisations les plus utilisées en pratique que sont : l'optimisation en espérance, de mesure de risque, robuste, robuste en distribution, avec contraintes de probabilité, et les processus de décision Markoviens.

2.1.1 Modélisation du processus de décision

Deux approches génériques existent pour ce qui concerne la modélisation du processus de décision : l'*optimisation stochastique multi-étapes* et les *processus de décision Markoviens*.

Le processus de décision de l'**optimisation stochastique multi-étapes** (OSM) se représente de façon générale comme suit :

$$\begin{aligned}\text{(OSM)} \quad y_0 = x_0(\xi_0) \xrightarrow{\xi_1} y_1 = x_1(\xi_0, \xi_1) \xrightarrow{\xi_2} y_2 = x_2(\xi_0, \xi_1, \xi_2) \xrightarrow{\xi_3} \cdots \\ \cdots \xrightarrow{\xi_T} y_T = x_T(\xi_0, \ldots, \xi_T).\end{aligned} \tag{2.1}$$

Le processus commence par prendre des décisions y_0 à l'étape 0 à partir de la donnée non aléatoire ξ_0. Puis, une réalisation ξ_1 des paramètres aléatoires d'étape 1 est observée, s'en suivent des décisions y_1 qui dépendent de ξ_0 et ξ_1 à travers une certaine fonction (politique) de décision $x_1(\cdot)$. Une réalisation ξ_2 des paramètres aléatoires d'étape 2 est ensuite observée, et des décisions y_2 sont prises, et ainsi de suite. Les fonctions $x_0(\cdot)$, $x_1(\cdot)$, ..., $x_T(\cdot)$ sont les solutions recherchées du problème d'optimisation. On observe qu'à l'étape t les décisions y_t sont fonction de toutes les réalisations aléatoires passées, $\xi_0, \ldots, \xi_t$, c.-à-d., de toute l'information accessible à cette étape. Le processus de décision s'arrête après un certain nombre d'étapes T fini. Pour simplifier les notations, nous notons dorénavant $\xi_{..t} := (\xi_0, \ldots, \xi_t)$ et $y_{..t} := (y_0, \ldots, y_t)$ les réalisations et décisions des étapes 0 à t ; lorsque $t = T$ on écrira ξ et y au lieu de $\xi_{..T}$ et $y_{..T}$, respectivement.

Les **processus de décision Markoviens** (PDM) sont des processus de contrôle stochastique en temps discret. Ils sont basés sur l'idée que le système évoluant dans l'environnement

stochastique peut être décrit à chaque instant t par un état s_t. Cet état inclut à lui seul toute l'information nécessaire à la prise de décisions et il évolue suivant une certaine dynamique stochastique. Le processus se représente alors de la façon suivante :

$$\begin{aligned}\text{(PDM)} \quad s_0 \rightsquigarrow y_0 = x_0(s_0) \xrightarrow{\xi_1} s_1 = f(s_0, y_0, \xi_1) \rightsquigarrow y_1 = x_1(s_1) \xrightarrow{\xi_2} \cdots \\ \cdots \rightsquigarrow y_{T-1} = x_{T-1}(s_{T-1}) \xrightarrow{\xi_T} s_T = f(s_{T-1}, y_{T-1}, \xi_T).\end{aligned} \tag{2.2}$$

Le processus commence dans l'état s_0 à l'étape 0, des décisions y_0 fonction de s_0 sont prises, puis une réalisation ξ_1 est observée ce qui fait évoluer le système dans le nouvel état $s_1 = f(s_0, y_0, \xi_1)$, où f est une fonction de transition connue, et ainsi de suite. On observe qu'à l'étape t la politique de décision $x_t(\cdot)$ ne dépend que de l'état courant s_t, et non de tous les états $s_0, \ldots, s_{t-1}$ visités depuis l'origine. Le processus peut s'arrêter après un nombre d'étapes T fini ou alors continuer infiniment ($T = \infty$).

Notons que les notations ci-dessus peuvent être différentes de celles généralement utilisées dans le littérature des processus de décision Markoviens. Cependant, dans le cas présent elles permettent de faire le parallèle entre ces processus et ceux décrits en (2.1). Généralement, les décisions sont notées a_t (pour *action*) plutôt que y_t, les fonctions de décision $\pi_t(\cdot)$ plutôt que $x_t(\cdot)$, et les paramètres aléatoires w_t plutôt que ξ_t.

Contrairement au processus de décision (2.1), la fonction de décision $x_t(\cdot)$ en (2.2) ne dépend que de l'information à l'étape t. La raison pour laquelle une telle simplification est possible tient dans l'hypothèse cruciale que le processus vérifie la propriété de Markov. Cette propriété engendre les trois conditions suivantes qui sont vérifiées à chaque étape : (i) la probabilité d'atteindre l'état $s_{t+1} = s'$ en étant dans l'état $s_t = s$ sous l'action des décisions $y_t = x_t(s)$ ne dépend que de s', s et y_t ; (ii) les revenus (ou coûts) reçus en transitant de l'état $s_t = s$ à $s_{t+1} = s'$ sous les décisions y_t ne dépendent que de s', s et y_t ; (iii) l'ensemble des décisions y_t admissibles ne dépend que de s_t.

La première condition implique que la distribution de l'incertitude ξ_{t+1} est indépendante des réalisations ou décisions passées conditionnellement à la connaissance de s_t et y_t. Cela diffère de l'optimisation stochastique multi-étapes où ξ_{t+1} peut dépendre de toutes les réalisations passées (y compris conditionnellement à ξ_t). L'optimisation stochastique a donc l'avantage de pouvoir considérer une plus grande classe de processus stochastiques, en particulier ceux pour lesquels la corrélation entre les étapes est persistante. Cependant, l'inconvénient qui en résulte est que les fonctions de décision dépendent de tout l'historique du processus. Ainsi, les problèmes d'optimisation stochastique multi-étapes sont généralement de grande taille (avec un grand nombre de variables de décision et de contraintes).

Les processus de décision Markoviens ont leurs origines dans les travaux de Bellman (1957b) et Howard (1960) et les problèmes d'optimisation stochastique multi-étapes dans ceux de Dantzig (1955) et Beale (1955).

Nous considérons par la suite que le problème contient d paramètres aléatoires et s décisions à chaque étape ($d \geq 1$, $s \geq 1$) ; ξ_t et y_t prennent donc leurs valeurs dans des sous-ensembles de $\mathbb{R}^d$ et $\mathbb{R}^s$, respectivement.

2.1.2 Modélisation du problème de décision

Dans cette section, nous décrivons les différentes formulations mathématiques des problèmes de décision multi-étapes sous incertitude, de façon suffisamment générale pour y inclure la plupart des approches classiques. Cette description permet de positionner les problèmes considérés dans un cadre plus large. (Par convention, nous exprimons les problèmes sous forme de *maximisation* (des revenus) ; les problèmes de minimisation (des coûts) s'y ramènent sans difficulté par changement de signe.)

Le processus de décision considéré dans cette thèse est celui de l'optimisation stochastique multi-étapes (2.1). Afin d'obtenir à partir de celui-ci un *problème* de décision, il est nécessaire de définir une fonction objectif à maximiser et des contraintes à satisfaire. Pour cela, nous introduisons :

- une **fonction de revenus**, $q(y;\xi)$, dépendant des décisions $y = (y_0, \ldots, y_T)$ et des réalisations aléatoires $\xi = (\xi_0, \ldots, \xi_T)$; on peut supposer par simplicité que celle-ci s'écrit comme une somme de revenus à chaque étape :

$$q(y;\xi) = \sum_{t=0}^{T} q_t(y_t, \xi_t); \tag{2.3}$$

- une **fonctionnelle de probabilité** pour la fonction objective, qui prend en variable les revenus aléatoires et renvoie un nombre réel (le critère à maximiser) ; dans notre cas cette fonctionnelle est l'espérance $\mathbb{E}_\xi[\cdot]$ sous la distribution du processus $\xi = (\xi_0, \ldots, \xi_T)$; la fonction objectif à maximiser est donc :

$$\mathbb{E}_\xi \left[\sum_{t=0}^{T} q_t(y_t; \xi_t) \right]; \tag{2.4}$$

- un **ensemble de contraintes** que les décisions doivent satisfaire avec une certaine probabilité $\alpha \in [0,1]$; dans notre cas les variables de décision satisfont :

$$y_0 \in Y_0(\xi_0) \subseteq \mathbb{R}^s \quad \text{et} \quad y_t \in Y_t(y_{t-1}; \xi_{..t}) \subseteq \mathbb{R}^s, \quad t = 1, \ldots, T, \tag{2.5}$$

presque sûrement (p.s.), c.-à-d., avec probabilité $\alpha = 1$.

Les problèmes de décision multi-étapes sous incertitude peuvent se formuler de deux façons équivalentes. La première est une formulation *fonctionnelle* exprimée sur un espace de fonctions (donc de dimension infinie). La deuxième est dite *dynamique*, car elle décompose le problème en une infinité de sous-problèmes de dimension finie. Dans les deux cas, le but est de trouver la meilleure politique de décision $(x_0, \ldots, x_T)$.

La **formulation fonctionnelle** de l'optimisation stochastique multi-étapes (F-OSM) s'écrit :

$$\text{(F-OSM)} \qquad \max_{(x_0,\ldots,x_T)} \quad \mathbb{E}_\xi \left[\sum_{t=0}^{T} q_t(x_t(\xi_{..t}); \xi_t) \right] \tag{2.6}$$

$$\text{sujet à :} \quad x_0(\xi_0) \in Y_0(\xi_0); \tag{2.7}$$

$$x_t(\xi_{..t}) \in Y_t(x_{t-1}(\xi_{..t-1}); \xi_{..t}), \ p.s., \ t = 1, \ldots, T. \tag{2.8}$$

La première ligne contient la fonction objective du problème à maximiser. Les deux lignes suivantes décrivent l'ensemble des fonctions de décision réalisables pour le problème. Ce problème a été étudié à l'origine par Rockafellar et Wets (1974) et Olsen (1976b,c).

La formulation dynamique consiste à décomposer la formulation fonctionnelle en plusieurs sous-problèmes. Cela est rendu possible par la fait que l'espérance peut elle-même être décomposée (théorème de l'espérance totale) et par certaines conditions techniques qui permettent de permuter les opérateurs d'espérances conditionnelles et de maximisation ; c'est *principe d'interchangeabilité* énoncé par exemple par Rockafellar et Wets (2009, Theorem 14.60). Ce principe permet de passer d'une formulation où la solution est une *fonction de décision* définie sur le support des paramètres aléatoires à une autre où les solutions sont des *vecteurs de décision* donnés pour chaque valeur possible de ces paramètres.

La **formulation dynamique** (D-OSM) correspondante est définie récursivement de la dernière à la première étape comme suit :

– on calcule alternativement par chaînage arrière :

$$Q_t^*(y_{t-1}; \xi_{..t}) = \max_{y_t \in Y_t(y_{t-1}; \xi_{..t})} \{q_t(y_t; \xi_t) + Q_t(y_t; \xi_{..t})\}, \quad t = T, \ldots, 1; \tag{2.9}$$

$$Q_t(y_t; \xi_{..t}) = \mathbb{E}[Q_{t+1}^*(y_t; \xi_{..t}, \xi_{t+1}) \,|\, \xi_{..t}], \qquad t = T, \ldots, 0; \tag{2.10}$$

avec par définition $Q_{T+1}^* \equiv 0$;

– à la première étape ($t = 0$), le problème global devient :

$$\max_{y_0 \in Y_0(\xi_0)} \{q_0(y_0; \xi_0) + Q_0(y_0; \xi_0)\}. \tag{2.11}$$

Sa valeur optimale est notée $Q_0^*(\xi_0)$.

Les fonctions Q_t^* et Q_t sont parfois appelées fonctions de recours optimal et fonction de recours espéré, respectivement[1]. Elles représentent les revenus maximums qu'il est possible d'obtenir de l'étape t à T lorsque les décisions futures sont prises de façon optimale. La formulation dynamique est étudiée par exemple dans Wets (1972). Nous référons aussi aux ouvrages de référence sur le sujet, par exemple Kall et Wallace (1994) et Birge et Louveaux (2011).

L'équivalence entre les deux formulations garantit que la valeur optimale de la formulation dynamique ($Q_0^*(\xi_0)$) est égale à celle de la formulation fonctionnelle. Elle garantit aussi que les décisions optimales sont identiques : les fonctions $(x_0^*, \ldots, x_T^*)$ sont optimales pour (F-OSM) si et seulement si elles sont optimales pour (D-OSM), c.-à-d.,

$$x_t^*(\xi_{..t}) \in \underset{y_t \in Y_t(x_{t-1}^*(\xi_{..t-1});\xi_{..t})}{\operatorname{argmax}} \{q_t(y_t;\xi_t) + Q_t(y_t;\xi_{..t})\}, \quad t = T, \ldots, 1; \tag{2.12}$$

$$x_0^*(\xi_0) \in \underset{y_0 \in Y_0(\xi_0)}{\operatorname{argmax}} \{q_0(y_0;\xi_0) + Q_0(y_0;\xi_0)\}. \tag{2.13}$$

Dans le cas des processus de décision Markoviens, la condition (ii) citée dans la section précédente implique que la fonction de revenus q_t dépend uniquement de la décision y_t et de l'état s_t : $q_t(y_t; s_t)$; de même la condition (iii) implique que l'ensemble des contraintes Y_t dépend uniquement de s_t : $Y_t(s_t)$.

La **formulation fonctionnelle** des processus de décision Markoviens (F-PDM) est :

$$\text{(F-PDM)} \qquad \max_{(x_0,\ldots,x_T)} \quad \mathbb{E}_\xi \left[\sum_{t=0}^{T} \gamma^t \, q_t(x_t(s_t); s_t) \right] \tag{2.14}$$

$$\text{sujet à :} \quad x_t(s_t) \in Y_t(s_t), \ p.s., \ t = 0, \ldots, T, \tag{2.15}$$

où $\gamma \in (0, 1]$ est le facteur de réduction qui permet à l'espérance d'être finie dans le cas où $T = \infty$.

La **formulation dynamique** (D-PDM) correspondante s'écrit :

$$V_t(s_t) = \max_{y_t \in Y_t(s_t)} \{q_t(y_t; s_t) + \gamma \, Q_t(y_t; s_t)\}, \quad t = T, \ldots, 0; \tag{2.16}$$

$$Q_t(y_t; s_t) = \mathbb{E}[V_{t+1}(f(s_t, y_t, \xi_{t+1})) \mid s_t, y_t], \qquad t = T-1, \ldots, 0; \tag{2.17}$$

avec par définition $V_{T+1} \equiv 0$,

[1] En anglais : *optimal recourse function* et *expected recourse function*.

Les fonctions V_t et Q_t sont généralement appelées fonction valeur de l'état et fonction valeur de l'action, respectivement[2]. La quantité $V_t(s_t)$ représente la valeur de l'état s_t à l'étape t alors que $Q_t(y_t; s_t)$ représente la valeur de la *décision* y_t dans l'état s_t à l'étape t. Pour une description plus détaillée des processus de décision Markoviens, nous référons à Bertsekas (2000).

2.1.3 Modélisations alternatives

Les modélisations introduites ci-dessus correspondent à un paradigme particulier de prise de décisions sous incertitude où la fonction objectif est l'espérance des revenus et où les contraintes doivent être satisfaites presque sûrement. Ce paradigme, bien qu'adapté à une grande quantité de problèmes en pratique, a certaines limites. Par exemple, le fait d'optimiser les revenus *en espérance* est adapté à une situation où le processus de décision est appliqué répétitivement, puisque la loi des grands nombres garantit alors que la moyenne empirique des revenus converge vers l'espérance. Par contre, optimiser en espérance offre peu de garanties dans le cas d'un processus de décision appliqué une seule fois. Cela provient du fait que l'espérance donne le même poids à des revenus extrêmement élevés ou, inversement, extrêmement faibles si les deux ont la même probabilité de se produire. Or, des revenus bien en deçà de la moyenne peuvent avoir des conséquences dramatiques en pratique et, par conséquent, doivent être évités à tout prix.

Optimisation averse au risque : Une façon de contourner ce problème, tout en continuant d'utiliser l'espérance comme fonctionnelle de probabilité, consiste à remplacer dans (2.6) la fonction de revenus $q(y, \xi)$ par $u(q(y; \xi))$ où $u(\cdot)$ est une fonction d'utilité qui tient compte des préférences du décideur en termes de risque. L'idée d'introduire une fonction d'utilité dans les problèmes de décision sous incertitude à son origine dans les travaux de Bernoulli (1738) pour résoudre le paradoxe de Saint-Pétersbourg. La théorie des fonctions d'utilité est formalisée dans un cadre probabiliste par Morgenstern et Von Neumann (1944). Ces derniers ont prouvé que si les préférences du décideur satisfont certains axiomes, alors il existe une fonction d'utilité $u : \mathbb{R} \to \mathbb{R}$ décrivant ses préférences. Les meilleures décisions sont alors celles qui maximisent l'espérance de la fonction d'utilité : $\mathbb{E}[u(q(y; \xi))]$. La limite de cette approche réside dans le fait qu'il est généralement impossible pour le décideur de déterminer exactement sa propre fonction d'utilité.

Une approche alternative de prise en compte du risque consiste à remplacer l'espérance par une *mesure de risque*, $\mathcal{R} : \mathcal{X} \to \mathbb{R}$, définie sur un espace de variables aléatoires $\mathcal{X}$. Par exemple, dans la théorie moderne du portefeuille, le modèle moyenne-variance de Markowitz

[2]En anglais : *state-value function* et *action-value function*.

(1952) identifie le risque d'un portefeuille avec sa variance (volatilité). Maximiser la moyenne des rendements du portefeuille tout en minimisant leur variance correspond à considérer une mesure de risque de la forme

$$\mathcal{R}(Z) = \beta\mathbb{E}[Z] - (1-\beta)\mathrm{Var}[Z], \tag{2.18}$$

où $\beta \in [0,1]$ est un paramètre qui balance l'importance de la moyenne par rapport à la variance. La limite de la variance comme mesure de risque est qu'elle pénalise de façon égale les gains et pertes extrêmes. Pour cela, d'autres mesures sont aujourd'hui prises en compte, comme par exemple la *Conditional Value-at-Risk* (CVaR) formulée par Rockafellar et Uryasev (2000) comme suit :

$$\mathcal{R}_\alpha(Z) = -\mathrm{CVaR}_\alpha(Z) = \mathbb{E}[Z \mid Z \leq F_Z^{-1}(\alpha)], \tag{2.19}$$

où $\alpha \in (0,1)$ est le niveau de confiance, Z est une variable aléatoire à densité et $F_Z^{-1}(\alpha)$ est le quantile α de la distribution de Z. Maximiser $\mathcal{R}_{0.1}(q(y,\xi))$ en y revient à chercher les décisions qui vont minimiser la moyenne des revenus du premier décile, c'est-à-dire les revenus les plus faibles ayant moins de 10% de chance de se produire.

L'approche moderne de la prise en compte de l'aversion au risque, qui formalise ces mesures dans un cadre commun, parle de mesures *cohérentes* lorsqu'elles satisfont un certain nombre d'axiomes. Nous référons à Artzner *et al.* (1999) pour une présentation générale et à Pflug et Pichler (2016, Chapitre 3) et Shapiro *et al.* (2014, Chapitre 6) pour une présentation dans le cadre de l'optimisation stochastique.

Optimisation robuste : Une autre approche de prise de décisions sous incertitude est l'*optimisation robuste*. Cette approche se différencie par le fait qu'elle ne suppose pas de distribution de probabilité, mais considère plutôt que les paramètres aléatoires appartiennent à certains *ensembles d'incertitude*. Le but est alors de calculer des décisions robustes, c.-à-d., qui offrent des garanties en terme de revenus obtenus dans le cas où la réalisation aléatoire est la pire possible dans l'ensemble d'incertitude. Aussi, en ne considérant aucune distribution de probabilité, cette approche permet de contourner le problème de l'estimation statistique de cette distribution et de l'erreur qui vient nécessairement avec.

Si on note Θ l'ensemble d'incertitude à l'intérieur duquel les paramètres aléatoires sont restreints, alors l'extension robuste du problème (F-OSM) consiste formellement à remplacer l'espérance (2.6) par

$$\min_{\xi\in\Theta}\left[\sum_{t=0}^{T} q_t(x_t(\xi_{..t});\xi_t)\right]; \tag{2.20}$$

et le «presque sûrement» des contraintes (2.7)-(2.8) par «$\forall \xi \in \Theta$». Les fonctions de décision optimales pour ce nouveau problème sont appelées les règles de décision robustes optimales. Ces règles sont généralement restreintes à des fonctions affines afin de ramener le problème robuste à un problème de dimension finie. Nous référons à Ben-Tal *et al.* (2009, Chapitre 14) pour une introduction détaillée sur l'approche robuste multi-étapes.

Les processus de décision Markoviens ont aussi une contre-partie robuste. Par exemple, dans Ben-Tal *et al.* (2009, Chapitre 13) les ensemble d'états $\mathcal{S}_0, \ldots, \mathcal{S}_T$ sont considérés finis et l'équation (2.17) de la formulation (D-PDM) est remplacée par son équivalent robuste :

$$Q_t(y_t; s_t) = \min_{p \in P(t, y_t, s_t)} \sum_{s_{t+1} \in \mathcal{S}_{t+1}} p(s_{t+1}) V_{t+1}(s_{t+1}), \quad t = T-1, \ldots, 0, \tag{2.21}$$

où $P(t, y_t, s_t)$ est un ensemble d'incertitude pour les probabilités de transition depuis l'état s_t sous la décision y_t à l'étape t.

Optimisation robuste en distribution : L'approche robuste permet de contourner le problème de l'estimation de la distribution du processus stochastique, mais cela est fait au prix d'optimiser par rapport au pire scenario de l'ensemble d'incertitude, ce qui peut mener en pratique à des décisions conservatrices. Une approche alternative, qui se situe à mi-chemin entre l'optimisation robuste et celle en espérance, est l'*optimisation robuste en distribution*. Dans ce cas, le maximum des revenus est calculé par rapport à la pire distribution dans un *ensemble d'ambiguïté* $\mathfrak{P}$ contenant un ensemble de distributions. Formellement, cela revient à remplacer l'espérance $\mathbb{E}_\xi[\cdot]$ dans (2.6) par

$$\min_{P \in \mathfrak{P}} \mathbb{E}_{\xi \sim P}[\cdot], \tag{2.22}$$

où $\xi \sim P$ signifie que les paramètres aléatoires ξ suivent la distribution P. La notion d'ambiguïté et de risque dans le choix d'une distribution de probabilité est discuté dans les travaux de Ellsberg (1961). En optimisation stochastique, les problèmes robuste en distribution ont été étudiés à l'origine par Žáčková (1966).

En pratique, la difficulté de cette approche réside dans le choix d'un ensemble d'ambiguïté approprié. Certains auteurs proposent de considérer toutes les distributions à une certaine distance $\delta > 0$ d'une distribution de référence P^*, i.e, $\mathfrak{P} = \{P : d(P, P^*) \leq \delta\}$, où d est une certaine distance (ou semi-distance). Par exemple, Pflug et Wozabal (2007) considère la distance de Wasserstein pour construire l'ensemble d'ambiguïté sur ce modèle. D'autres auteurs considèrent des ensembles d'ambiguïté contenant toutes les distributions qui ont un certain nombre de moments en commun. Cette approche est suivie par exemple par Delage

et Ye (2010).

Pour une introduction détaillée sur les problèmes robustes en distribution dans le cadre de l'optimisation stochastique multi-étapes, nous référons à Pflug et Pichler (2016, Chapitre 7). La robustesse en distribution peut aussi être appliquée au cas des processus de décision Markoviens (Xu et Mannor, 2010; Wiesemann *et al.*, 2013).

Contrairement à l'optimisation stochastique en espérance, les problèmes d'optimisation robustes et averses au risque utilisent des fonctionnelles de probabilité autres que l'espérance $\mathbb{E}_\xi[\cdot]$ pour la définition la fonction objectif. Contrairement à cette dernière, ces fonctionnelles ne sont pas nécessairement décomposables par étape, et dans ce cas le principe d'interchangeabilité ne permet pas de décomposer la formulation fonctionnelle en formulation dynamique. C'est le problème de la *consistance temporelle* en optimisation sous incertitude : une fonction de décision optimale est dite consistante (en temps) si les décisions qu'elle produit aux étapes $t = 1, \ldots, T$ restent optimales quelque que soient les réalisations du processus stochastique. En pratique, la consistance temporelle n'est pas garantie si, par exemple, $\mathcal{R}(\cdot)$ n'est pas strictement monotone, c.-à-d., si $\mathcal{R}(\cdot)$ ne vérifie pas : $Z < Z'$ *p.s.* $\Rightarrow \mathcal{R}(Z) < \mathcal{R}(Z')$. (On note que l'espérance est strictement monotone.) Le problème de la consistance temporelle est étudié activement, nous référons à Pflug et Pichler (2016, Chapitre 5) et Shapiro (2018) pour une présentation générale sur le sujet.

Optimisation sous contraintes de probabilité : Jusqu'à présent, nous avons considéré des problèmes de décision où les contraintes sont vérifiées *presque sûrement*. En pratique, il peut être extrêmement coûteux, voire impossible, de garantir la réalisabilité des décisions pour essentiellement toutes les réalisations des paramètres aléatoires. Dans ce cas, il est possible de relaxer cette condition en imposant que les contraintes doivent être satisfaites au minimum avec une certaine probabilité $\alpha \in (0, 1)$. Ce type de contraintes, appelé *contraintes de probabilité*, consiste formellement à remplacer la condition (2.5) par

$$\mathbb{P}(y_t \in Y_t(y_{t-1}; \xi_{..t})) \geq \alpha, \ t = 1, \ldots, T. \tag{2.23}$$

Ces contraintes ont été introduites par Charnes *et al.* (1958) .

Les contraintes de probabilité peuvent être vues comme un cas particulier des *contraintes en espérance*, où la condition (2.5) est remplacée par :

$$\mathbb{E}[G_t(y_t, y_{t-1}, \xi_{..t})] \geq 0, \quad t = 1, \ldots, T, \tag{2.24}$$

pour certaines fonctions $G_t : \mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^{d(t+1)} \to \mathbb{R}^{k_t}$, où $k_t \geq 1$ représente le nombre de

contraintes à $t = 1, \ldots, T$. Le cas des contraintes de probabilité se retrouve en posant

$$G_t(y_t, y_{t-1}, \xi_{..t}) = \mathbf{1}_{Y_t(y_t;\xi_{..t})}(y_t) - \alpha, \tag{2.25}$$

où $\mathbf{1}_{Y_t}(\cdot)$ dénote la fonction indicatrice de l'ensemble Y_t. Cependant, la discontinuité de G_t dans le cas (2.25) rend la résolution des problèmes avec contraintes de probabilité plus difficile numériquement que lorsque G_t est une fonction continue (possiblement aussi linéaire ou convexe). Le cas des contraintes en espérance apparait par exemple lorsque le décideur cherche à maximiser en espérance plusieurs objectifs et que par choix de modélisation certains sont mis dans les contraintes (voir, p. ex., Ruszczyński et Shapiro (2003c, Exemple 2). Nous référons à Prékopa (2003) pour une description détaillée des problèmes avec contraintes probabilistes.

Les problèmes considérés dans cette thèse : Maintenant que nous avons décrit la plupart des approches existantes en prise de décisions multi-étapes sous incertitude, nous pouvons positionner plus précisément les problèmes que nous considérerons dans cette thèse. Il s'agit des problèmes où :

- le **processus de décision** est celui de l'optimisation stochastique multi-étapes (2.1) où $t = 0, 1, \ldots, T$ avec $T < \infty$; les étapes correspondent à des instants ponctuels arbitraires, la durée réelle entre deux étapes successives peut donc être variable ;
- les **variables de décision** sont en nombre quelconque et peuvent être continues ou entières ;
- le **processus stochastique** contient un nombre quelconque de paramètres aléatoires à chaque étape ;
- le **modèle stochastique** est connu analytiquement (possiblement estimé par des outils statistiques à partir de données historiques) et il n'est pas modifiable par les décisions prises ; les distributions de probabilité peuvent être discrètes ou continues, à support bornée ou non ;
- la **fonctionnelle de probabilité** de la fonction objectif est l'espérance ; il peut s'agir de l'espérance d'une fonction de revenus (neutre au risque) ou d'une fonction d'utilité (averse au risque) ; dans les deux cas cette fonction est notée simplement $q(y; \xi)$;
- la **fonction de revenus** (ou d'utilité) est essentiellement quelconque (elle peut être non-linéaire, non-convexe, non-différentiable, etc.) ;
- les **contraintes** doivent être vérifiées presque sûrement (et non en probabilité ou en espérance).

La formulation fonctionnelle (F-OSM) et dynamique (D-OSM) sont donc les deux formulations qui nous serviront alternativement de référence pour exprimer les problèmes d'optimisation stochastique multi-étapes dans les chapitres suivants. En particulier, le Chapitre 5 contient un ensemble de conditions suffisantes venant compléter la description ci-dessus afin d'assurer que les problèmes soient bien définis mathématiquement.

Nous décrivons maintenant, dans les deux sections à venir, les méthodes et applications utilisées en optimisation stochastique multi-étapes.

2.2 Les méthodes de l'optimisation stochastique multi-étapes

Les différentes étapes menant de la modélisation d'un problème réel à sa résolution sont représentées dans la Figure 2.1. Ces étapes consistent à : (1) approximer le problème original en le discrétisant avec un arbre de scénarios pour le ramener à un problème de dimension fini ; (2) résoudre le problème approché par des méthodes de l'optimisation déterministe ; (3) transformer les décisions optimales du problème approché en politique réalisable pour le problème original et évaluer la qualité de cette dernière.

2.2.1 Génération d'arbres de scénarios

Un *arbre de scénarios* est une représentation du processus stochastique construite à partir d'un sous-ensemble fini de ses réalisation. Celle-ci est caractérisée par (i) une structure en forme d'arbre avec racine composée d'un ensemble (fini) de nœuds et d'arêtes, et (ii) un ensemble de points (sur les nœuds) et de poids (sur les arêtes) qui représentent des scénarios (ou trajectoires) possibles du processus. La nécessité de construire une structure d'arbre est ce qui différence fondamentalement les problèmes *d'optimisation stochastique* des problèmes *d'intégration numérique*. En effet, pour ces derniers les scénarios ont uniquement une origine commune et sont rangés sous forme «de peigne», c'est-à-dire sans lien aux étapes intermédiaires. Dans le cas de l'optimisation stochastique, la structure en forme d'arbre partant de la racine jusqu'aux feuilles modélise l'évolution progressive de l'information à travers les étapes, ce qui a pour but de garantir que les décisions prises ne sont pas (ou peu) anticipatives.

Il existe de nombreuses méthodes de construction d'arbres de scénarios. Celles-ci rentrent généralement dans deux catégories : les premières génèrent uniquement des scénarios étant donné une structure d'arbre fixée à l'avance, tandis que les secondes permettent de générer des scénarios en même temps qu'une structure d'arbre adéquate. Dans la première catégorie, on trouve de nombreuses méthodes provenant du domaine de l'intégration numérique, que nous décrivons maintenant.

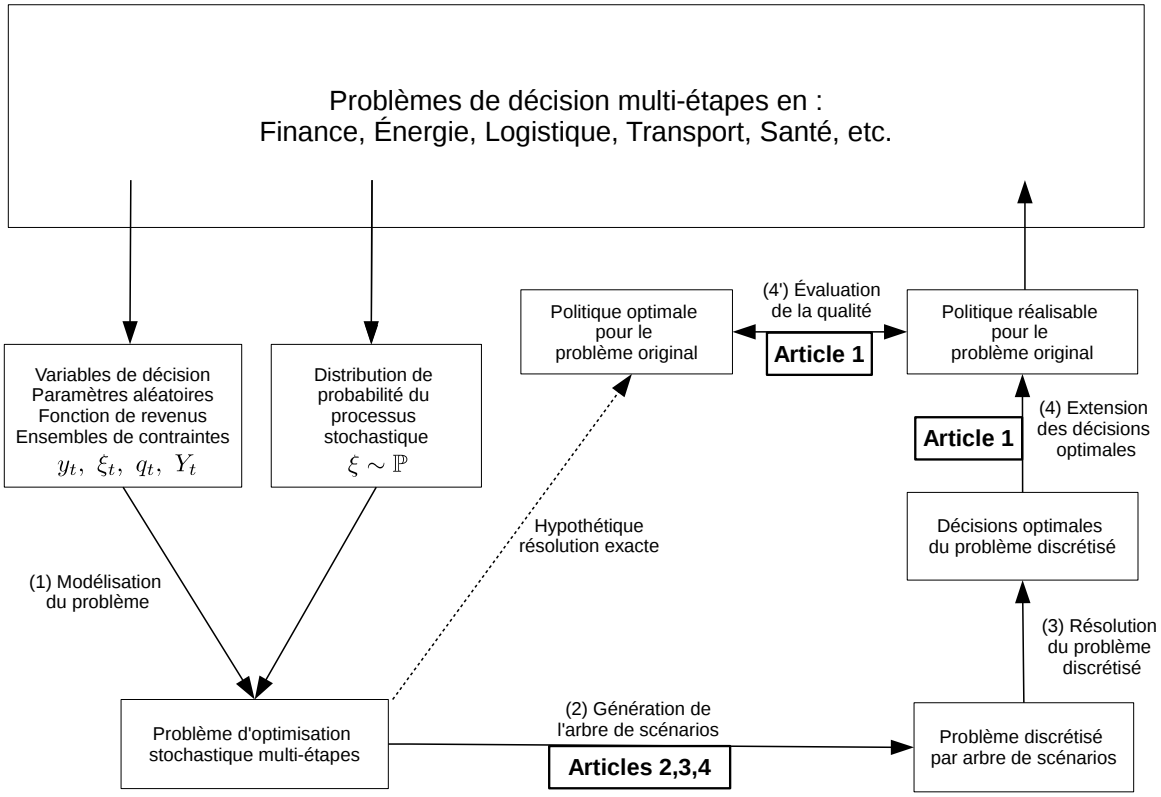


Figure 2.1 Les différentes étapes (1) à (4) de la modélisation à la résolution d'un problème d'optimisation stochastique multi-étapes et positionnement des articles.

Les **méthodes Monte-Carlo (MC)** sont les plus répandues pour générer des scénarios. Elles consistent à échantillonner aléatoirement la distribution de probabilité des paramètres aléatoires. La convergence de l'erreur d'estimation vers zéro (c.-à-d., la consistance de l'estimateur) est assurée sous les hypothèses de la *loi des grands nombres*. Le taux de convergence est donné par $\mathrm{Var}(f)N^{-1/2}$, où $\mathrm{Var}(f)$ est la variance de la fonction à intégrer et N est le nombre de points d'échantillonnage considérés. Celui-ci a l'avantage d'être indépendant du nombre de paramètres aléatoires. Cependant, son inconvénient est qu'il est relativement lent par rapport à un taux en $O(N^{-1})$ (ou plus) pouvant être atteint en faible dimension par des méthodes de discrétisation déterministe. Il existe de nombreuses techniques pour réduire le terme en $\mathrm{Var}(f)$ et ainsi accélérer la convergence de l'erreur (celles-ci sont connues sous le nom de *réduction de variance*). On peut citer, par exemple, celles utilisant des variables de contrôles ou antithétiques, l'échantillonnage d'hypercube latin, stratifié ou préférentiel. Nous référons à Lemieux (2009) pour une présentation générale des méthodes MC dans un cadre général d'intégration numérique.

Dans le cadre des problèmes d'optimisation stochastique, les méthodes MC ont été les premières à être étudiées pour des problèmes à une étape de décision de la forme :

$$\min_{x \in X \subseteq \mathbb{R}^s} \{f(x) = \mathbb{E}[F(x, \xi)]\}. \tag{2.26}$$

Le concept de *sample average approximation (SAA)* a été introduit pour référer au problème d'optimisation échantillonné par MC :

$$\text{(SAA)} \qquad \min_{x \in X \subseteq \mathbb{R}^s} \left\{ \widehat{f}_N(x) = \sum_{i=1}^{N} F(x, \xi^j) \right\}, \tag{2.27}$$

pour un échantillon de N réalisations (scénarios) $\xi^1, \ldots, \xi^N$ tirées de façon indépendante suivant la distribution de ξ. La plupart des propriétés de base de MC (consistance des estimateurs, distribution asymptotique de l'erreur suivant une loi normale) se transportent dans le cadre des problèmes stochastiques ; nous référons à Shapiro *et al.* (2014, Section 5.1) pour une revue générale de ces propriétés. Pour les problèmes multi-étapes, Shapiro (2006) a montré que les méthodes MC appliquées sur des structures d'arbres régulières nécessitent un nombre de scénarios exponentiellement grand en fonction du nombre d'étapes. Pour améliorer la qualité d'estimation, Kouwenberg (2001) propose une procédure (*adjusted random sample*) visant à modifier l'échantillon aléatoire $\xi^1, \ldots, \xi^N$ de sorte à ce qu'il reproduise certains moments de la distribution originale.

Les **méthodes quasi-Monte Carlo (QMC)** cherchent à améliorer le taux de convergence des méthodes MC en considérant des ensembles de points déterminés de façon déterministe

afin de couvrir plus uniformément l'espace d'échantillonnage que ne le fait MC. La mesure d'uniformité est appelée *discrépance* et les points générés par QMC les *points de faible discrépance*. Il existe plusieurs notions différentes de discrépance (p. ex., la *discrépance-étoile* ou la *discrépance-L_2*) et des familles différentes de points de faible discrépance (p. ex., les *règles de réseau* ou les *réseaux digitaux*). La mesure de discrépance d'un ensemble de points est reliée au concept d'*erreur pire-cas* pour l'intégration de certaines fonctions dans des espaces de Hilbert à noyau reproduisant, comme expliqué dans Dick et Pillichshammer (2010, Chapitre 2). Le taux de convergence de QMC est généralement en $O(N^{-1}(\log N)^d)$, où d est le nombre de paramètres aléatoires, ce qui signifie que, contrairement à MC, ces méthodes sont sensibles à la dimension d. On note que $N^{-1}(\log N)^d$ décroit *asymptotiquement* vers zéro plus rapidement que $N^{-1/2}$ pour tout $d \geq 1$, mais que ce comportement asymptotique cache en réalité un inconvénient majeur pour des ensembles finis de points : la dépendance en $(\log N)^d$ implique que le comportement asymptotique apparait en pratique qu'à partir de valeurs N extrêmement grandes. Par exemple, pour $d = 5$, on a que $N^{-1}(\log N)^d < N^{-1/2}$ seulement à partir de $N \geq 10^{15}$, ce qui rend impraticable les méthodes QMC standards en haute dimension. Pour résoudre ce problème, Sloan et Woźniakowski (1998) ont introduit la notion d'espaces *pondérés* de Hilbert à noyaux reproduisant afin de supprimer la dépendance en d et obtenir un taux arbitrairement proche de $O(N^{-1})$. L'appartenance d'une fonction f à ce type d'espace signifie en pratique que la variabilité de f suivant les axes de haute dimension diminue fortement à mesure que le nombre de dimension augmente.

Il est aussi possible de considérer des ensembles de points de faible discrépance *randomisé* (RQMC). La randomisation permet d'obtenir un estimateur non-biaisé (comme dans le cas MC) tout en gardant une uniformité supérieure à MC. Cela permet d'avoir des taux de convergence en $O(N^{-3/2}(\log N)^c)$ (pour un certain $c > 0$) et même $O(N^{-\alpha}(\log N)^c)$ avec $\alpha > 1$ arbitrairement grand pour des fonctions suffisamment lisses ; nous référons à Dick et Pillichshammer (2010) et Dick *et al.* (2013) pour une présentation générale des méthodes QMC.

Dans le cadre de l'optimisation stochastique, Koivu (2005) montre que sous certaines conditions les méthodes QMC garantissent l'épi-convergence de la fonction objectif discrétisée d'un problème statique à une étape vers la fonction objectif du problème original. L'épi-convergence (aussi étudiée dans Pennanen et Koivu (2002)) implique alors que les valeur et solution optimales du problème discrétisé convergent presque sûrement vers leurs équivalents exacts. Koivu (2005) applique les méthodes QMC randomisé à des problèmes à une étape de gestion de portefeuille contenant 10 actifs. En comparant avec MC et plusieurs techniques de réduction de variance, ses résultats numériques montrent que QMC randomisé permet de réduire de façon significative la variance et le biais de la valeur optimale. Homem-de Mello

(2008) analyse de façon théorique l'effet des méthodes QMC sur les taux de convergence des estimateurs de la valeur et de la solution optimales. Il conclut que, sous certaines conditions, ces taux sont identiques à ceux de l'estimateur point par point de la fonctions objectif. En d'autres termes, si l'erreur d'approximation de l'espérance est de l'ordre de $O(N^{-1})$, alors il est possible d'espérer que les erreurs d'approximation de la valeur et de la solution optimales du problème convergent à des taux similaires.

Pour des problèmes à plus haute dimension, Drew et Homem-de Mello (2006) proposent d'utiliser les méthodes QMC pour échantillonner un certain sous-ensemble de variables et de combiner cela avec un échantillonnage MC pour les autres variables. Cela fait écho à la notion de *dimension effective*, qui correspond (informellement) au nombre de variables contiennent la grande majorité de la variabilité totale de la fonction. Les auteurs proposent deux heuristiques pour déterminer les variables les plus importantes du problème (celles avec le plus de variabilité). Leurs résultats numériques montrent que cette technique combinée QMC-MC augmente la qualité d'échantillonnage. La taille de leurs problèmes est cependant relativement petite (3 à 5 variables aléatoires).

La question de savoir si les méthodes QMC peuvent être efficaces pour des problèmes d'optimisation stochastique à grand nombre de paramètres aléatoires est analysée par Leövey et Römisch (2015) et Heitsch *et al.* (2016) dans le cadre des problèmes à deux étapes. Les auteurs montrent qu'il est possible de réduire la dimension effective du problème par des techniques comme la factorisation de Cholesky ou l'analyse par composante principale, afin d'obtenir des taux de convergence proches de $O(N^{-1})$. Leurs expériences numériques utilisent des méthodes QMC randomisé (*scrambled Sobol* et *randomly shifted lattice rules*) pour résoudre un problème de planification de la production avec 100 et 200 variables aléatoires. Les résultats montrent que les taux de convergence de ces deux méthodes sont largement supérieurs à MC. A notre connaissance, une telle analyse n'a pour l'instant pas été menée dans le cas de problèmes multi-étapes.

Les **méthodes de quantification optimale (OQ)**, tout comme les méthodes QMC, ont pour but de générer des ensembles de points déterministes qui couvrent plus uniformément l'espace d'échantillonnage que MC. Contrairement à QMC, les méthodes OQ cherchent à remplacer la distribution de probabilité originale par une distribution de taille inférieure *optimale*, dans le sens de celle qui minimise une certaine distance entre distributions de probabilité. Cette approche a des origines dans plusieurs domaines de recherche, comme la théorie de l'information et le partitionnement des données. Nous référons à Graf et Luschgy (2007) pour une introduction générale à ce sujet. Dans le cadre de l'intégration numérique, elle est considérée à l'origine par Pagès (1998) et est appliquée à des problèmes de finance par

Pagès *et al.* (2004). Le taux de convergence des méthodes OQ pour des fonctions de classe $\mathcal{C}^\alpha$ est généralement en $O(N^{-\alpha/d})$, ce qui signifie que celui-ci est sensible au nombre d de variables et nécessite que les fonctions soient suffisamment régulières (comme pour QMC). Contrairement aux méthodes QMC, dont les poids associés aux points sont égaux et de valeur $1/N$, les méthodes OQ cherchent à la fois les points et les poids (les probabilités) de la distribution optimale. Cette caractéristique confère aux méthodes OQ un avantage par rapport à QMC (et MC) dans le cas d'ensembles de petites tailles N. De plus, les points et poids obtenus par OQ sont calculables pour n'importe quelle valeur de N, alors que les méthodes QMC ont parfois des garanties de qualité seulement pour des valeurs spécifiques de N (comme les puissances $N = 2^i$, $i = 1, 2 \ldots$, pour les réseaux digitaux).

Les méthodes OQ ont été considérées dans le cadre de l'optimisation stochastique par Pflug (2001). L'auteur utilise la *distance de Wasserstein d'ordre 1*, W_1, définie par

$$W_1(P,Q) = \inf_{(X,Y)} \{\mathbb{E}[\|X-Y\|] : X \sim P,\, Y \sim Q,\, (X,Y) \text{ de distribution quelconque}\}\,. \tag{2.28}$$

Cette distance est reliée au problème de transport optimal de masse étudié à l'origine par Monge (1781), puis par Kantorovitch (1958), et aujourd'hui appelé *problème (du transport) de Monge-Kantorovitch*. L'intérêt de cette distance, dans le cadre de l'intégration numérique et des problèmes d'optimisation stochastique, réside dans le fait qu'elle possède une définition duale équivalente qui la relie à la notion d'erreur d'intégration pire-cas dans certains espaces de fonctions (les espaces de fonctions lipschitziennes). Il s'agit du *théorème de Kantorovitch et Rubinstein* (Kantorovich et Rubinstein, 1958) qui permet d'écrire :

$$W_1(P,Q) = \sup_f \left\{ \int f(x)P(dx) - \int f(x)Q(dx) : L_1(f) \leq 1 \right\}, \tag{2.29}$$

où $L_1(f)$ est la constant de Lipschitz f. Cette définition permet d'avoir des garanties en termes d'erreur entre la fonction objectif exacte et discrétisée comme illustré dans l'article Pflug (2001). Ces définitions se généralisent facilement au cas de fonctions f définies sur des espaces plus généraux, comme sur des espaces métriques muni d'une distance quelconque $d(\cdot,\cdot)$. Dans l'article Hochreiter et Pflug (2007), les auteurs montrent que le problème de minimisation de la distance de Wasserstein se ramène à un problème de *facility location* ou de *nested facility location* (pour les problèmes multi-étapes) pour lequel des approches heuristiques existent. D'autres familles de distances peuvent aussi être utilisées en optimisation stochastique, nous référons à Pflug et Pichler (2011) pour une description générale à ce sujet.

Les **méthodes de correspondance des moments** (*moment-matching*) génèrent des ensembles de points et de poids qui ont un certain nombre de propriétés statistiques identiques

à la distribution originale. Généralement, il s'agit des quatre premiers moments (moyenne, écart-type, skewness, kurtosis) et de la matrice de covariance. Cette approche est développée par Høyland et Wallace (2001) etHøyland *et al.* (2003) dans le cadre des problèmes d'optimisation stochastique multi-étapes. L'intérêt de cette approche réside dans le fait qu'en pratique la distribution du processus stochastique peut être difficilement estimable, alors que certaines propriétés comme la moyenne ou la covariance peuvent facilement l'être à partir de données historiques. Les méthodes de correspondance des moments permettent alors de générer des arbres de scénarios qui vont à minima reproduire ces propriétés. Elles sont donc utiles dans un cas où le décideur a une connaissance imparfaite du modèle stochastique décrivant son environnement. De plus, pour certains problèmes spécifiques, il est connu que les solutions optimales dépendent uniquement de certaines propriétés statistiques de la distribution, comme la moyenne et la covariance dans le problème de gestion de portefeuille de Markowitz (1952). Cependant, l'inconvénient de cette méthode, soulevé par exemple par Pflug et Pichler (2011), est que la connaissance d'un certain nombre de moments (même infini) ne permet pas de caractériser de façon unique une distribution de probabilité. Par conséquent, dans un cas général il n'y a pas de garanti de convergence vers zéro de l'erreur (c.-à-d., de consistance de la méthode).

Nous décrivons maintenant les méthodes appartenant à la seconde catégorie, c.-à-d., celles qui génèrent des scénarios en même temps qu'une structure d'arbre.

Les **méthodes de réduction de scénarios** génèrent des arbres de scénarios selon la stratégie suivante : à partir d'un ensemble fini de scénarios (potentiellement très grand), elles combinent (réduisent) ces derniers entre eux afin d'obtenir un arbre de scénarios de taille donnée qui minimise une certaine distance entre processus stochastiques. Les scénarios originaux peuvent être générés à partir d'un modèle stochastique ou simplement provenir de données historiques, et ils peuvent être sous forme de peigne (avec uniquement la racine de commun) ou d'arbre. On distingue deux familles de réduction de scénarios basées sur des distances distinctes : la distance entre filtrations (*filtration distance*) (Heitsch et Römisch, 2009) et la distance *nested* (Pflug et Pichler, 2012). Il est important de noter qu'il s'agit de distances entre *processus stochastiques* (donc multi-étapes), à l'inverse des distances précédentes qui étaient entre *distributions de probabilité* (donc à une étape donnée). La distance *nested*, par exemple, est construite en tant que généralisation multi-étapes de la distance de Wasserstein. La consistance de l'approximation par arbres de scénarios est assurée, sous certaines hypothèses, par des théorèmes de *stabilité multi-étapes*. Ces résultats (cf. Heitsch et Römisch (2009, Théorème 3.1) et Pflug et Pichler (2012, Théorème 11)) garantissent que la valeur optimale du problème discrétisé converge vers celle du problème originale à mesure que la distance entre le processus original et celui induit par l'arbre de scénarios converge vers zéro.

Les méthodes de réduction de scénarios permettent de générer des structures d'arbres générales, donc potentiellement des structures non-symétriques plus adaptées à l'approximation du processus stochastique original.

Les **méthodes d'échantillonnage préférentiel séquentiel** (EPS) peuvent également être utilisées pour générer des arbres de scénarios non-symétriques. L'EPS, utilisé à l'origine pour l'échantillonnage de processus stochastique, agit comme l'échantillonnage conditionnelle à la différence que les distributions suivant lesquelles les réalisations sont tirées diffèrent des distributions conditionnelles. Le but est alors d'accroitre la qualité d'échantillonnage, c.-à-d., d'avoir des scénarios plus représentatifs qui exploitent les propriétés de la distribution et du problème d'optimisation. Cette procédure est considérée à l'origine par Chen *et al.* (1998); Dempster et Thompson (1999); Dupačová *et al.* (2000). Elle est appliquée à un problème de finance par Dempster (2006). L'EPS nécessite un critère pour guider la génération de la structure de l'arbre. Un critère pertinent, d'après les travaux ci-dessus, est l'EVPI (*expected value of perfect information*). Le critère EVPI mesure, à chaque nœud de l'arbre, l'importance de l'information aléatoire future dans la prise de décisions présente. Si l'EVPI est élevé, cela signifie que les réalisations aléatoires futures influencent énormément les décisions présentes. Il est alors nécessaire d'avoir une structure qui branche beaucoup à ce nœud, afin que la description du futur soit précise. A l'inverse, si l'EVPI est faible, alors les réalisations futures ont peu (ou pas) d'influence sur les décisions présentes et il n'est alors pas nécessaire de brancher à partir de ce nœud, un seul scénario sortant peut être suffisant. Le critère EVPI ne pouvant se calculer qu'à partir d'une solution donnée, cette méthode de génération d'arbres de scénarios est itérative : elle démarre d'une structure spécifiée arbitrairement et la raffine au fur et à mesure des itérations en se basant sur les nouvelles valeurs d'EVPI. Dans ce sens, elle peut être coûteuse en temps de calcul. La consistance de la méthode est discutée dans Dempster (2006, Section 5).

Il existe d'autres familles de méthodes qui génèrent itérativement une *suite* d'arbres de scénarios, dont chaque élément est une version plus raffinée du précédent. Par exemple, Frauendorfer (1996) considère deux suites d'arbres de scénarios, appelés *barycentriques*, bornant inférieurement et supérieurement la valeur optimale du problème original. L'auteur prouve que, sous certaines conditions, ces approximations successives convergent vers la valeur optimale du problème original. Un autre type d'algorithme de raffinement d'arbres est proposé par Edirisinghe (1999). Cet algorithme agrège les contraintes de non-anticipativité afin d'obtenir une suite de problèmes discrétisés bornant inférieurement le problème original. Dans Casey et Sen (2005), un algorithme qui construit une suite de problèmes menant à la solution optimale originale est proposé pour une certaine classe de problèmes multi-étapes linéaires. Leur méthode à l'avantage de fournir des politiques de décision pour le problème original qui

étendent les décisions obtenues avec l'arbre de scénarios, et dont il est possible de calculer la probabilité de réalisabilité.

2.2.2 Résolution du problème discrétisé

Dans cette partie, nous décrivons l'idée générale de deux des algorithmes les plus utilisés pour résoudre le problème multi-étapes discrétisé par arbre de scénarios : l'algorithme *nested L-shaped* (ou *nested Benders decomposition*) et l'algorithme de *progressive hedging*. Pour cela, nous nous plaçons dans le cadre de l'optimisation convexe (c.-à-d., la fonction objectif est convexe en fonction des décisions et l'ensemble des contraintes aussi). Nous référons à Birge et Louveaux (2011, Chapitre 6) et Ruszczyński (2003) pour une description plus détaillée ainsi que pour la présentation d'autres méthodes.

L'algorithme **nested L-shaped**, introduit par Louveaux (1980) dans le cas quadratique et par Birge (1985) dans le cas linéaire, est une extension de la méthode L-shaped introduite pour les problèmes linéaires stochastiques à deux étapes par Van Slyke et Wets (1969). Cette dernière est elle-même une application de la méthode des *plans coupants* (*cutting plane*) proposée par Benders (1962) pour résoudre des problèmes linéaires (déterministes) à deux étapes.

L'idée de l'algorithme est de construire itérativement à chaque nœud de l'arbre une approximation de la fonction de recours (2.9) à l'aide de *coupes d'optimalité* ainsi qu'une approximation de l'ensemble réalisable à l'aide de *coupes de réalisabilité*. Ces coupes fournissent des approximations linéaires successives qui vont, après un nombre fini d'itérations, décrire exactement la fonction de recours et l'ensemble réalisable, de par la structure polyédrale de ces derniers. Par conséquent, il est prouvé que cet algorithme converge après un nombre fini d'itérations vers la solution optimale du problème si celle-ci existe, ou alors conclut à la non-réalisabilité du problème.

Un inconvénient de cet algorithme réside dans le fait que l'ensemble des coupes devant être stockées peut rapidement devenir très grand si l'arbre de scénarios contient beaucoup de nœuds, ce qui est généralement le cas pour les problèmes à grand nombre d'étapes. Pour remédier à cela, il est possible d'utiliser des techniques de *partage de coupes* (*cut sharing*). Celles-ci sont basées sur l'idée que deux fonctions de recours à la même étape mais correspondant à des scénarios différentes sont identiques –et donc partagent les mêmes coupes– si la discrétisation sortant de ces scénarios est identique. En d'autres termes, si deux nœuds ont des sous-arbres identiques, alors leur coupes peuvent être partagées. Le plus grand partage de coupes se produit lorsque le processus stochastique est indépendant entre les étapes, c.-à-d., lorsque $(\xi_{t+1}, \ldots, \xi_T)$ est indépendant de $(\xi_0, \ldots, \xi_t)$ pour tout t. Dans ce cas, toutes les fonc-

tions de recours à une même étape sont identiques. Cela permet d'avoir un seul ensemble de coupes à chaque étape partagé par toutes les fonctions de recours, et ainsi d'accélérer significativement la convergence de la méthode. Cette technique apparait dans l'approche SDDP (*stochastic dual dynamic programming*) proposée par Pereira et Pinto (1991) pour résoudre des problèmes de planification à grand nombre d'étapes.

L'algorithme de **progressive hedging** a été introduit par Rockafellar et Wets (1991). Il consiste à relaxer les contraintes de non-anticipativité (liants les scénarios) au sein d'un Lagrangien *augmenté* par un terme quadratique, de sorte à avoir des sous-problèmes séparables par scénarios. Les sous-problèmes –qui sont déterministes car exprimés pour un scénario donné– sont alors résolus indépendamment les uns des autres. Les solutions optimales, qui sont anticipatives car dépendant des scénarios et non des nœuds, sont alors corrigées de sorte à ce que les décisions provenant de scénarios indistinguables jusqu'à une certaine étape deviennent identiques. Cette correction est faite en projetant orthogonalement les vecteurs de décision sur le sous-espace linéaire des politiques non-anticipatives. Contrairement, cela est fait en prenant l'espérance conditionnelle de toutes les décisions provenant de scénarios indistinguables jusqu'à une certaine étape. Puis, les multiplicateurs de Lagrange sont mis à jour à l'aide d'un paramètre de pénalité et les sous-problèmes sont résolus à nouveau. Sous certaines conditions, cet algorithme converge vers la solution optimale.

L'avantage de cet algorithme est qu'il décompose le problème discrétisé en autant de sous-problèmes indépendants qu'il existe de scénarios. Ces problèmes peuvent alors être résolus parallèlement sur différents processeurs afin d'accélérer substantiellement la convergence de la méthode. Son inconvénient pratique est que l'algorithme n'offre pas d'ajustement naturel pour le paramètre de pénalité à chaque itération. La question de la parallélisation, de l'ajustement du paramètre de pénalité et de leurs effets sur les performances de l'algorithme est discuté par exemple par Mulvey et Vladimirou (1991) dans le cadre des réseaux généralisés stochastiques.

2.2.3 Mise en pratique des décisions de l'arbre de scénarios

Après la génération de l'arbre de scénarios et la résolution du problème discrétisé, l'étape suivante dans la mise en pratique de l'optimisation stochastique multi-étapes (cf. Figure 2.1) consiste à transformer les décisions optimales de l'arbre en politique réalisable pour le problème original et à analyser la qualité de cette dernière. Cette étape n'est pas triviale. En effet, les décisions de l'arbre (autres que celles du nœud racine) dépendent des scénarios choisis. En pratique, ces scénarios ont de grandes chances de ne jamais apparaitre car, si le processus a une distribution continue, alors l'ensemble des scénarios est de probabilité nulle. Pour contourner ce problème, certains auteurs comme Kouwenberg (2001), Chiralaksanakul

et Morton (2004) et Hilli et Pennanen (2008) recommandent de mettre en pratique uniquement les décisions du nœud racine, et de résoudre à nouveau le problème discrétisé à chaque nouvelle étape. Cet type de mise en pratique est dit en *horizon fuyant* ou en *horizon rétrécissant*[3] suivant que l'horizon T soit repoussé d'une étape (le problème est toujours à $T+1$ étapes) ou réduit d'une étape (le problème est alors à $T-t+1$ étapes) à chaque nouvelle résolution. Cette approche génère une politique de décision pour le problème original dont la qualité peut être estimable. Pour cela, il suffit de générer un ensemble de scénarios indépendants tirés directement depuis la distribution du processus (*out-of-sample*) et de calculer la moyenne des revenus obtenus par horizon roulant ou rétrécissant pour chaque scénario.

Cette approche a cependant deux inconvénients. Tout d'abord, elle ne garantit pas la réalisabilité «presque sûre» de la politique, à moins que le problème vérifie la propriété de *recours relativement complet*. Celle-ci offre la garantie que pour chaque décision réalisable à une étape donnée, il existe des décisions réalisables aux étapes suivantes quelles que soient les réalisations du processus. En d'autres termes, sans cette propriété, la suite de décisions obtenues par horizon fuyant/roulant jusqu'à l'étape t peut mener à un «cul-de-sac», c.-à-d., l'inexistance de décision réalisable à $t+1$. Un autre inconvénient majeur est la nécessité de résoudre à nouveau le problème pour chaque scénario. Une estimation *out-of-sample* précise nécessite la génération de plusieurs centaines ou milliers de scénarios, ce qui peut être extrêmement coûteux en temps de calcul.

Une autre approche, développée par Defourny *et al.* (2013), consiste à reconstruire intégralement (sur l'ensemble de tous les paramètres aléatoires) une politique de décision pour le problème original à partir des décisions optimales de l'arbre. Dans cet article, l'auteur utilise des méthodes de régression provenant du domaine de l'apprentissage machine pour construire une fonction de décision à partir d'un échantillon fini de vecteurs de décision. Plusieurs arbres de scénarios sont générés afin d'obtenir plusieurs candidates pour la meilleure politique. Celle-ci est alors déterminée en fonction des revenus qu'elle fournit en espérance, estimés par un ensemble de scénarios indépendants. A l'inverse de l'approche par horizon fuyant/rétrécissant, dans ce cas l'estimation *out-of-sample* n'est pas coûteuse puisque les décisions sont accessibles pour chaque scénario simplement en évaluant une fonction en un point et non en résolvant à nouveau le problème d'optimisation.

Dans un problème de maximisation de l'espérance des revenus, chaque politique réalisable du problème original fournit des revenus inférieurs (en moyenne) à ceux de la politique optimale. Dans ce sens, chaque politique fournit une borne *inférieure* (plus ou moins bonne) sur la valeur optimale du problème. Afin d'estimer la qualité de cette borne, et ainsi d'en déduire la

[3]En anglais : *rolling horizon* et *shrinking horizon*, respectivement.

qualité de la politique correspondant, Mak *et al.* (1999) propose d'utiliser le fait que la valeur optimale du problème discrétisé par MC est (en moyenne) une borne *supérieure* sur celle du problème original. Ainsi, en estimant la distance entre les deux bornes, il est possible de déduire (avec un certain intervalle de confiance) le manque de revenus d'une politique donnée par rapport à l'optimalité (c.-à-d., le *gap* d'optimalité). Cette approche est aussi développée par Bayraksan et Morton (2009).

Ceci montre que pour estimer la qualité d'une politique de décision donnée (qu'elle soit obtenue à l'aide d'un arbre de scénarios ou tout autre méthode), il est nécessaire d'avoir une estimation précise de la valeur optimale *exacte* du problème original. Il est donc naturel de demander que, en plus de fournir des bonnes décisions, l'arbre de scénarios fournisse aussi une estimation précise de cette valeur optimale. La plupart des méthodes de génération d'arbres sont basées sur l'idée de développer une discrétisation qui minimise autant que possible l'erreur entre les valeurs optimales exacte et approchée. Lorsqu'il est possible d'estimer cette erreur (ou sa distribution si l'erreur est aléatoire), il est possible de déduire, pour un problème donnée, quelle est la meilleure méthode, pour ce qui concerne l'estimation de la valeur optimale. Lorsque ce n'est pas possible, Kaut (2012) propose d'estimer la pertinence d'une méthode à travers sa stabilité interne (*in-sample*). Une méthode a une stabilité interne si, lorsqu'elle est utilisée de façon répétée, elle génère des problèmes discrétisés dont les valeurs optimales sont relativement proches les unes des autres. En effet, une méthode qui mène à des valeurs optimales très différentes (en termes d'écart-type par exemple) n'offre aucune garantie en pratique. Ceci concerne les méthodes *stochastiques*, qui génèrent des arbres différents à chaque fois. Pour les méthodes déterministes (qui génèrent toujours le même arbre), Kaut (2012) propose de comparer les valeurs optimales obtenues avec des structures légèrement différentes (avec plus ou moins de scénarios).

2.3 Les applications de l'optimisation stochastique multi-étapes

Les méthodes de l'optimisation stochastiques ont été appliquées avec succès dans des domaines très variés. Dans la suite nous décrivons en particulier des applications en finance, énergie, logistique et santé.

En finance, une des premières applications à succès est le modèle de gestion actif-passif (GAP) connu sous le nom de *Russell-Yasuda Kasai* (Carino *et al.*, 1994). Dans ce modèle, des décisions d'investissement sont prises par une entreprise d'assurance japonaise afin de produire un flux de revenus qui va contre-balancer le flux passif dont elle fait face à travers ses politiques d'assurance. Ce dernier est aléatoire de même que les retours sur investissement. Carino *et al.* (1994) expriment ce modèle sous forme d'un problème d'optimisation stochas-

tique à six étapes avec une fonction de coûts polyédrale (convexe linéaire par morceaux). Le problème est résolu par un arbre de scénarios avec 256 ($8 \times 4^2 \times 2 \times 1$) scénarios et un facteur de branchement décroissant : 8,4,4,2,1. Les scénarios sont générées soit aléatoirement (avec ou sans dépendance entre les étapes), soit déterminés manuellement par le décideur. La réduction des scénarios sous forme d'arbre se fait en garantissant que la moyenne et l'écart-type correspondent à ceux des distributions originales. Au final, les auteurs reportent que la stratégie d'investissement obtenue par le modèle *Russell-Yasuda Kasai* fournit des revenus supplémentaires de 79 millions de dollars dans les deux premières années d'utilisation. Cela s'explique par la capacité du modèle à réagir de façon dynamique aux événements aléatoires, à l'inverse des modèles statiques de gestion de portefeuilles.

D'autres modèles GAP ont depuis été développés : Consigli et Dempster (1998) proposent un modèle générique de gestion actif-passif nommé CALM (pour *computer-aided asset/liability management*) et l'appliquent à un problème de fond de pension. Leurs arbres de scénarios ont 10 étapes et leur taille varie de 16 ($2^4 \times 1^5$) à 2688 ($7 \times 3 \times 2^7$) scénarios générés indépendamment ou par échantillonnage conditionnel. Kouwenberg (2001) développe un modèle GAP pour un fond de pension hollandais. L'auteur utilise un seul arbre de scénarios à 6 étapes et 5760 ($10 \times 6^2 \times 4^2$) scénarios générés par trois procédures différentes : échantillonnage aléatoire, échantillonnage ajusté (avec variables antithétiques et ajustement à l'écart-type), et correspondance des moments (moyenne et matrice de covariance). Les performances de la méthode sont estimées par horizon roulant. La méthode de correspondance des moments est aussi utilisée par Topaloglou *et al.* (2008a) pour un problème de gestion de portefeuille international. Leur arbre de scénarios à 3 étapes et 15000 (150×100) scénarios générés de sorte à reproduire les quatre premiers moments de la distribution et les corrélations. Plus récemment, une comparaison de plusieurs méthodes de génération d'arbres de scénarios a été réalisée par Oliveira *et al.* (2018) dans le cadre des problèmes de GAP. Nous référons à ces articles pour plus détails. Pour une revue plus complète des problèmes de finance traités par optimisation stochastique multi-étapes, nous référons par exemple à Yu *et al.* (2003), Ziemba (2003) et Consigli *et al.* (2017).

Notons que la génération d'arbres de scénarios pour des problèmes de gestion de portefeuilles financiers est contraint par des conditions de *non-arbitrage* (Klaassen, 1997, 2002). Celles-ci imposent des restrictions sur la structure d'arbre et les scénarios utilisés, comme par exemple un nombre de branchements minimum par nœud (Geyer *et al.*, 2010). Pour contourner ce problème, une méthode de réduction de scénarios permettant de maintenir l'absence d'arbitrage a été développée par exemple par Klaassen (1998). Cette méthode est cependant critiquée par Geyer *et al.* (2010) pour l'erreur qu'elle introduit dans le calcul du portefeuille optimal. Dans le cadre du marché de l'électricité, Kovacevic (2018) développe des conditions néces-

saires et suffisantes pour garantir l'absence d'arbitrage et analyse leurs implications dans la construction d'arbres de scénarios et l'évaluation de contrat.

Pour ce qui concerne les problèmes d'évaluation d'options avec un droit d'exercice continue (américaines) ou discret (bermudiennes), de nombreuses méthodes de génération d'arbres de scénarios *recombinants* ont été développées. En effet, le prix d'une option peut se représenter sous la forme d'un problème de temps d'arrêt, qui consiste à trouver la politique d'exercice (l'arrêt du contrat) qui maximise l'espérance des revenus. Cette espérance est calculé suivant la distribution *neutre au risque* de l'actif sous-jacent de sorte à garantir l'absence d'*arbitrage* (Harrison et Kreps, 1979). Cette propriété des problèmes d'évaluation d'option permet de considérer des structures de discrétisation quelconques, contrairement aux problèmes de gestion de portefeuilles où l'espérance est calculée suivant la distribution *réelle* de l'actif, et est donc sujet à arbitrage. Deux exemples de structures recombinantes utilisées pour l'évaluation d'options américaines est l'*arbre binomial* de Cox *et al.* (1979) et la méthode de *mesh* stochastique de Broadie et Glasserman (2004). Le caractère recombinant permet de réduire la complexité de la méthode en fonction du nombre d'étapes. En effet, le nombre de nœuds des arbres binomiaux augmente de façon quadratique et celui des *meshs* de façon linéaire. Cependant, elle ne peuvent être utilisées telles quelles que lorsque les décisions sont indépendantes du chemin suivi par le processus stochastique. C'est le cas par exemple des options *vanilles*, mais cela n'est plus vrai pour les options *exotiques* (p. ex., *lookbacks* ou asiatiques). Pour une revue des méthodes numériques pour l'évaluation d'option, nous référons à Glasserman (2003), Hull (2017) et Brandimarte (2013).

Dans le domaine de l'énergie, de nombreux travaux ont été menés en optimisation multi-étapes de la production d'électricité et en gestion de portefeuille d'énergie. Par exemple, Growe-Kuska *et al.* (2003) appliquent les méthodes de réduction de scénarios pour discrétiser les prix spots du marché et la charge électrique pour le problème de gestion de portefeuille d'un fournisseur d'électricité allemand. Fleten *et al.* (2002) considèrent un problème similaire dans la région nordique (Danemark, Finlande, Norvège et Suède). Dans leur modèle, l'incertitude est présente dans les prix et les apports naturels en eaux dans les réservoirs. Les arbres de scénarios qu'ils utilisent ont cinq étapes (sur un horizon de deux ans) et 256 (4^4) scénarios générés par correspondance des moments (les trois premiers et les corrélations). De façon générale, nous référons aux ouvrages de Rebennack *et al.* (2010), Bertocchi *et al.* (2011), et Kovacevic *et al.* (2013) qui regroupent de nombreuses applications et développements méthodologiques portant sur les modèles multi-étapes en énergie.

Les modèles d'optimisation stochastique multi-étapes appliqués à la logistique sont relativement moins fréquents qu'en finance ou en énergie. Cela peut être due au fait que les problèmes

rencontrés en logistique sont essentiellement en variables binaires et entières, ce qui augmente substantiellement la complexité de résolution de leur version stochastique multi-étapes, en comparaison des versions déterministes ou à deux étapes. Cela peut aussi provenir du fait qu'il est généralement difficile de modéliser l'environnement stochastique d'un problème de logistique, à l'inverse des problèmes de finance ou d'énergie qui possèdent généralement des bases de données historiques et des communautés de recherche travaillant à la modélisation de leur aléa. Cet inconvénient peut mener plus naturellement à des formulations robustes ou à deux étapes. De nombreuses applications existent cependant. On peut citer par exemple les problèmes de *capacity planning* (Ahmed et Sahinidis, 2003; Huang et Ahmed, 2009), *supply chain design* (Nickel *et al.*, 2012; Xie et Huang, 2018), *lot sizing* (Brandimarte, 2006; Varas *et al.*, 2018), *vehicle routing* (Dror, 1993; Hvattum *et al.*, 2006). Nous référons à ces articles et aux références qu'ils contiennent pour une revue plus complète des applications en logistique. Nous référons aussi à Powell et Topaloglu (2003) pour une introduction sur les approches de programmation stochastique en logistique et transport.

En santé, Colvin et Maravelias (2008) et Zeng et Cremaschi (2017) appliquent les méthodes d'optimisation stochastique multi-étapes pour résoudre des problèmes de planification d'études cliniques. Dans leur modèle, l'incertitude porte sur le résultat des tests (réussite ou échec à trois phases possibles du test). Leur arbres de scénarios ont 13 et 11 étapes, respectivement.

CHAPITRE 3 DÉMARCHE ET ORGANISATION DE LA THÈSE

Le positionnement des quatre articles dans le cadre général de l'optimisation stochastique multi-étapes est représenté dans la Figure 2.1. Nous décrivons brièvement leur contenu ci-dessous.

3.1 Premier article

Le premier article est indépendant des trois autres. Il se place à la fin de la chaîne d'étapes qui mènent de la modélisation d'un problème à sa résolution pratique. Concrètement, cet article s'intéresse à la mise en pratique des décisions obtenues à l'aide d'arbres de scénarios. Nous considérons, en tant qu'agent décideur, que nous avons en notre possession différentes méthodes de génération d'arbres de scénarios et nous souhaitons répondre à deux problématiques pratiques :

(i) Comment construire une politique de décision à partir des décisions qu'elles fournissent ?

(ii) Comment évaluer la qualité de cette politique, si possible en comparaison de la politique optimale originale ?

Pour répondre à ces deux questions, nous développons un cadre mathématique, basé sur le concept de *procédure d'extension*, qui permet à l'agent décideur de comparer des méthodes différentes dans un sens pertinent d'un point de vue pratique afin de choisir la plus adaptée au problème.

3.2 Deuxième article

Le deuxième article fournit la pierre angulaire de la nouvelle méthode de génération d'arbres de scénarios que nous développons. Dans un premier temps, nous montrons, à travers plusieurs résultats théoriques, que la variabilité des fonctions de recours optimal est une caractéristique importante du problème qui influence la capacité d'approximation de l'arbre de scénarios. Dans un second temps, nous développons une nouvelle approche de génération d'arbres qui utilise la connaissance de la distribution de variabilité à travers les étapes et les scénarios pour construire des arbres adaptés aux problèmes. Les structures d'arbres considérées sont cependant contraintes d'avoir un certain nombre de nœuds par étape fixé à l'avance, ce qui n'est pas totalement satisfaisant au vu de la question (2.2) de la Section 1.3. Nous montrons, à travers des résultats numériques préliminaires, que les arbres obtenus peuvent grandement réduire l'erreur d'approximation par rapport à des arbres à structure symétrique.

3.3 Troisième article

Le troisième article poursuit le développement théorique du deuxième. Dans un premier temps, il introduit les deux concepts fondamentaux que nous avions décrits dans les objectifs de cette recherche (cf. Section 1.3). Le premier est le concept de *set of guidance functions* qui représente la quantité $\Gamma(p)$ contenant quantitativement l'information nécessaire à la génération d'arbre de scénarios pour le problème p. Le deuxième est celui de *figure of demerit* qui représente la mesure $\mathcal{M}$ d'adéquation entre le problème et l'arbre de scénarios. Nous montrons que cette mesure est en effet pertinente pour la génération d'arbres de scénarios car elle constitue une borne sur l'erreur de valeur optimale. Ce résultat peut être interprété similairement aux théorèmes de stabilité multi-étapes qui sont utilisés pour trouver un bon arbre de scénarios. La différence avec ces derniers est que la mesure $\mathcal{M}$ a vocation à tenir compte de la structure du problème (à travers l'ensemble Γ), ce qui n'est pas le cas des distances multi-étapes entre distributions de probabilité (nested distance ou filtration distance) qui bornent aussi l'erreur et que nous avons décrit dans la section de revue de littérature. Du fait de tenir compte de l'intégralité du problème d'optimisation pour construire l'arbre de scénarios, notre approche partage une idée similaire avec les approches basées sur le critère EVPI ou sur le raffinement d'arbres (qui sont décrites à la fin de la Section 2.2.1). Cependant, à la différence de ces dernières, nous ne souhaitons pas construire l'arbre de scénarios par résolutions successives du problème approché car cela engendre un coût computationnel significatif. Finalement, dans un second temps nous montrons comment la mesure $\mathcal{M}$ peut être minimisée en pratique dans le cas de trois ensembles $\Gamma(p)$ de formes différentes. Dans les trois cas, des arbres de scénarios sont construits et leurs structures sont analysées en relation avec la forme de $\Gamma(p)$. Différentes méthodes de discrétisation provenant du domaine de l'intégration numérique peuvent être considérées pour générer les points et les poids de l'arbre (p. ex. quasi-Monte Carlo ou la quantification optimale). Dans ce cas, notre méthode offre à l'agent décisionnaire une approche systématique pour trouver la meilleure structure d'arbre non nécessairement symétrique, alors que jusqu'à présent la tendance pratique consistait à se restreindre à des arbres symétriques puisque ces méthodes de discrétisation n'incluent pas naturellement une procédure de calcul de structure.

3.4 Quatrième article

Le quatrième article met en pratique les concepts théoriques développés dans les deux précédents. Pour cela, nous considérons un problème d'évaluation d'options asiatiques à style d'exercice bermudien. Le style bermudien se différencie du style américain en cela qu'il per-

met d'exercer l'option uniquement à certaines dates (étapes) déterminées par avance. Les revenus de l'option asiatique ont la caractéristique de dépendre de tout l'historique de prix de l'actif sous-jacent à travers sa valeur moyenne. Cette dépendance a pour conséquence de réduire la variabilité de son prix à mesure que les étapes avancent. Ainsi, les fonctions de recours du problème exhibent des variations de leur variabilité à travers les étapes et les scénarios, ce qui fait de cette option un exemple approprié pour illustrer notre méthode. Puisque les variations proviennent essentiellement de la forme particulière de la fonction de revenus, et non de la distribution de probabilité de l'actif sous-jacent (qui suit un mouvement Brownien géométrique), cet exemple démontre en plus l'intérêt de considérer une approche de génération d'arbres *guidée par le problème* plutôt que par la distribution du processus stochastique.

Après avoir exprimé analytiquement le représentant $\Gamma(p)$ pertinent pour le problème, nous construisons les arbres de scénarios qui minimisent la mesure $\mathcal{M}$. Nous observons que ceux-ci sont en effet appropriés au problème car ils réduisent significativement l'erreur par rapport à des arbres symétriques qui ne tiennent pas compte des caractéristiques du problème. De plus, l'analyse des taux de convergence démontre un effet de réduction du nombre d'étapes effectif du problème.

CHAPITRE 4 ARTICLE 1: QUALITY EVALUATION OF SCENARIO-TREE GENERATION METHODS FOR SOLVING STOCHASTIC PROGRAMMING PROBLEMS

Julien Keutchayan[1,2], Michel Gendreau[1,2], Antoine Saucier[1]

[1]Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Canada.

[2]Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Canada.

Article publié dans la revue *Computational Management Science (Volume 14, Issue 3, pp 333–365)* le 1 juillet 2017.

Abstract. This paper addresses the generation of scenario trees to solve stochastic programming problems that have a large number of possible values for the random parameters (possibly infinitely many). For the sake of the computational efficiency, the scenario trees must include only a finite (rather small) number of scenarios, therefore, it provides decisions only for some values of the random parameters. To overcome the resulting loss of information, we propose to introduce an *extension procedure*. It is a systematic approach to interpolate and extrapolate the scenario-tree decisions to obtain a decision policy that can be implemented for any value of the random parameters at little computational cost. To assess the quality of the scenario-tree generation method and the extension procedure (STGM-EP), we introduce three generic quality parameters that focus on the quality of the decisions. We use these quality parameters to develop a framework that will help the decision-maker to select the most suitable STGM-EP for a given stochastic programming problem. We perform numerical experiments on two case studies. The quality parameters are used to compare three scenario-tree generation methods and three extension procedures (hence nine couples STGM-EP). We show that it is possible to single out the best couple in both problems, which provides decisions close to optimality at little computational cost.

4.1 Introduction

Stochastic programming is a mathematical programming framework used to formulate and solve sequential decision-making problems under uncertainty. It relies on the assumption that the probability distribution of the random parameters is known (possibly inferred from

data), and that the information about these parameters becomes available stage by stage. If the distribution is supported by a finite number of points, small enough for a tractable computation, then all the random outcomes can be represented in a so-called *scenario tree*, and solving the stochastic program on the scenario tree provides the optimal decisions for every outcome. In this context, stochastic programming has proved to be a powerful framework to solve problems in energy, transportation, logistic, finance, etc.; see, e.g., Wallace et Fleten (2003), Schultz *et al.* (2003), Yu *et al.* (2003), Louveaux (1998) and Powell et Topaloglu (2003). The situation becomes more complicated if the random parameters take a large (possibly infinite) number of values, since stochastic programming problems are then large-scale optimization problems (possibly infinite dimensional problems) that are typically impossible to solve analytically or computationally in a reasonable time. In that case, the scenario tree is built with a finite subset of scenarios, obtained by discretizing the stochastic process that models the random parameters across the stages. Many discretization schemes for generating scenario trees have been developed in the literature; we will cite some important references in Section 4.1.2. The scenario-tree generation method enables the decision-maker to obtain estimates of the optimal value and the optimal solutions of the stochastic program, but two questions remain open for the decision-maker:

- How to implement the optimal solutions that are scenario dependent? Apart from the first-stage decisions, which are not scenario dependent, all subsequent stage decisions depend on the scenarios, and therefore they may not be implementable if the real-world realization of the stochastic process does not coincide with a scenario in the tree.

- How to tell which method provides the best quality decisions for a given problem? Methods are typically built from mathematical results on the optimal-value error (consistency, rate of convergence, etc.), but it is unclear whether a good optimal-value estimate systematically implies good quality decisions. Additionally, the claimed effectiveness may be guaranteed under assumptions that are not fulfilled in practice. Also, it may hide some unknown quantities (e.g., the implied constant in a big-O notation for the rate of convergence) that prevents the decision-maker from knowing with certainty that the best method from a theoretical point of view will be the most efficient when put into practice.

In this paper, we show that both questions are inherently linked and we propose a mathematical framework that answers both of them.

The remainder of this paper is organized as follows: In Section 4.1.1 and 4.1.2, we introduce the notation to describe the stochastic programming problem and the scenario-tree formu-

lation. In Section 4.1.3, we describe with more details the motivation of our approach. In Section 4.2, we develop the quality evaluation framework, and we provide in Section 4.3 the statistical tools (estimators, confidence intervals) to put it into practice. We present actual extension procedures in Section 4.4, and we apply them in the two case studies in Section 4.5. Finally, Section 4.6 concludes the paper.

4.1.1 Stochastic programming problem formulation

We consider a stochastic programming problem with a time horizon $T \in \mathbb{N}^*$ and integer time stages t ranging from 0 to T. The stagewise evolution of the random parameters is represented by a stochastic process $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_T)$, defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, where $\boldsymbol{\xi}_t$ is a random vector with d_t components that represent the random parameters revealed in period $(t-1, t)$. We define $\boldsymbol{\xi}_{..t} := (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_t)$ the partial stochastic process up to stage t, and we denote the supports of $\boldsymbol{\xi}_t$, $\boldsymbol{\xi}_{..t}$, and $\boldsymbol{\xi}$ by Ξ_t, $\Xi_{..t}$, and Ξ, respectively. Throughout this paper, random quantities are always written in bold font, while their realizations are written with the same symbols in normal font.

At each stage $t \in \{1, \ldots, T\}$, the decisions must be based only on the information available at this stage in order to be *non-anticipative*. Thus, the stage-t decision function, denoted by x_t, is defined as

$$\begin{aligned} x_t : \Xi_{..t} &\to \mathbb{R}^{s_t} \\ \xi_{..t} &\mapsto x_t(\xi_{..t}), \end{aligned} \tag{4.1}$$

where s_t is the number of decisions to be made at stage t (for the sake of clarity, we consider that $s_t = s$ and $d_t = d$ for all t). The decisions at stage 0 are represented in a vector $x_0 \in \mathbb{R}^s$. We assume that each decision function belongs to an appropriate space of measurable functions, e.g., the space $\mathcal{L}_p(\Xi_{..t}; \mathbb{R}^s)$ of p-integrable functions for $p \in [1, +\infty]$. The *decision policy* (or simply *policy*) is denoted by x and is the collection of all the decision vector/functions from stage 0 to stage T, i.e., $x = (x_0, x_1, \ldots, x_T) \in \mathbb{R}^s \times \Pi_{t=1}^T \mathcal{L}_p(\Xi_{..t}; \mathbb{R}^s)$ or equivalently

$$x(\boldsymbol{\xi}) = (x_0, x_1(\boldsymbol{\xi}_1), \ldots, x_t(\boldsymbol{\xi}_{..t}), \ldots, x_T(\boldsymbol{\xi})). \tag{4.2}$$

The set of feasible decision vectors (or simply feasible set) is denoted by X_0 at stage 0 and by $X_t(x_{..t-1}(\xi_{..t-1}); \xi_{..t})$ at stage $t \in \{1, \ldots, T\}$; the latter notation emphasizes that it may depend on the realization $\xi_{..t} \in \Xi_{..t}$ and on the decisions $x_{..t-1}(\xi_{..t-1}) := (x_0, ..., x_{t-1}(\xi_{..t-1}))$ prior to t. A decision policy is *feasible* if it yields a feasible decision vector at every stage

with probability one. We assume that the space of feasible policies is nonempty and that the decisions made at each stage do not alter the probability distribution of the stochastic process. We emphasize that our modelization can include integrity constraints.

We introduce a revenue function $q(x(\xi);\xi)$ that represents the total revenues obtained from stage 0 to T for a policy x and a realization ξ of the stochastic process. We are interested in the dependence of the first moment of $q(x(\boldsymbol{\xi});\boldsymbol{\xi})$ with respect to the decision policy, i.e., in the functional $Q(x) := \mathbb{E}[q(x(\boldsymbol{\xi});\boldsymbol{\xi})]$; we suppose that $Q(x)$ is well-defined for any feasible policy x.

The stochastic programming problem consists in finding a feasible and non-anticipative policy that maximizes $Q(\cdot)$, which means finding x^* of the form (4.2) satisfying

$$
\begin{aligned}
Q(x^*) \;=\; & \max_{x=(x_0,\ldots,x_T)} \mathbb{E}[q(x(\boldsymbol{\xi});\boldsymbol{\xi})] && (4.3)\\
s.t. \quad & x_0 \in X_0; && (4.4)\\
& x_1(\boldsymbol{\xi}_1) \in X_1\Big(x_0;\boldsymbol{\xi}_1\Big),\ w.p.1; && (4.5)\\
& x_t(\boldsymbol{\xi}_{..t}) \in X_t\Big(x_{..t-1}(\boldsymbol{\xi}_{..t-1});\boldsymbol{\xi}_{..t}\Big),\ w.p.1,\ \forall t \in \{2,\ldots,T\}. && (4.6)
\end{aligned}
$$

Constraints (4.5) and (4.6) hold with probability one (w.p.1). The policy x^* is called an *optimal decision policy* and $Q(x^*)$ is the *optimal value* of the stochastic programming problem. Some additional conditions should be added to ensure that there exists at least one optimal decision policy; see, e.g., Rockafellar et Wets (1974).

4.1.2 Scenario tree and scenario-tree deterministic program

In most problems the optimal value $Q(x^*)$ and the optimal decision policy x^* are difficult to compute exactly, or approximately within a sufficiently small error, as shown in Dyer et Stougie (2006) and Hanasusanto *et al.* (2016). For this reason, approximate solution methods have been developed, such as the large family of scenario-tree generation methods. We refer the reader to the following references for a general presentation on stochastic programming and solution methods: Birge et Louveaux (2011), Ruszczyński et Shapiro (2003a), Schultz (2003), and Defourny *et al.* (2011).

A scenario-tree generation method builds a scenario-tree deterministic program from a finite subset of realizations of $\boldsymbol{\xi}$ (called *scenarios*). The scenarios are obtained through a discretization scheme, which can be performed using many possible techniques, and are organized in a tree structure to approximate the stagewise evolution of information (called *filtration*) of the stochastic process. Some of the most popular works on scenario generation are: Shapiro

et Homem-de Mello (1998) and Mak *et al.* (1999) on the Monte Carlo method; Pennanen et Koivu (2002), Drew et Homem-de Mello (2006), and Leövey et Römisch (2015) on integration quadrature and quasi-Monte Carlo; Høyland et Wallace (2001) and Høyland *et al.* (2003) on moment-matching; Pflug (2001), Pflug et Pichler (2012), and Pflug et Pichler (2015) on optimal quantization; Dupačová *et al.* (2003) and Heitsch et Römisch (2009) on scenario reduction; Frauendorfer (1996) and Edirisinghe (1999) on bound-based approximations; Chen et Mehrotra (2008) and Chen *et al.* (2015) on sparse grid quadrature rules.

All the above methods provide a procedure to generate scenarios from $\boldsymbol{\xi}$, but only a few also provide a systematic approach to generate a *tree structure*, i.e., to organize the scenarios in a structure with branchings at every stage. In most cases, the choice of a tree structure is left to the decision-makers themselves, who may choose it empirically. Throughout this paper, we use the term *scenario-tree generation method* to name a procedure that generates a set of scenarios organized in a tree structure; this includes the case where the structure is chosen beforehand by the decision-maker. The remainder of this section introduces the notation for the scenario tree and the scenario-tree deterministic program.

A scenario tree is a rooted tree structure $\mathcal{T} = (\mathcal{N}, \mathcal{E})$, with (finite) node set $\mathcal{N}$, edge set $\mathcal{E}$, and root node n_0. The structure is such that T edges separate the root from any of the leaves. We denote by $C(n)$, $a(n)$, and $t(n)$, respectively, the children nodes of n, the ancestor node of n, and the stage of n (i.e., the number of edges that separate n from n_0). We also denote $\mathcal{N}^* := \mathcal{N} \setminus \{n_0\}$ and $\mathcal{N}_t := \{n \in \mathcal{N} \,|\, t(n) = t\}$. Each node $n \in \mathcal{N}^*$ carries a discretization point ζ^n of $\boldsymbol{\xi}_{t(n)}$ and a weight $w^n > 0$. The latter represents the weight of n with respect to its sibling nodes. The weight of n with respect to whole scenario tree, denoted by W^n, is the product of all w^m for m on the path from n_0 to n. We emphasize that we let the weights be any positive real values to cover a large setting of discretization schemes. We denote by $\zeta^{\cdot\cdot n}$ the sequence of discretization points on the path from n_0 to n; hence $\zeta^{\cdot\cdot n}$ is a discretization point of $\boldsymbol{\xi}_{\cdot\cdot t(n)}$.

The scenario-tree approach proceeds as follows: the vector $\widehat{x}^n \in \mathbb{R}^s$ is the decision at node $n \in \mathcal{N}$ and the sequence $\widehat{x}^{\cdot\cdot n} := (\widehat{x}^{n_0}, \ldots, \widehat{x}^n)$ denotes the decision vectors on the path from n_0 to n. The scenario-tree deterministic program is written as

$$\widehat{Q}^* := \max_{\{\widehat{x}^n \,:\, n \in \mathcal{N}\}} \sum_{l \in \mathcal{N}_T} W^l \, q(\widehat{x}^{\cdot\cdot l}; \zeta^{\cdot\cdot l}) \tag{4.7}$$

$$s.t. \quad \widehat{x}^{n_0} \in X_0; \tag{4.8}$$

$$\widehat{x}^n \in X_1\Big(\widehat{x}^{n_0}; \zeta^n\Big), \ \forall n \in \mathcal{N}_1; \tag{4.9}$$

$$\widehat{x}^n \in X_t\Big(\widehat{x}^{\cdot\cdot a(n)}; \zeta^{\cdot\cdot n}\Big), \ \forall n \in \mathcal{N}_t, \ \forall t \in \{2, \ldots, T\}, \tag{4.10}$$

The optimal decision vector at each node is denoted by $\widehat{x}^{n*}$ and, for convenience, we define $\widehat{x}^* := \{\widehat{x}^{n*} \,|\, n \in \mathcal{N}\}$ and we refer to it as the *tree optimal policy*.

We emphasize that if the stochastic program cannot be solved exactly, which is our case of interest in this paper, then the outputs $\widehat{x}^*$ and $\widehat{Q}^*$ of the scenario-tree approach are approximations of x^* and $Q(x^*)$, respectively.

4.1.3 Motivations and extension procedure formulation

Scenario trees have proved to be a useful approach for a wide class of stochastic programming problems (see the references in the Introduction). However, as pointed out in Ben-Tal *et al.* (2009), this approach fails to provide decisions for all values of the random parameters. The reason is that the decisions at stage t are only available for the set $\{\zeta^{\cdot n} \,|\, n \in \mathcal{N}_t\}$, which is a proper subset of $\Xi_{\cdot t}$. If the stochastic process has a continuous distribution, then the former set has probability zero, and therefore the real-world realization of the stochastic process never coincides with a scenario in the tree. In that case, only the stage-0 decision, which is not scenario dependent, can be implemented by the decision-maker. This raises the first question written in the Introduction: How to implement the optimal solutions that are scenario dependent?

An attempt to answer this question, proposed for instance in Kouwenberg (2001), Chiralaksanakul et Morton (2004), and Hilli et Pennanen (2008), consists in solving dynamically the scenario-tree deterministic program on a shrinking horizon in order to implement the stage-0 decision recursively at every stage. However, a drawback of this approach is its computational cost. It requires as many solutions as the total number of stages, and the procedure must be carried out all over again for each new realization ξ. With this approach, it can be computationally costly to perform an out-of-sample test, which is an evaluation of the tree decisions on a set of scenarios directly sampled from $\boldsymbol{\xi}$, since it is typically required to test the decisions on thousands of realizations for a reliable accuracy. Therefore, a satisfactory answer to the first question would be to find a way to provide decisions for *any* value of the random parameters, in a manner that allows a thorough out-of-sample test.

The second question raised in the introduction is concerned with the choice of the scenario-tree generation method. Methods are usually developed with the goal to control the optimal-value error $|Q(x^*) - \widehat{Q}^*|$. But as far as the decisions are concerned, it is unclear whether a small value of the error always implies a tree optimal policy $\widehat{x}$ close to the optimal decision policy x^*. Additionally, the notion of closeness between the two policies is in itself difficult to define, because $\widehat{x}$ is a finite set of vectors whereas x^* is a sequence of functions. For this reason, the focus is sometimes made on controlling the distance between the two stage-0

decision vectors $\widehat{x}^{n_0}$ and x_0^*. This is done for instance in Pennanen et Koivu (2005), where the scenario trees are generated in a way that guarantees the convergence of $\widehat{x}^{n_0}$ toward x_0^* as the number of scenarios increases. However, such approaches address only the quality of the stage-0 decisions. They ignore the decisions in the following stages, as if those decisions were irrelevant or irremediably out of reach for an evaluation. We do not believe so.

In this paper, we completely depart from the view of the references above. From the tree optimal policy $\widehat{x}$, we intend to recover a decision policy of the form (4.2) in order to treat the decisions of the scenario tree as a candidate solution of the stochastic programming problem. We do so as follows: after solving the program (4.7)-(4.10), we extrapolate and interpolate the tree optimal decisions outside the set of the scenarios. We refer to this as an *extension procedure* and to the resulting policy as an *extended tree policy*. The extended tree policy (formalized in Definition 4.1.1) is defined over all possible realizations of the stochastic process, and it coincides with the tree optimal policy on the scenarios of the tree.

Definition 4.1.1. Let $\widehat{x}^* = \{\widehat{x}^{n*} \,|\, n \in \mathcal{N}\}$ be a tree optimal policy. An *extended tree policy* for $\widehat{x}^*$ is a decision policy $\widetilde{x} = (\widetilde{x}_0, \ldots, \widetilde{x}_T)$, where $\widetilde{x}_0 = \widehat{x}^{n_0*}$ and for every $t \in \{1, \ldots, T\}$, $\widetilde{x}_t$ is defined as in (4.1) and satisfies

$$\widetilde{x}_t(\zeta^{\cdot n}) = \widehat{x}^{n*}, \quad \text{for all } n \in \mathcal{N}_t. \tag{4.11}$$

The extension procedure enables a thorough out-of-sample test of the policy, because the decisions are available merely through the evaluation of a function at a particular point, which can be carried out many times at little cost. Since the extended tree policy $\widetilde{x}$ is a candidate solution of the stochastic program, it can be compared with the optimal policy x^*. A natural comparison criterion, which is highly relevant for the decision-maker, is to compare the value $Q(\widetilde{x})$ with $Q(x^*)$. Although this idea provides the basis for our quality evaluation framework, we show that it must be addressed with care because the extended tree policy may not satisfy the feasibility requirement.

Our quality evaluation framework is based on three quality parameters that will enable the decision-maker to compare couples of scenario-tree generation method and extension procedure (STGM-EP) in order to select the best one for a given problem. We refer to Kaut et Wallace (2007) for a discussion on the evaluation of scenario-tree generation methods for two-stage problems, which provided the inspiration for this paper. We also refer to the works of Defourny *et al.* (2013) where some extension techniques are introduced using regression tools from machine learning. However, their approach and ours differ in several aspects, the main one being their desire to select the best *policy*, while we want to select the best *method*

(i.e., the whole family of policies that can be obtained by the method; see Remark 4.1.1). The reason why we focus on methods rather than on policies lies in the way scenario trees are used in practice. Very often, problems are solved regularly with different data to infer the random parameters, and therefore a new policy must be computed for each new data set. In that case, our quality evaluation framework requires to run the selection test just on one data set. Then, the selected STGM-EP can be used to compute a policy for each new data set without additional comparison tests. We will discuss this point in more details after the definition of the selection criterion in Section 4.2.3.

Remark 4.1.1. The quality evaluation framework developed in this paper can be applied to any scenario-tree generation method. However, the future mathematical developments require to differentiate two categories of methods that we refer to as *stochastic* and *deterministic*. A method is said to be stochastic if it uses some random sampling techniques, which implies that the output scenario trees are different every time the method is carried out. Conversely, it is said to be deterministic if it always generates the same scenario tree, i.e., if it provides a unique tree structure and a unique set of discretization points and weights.

Since different scenario trees imply different tree optimal policies, and therefore different extended tree policies, we need to be able to consider somehow all the possible extended tree policies that a stochastic method may yield. All these policies can be seen as the realizations of a *random* extended policy $\widetilde{\boldsymbol{x}}$ (denoted in bold font to distinguish it from a particular realization $\widetilde{x}$; see Figure 4.1). Therefore, by $\widetilde{\boldsymbol{x}}$ we denote the family of all decision policies that are obtained in a random manner by a particular STGM-EP.

The goal of this paper is to assess the quality of methods. For this reason, we shall focus on studying the quality of $\widetilde{\boldsymbol{x}}$ rather than a specific realization of it. To this end, we will denote by $\mathbb{E}_{\widetilde{\boldsymbol{x}}}[\cdot]$ the expectation operator taken with respect to the probability measure of $\widetilde{\boldsymbol{x}}$. This measure is defined on the infinite dimensional space of decision policies, i.e., $\mathbb{R}^s \times \Pi_{t=1}^{T} \mathcal{L}_p(\Xi_{..t}; \mathbb{R}^s)$, which makes it a highly complicated mathematical object. However, we do not further develop this point in this paper because, as far as the decision-maker is concerned, the only important matter is that the probability measure of $\widetilde{\boldsymbol{x}}$ can be sampled by generating several scenario trees, and hence the expectation can be estimated by a finite sum; the statistical properties of such estimation will be studied in Section 4.3.

4.2 Quality parameters

In this section, we introduce the quality parameters that assess any scenario-tree generation method and extension procedure. We assume the following condition holds throughout this

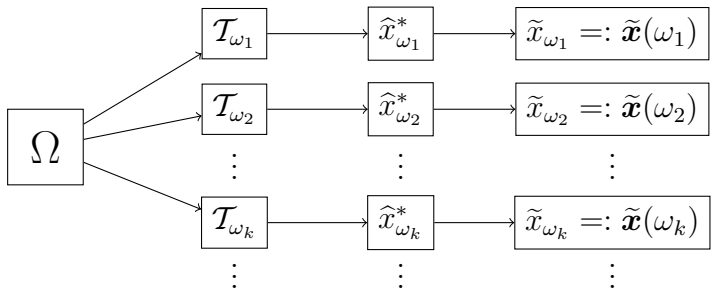


Figure 4.1 A *stochastic* STGM-EP yields a *random* extended tree policy $\widetilde{\boldsymbol{x}}$. As a random element, this policy can be seen as a map $\widetilde{\boldsymbol{x}} : \Omega \to \mathbb{R}^s \times \Pi_{t=1}^T \mathcal{L}_p(\Xi_{..t}; \mathbb{R}^s)$ obtained through the composition of several transformations represented by the arrows.

section:

C1. The stochastic program (4.3)-(4.6) and the scenario-tree deterministic program (4.7)-(4.10) each have an optimal decision policy.

The framework developed in the section works for both stochastic and deterministic methods. However, for the sake of conciseness, it is expressed for stochastic methods only. The equivalent results for the deterministic ones are easy to deduce by removing the expectation $\mathbb{E}_{\widetilde{\boldsymbol{x}}}[\cdot]$ and by substituting $\widetilde{\boldsymbol{x}}$ with its unique realization $\widetilde{x}$.

4.2.1 Probability of feasibility and conditional revenues

It follows from Definition 4.1.1 that an extended tree policy yields feasible decisions at the root node and for any realization $\xi_{..t}$ that coincides with a discretization sequence $\zeta^{\cdot\cdot n}$ for a node $n \in \mathcal{N}_t$ in the scenario tree. For any other realization the feasibility is not guaranteed and will depend on the considered STGM-EP. The first two features of the extended tree policy that we want to assess is its probability of feasibility at every stage and its conditional revenues given the feasibility.

Consider a random extended tree policy $\widetilde{\boldsymbol{x}}$, and let $\widetilde{x}$ be a realization of $\widetilde{\boldsymbol{x}}$. The subset of $\Xi_{..t}$ on which $\widetilde{x}$ provides feasible decisions from stage 0 to t, denoted by $\widetilde{\Xi}_{..t}(\widetilde{x})$, is defined as

$$\widetilde{\Xi}_1(\widetilde{x}) = \Big\{\xi_1 \in \Xi_1 \,|\, \widetilde{x}_1(\xi_1) \in X_1(\widetilde{x}_0; \xi_1)\Big\}, \tag{4.12}$$

and

$$\widetilde{\Xi}_{..t}(\widetilde{x}) = \Big\{\xi_{..t} \in \Xi_{..t} \,\Big|\, \xi_{..t-1} \in \widetilde{\Xi}_{..t-1}(\widetilde{x}),\ \widetilde{x}_t(\xi_{..t}) \in X_t\Big(\widetilde{x}_{..t-1}(\xi_{..t-1}); \xi_{..t}\Big)\Big\}, \tag{4.13}$$

for each $t \in \{2, \ldots, T\}$. Thus, the probability that $\widetilde{x}$ provides feasible decisions from stage 0 to t is $\mathbb{P}[\boldsymbol{\xi}_{..t} \in \widetilde{\Xi}_{..t}(\widetilde{x})]$. When considering the random policy $\widetilde{\boldsymbol{x}}$, the set $\widetilde{\Xi}_{..t}(\widetilde{\boldsymbol{x}}(\omega))$ varies depending on the outcome $\omega \in \Omega$ (see Figure 4.1). Taking into account the randomness of $\widetilde{\boldsymbol{x}}$ leads to the following definition of the quality parameters $p(t)$ and CR.

Definition 4.2.1. (i) The probability $p(t)$ that a random extended tree policy $\widetilde{\boldsymbol{x}}$ yields feasible decisions up to stage $t \in \{0, \ldots, T\}$ is given by $p(0) = 1$ and

$$p(t) = \mathbb{P}_{(\boldsymbol{\xi}, \widetilde{\boldsymbol{x}})}[\boldsymbol{\xi}_{..t} \in \widetilde{\Xi}_{..t}(\widetilde{\boldsymbol{x}})]. \tag{4.14}$$

(ii) The conditional revenues CR obtained with $\widetilde{\boldsymbol{x}}$ when it yields feasible decisions up to the end of the optimization horizon is given by

$$\mathrm{CR} = \mathbb{E}_{(\boldsymbol{\xi}, \widetilde{\boldsymbol{x}})}[q(\widetilde{\boldsymbol{x}}(\boldsymbol{\xi}); \boldsymbol{\xi}) \,|\, \boldsymbol{\xi} \in \widetilde{\Xi}_{..T}(\widetilde{\boldsymbol{x}})]. \tag{4.15}$$

The value CR is well-defined provided that $p(T) > 0$.

The sequence $(p(0), p(1), \ldots, p(T))$, non-increasing by definition of $\widetilde{\Xi}_{..t}(\widetilde{\boldsymbol{x}})$, provides information about the stagewise evolution of the size of the stage-t feasible region $\widetilde{\Xi}_{..t}(\widetilde{\boldsymbol{x}})$ (as a sequence of numbers ranging from 0 to 1) embedded in the support $\Xi_{..t}$.

Although CR is a natural quantity to compute, its interpretation can be tricky. By definition of the conditional expectation as a ratio of an expectation and a probability, the values of CR are inherently linked with those of $p(T)$. If $p(T)$ is less than one, then the conditional revenues are computed on a subset of random parameters, and therefore their values can be larger than the optimal ones $Q(x^*)$. Typically, it will be observed in the numerical experiments that the lower $p(T)$ the larger CR, which means that CR becomes almost irrelevant when $p(T)$ is much smaller than one, as it gives the expected revenues in a world where pessimistic scenarios are ignored. Conversely, if $p(T)$ is close to one, then CR is forced not to exceed $Q(x^*)$ too much (and in the limit $p(T) \to 1$, $Q(x^*)$ is an upper bound on CR), therefore its value is meaningful for the decision-maker.

When considering two STGM-EPs, denoted by A and B, a decision-maker will select A if the respective quality parameters satisfy:

$$p_A(T) > p_B(T) \quad \text{and} \quad \mathrm{CR}_A > \mathrm{CR}_B, \quad \text{with} \quad p_A(T) \geq \alpha, \tag{4.16}$$

where $\alpha \in (0, 1]$ is the feasibility threshold that the decision-maker considers as satisfactory. As we will see in the numerical experiments, the selection criterion (4.16) allows to put aside scenario-tree generation methods and extension procedures of poor quality. However, for the

reason explained above, it is not always conclusive and it may not allow to single out the best method out of a group of good methods. In Section 4.2.3, we introduce a third quality parameter leading to a more robust selection criterion. To this end, we first address the feasibility restoration of the extended tree policy.

4.2.2 Feasibility restoration

In a real-world application, the extended tree policy $\widetilde{x}$ can be used all the way to the end of the optimization horizon provided the real-word realization ξ satisfies $\xi \in \widetilde{\Xi}_{..T}(\widetilde{x})$. If this condition does not hold, then there exists a stage $t^* \geq 0$ such that $\xi_{..t} \notin \widetilde{\Xi}_{..t}(\widetilde{x})$ for every $t > t^*$, and therefore the decision-maker has to find alternative feasible decisions from stage t^*+1 to T. This is known in the literature as the *feasibility restoration* problem. A necessary condition for restoring the feasibility is that the decisions $(\widetilde{x}_0, \ldots, \widetilde{x}_{t^*}(\xi_{..t^*}))$ do not lead to an empty feasible sets from stage t^*+1 to T. This is guaranteed if we assume that the following condition holds:

C2. The stochastic programming problem has a relatively complete recourse at every stage.

There are several approaches to address the feasibility restoration. In the works of Küchler et Vigerske (2010) and Defourny *et al.* (2013), the feasibility is restored by projecting the infeasible decision on the feasible set, which is done by solving non-linear optimization problems. In this paper, in order to proceed with the idea that the decisions must be available at little computational cost, we investigate the possibility that the decision-makers have the ability to fix any infeasibility by their own empirical knowledge on the problem. In other words, we assume that the following condition holds:

C3. The decision-maker possesses a *recourse policy*, obtained empirically, that always provides feasible decisions.

We model the recourse policy as a sequence $r = (r_1, \ldots, r_T)$, where the stage-t recourse function r_t takes a realization $\xi_{..t}$ and a sequence of decisions $x_{..t-1}(\xi_{..t-1})$ and yields a feasible decision vector, i.e., $r_t(x_{..t-1}(\xi_{..t-1}); \xi_{..t}) \in X_t(x_{..t-1}(\xi_{..t-1}); \xi_{..t})$. We emphasize that the definition of r_t differs from the definition of x_t in (4.1), since r_t depends also on the previous decisions.

The implementation of $\widetilde{x}$ (if feasible) and r (otherwise) yields a new decision policy, called the *feasible extended tree policy*, that provides feasible decisions at every stage and for every realization of the stochastic process. Definition 4.2.2 provides its explicit construction.

Definition 4.2.2. Let $\widetilde{x}$ be an extended tree policy and r a recourse policy. The *feasible extended tree policy* $\overline{x}$ resulting from $\widetilde{x}$ and r is given by $\overline{x}_0 = \widetilde{x}_0$ and recursively from $t = 1$ to $t = T$ by

$$\overline{x}_t(\xi_{..t}) = \begin{cases} \widetilde{x}_t(\xi_{..t}) & \text{if } \xi_{..t} \in \widetilde{\Xi}_{..t}(\widetilde{x}), \\ r_t(\overline{x}_{..t-1}(\xi_{..t-1}); \xi_{..t}) & \text{otherwise,} \end{cases} \tag{4.17}$$

where for $t = 1$ the term $\overline{x}_{..0}(\xi_{..0})$ corresponds to $\overline{x}_0$.

A stochastic STGM-EP yields a feasible extended tree policy that is random and is denoted by $\overline{\boldsymbol{x}}$ (see Remark 4.1.1 and Figure 4.1).

4.2.3 Distance between methods and selection criterion

The expected revenues obtained by implementing the feasible extended tree policy $\overline{x}$ is $Q(\overline{x}) = \mathbb{E}_{\boldsymbol{\xi}}[q(\overline{x}(\boldsymbol{\xi}); \boldsymbol{\xi})]$. Since all realizations $\overline{x}$ of the random policy $\overline{\boldsymbol{x}}$ are feasible and non-anticipative, we have that

$$Q(\overline{\boldsymbol{x}}) = \mathbb{E}_{\boldsymbol{\xi}}[q(\overline{\boldsymbol{x}}(\boldsymbol{\xi}); \boldsymbol{\xi})] \leq Q(x^*), \quad w.p.1. \tag{4.18}$$

We emphasize that the left-hand side of the inequality is a random variable, which is why the inequality holds *with probability one*. We see from (4.18) that every realization $\overline{x}$ of $\overline{\boldsymbol{x}}$ provides a lower bound $Q(\overline{x})$ of $Q(x^*)$. The nonnegative value $Q(x^*) - Q(\overline{x})$ provides a relevant measure of quality of $\overline{x}$. However, in general $Q(x^*)$ is not known, hence the computation may be done with $\widehat{Q}^*$ or an upper bound on $Q(x^*)$ rather than with $Q(x^*)$. This approach was used in the particular case of the Monte Carlo method for two-stage problems by Mak *et al.* (1999), and was developed further by Bayraksan et Morton (2009).

In this paper, we are interested in assessing the quality of methods, therefore, we must address the quality of $\overline{\boldsymbol{x}}$ rather than a specific realization $\overline{x}$. The inequality (4.18) still holds if we take the expectation of the left-hand side. This leads to the following definition of the distance $d(\overline{\boldsymbol{x}}, x^*)$ between the feasible extended tree policy $\overline{\boldsymbol{x}}$ and the optimal policy x^*.

Definition 4.2.3. The distance $d(\overline{\boldsymbol{x}}, x^*)$ between the feasible extended tree policy $\overline{\boldsymbol{x}}$ and the optimal policy x^* of the stochastic program (4.3)-(4.6) is given by

$$d(\overline{\boldsymbol{x}}, x^*) = Q(x^*) - \mathbb{E}_{\overline{\boldsymbol{x}}}\Big[Q(\overline{\boldsymbol{x}})\Big] \geq 0. \tag{4.19}$$

This distance defines a concept of "optimal choice" for the selection of the scenario-tree generation method and the extension procedure, in the sense of the following proposition.

Proposition 4.2.4. *A scenario-tree generation method and an extension procedure yield a random policy* $\overline{\boldsymbol{x}}$ *that is optimal with probability one for the stochastic program* (4.3)-(4.6) *if and only if* $d(\overline{\boldsymbol{x}}, x^*) = 0$.

Proof. The proof is straightforward: we have that $d(\overline{\boldsymbol{x}}, x^*) = 0$ if and only if $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(x^*) - Q(\overline{\boldsymbol{x}})] = 0$, and by the inequality (4.18), this is equivalent to $Q(x^*) = Q(\overline{\boldsymbol{x}})$ with probability one. By construction, the random policy $\overline{\boldsymbol{x}}$ is feasible and non-anticipative with probability one, which completes the proof. □

Thus, the distance $d(\overline{\boldsymbol{x}}, x^*)$ measures how far (in terms of revenues) the considered STGM-EP is from an ideal method that would *always* provide the optimal decisions. In applications, the value $d(\overline{\boldsymbol{x}}, x^*)$ represents the expected missing revenues that results from the implementation of any (randomly chosen) realization $\overline{x}$ of $\overline{\boldsymbol{x}}$.

It follows from Proposition 4.2.4 that a decision-maker will select the STGM-EP that provides the smallest value of $d(\overline{\boldsymbol{x}}, x^*)$. This selection criterion assesses the absolute quality of a method, i.e., in comparison to the ideal method. In applications, $Q(x^*)$ is typically not known, hence the decision-maker will rather compare the relative efficiency of two methods, as shown in Definition 4.2.5.

Definition 4.2.5 (Selection criterion)**.** Let A and B be two couples of scenario-tree generation method and extension procedure that yield random policies $\overline{\boldsymbol{x}}_A$ and $\overline{\boldsymbol{x}}_B$, respectively. We say that A is better than B for the stochastic programming problem if

$$\mathbb{E}_{\overline{\boldsymbol{x}}_A}[Q(\overline{\boldsymbol{x}}_A)] > \mathbb{E}_{\overline{\boldsymbol{x}}_B}[Q(\overline{\boldsymbol{x}}_B)]. \tag{4.20}$$

Criterion (4.20) guarantees that the whole family of policies obtained by A is of better quality on average than those obtained by B. For this reason, we say that the selection criterion (4.20) is in the *average-case setting*. This setting is particularly relevant when the problem is solved regularly, each time with new data to infer the distribution of the random parameters. In that case, one runs the selection test just once, and uses the selected STGM-EP to obtain a new decision policy for each new data set. It is reasonable to expect the selected STGM-EP to perform well on the other data sets, provided of course the latter do not change the distribution too much, i.e., provided they affect the parameters of the distribution and not the very nature of the distribution itself. Moreover, the law of large numbers guarantees that the average performance of a method is a relevant quality criterion for a problem solved regularly.

It is worth noting an alternative selection setting that suits better decision-makers who intend to solve always the *exact* same problem. We call it a comparison in the *best-case setting* because it is based on the best policy $\overline{\boldsymbol{x}}(\omega)$, for all $\omega \in \Omega$, that a STGM-EP yields. In the best-case setting, A is better than B if

$$\sup_{\omega\in\Omega} Q(\overline{\boldsymbol{x}}_A(\omega)) > \sup_{\omega\in\Omega} Q(\overline{\boldsymbol{x}}_B(\omega)). \tag{4.21}$$

The interpretation of (4.21) is the following: if the decision-maker has unlimited computational resources and carries out A and B enough times, then *eventually* A will yield a policy with expected revenues higher than those obtained with any policy yielded by B. An estimator for $\sup_{\omega\in\Omega} Q(\overline{\boldsymbol{x}}(\omega))$ is $\max_{k=1,\ldots,K} Q(\overline{x}^k)$, where $\overline{x}^1, \ldots, \overline{x}^K$ are K realizations of $\overline{\boldsymbol{x}}$, which links the selection criterion (4.21) with the policy selection technique developed by Defourny *et al.* (2013). The drawback of (4.21) lies in the potentially long time required to find the best policy. This prevents the criterion to be used when the problem is solved regularly with different data, since the decision-maker will have to go through the selection test to find the best policy for each new data set.

The criterion (4.20) in the average-case setting can be slightly modified to assess not only the expected revenues, but also the stability of the STGM-EP with regard to its repeated use. We say that a method is *stable* if it yields decision policies that are close to each other in terms of expected revenues (otherwise it is called *unstable*). The importance of stability in the evaluation of scenario-tree generation methods has been discussed in Kaut et Wallace (2007). In our framework, a measure of stability is provided by the variance $\mathrm{Var}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$; the lower the variance, the more stable the method. A decision-maker may substitute in (4.20) the expectation $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$ with $\delta\,\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})] - (1-\delta)\mathrm{Var}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$, for $\delta \in (0,1)$, to include a measure of stability in the selection criterion. Obviously, deterministic methods are the most stable, since $\overline{\boldsymbol{x}}$ is not random and therefore $\mathrm{Var}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})] = 0$.

4.3 Statistical estimation of the quality parameters

In this section, we show how to estimate the quality parameters introduced in Section 4.2. After introducing their statistical estimators in Section 4.3.1, we derive in Section 4.3.2 their confidence intervals, and in Section 4.3.3 we provide an algorithmic procedure to find the optimal sample sizes, defined as those minimizing the confidence interval bound for a given computational time.

4.3.1 Estimators

The quality parameters are estimated by sampling the probability distribution of $\boldsymbol{\xi}$ and the probability measure of $\widetilde{\boldsymbol{x}}$ (or $\overline{\boldsymbol{x}}$). Since the latter is sampled by generating several scenario trees (see Figure 4.1), and by solving the deterministic program (4.7)-(4.10) for each one, it is typically more costly to sample $\widetilde{\boldsymbol{x}}$ (or $\overline{\boldsymbol{x}}$) than $\boldsymbol{\xi}$. For this reason, it is relevant to consider an estimator that samples K times the random policy and $K \times M$ times the stochastic process, and to leave a degree of freedom in choosing how to balance the relative values of K and M to maximize the efficiency of the estimators.

We define the estimators of $p(t)$, CR, and $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$ as follows:

$$\widehat{p(t)}_{K,M} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{m=1}^{M}\mathbf{1}_{\widetilde{\Xi}_{..t}(\widetilde{x}^k)}(\xi_{..t}^{k,m}), \tag{4.22}$$

$$\widehat{\mathrm{CR}}_{K,M} = (\widehat{p(T)}_{K,M})^{-1}\left(\frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{m=1}^{M}q(\widetilde{x}^k(\xi^{k,m});\xi^{k,m})\mathbf{1}_{\widetilde{\Xi}_{..T}(\widetilde{x}^k)}(\xi^{k,m})\right), \tag{4.23}$$

$$\widehat{\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]}_{K,M} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{m=1}^{M}q(\overline{x}^k(\xi^{k,m});\xi^{k,m}), \tag{4.24}$$

where $\{\widetilde{x}^k \,|\, k = 1,\ldots,K\}$ and $\{\overline{x}^k \,|\, k = 1,\ldots,K\}$ are two sets of K independent and identically distributed (i.i.d.) realizations of $\widetilde{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$, respectively; $\{\xi^{k,m} \,|\, k = 1,\ldots,K; m = 1,\ldots,M\}$ is a set of $K \times M$ i.i.d. sample points of $\boldsymbol{\xi}$; $\xi_{..t}^{k,m}$ is the shorthand for $(\xi_1^{k,m},\ldots,\xi_t^{k,m})$; and the notation $\mathbf{1}_U(\cdot)$, for some set U, is the indicator function:

$$\mathbf{1}_U(u) := \begin{cases} 1 & \text{if } u \in U; \\ 0 & \text{otherwise.} \end{cases} \tag{4.25}$$

We emphasize that each estimator (4.22)-(4.24) is computed using $K \times M$ out-of-sample scenarios.

4.3.2 Confidence interval

To derive a confidence interval for the quality parameters, it is convenient to introduce a single notation for them, and to do all the mathematical developments with it. To this end, we define the quantity of interest θ that we want to estimate:

$$\theta := \mathbb{E}[\phi(\boldsymbol{x},\boldsymbol{\xi})], \tag{4.26}$$

where $\phi : \left(\mathbb{R}^s \times \Pi_{t=1}^T \mathcal{L}_p(\Xi_{..t}; \mathbb{R}^s)\right) \times \Xi \to \mathbb{R}$ is a map whose definition varies depending on the considered quality parameter, and $\boldsymbol{x}$ denotes a random policy being either $\widetilde{\boldsymbol{x}}$ or $\overline{\boldsymbol{x}}$. Throughout this section, for the sake of clarity, we do not add the subscript $(\boldsymbol{x}, \boldsymbol{\xi})$ to the expectation, probability, variance, and covariance operators. If $\theta = p(t)$, we have

$$\phi(\widetilde{\boldsymbol{x}}, \boldsymbol{\xi}) = \mathbf{1}_{\widetilde{\Xi}_{..t}(\widetilde{\boldsymbol{x}})}(\boldsymbol{\xi}_{..t}), \tag{4.27}$$

and if $\theta = \mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$,

$$\phi(\overline{\boldsymbol{x}}, \boldsymbol{\xi}) = q(\overline{\boldsymbol{x}}(\boldsymbol{\xi}); \boldsymbol{\xi}). \tag{4.28}$$

As for CR, it is the ratio of two expectations of the form (4.26). Following the definition of the estimators (4.22)-(4.24), we define the estimator $\widehat{\theta}_{K,M}$ of θ as follows:

$$\widehat{\theta}_{K,M} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{M} \sum_{m=1}^{M} \phi(x^k, \xi^{k,m}), \tag{4.29}$$

where $\{x^k\}$ is a set of K i.i.d. realizations of $\boldsymbol{x}$, and the sample points $\{\xi^{k,m}\}$ are defined as above. It is immediate to see that $\widehat{\theta}_{K,M}$ is an unbiased and consistent estimator of θ. To derive a confidence interval for θ, we assume that the following condition holds:

C4. The random variable $\phi(\boldsymbol{x}, \boldsymbol{\xi})$ is square-integrable: $\mathbb{E}[\phi(\boldsymbol{x}, \boldsymbol{\xi})^2] < +\infty$.

The following proposition provides a confidence interval for θ; the notation $[a \pm b]$ is a shorthand for the interval $[a - b, a + b]$.

Proposition 4.3.1. *Assume condition C4 holds. Then, a 100(1 - α)% asymptotic confidence interval for θ is*

$$\mathcal{I}_{K,M}^{1-\alpha} = \left[\widehat{\theta}_{K,M} \pm z_{1-\alpha/2}\left(\frac{\beta + \gamma(M-1)}{KM}\right)^{1/2}\right], \tag{4.30}$$

where z_α denotes the α-level quantile of a standard normal distribution and β and γ are given by

$$\beta = \mathrm{Var}[\phi(\boldsymbol{x}, \boldsymbol{\xi})], \tag{4.31}$$

$$\gamma = \mathrm{Cov}[\phi(\boldsymbol{x}, \boldsymbol{\xi}^1), \phi(\boldsymbol{x}, \boldsymbol{\xi}^2)], \tag{4.32}$$

with $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ two i.i.d copies of $\boldsymbol{\xi}$.

Proof. Consider the random variables

$$U_k := \frac{1}{M} \sum_{m=1}^{M} \phi(x^k, \xi^{k,m}), \quad \text{for all } k \in \{1, \ldots, K\}. \tag{4.33}$$

These random variables are i.i.d. because of the i.i.d. assumption on $\{x^k\}$ and $\{\xi^{k,m}\}$. We shall now verify that $\mathbb{E}[U_k^2] < \infty$ to apply the central limit theorem to $\frac{1}{K}\sum_{k=1}^K U_k$. We have

$$\mathbb{E}[U_k^2] = \frac{1}{M^2}\Bigg[\sum_{m=1}^{M}\mathbb{E}\Big[\phi(x^k,\xi^{k,m})^2\Big] + \sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\mathbb{E}\Big[\phi(x^k,\xi^{k,m})\,\phi(x^k,\xi^{k,m'})\Big]\Bigg], \tag{4.34}$$

The expectation in the double sum is bounded by $\mathbb{E}[\phi(x^k,\xi^{k,m})^2]$ by Cauchy-Schwarz inequality. Therefore, Condition C4 implies that $\mathbb{E}[U_k^2]$ is finite for any k.

The central limit theorem applied to $\frac{1}{K}\sum_{k=1}^K U_k$ yields the following convergence:

$$\mathbb{P}\Bigg(\Bigg|\frac{K^{1/2}}{\mathrm{Var}[U_1]^{1/2}}\Big(\frac{1}{K}\sum_{k=1}^{K}U_k-\theta\Big)\Bigg|\leq z_{1-\alpha/2}\Bigg)\underset{K\to+\infty}{\longrightarrow}\mathbb{P}(|Z|\leq z_{1-\alpha/2}) = 1-\alpha, \tag{4.35}$$

where Z follows a standard normal distribution and z_α denotes the α-level quantile of Z. Thus, a 100(1 - α)% asymptotic confidence interval for θ is

$$\mathcal{I}_{K,M}^{1-\alpha} = \Bigg[\widehat{\theta}_{K,M} \pm z_{1-\alpha/2}\frac{\mathrm{Var}[U_1]^{1/2}}{K^{1/2}}\Bigg]. \tag{4.36}$$

The quantity $\mathrm{Var}[U_1]$ can be simplified using the fact that the random variables $\phi(x^1,\xi^{1,m})$, for all $m\in\{1,\dots,M\}$, are i.i.d:

$$\mathrm{Var}[U_1] = \frac{1}{M^2}\sum_{m=1}^{M}\mathrm{Var}[\phi(x^1,\xi^{1,m})] + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\mathrm{Cov}[\phi(x^1,\xi^{1,m}),\phi(x^1,\xi^{1,m'})] \tag{4.37}$$

$$= \frac{1}{M}\mathrm{Var}[\phi(x^1,\xi^{1,m})] + \frac{M-1}{M}\mathrm{Cov}[\phi(x^1,\xi^{1,1}),\phi(x^1,\xi^{1,2})]. \tag{4.38}$$

Finally, defining $\beta = \mathrm{Var}[\phi(x^1,\xi^{1,m})]$ and $\gamma = \mathrm{Cov}[\phi(x^1,\xi^{1,1}),\phi(x^1,\xi^{1,2})]$, and combining (4.36) and (4.38), yields the confidence interval

$$\mathcal{I}_{K,M}^{1-\alpha} = \Bigg[\widehat{\theta}_{K,M} \pm z_{1-\alpha/2}\Big(\frac{\beta+\gamma(M-1)}{KM}\Big)^{1/2}\Bigg]. \tag{4.39}$$

□

The quantities β and γ, defined in (4.31) and (4.32), are not available analytically for the same reason as θ. They can be estimated through the following consistent and unbiased

estimators $\widehat{\beta}_{K,M}$ and $\widehat{\gamma}_{K,M}$ that use the same sample sets $\{x^k\}$ and $\{\xi^{k,m}\}$ as $\widehat{\theta}_{K,M}$:

$$\widehat{\beta}_{K,M} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{m=1}^{M}\phi(x^k,\xi^{k,m})^2 - (\widehat{\theta}_{K,M})^2, \tag{4.40}$$

$$\widehat{\gamma}_{K,M} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{m=1}^{M}\phi(x^k,\xi^{k,m})\right)^2 - (\widehat{\theta}_{K,M})^2. \tag{4.41}$$

Finally, let us observe that a deterministic STGM-EP always satisfies $\gamma = 0$. Indeed, a deterministic STGM-EP yields a nonrandom policy (i.e., $\boldsymbol{x}(\omega) = x$ for all $\omega \in \Omega$), therefore $\gamma = \mathrm{Cov}[\phi(x,\boldsymbol{\xi}^1),\phi(x,\boldsymbol{\xi}^2)] = 0$, because $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ are independent. As a result, when applied to a deterministic STGM-EP, the estimator and the confidence interval will be set with $K = 1$ and $\gamma = 0$.

4.3.3 Optimal sample size selection

The bound of the confidence interval (4.30) depends on the two sample sizes K and M. There is a degree of freedom in choosing how to balance the values of K and M, and this choice will clearly affect the estimation quality of θ. We propose a systematic procedure to find K and M in an optimal way, i.e., to minimize the bound of the confidence interval for a given computational time.

We denote by t_0 and t_1 the times required to obtain one realization x^k of $\boldsymbol{x}$ and $\xi^{k,m}$ of $\boldsymbol{\xi}$, respectively, and by t_2 the time required to compute the quantity $\phi(x^k,\xi^{k,m})$. Thus, the whole computation of $\widehat{\theta}_{K,M}$ takes $Kt_0 + KM(t_1+t_2)$ units of time. We denote by $\tau > 0$ the total computational time available to the decision-maker (of course it is required that $\tau \geq t_0 + t_1 + t_2$).

The optimal sample sizes K^* and M^* that minimize the confidence interval bound for a computational time τ are the optimal solutions of the following program:

$$P_\tau(\beta,\gamma): \quad \min_{K,M} \quad \frac{\beta+\gamma(M-1)}{KM} \tag{4.42}$$

$$s.t. \quad Kt_0 + KM(t_1+t_2) \leq \tau, \tag{4.43}$$

$$K \in \mathbb{N}^*, M \in \mathbb{N}^*. \tag{4.44}$$

It is also possible to consider the reverse problem, i.e., to find K^* and M^* that minimize the computational time required to have the value $\frac{\beta+\gamma(M-1)}{KM}$ lower than some target $v > 0$. In

that case, K^* and M^* are the optimal solutions of the following program:

$$P_v(\beta,\gamma): \quad \min_{K,M} \quad Kt_0 + KM(t_1+t_2) \tag{4.45}$$

$$s.t. \quad \frac{\beta+\gamma(M-1)}{KM} \leq v, \tag{4.46}$$

$$K \in \mathbb{N}^*, M \in \mathbb{N}^*. \tag{4.47}$$

We describe now the procedure that we propose to estimate the quality parameters of a stochastic STGM-EP:

(i) compute the estimators $\widehat{\beta}_{K_0,K_0}$ and $\widehat{\gamma}_{K_0,K_0}$ for two values $K_0 \geq 2$ and $M_0 \geq 2$ (set empirically to have a fast but fairly accurate estimation), and estimate t_0, t_1, and t_2;

(ii) solve the program $P_\tau(\widehat{\beta}_{K_0,M_0}, \widehat{\gamma}_{K_0,M_0})$ for the time limit τ, and retrieve (K^*, M^*);

(iii) compute $\widehat{\theta}_{K^*,M^*}$, $\widehat{\beta}_{K^*,M^*}$, $\widehat{\gamma}_{K^*,M^*}$, and derive the confidence interval $\mathcal{I}^{1-\alpha}_{K^*,M^*}$.

A decision-maker interested in having highly reliable estimates, regardless of the computational cost, can substitute in step (ii) the program $P_\tau(\widehat{\beta}_{K_0,M_0}, \widehat{\gamma}_{K_0,M_0})$ with $P_v(\widehat{\beta}_{K_0,M_0}, \widehat{\gamma}_{K_0,M_0})$ for some variance target $v > 0$.

As for the estimation of the quality parameters of a deterministic STGM-EP, it is done by setting $K = 1$ and by letting M be as large as possible within the computational time limit.

In Section 4.5.4 of the numerical experiments, we will prove the relevance of the optimal sample size selection by comparing the efficiency of the estimator (4.29) with a more classical estimator that samples $\boldsymbol{x}$ and $\boldsymbol{\xi}$ together.

4.4 Proposed procedures to extend the tree policy

4.4.1 Nearest-neighbor extension

The nearest-neighbor (NN) extension assigns to $\widetilde{x}_t(\xi_{..t})$ the value of the decisions $\widehat{x}^{n*}$ at the node $n \in \mathcal{N}_t$ nearest to $\xi_{..t}$. Several nearest-neighbor extensions can be defined depending on (i) the metric used to define the distance between $\xi_{..t}$ and $\zeta^{\cdot\cdot n}$, and (ii) the subset of $\mathcal{N}_t$ in which the nearest node is searched. For (i), we choose the stage-t Euclidean metric $\|\cdot\|_t$ defined as

$$\|u\|_t = \left(\sum_{i=1}^{t}\sum_{j=1}^{d}[(u_i)_j]^2\right)^{1/2}, \tag{4.48}$$

where $u = (u_1, \dots, u_t) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$ and $(u_i)_j$ denotes the j-th component of u_i; for convenience, we also denote $\|u_i\|^2 = \sum_{j=1}^d [(u_i)_j]^2$. For (ii), two relevant choices exist. The first is to search the nearest node among the whole stage-t nodes $\mathcal{N}_t$; the second is to search only among the children nodes $C(m)$, where $m \in \mathcal{N}_{t-1}$ is the node corresponding to the decisions made at stage $t-1$. We refer to the first choice as NN-AT (AT standing for *across tree*) and to the second as NN-AC (*across children*). We note that both extensions yield the same stage-1 decision function, therefore, they coincide for two-stage problems.

Nearest-neighbor extension across tree (NN-AT)

The extension *across tree* searches for the nearest node among the whole of $\mathcal{N}_t$. It allows to switch among the branches of the scenario tree, hence it is less sensitive than NN-AC to the decisions made at the previous stages. The node n nearest to the realization $\xi_{..t}$ is chosen by comparing the whole history of the random parameters up to stage t, by means of the norm (4.48). Thus, the set $\Xi_{..t}$ is partitioned into $|\mathcal{N}_t|$ Voronoï cells $V^n_{t,\mathrm{AT}}$ defined as

$$V^n_{t,\mathrm{AT}} = \Big\{\xi_{..t} \in \Xi_{..t} \;\Big|\; \forall r \in \mathcal{N}_t \setminus \{n\},\ \|\xi_{..t} - \zeta^{..n}\|_t < \|\xi_{..t} - \zeta^{..r}\|_t\Big\}, \tag{4.49}$$

for all $n \in \mathcal{N}_t$. On each cell $V^n_{t,\mathrm{AT}}$, the decision function is constant and yields the decision $\widehat{x}^{n*}$. Thus, the stage-t decision function is piecewise constant on $\Xi_{..t}$ and takes the form

$$\widetilde{x}^{\mathrm{AT}}_t(\xi_{..t}) = \sum_{n \in \mathcal{N}_t} \widehat{x}^{n*}\, \mathbf{1}_{V^n_{t,\mathrm{AT}}}(\xi_{..t}), \quad \forall \xi_{..t} \in \Xi_{..t}, \tag{4.50}$$

where $\mathbf{1}_{V^n_{t,\mathrm{AT}}}(\cdot)$ is the indicator function (defined in (4.25)).

Nearest-neighbor extension across children (NN-AC)

The extension *across children* searches for the nearest node at stage t among the set $C(m)$, where $m \in \mathcal{N}_{t-1}$ is the node of the decisions at the previous stage. This extension is computationally advantageous because it does not require to partition the whole set $\Xi_{..t}$ at every stage, but only its subset $V^m_{t,\mathrm{AC}} \times \Xi_t$ with $V^m_{t,\mathrm{AC}}$ the Voronoï cell of m. This set is partitioned into $|C(m)|$ Voronoï cells $V^n_{t,\mathrm{AC}}$ defined as

$$V^n_{t,\mathrm{AC}} = V^m_{t,\mathrm{AC}} \times \Big\{\xi_t \in \Xi_t \;\Big|\; \forall r \in C(m) \setminus \{n\},\ \|\xi_t - \zeta^n\| < \|\xi_t - \zeta^r\|\Big\}, \tag{4.51}$$

for all $m \in C(n)$. The stage-t decision function is piecewise constant on $\Xi_{..t}$ and takes the form

$$\widetilde{x}_t^{\mathrm{AC}}(\xi_{..t}) = \sum_{n \in C(m)} \widehat{x}^{n*}\, \mathbf{1}_{V^n_{t,\,\mathrm{AC}}}(\xi_{..t}), \quad \forall \xi_{..t} \in V^m_{t,\,\mathrm{AC}} \times \Xi_t. \tag{4.52}$$

We illustrate the stage-1 decision function of the nearest-neighbor extension in Figure 4.2 (a), and the Voronoï cells in Figure 4.3 (a)-(b); they correspond to the scenario tree represented in Figure 4.4.

4.4.2 N-nearest-neigbhor-weighted extension (NNNW)

The nearest-neighbor extension can be generalized to a weighted average over the N-*nearest* nodes (denoted by NNNW). To define it formally, let us denote by $\mathcal{V}_N(\xi_{..t}) \subseteq \mathcal{N}_t$, for $N \geq 2$, the set of the N-nearest stage-t nodes to $\xi_{..t}$ for the metric (4.48). The stage-t decision function of the NNNW extension is given by

$$\widetilde{x}_t(\xi_{..t}) = \sum_{n \in \mathcal{V}_N(\xi_{..t})} \lambda^n(\xi_{..t})\, \widehat{x}^{n*}, \quad \forall \xi_{..t} \in \Xi_{..t}, \tag{4.53}$$

where $\lambda^n(\cdot)$ is a weight function that we define as

$$\lambda^n(\xi_{..t}) = \left[\sum_{l \in \mathcal{V}_N(\xi_{..t})} \prod_{m \in \mathcal{V}_N(\xi_{..t}) \setminus \{l\}} \|\xi_{..t} - \zeta^{..m}\|_t \right]^{-1} \prod_{m \in \mathcal{V}_N(\xi_{..t}) \setminus \{n\}} \|\xi_{..t} - \zeta^{..m}\|_t, \tag{4.54}$$

for every $n \in \mathcal{V}_N(\xi_{..t})$. This definition is justified by the fact that $\lambda^n(\cdot)$ satisfies:

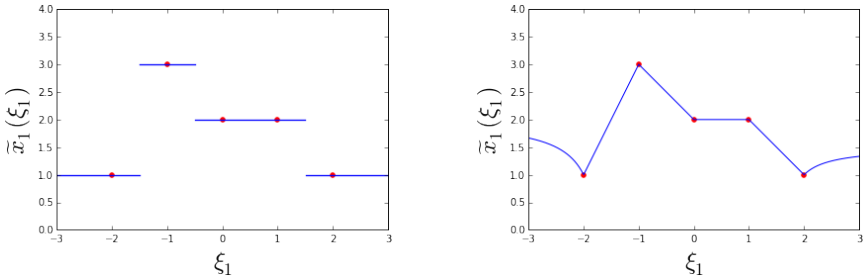
(i) $\lambda^n(\cdot) \geq 0$;

(ii) $\sum_{n \in \mathcal{V}_N(\xi_{..t})} \lambda^n(\xi_{..t}) = 1$ for every $\xi_{..t} \in \Xi_{..t}$;

(iii) $\lambda^n(\zeta^{..n}) = 1$ and $\lambda^m(\zeta^{..n}) = 0$ for every $m \in \mathcal{V}_N(\zeta^{..n}) \setminus \{n\}$.

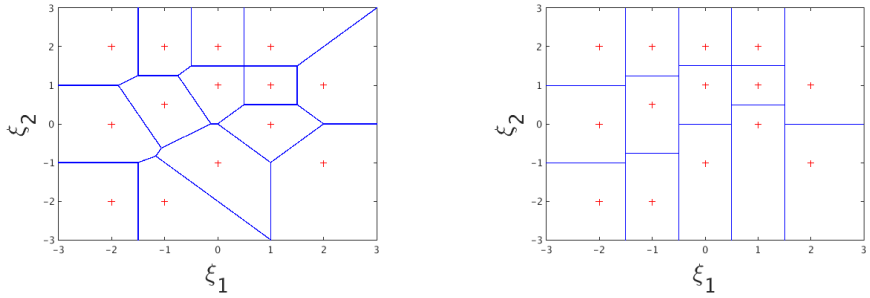
The properties (i) and (ii) imply that $\widetilde{x}_t(\xi_{..t})$ is a convex combination of the tree decisions $\widehat{x}^{n*}$ for $n \in \mathcal{V}_N(\xi_{..t})$, and (iii) ensures that $\widetilde{x}_t(\cdot)$ satisfies $\widetilde{x}_t(\zeta^{..n}) = \widehat{x}^{n*}$, which is a requirement of Definition 4.1.1. We note that since $\widetilde{x}_t(\xi_{..t})$ is a convex combination of $\widehat{x}^{n*}$, it may fail to provide integer values even if all $\widehat{x}^{n*}$ are integers. For this reason, the NNNW extension cannot be used directly for integer programs, unless some techniques are introduced to restrict $\widetilde{x}_t(\cdot)$ to a set of integers.

An illustration of this extension is displayed in Figure 4.2 (b), for the scenario tree in Figure 4.4.



(a) NN (b) 2NNW

Figure 4.2 First-stage extended decision functions of NN (a) and 2NNW (b), for the scenario tree in Figure 4.4.



(a) NN-AT (b) NN-AC

Figure 4.3 Voronoï cells (4.49) and (4.51) in the support of $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ for NN-AT (a) and NN-AC (b), and for the scenario tree in Figure 4.4. The +-markers are the sites of the cells.

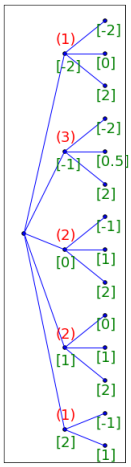


Figure 4.4 An example of a 3-stage scenario tree ($T = 2$). The values in bracket are the discretization points ζ^n for $n \in \mathcal{N}_1 \cup \mathcal{N}_2$. The values in parenthesis are the optimal decisions $\widehat{x}^{n*}$ for $n \in \mathcal{N}_1$ (only shown at stage 1).

4.5 Numerical experiments

4.5.1 Preliminaries of the numerical experiments

In this section, we apply the quality evaluation framework developed in Section 4.2, 4.3, and 4.4 on two case studies: a two-stage newsvendor problem and a four-stage multi-product assembly problem; for the latter, we use the same data as in Defourny *et al.* (2013). We generate the scenario trees by three different methods: optimal quantization (OQ), randomized quasi-Monte Carlo (RQMC), and Monte Carlo (MC) (see the corresponding references in Section 4.1.2). The tree structures are chosen beforehand with constant branching coefficients, i.e., $|C(n)|$ is constant for all $n \in \mathcal{N} \setminus \mathcal{N}_T$. The tree decisions are extended by the three extension procedures introduced in Section 4.4: nearest-neigbhor across tree (NN-AT), nearest-neigbhor across children (NN-AC), and two-nearest-neigbhor-weighted (2NNW). The resulting STGM-EPs (summarized in Table 4.1) are compared by means of the quality parameters, and the selection of the best method is done in the average-case setting.

The generation of scenario trees for both case studies is based on the discretization of a standard normal $\mathcal{N}(0,1)$ distribution. Discretization by the OQ method is done by Algorithm 2 in Pflug et Pichler (2015), which minimizes the Wasserstein distance of order 2 between the $\mathcal{N}(0,1)$ distribution and its approximation sitting on finitely many points. This method provides a set of discretization points along with the corresponding probabilities.

Discretization by the RQMC method is done by the technique of *randomly shifted lattice rules* (see, e.g, L'Ecuyer et Lemieux (2000) and Sloan *et al.* (2002)). This technique randomizes a low discrepancy set of N points in $[0,1]$ and transforms it with the inverse $\mathcal{N}(0,1)$ cumulative $\phi^{-1} : (0,1) \to \mathbb{R}$. The output set of points is

$$\Big\{\phi^{-1}(\{i/N + u\}) \,|\, i = 0, \ldots, N-1\Big\}, \tag{4.55}$$

where u is a realization of a uniform distribution in $[0,1]$ and $\{\cdot\}$ is the fractional part function. The weight corresponding to each point is set to $1/N$, as it is customary in quasi-Monte Carlo. The set of points (4.55) enjoys two important properties, making it interesting for discretization the $\mathcal{N}(0,1)$ distribution: (i) each point has a marginal $\mathcal{N}(0,1)$ distribution, (ii) the points are not independent of each other, they cover uniformly the part of the support of $\mathcal{N}(0,1)$ where the probability mass is concentrated. We note that each new realization u implies a new set of points, therefore, RQMC is a *stochastic* scenario-tree generation method. We refer for instance to Koivu (2005) for the use of the RQMC method in stochastic programming.

Discretization by the MC method provides the so-called *Sample Average Approximation* (see Shapiro (2003)). Although MC is known by theoreticians to be less efficient than RQMC and OQ for sampling in small dimension, it is often used by practitioners because it remains the most natural and easiest way to generate scenarios. For this reason, we include it in the evaluation test.

The numerical experiments are implemented in Python 2.7.4 on a Linux machine with Intel Xeon X5472 @ 3.00GHz. We use CPLEX 12.6.1.0 with default setting to solve the scenario-tree deterministic programs.

Table 4.1 STGM-EPs considered in the numerical experiments.

Notation	**Description**
NN-AC	nearest-neighbor extension across children
NN-AT	nearest-neighbor extension across tree
2NNW	2-nearest-neighbor-weighted extension across tree

(a) Extension procedures

Notation	**Description**
OQ	optimal quantization method
RQMC	randomized quasi-Monte Carlo method
MC	Monte Carlo method

(b) Scenario-tree generation methods

4.5.2 Case study 1: the newsvendor problem

The newsvendor problem is stated as follows: A newsvendor buys to a supplier x_0 newspapers at stage 0 at a fixed price a. At stage 1, the newsvendor sells $x_{1,1}$ newspapers at price b and returns $x_{1,2}$ to the supplier, the latter pays c for each newspaper returned. Demand for newspapers is given by a positive random variable $\boldsymbol{\xi}_1$. Although the decisions involve integer variables, it is customary to relax the integrity constraints and to consider the corresponding continuous problem (see, e.g., Birge et Louveaux (2011)). The two-stage stochastic program takes the form

$$\max_{(x_0, x_{1,1}, x_{1,2})} \quad -a\, x_0 + \mathbb{E}[b\, x_{1,1}(\boldsymbol{\xi}_1) + c\, x_{1,2}(\boldsymbol{\xi}_1)] \tag{4.56}$$

$$\text{s.t.} \quad x_{1,1}(\boldsymbol{\xi}_1) \leq \boldsymbol{\xi}_1; \tag{4.57}$$

$$x_{1,1}(\boldsymbol{\xi}_1) + x_{1,2}(\boldsymbol{\xi}_1) \leq x_0; \tag{4.58}$$

$$x_0 \in \mathbb{R}_+,\ x_{1,1}(\boldsymbol{\xi}_1) \in \mathbb{R}_+,\ x_{1,2}(\boldsymbol{\xi}_1) \in \mathbb{R}_+. \tag{4.59}$$

The parameters are set to $a = 2$, $b = 5$, and $c = 1$. The demand $\boldsymbol{\xi}_1$ follows a log-normal distribution, i.e., $\log(\boldsymbol{\xi}_1)$ follows a $\mathcal{N}(\mu, \sigma^2)$ distribution, the mean is set to $\mu = \log(200)$ and the variance to $\sigma^2 = 1/2$ (these values for the parameters are taken from Proulx (2014)).

The optimal value of (4.56)-(4.59) rounded off to the second decimal is $Q(x^*) = 500.25$. The optimal stage-1 decision functions are $x^*_{1,1}(\boldsymbol{\xi}_1) = \min(x^*_0, \boldsymbol{\xi}_1)$ and $x^*_{1,2}(\boldsymbol{\xi}_1) = \max(x^*_0 - \boldsymbol{\xi}_1, 0)$.

The numerical experiments are performed with the methods in Table 4.1 and for scenario trees with 5, 20, 40, and 80 scenarios. Although the resulting scenario trees have small sizes, we will see that clear conclusions can be drawn from them. Moreover, we wish to assess the quality of methods with small scenario sizes, because a method that performs well with few scenarios can also be used to solve a generalization of the problem with more stages. Indeed, having good quality decisions with only 5 scenarios opens the door to the solution of the problem with, e.g., 10 stages, since a 10-stage scenario tree with a branching of 5 nodes at every stage remains tractable ($5^{10-1} \simeq 2 \times 10^6$ nodes). However, if 80 scenarios are required to obtain good quality decisions for the problem with 2 stages, then the same problem extended to 10 stages is intractable for the tested method ($80^{10-1} \simeq 10^{17}$ nodes).

Optimal selection of sample sizes:

The quality parameters are estimated using the procedure described in Section 4.3.3, for a computational time limit of one hour for each STGM-EP. The optimal sample sizes K^* and M^* are displayed in Table 4.2 (since the extensions *across tree* or *across children* coincide for two-stage problems, we remove the suffix -AT and -AC).

We have that $K^* = 1$ for the optimal quantization method, since it is a deterministic way to generate scenario trees (see Remark 4.1.1 and the discussion at the end of Section 4.3.2). We emphasize that each value $K^* \times M^*$ is the number of out-of-sample scenarios used to test the corresponding STGM-EP.

Quality parameters $p(1)$ and CR:

The probability of feasibility $p(1)$ and the conditional revenues CR given the whole feasibility of the extended tree policy $\widetilde{\boldsymbol{x}} = (\widetilde{\boldsymbol{x}}_0, \widetilde{\boldsymbol{x}}_{1,1}, \widetilde{\boldsymbol{x}}_{1,2})$ take the form (see Definition 4.2.1):

$$p(1) = \mathbb{P}_{(\boldsymbol{\xi}_1, \widetilde{\boldsymbol{x}})}[\boldsymbol{\xi}_1 \in \widetilde{\Xi}_1(\widetilde{\boldsymbol{x}})], \tag{4.60}$$

$$\mathrm{CR} = \mathbb{E}_{(\boldsymbol{\xi}_1, \widetilde{\boldsymbol{x}})}[-a\,\widetilde{\boldsymbol{x}}_0 + b\,\widetilde{\boldsymbol{x}}_{1,1}(\boldsymbol{\xi}_1) + c\,\widetilde{\boldsymbol{x}}_{1,2}(\boldsymbol{\xi}_1) \,|\, \boldsymbol{\xi}_1 \in \widetilde{\Xi}_1(\widetilde{\boldsymbol{x}})], \tag{4.61}$$

Table 4.2 Optimal sample sizes K^* and M^* for a limit of 1h of computation for each STGM-EP. The values M^* are rounded-off to the nearest 10^6 for OQ; the values K^* are rounded-off to the nearest 10^3 for RQMC and MC.

	5 scen.		20 scen.	
	K^*	M^*	K^*	M^*
OQ-NN	1	19×10^6	1	9×10^6
OQ-2NNW	1	10×10^6	1	6×10^6
RQMC-NN	168×10^3	32	64×10^3	50
RQMC-2NNW	161×10^3	19	66×10^3	28
MC-NN	179×10^3	25	79×10^3	24
MC-2NNW	152×10^3	23	62×10^3	34

	40 scen.		80 scen.	
	K^*	M^*	K^*	M^*
OQ-NN	1	5×10^6	1	3×10^6
OQ-2NNW	1	4×10^6	1	2×10^6
RQMC-NN	46×10^3	19	20×10^3	36
RQMC-2NNW	43×10^3	21	20×10^3	32
MC-NN	39×10^3	38	21×10^3	28
MC-2NNW	33×10^3	46	18×10^3	41

where $\widetilde{\Xi}_1(\widetilde{x}) = \{\xi_1 \in \Xi_1 \,|\, (\widetilde{x}_0, \widetilde{x}_{1,1}(\xi_1), \widetilde{x}_{1,2}(\xi_1)) \text{ satisfy } (4.57), (4.58), (4.59)\}$. The estimates of $p(1)$ and CR are displayed in Table 4.3. To facilitate the comparison with the optimal value $Q(x^*)$, the estimates of CR are given in percentage of $Q(x^*)$ and are denoted in the table by $\mathrm{CR}_{\%}$.

It follows from the estimates in Table 4.3 that a clear hierarchy exists among the three scenario-tree generation methods: OQ is better than RQMC which, in turn, are better than MC, as well as between the two extension procedures: 2NNW is better than NN. This hierarchy can be schematized as

$$\text{OQ} > \text{RQMC} > \text{MC} \quad \text{and} \quad \text{2NNW} > \text{NN}. \tag{4.62}$$

The couple OQ-2NNW with only 20 scenarios yields an extended tree policy that is very close to the optimal policy in terms of feasibility (99.8% of the time) and expected revenues (100.2% of the optimal ones); we recall that $\mathrm{CR}_{\%}$ may be greater than 100% when $p(1) < 1$, because $\mathrm{CR}_{\%}$ computes the expected revenues on a subset of values of the random parameters. This nearly achieves the goal we formulated in Section 4.1.3, i.e, to introduce the extension

Table 4.3 Estimates of the quality parameters $p(1)$ and $\mathrm{CR}_{\%}$. Data in bold font single out the STGM-EPs that satisfy $p(1) \geq 0.98$ and $\mathrm{CR} \geq 99\% \times Q(x^*)$, which can be considered as satisfactory. Confidence intervals are not displayed for the sake of clarity; the widest 95% confidence interval for each column from left to right are: ± 0.0009; ± 0.11; ± 0.001; ± 0.2; ± 0.0014; ± 0.3; ± 0.0017; ± 0.3.

	5 scen.		20 scen.		40 scen.		80 scen.	
	$p(1)$	$\mathrm{CR}_{\%}$	$p(1)$	$\mathrm{CR}_{\%}$	$p(1)$	$\mathrm{CR}_{\%}$	$p(1)$	$\mathrm{CR}_{\%}$
OQ-NN	0.618	102.1	0.620	114.4	0.622	116.8	0.623	118.2
OQ-2NNW	0.957	101.8	**0.998**	**100.2**	**0.999**	**100.1**	**0.997**	**100.3**
RQMC-NN	0.613	112.8	0.602	118.7	0.612	119.1	0.617	119.1
RQMC-2NNW	0.895	109.1	0.961	104.8	0.977	103.0	**0.985**	**102.0**
MC-NN	0.610	100.1	0.612	113.5	0.616	116.4	0.619	117.8
MC-2NNW	0.757	101.9	0.790	106.3	0.798	107.3	0.804	107.5

procedure to recover a decision policy of the original stochastic program, and to find the STGM-EP providing the closest policy to x^*.

The second best couple, namely RQMC-2NNW, provides good quality decisions from 80 scenarios. However, even for this many scenarios, the probability of feasibility is statistically significantly below that of OQ-2NNW. Any other couple (including that made by combining MC and 2NNW) yields decisions with low probability of feasibility (less than 80%). In that case, the conditional revenues CR cannot be trusted, as we explained in the discussion following Definition 4.2.1.

Quality parameter $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$:

Since the stage-1 recourse decisions are known analytically for the newsvendor problem, a decision-maker is interested in the stage-0 decision only. For this reason, we assess the quality of the feasible extended tree policy $\overline{\boldsymbol{x}} = (\widetilde{\boldsymbol{x}}_0, r_{1,1}, r_{1,2})$, where $(r_{1,1}, r_{1,2})$ are the recourse decisions given by

$$r_{1,1}(x_0; \xi_1) = \min(x_0, \xi_1), \tag{4.63}$$

$$r_{1,2}(x_0; \xi_1) = \max(x_0 - \xi_1, 0). \tag{4.64}$$

The quality of $\overline{\boldsymbol{x}}$ is assessed through the expected revenues that it yields, i.e., through the third quality parameter given by

$$\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})] = \mathbb{E}_{(\boldsymbol{\xi}_1, \widetilde{\boldsymbol{x}}_0)}[-a\,\widetilde{\boldsymbol{x}}_0 + b\,r_{1,1}(\widetilde{\boldsymbol{x}}_0; \boldsymbol{\xi}_1) + c\,r_{1,2}(\widetilde{\boldsymbol{x}}_0; \boldsymbol{\xi}_1)]. \tag{4.65}$$

Table 4.4 displays the estimates of $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$ in percentage of $Q(x^*)$, along with the corresponding 95% confidence interval bounds (column $\pm\text{CI}_{95\%}$). Since only the stage-0 decision is assessed by (4.65), the estimates do not depend on an extension procedure.

Table 4.4 Estimates of $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$ with $\overline{\boldsymbol{x}} = (\widetilde{\boldsymbol{x}}_0, r_{1,1}, r_{1,2})$ (given in percentage of $Q(x^*)$).

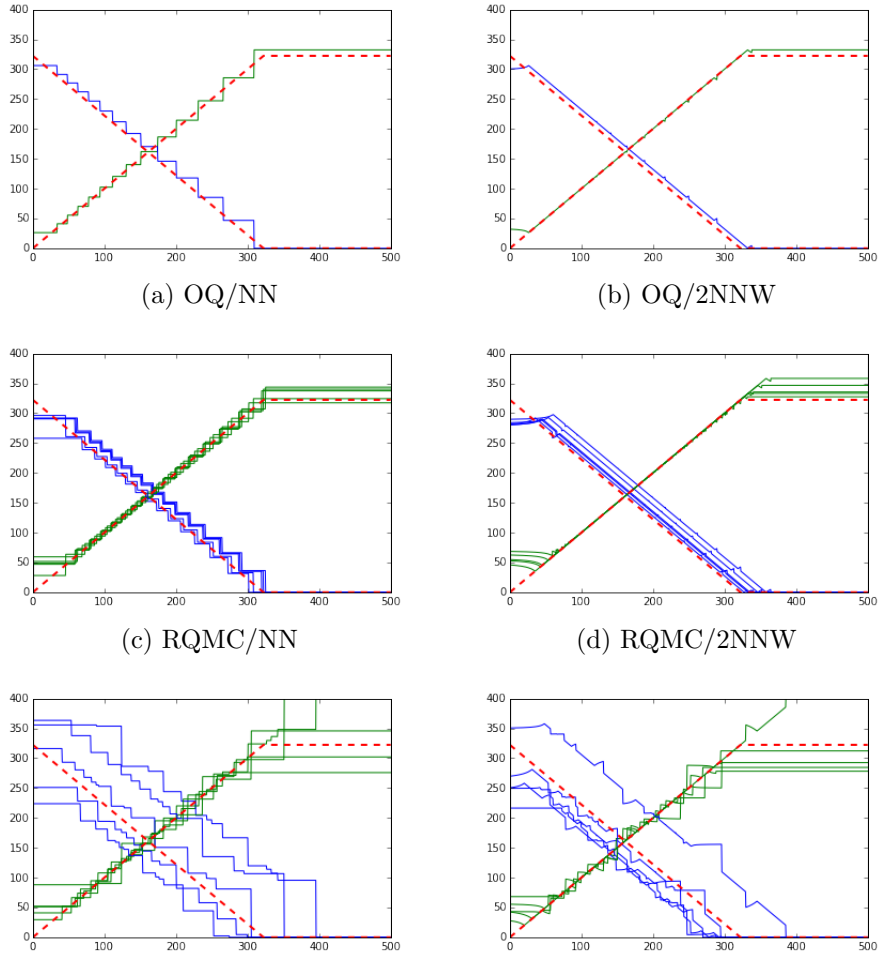
	5 scen.		20 scen.		40 scen.		80 scen.	
	Est.	$\pm\text{CI}_{95\%}$	Est.	$\pm\text{CI}_{95\%}$	Est.	$\pm\text{CI}_{95\%}$	Est.	$\pm\text{CI}_{95\%}$
OQ	99.80	0.04	99.96	0.05	99.95	0.07	100.01	0.09
RQMC	98.71	0.08	99.78	0.12	100.00	0.22	99.92	0.25
MC	91.44	0.11	97.22	0.15	98.62	0.17	99.35	0.27

The OQ method achieves 99.8% of the optimal expected revenues with *only 5 scenarios*, while 20 scenarios are needed for the RQMC method to reach this value. From 40 scenarios, the distinction between OQ and RQMC is made impossible by the statistical error (though small: less than ±0.25%), and both methods yield a decision close to optimality. The MC method yields the lowest quality stage-0 decision, which is statistically significantly below the OQ and RQMC methods for all tested scenario sizes. However, as far as the stage-0 decision is concerned, the use of MC with 80 scenarios is acceptable since the expected revenues are greater than 99% of the optimal ones. This last observation, combined with the estimates of $p(1)$ in Table 4.3, shows that the drawback of MC lies mostly in the poor quality of its stage-1 decision functions. This statement is supported by the graphical analysis done in the next paragraph.

Plots of the stage-1 decision functions

The extended stage-1 decision functions $\widetilde{x}_{1,1}$ and $\widetilde{x}_{1,2}$ are displayed for each STGM-EP in Figure 4.5, and are compared with their optimal counterparts $x^*_{1,1}$ and $x^*_{1,2}$. This graphical comparison allows to understand why some STGM-EPs provide expected revenues higher than others.

As expected from the above quantitative analysis, the couple OQ-2NNW in (b) yields stage-1 decisions that fit very well the optimal ones. The approximation quality is also quite good for RQMC-2NNW in (d). As for MC, in (e) and (f), it appears that the functions have a large variability due to the absence of a variance reduction technique, and an erratic behavior due to a discretization scheme that covers the support of $\boldsymbol{\xi}_1$ less uniformly than RQMC and OQ.



(a) OQ/NN (b) OQ/2NNW

(c) RQMC/NN (d) RQMC/2NNW

(e) MC/NN (f) MC/2NNW

Figure 4.5 Plots of $\widetilde{\boldsymbol{x}}_{1,1}$ and $\widetilde{\boldsymbol{x}}_{1,2}$ (solid lines) compared with $x^*_{1,1}$ and $x^*_{1,2}$ (dashed lines) for 20 scenarios. For RQMC and MC, five realizations of $\widetilde{\boldsymbol{x}}_{1,1}$ and $\widetilde{\boldsymbol{x}}_{1,2}$ are displayed on the same figure. The x-axis is the demand ξ_1.

4.5.3 Case study 2: the multi-product assembly problem

We consider the four-stage multi-product assembly problem presented in Defourny *et al.* (2013), Section 4.1, with the same numerical parameters and stochastic process. The stochas-

tic program takes the form:

$$\max_{(x_0,\dots,x_3)} \quad -c_0^\top x_0 + \mathbb{E}\Big[-c_1^\top x_1(\boldsymbol{\xi}_1) - c_2^\top x_2(\boldsymbol{\xi}_{..2}) + c_3^\top x_3(\boldsymbol{\xi}_{..3})\Big] \tag{4.66}$$

$$\text{s.t.} \quad \sum_{j=1}^{8} A_{i,j}\, x_{1,j}(\boldsymbol{\xi}_1) \le x_{0,i}, \quad \forall i \in \{1,\dots,12\}; \tag{4.67}$$

$$\sum_{k=1}^{5} B_{j,k}\, x_{2,k}(\boldsymbol{\xi}_{..2}) \le x_{1,j}(\boldsymbol{\xi}_1), \quad \forall j \in \{1,\dots,8\}; \tag{4.68}$$

$$x_{3,k}(\boldsymbol{\xi}_{..3}) \le \max(0, b_{k,0} + \sum_{t=1}^{3} b_{k,t}\boldsymbol{\xi}_t), \quad \forall k \in \{1,\dots,5\}; \tag{4.69}$$

$$x_{3,k}(\boldsymbol{\xi}_{..3}) \le x_{2,k}(\boldsymbol{\xi}_{..2}), \quad \forall k \in \{1,\dots,5\}; \tag{4.70}$$

$$x_0 \in \mathbb{R}_+^{12},\ x_1(\boldsymbol{\xi}_1) \in \mathbb{R}_+^{8},\ x_2(\boldsymbol{\xi}_{..2}) \in \mathbb{R}_+^{5},\ x_3(\boldsymbol{\xi}_{..3}) \in \mathbb{R}_+^{5}. \tag{4.71}$$

The interpretation of this problem is as follows: At stage 0, a quantity $x_{0,i}$ of a product i is purchased at a price $c_{0,i}$ per unit. At stage 1, a quantity $x_{1,j}$ of a product j is produced at a cost $c_{1,j}$ per unit from the quantity of products available at the previous stage, and in proportions given by the matrix A in (4.67). The same production scheme happens again at stage 2, with the quantity $x_{2,k}$, the cost $c_{2,k}$, and in proportions given by the matrix B in (4.68). At stage 3, a quantity $x_{3,k}$ of the final product k is sold at price $c_{3,k}$ per unit, and the demand for k is $\max(0, b_{k,0} + \sum_{t=1}^{3} b_{k,t}\boldsymbol{\xi}_t)$ where $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ are three i.i.d. $\mathcal{N}(0,1)$ variables.

The optimal value of the problem is estimated by Defourny *et al.* (2013) at about 375; we refer to their paper for more details, in particular for the values of the parameters. The numerical experiments are performed with the STGM-EPs in Table 4.1 and for scenario trees with 125, 512, and 1000 scenarios (corresponding to the branching coefficients 5, 8, and 10, respectively).

Optimal selection of sample sizes:

The quality parameters are estimated using the procedure described in Section 4.3.3, for a computational time limit of two hours for each STGM-EP. Table 4.5 displays the optimal sample sizes K^* and M^*. The value $K^* \times M^*$, which is the number of out-of-sample scenarios used to test the corresponding STGM-EP, is larger for the couples involving the extension NN-AC, since it is computationally less costly than the other two extensions (it does not require to partition the whole support of the random parameters at every stage).

Table 4.5 Optimal sample sizes K^* and M^* for a limit of 2h of computation for each STGM-EP. The values M^* are rounded-off to the nearest 10^6 for OQ.

	125 scen.		512 scen.		1000 scen.	
	K^*	M^*	K^*	M^*	K^*	M^*
OQ–NN-AC	1	11×10^6	1	9×10^6	1	7×10^6
OQ–NN-AT	1	6×10^6	1	3×10^6	1	2×10^6
OQ–2NNW	1	4×10^6	1	2×10^6	1	2×10^6
RQMC–NN-AC	4613	376	873	491	298	474
RQMC–NN-AT	4783	155	825	301	286	315
RQMC–2NNW	4639	139	792	409	275	323
MC–NN-AC	5079	172	842	707	294	457
MC–NN-AT	4829	147	863	159	278	318
MC–2NNW	4764	118	786	430	280	327

Quality parameters $p(t)$ and CR:

We compute the quality parameters $p(t)$ and CR for an extended decision policy of the form $\widetilde{\boldsymbol{x}} = (\widetilde{\boldsymbol{x}}_0, \widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, r_3)$, where r_3 is the optimal recourse function at stage 3 available analytically for this four-stage assembly problem. Although the use of a recourse policy was not initially introduced for the quality parameters $p(t)$ and CR, in this problem it is readily deducible from the constraints (4.69)-(4.70), hence we take it into account in the computation of $p(t)$ and CR. The optimal recourse decisions at stage 3 are given by

$$r_{3,k}(x_{..2}; \xi_{..3}) = \min\left(x_{2,k}, \max(0, b_{k,0} + \sum_{t=1}^{3} b_{k,t}\xi_t)\right), \quad \forall k \in \{1, \ldots, 5\}. \tag{4.72}$$

It follows that the probability of feasibility does not reduce from stage 2 to 3, i.e., we have $p(3) = p(2)$. Additionally, we note that all three extension procedures yield stage-1 decisions that satisfy the constraint (4.67), hence we also have $p(1) = 1$ for all STGM-EPs. We display in Table 4.6 the estimates of $p(2)$ and CR.

We note that the extension NN-AC satisfies *exactly* $p(2) = 1$, because the decisions that it yields always satisfy the constraint (4.68). The estimates in Table 4.6 clearly show the following hierarchy among the scenario-tree generation methods and extension procedures:

$$\text{OQ} > \text{RQMC} > \text{MC} \quad \text{and} \quad \text{NN-AC} > \text{NN-AT} > \text{2NNW}. \tag{4.73}$$

The hierarchy remains the same as the first case study for the scenario-tree generation methods. However, for the extension procedure, we have now that the nearest neighbor (NN) extensions yield better quality decisions than the extension with the 2 nearest neighbors

Table 4.6 Estimates of the quality parameters $p(2)$ and CR for an extended decision policy of the form $\widetilde{\boldsymbol{x}} = (\widetilde{\boldsymbol{x}}_0, \widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, r_3)$. Data in bold font single out the STGM-EPs that satisfy $p(2) \geq 0.98$. Parentheses in the CR-column include the 95% confidence intervals. In the $p(2)$-column this interval is not shown for the sake of clarity and because it is typically smaller than the last decimal displayed.

	125 scen.		512 scen.		1000 scen.	
	$p(2)$	CR	$p(2)$	CR	$p(2)$	CR
OQ–NN-AC	**1**	**366.1** (±0.4)	**1**	**370.7** (±0.5)	**1**	**372.8** (±0.5)
OQ–NN-AT	**0.986**	**366.3** (±0.6)	**0.986**	**370.9** (±0.8)	**0.985**	**371.9** (±0.9)
OQ–2NNW	0.402	352.3 (±1.2)	0.396	426.0 (±1.6)	0.394	365.1 (±1.8)
RQMC–NN-AC	**1**	**349.8** (±1.0)	**1**	**365.7** (±2.0)	**1**	**369.3** (±3.4)
RQMC–NN-AT	0.958	350.4 (±1.5)	0.958	367.3 (±2.6)	0.958	374.3 (±4.3)
RQMC–2NNW	0.334	3.4 (±2.6)	0.414	283.6 (±4.0)	0.414	351.3 (±7.6)
MC–NN-AC	**1**	**296.7** (±1.4)	**1**	**331.7** (±1.7)	**1**	**343.8** (±3.5)
MC–NN-AT	0.871	300.1 (±1.7)	0.873	340.2 (±3.8)	0.875	354.8 (±4.53)
MC–2NNW	0.356	93.0 (±3.0)	0.442	310.1 (±3.7)	0.446	383.2 (±7.1)

(2NNW). The reason for this could be that the optimal decision functions at stage 1 and 2 are close to piecewise constant, and therefore the NN extension is more suitable for approximating them.

In particular, we observe that the couple OQ–NN-AC with 1000 scenarios yields a decision policy with 100% probability of feasibility and expected revenues estimated at 372.8 ± 0.5, hence only about 0.6% away from optimality (assuming it is equal to 375, as estimated by Defourny *et al.* (2011)). As a result, it is possible to build a decision policy at almost optimality for the four-stage multi-product assembly problem using the optimal quantization method for generating the scenario tree and the nearest-neighbor extension across children for extending the decisions out of the set of scenarios.

Expected value solution

We end this section by showing that the four-stage multi-product assembly problem should not be solved by the deterministic approach, known as the *expected value solution* (see Birge et Louveaux (2011), Chapter 4), that consists in solving the program (4.66)-(4.71) where $\boldsymbol{\xi}$ is substituted by its expectation (hence resulting in a deterministic program). Since the random parameters enter the constraints at stage 3, the decisions obtained by such approach are feasible from stage 0 to 2. At stage 3, we complete these decisions by the recourse functions (4.72) to obtain a feasible decision policy, which can then be compared with the decision policies built from the extension procedures. The expected revenues obtained by

this approach (also known as the *expected result of using the expected value solution*) are estimated at 263 ± 1, with 95% confidence. We see that this value is much smaller than the expected revenues obtained by any of the scenario-tree generation method combined with the extension NN-AC (see Table 4.6).

4.5.4 Efficiency of the sample size selection technique

We end the numerical experiments by showing that the estimator $\widehat{\theta}_{K,M}$ defined in (4.29) and the optimal sample sizes selection introduced in Section 4.3 provide an efficient way for estimating the quality parameters. In particular, we want to show that it is more efficient that the classical estimator $\widetilde{\theta}_N$ that samples N times the random policy *and* the stochastic process, as opposed to $\widehat{\theta}_{K,M}$, which samples K times the random policy and $K \times M$ times the stochastic process. In the following, we use the notation introduced in Section 4.3.2 and 4.3.3.

Recall that we want to estimate an expectation of the form $\theta = \mathbb{E}[\phi(\boldsymbol{x}, \boldsymbol{\xi})]$. To this end, we define the estimator $\widetilde{\theta}_N$ as follows:

$$\widetilde{\theta}_N = \frac{1}{N} \sum_{n=1}^{N} \phi(x^n, \xi^n), \tag{4.74}$$

where $\{(x^n, \xi^n) \,|\, n = 1, \ldots, N\}$ is a set of N i.i.d. samples of $(\boldsymbol{x}, \boldsymbol{\xi})$. To compare the efficiency of $\widehat{\theta}_{K,M}$ and $\widetilde{\theta}_N$ for estimating θ, we compare their variances. They are given by

$$\mathrm{Var}(\widehat{\theta}_{K,M}) = \frac{\beta + \gamma(M-1)}{KM} \quad \text{and} \quad \mathrm{Var}(\widetilde{\theta}_N) = \frac{\beta}{N}, \tag{4.75}$$

where β and γ are defined in (4.31) and (4.32). More specifically, we want to compare $\mathrm{Var}(\widehat{\theta}_{K^*,M^*})$ and $\mathrm{Var}(\widetilde{\theta}_{N^*})$, where (K^*, M^*) is given by the optimal sample size selection technique described in Section 4.3.3, and N^* is the number of sample points (x^n, ξ^n) that can be obtained within the same time limit τ. Since the computation of $\widetilde{\theta}_N$ takes $N(t_0 + t_1 + t_2)$ units of time, we simply have $N^* = \lfloor \tau/(t_0 + t_1 + t_2) \rfloor$.

The variances are compared in the context of the first case study, where the quality parameter $\mathbb{E}_{\overline{\boldsymbol{x}}}[Q(\overline{\boldsymbol{x}})]$ was computed in Table 4.4 with a time limit of one hour for each scenario-tree generation method. Table 4.7 displays the ratio $\mathrm{Var}(\widetilde{\theta}_{N^*})/\mathrm{Var}(\widehat{\theta}_{K^*,M^*})$ for the RQMC and MC methods (the comparison is not relevant for the OQ method because $K^* = 1$). We see that the ratio ranges from about 8 to 16, hence $\mathrm{Var}(\widetilde{\theta}_{N^*})$ is typically one order of magnitude larger than $\mathrm{Var}(\widehat{\theta}_{K^*,M^*})$. This comparison shows that the estimator $\widehat{\theta}_{K^*,M^*}$ achieves a higher accuracy than $\widetilde{\theta}_{N^*}$ when the computational time is limited.

Table 4.7 Estimates of $\mathrm{Var}(\widetilde{\theta}_{N^*})/\mathrm{Var}(\widehat{\theta}_{K^*,M^*})$.

	5 scen.	20 scen.	40 scen.	80 scen.
RQMC	12.39	16.48	9.18	14.52
MC	7.84	10.11	14.06	12.25

4.6 Conclusion

In this paper, we introduced a quality evaluation framework, whose goal is to help the decision-maker to find the most suitable scenario-tree generation method and extension procedure (STGM-EP) to solve a given stochastic programming problem. From the scenario-tree optimal decisions, which are given for a finite set of scenarios only, the extension procedure builds a decision policy of the original problem. The framework consists of several quality parameters that assess the decisions yielded by a STGM-EP, and of several selection criteria that find the most suitable STGM-EP for a given problem in different settings. We focus in particular on the average-case setting, because we think that it is the most relevant when the problem is solved regularly with different data for the random parameters, which is the way scenario trees are often used in practice. The framework also includes the statistical tools (estimators, confidence intervals, sampling strategy) needed for an efficient application on a real-world problem. Overall, this newly introduced framework can be applied to a great deal of problems (two-stage or multistage, linear or nonlinear, continuous or integer variables) and to all scenario-tree generation methods, as far as we are aware.

We apply this framework on a two-stage newsvendor problem and a four-stage multi-product assembly problem. We demonstrate that simple extension procedures, such as the ones that set the decisions to the value of the nearest-neigbhor nodes in the scenario tree, provide good quality decisions at little computational cost. The application of the average-case selection criterion reveals that among the three tested methods, namely the Monte Carlo method, the randomized quasi-Monte Carlo method, and the optimal quantization method, the last one yields the highest quality decisions in both case studies. Randomized quasi-Monte Carlo method also yields good quality decisions, but it typically requires more scenarios to achieve a similar quality than the optimal quantization method. The quality of the Monte Carlo method is always statistically significantly below the other two methods for the small scenario sizes. The fact that the Monte Carlo method requires much more scenarios to provide acceptable quality decisions confirms that it is not suitable for multistage stochastic programming.

The quality evaluation framework was developed for stochastic programming problems with

an objective function given by an expectation. We think that future work should look for a generalization to objective functions that also include some risk measures. More extension procedures should also be investigated. Several techniques other than interpolation and extrapolation can be considered, such as regression and curve fitting. Another important future work would be the quality evaluation of decision policies on the so-called *rare events* (also known as *black swan events*). Methods that build scenario trees by taking into account the stochastic process, but not the variation of the revenue function, may not incorporate those events in the tree. This result in overoptimistic policies that may provide decisions with disastrous consequences should one of these events occur.

Overall, we hope that the approach introduced in this paper will bring more attention to the importance of the evaluation of decision policies and scenario-tree generation methods. We believe that quality evaluation techniques should eventually be included in every stochastic optimization software to help practitioners making better decisions.

CHAPITRE 5 ARTICLE 2: ON THE SCENARIO-TREE OPTIMAL-VALUE ERROR FOR STOCHASTIC PROGRAMMING PROBLEMS

Julien Keutchayan[1,2], David Munger[1,2], Michel Gendreau[1,2]

[1]Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Canada.

[2]Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Canada.

Article révisé soumis à la revue *Mathematics of Operations Research* le 20 juin 2018 après une première soumission le 20 décembre 2017.

Abstract. Stochastic programming problems generally lead to large-scale intractable programs if the number of possible outcomes for the random parameters is large or if the problem has many stages. A way to address these problems at lower computational cost is provided by the scenario-tree generation methods, which construct approximate problems from a finite subset of scenarios. When considering a general problem, the number of scenarios required to keep the optimal-value error within a given range grows exponentially with the number of random parameters and stages, which may lead to approximate problems that are themselves intractable. To overcome this fast growth of complexity, there is a need to look for scenario-tree generation methods tailored to specific classes of problems. In this paper, we provide a theoretical basis to develop such methods by studying the optimal-value error in the context of a general stochastic programming problem. We derive specifically two main results: an error decomposition and an error upper bound, both written as a sum of lower-level errors made at each node of the scenario tree. These two results provide the cornerstone to a new approach that takes into account the structure of the problem to generate suitable scenario trees.

5.1 Introduction

Stochastic programming is a subarea of mathematical programming used to model optimization problems involving uncertain parameters. The uncertain parameters are modeled by random vectors whose probability distributions are generally inferred from available data, and the objective function to be maximized is modeled as an expectation of a revenue or utility function; see, e.g., King et Wallace (2012). Stochastic programming problems are

found in various fields of applications, we refer for instance to: Yu *et al.* (2003) and Ziemba (2003) for applications in finance; Bertocchi *et al.* (2011) and Kovacevic *et al.* (2013) for applications in energy production and trading; Powell et Topaloglu (2003) and Yen et Birge (2006) for applications in transport and logistic; Beraldi *et al.* (2004) and Colvin et Maravelias (2008) for applications in medicine. For a general presentation on the theory and the solution methods in stochastic programming, we refer to Prékopa (1995), Ruszczyński et Shapiro (2003a) and Birge et Louveaux (2011).

Stochastic programming problems are generally highly computationally challenging to solve exactly, as shown in Dyer et Stougie (2006) and Hanasusanto *et al.* (2016). A way to address them at smaller computational cost consists in constructing an approximate problem with a finite subset of realizations obtained by discretizing the random parameters. These realizations are called *scenarios* and this solution approach is known as *scenario-tree generation*. The generation of scenario trees is subject to a trade-off. On the one hand, scenario trees must include a small number of scenarios to ensure that the approximate problem is solvable in reasonable times by optimization tools. On the other hand, this number must be large enough to ensure that the approximate problem provides accurate estimates of the optimal value and solutions of the original problem. This trade-off is fairly easy to satisfy if the problem has a reasonable size, i.e., a reasonable number of stages and random parameters per stage. However, as the size of the problem increases, the trade-off is more and more difficult to manage and, currently, problems with many stages and random parameters are generally out of reach of scenario-tree generation methods. An important challenge in stochastic programming is therefore the design of efficient scenario trees for such problems.

Many methods based on different theories have been proposed to generate scenario trees, we refer in particular to the following works: Shapiro et Homem-de Mello (1998) and Shapiro (2003) on *Monte Carlo methods*; Pennanen (2009), Koivu (2005) and Leövey et Römisch (2015) on *quasi-Monte Carlo methods* and *numerical integration rules*; Høyland et Wallace (2001) and Høyland *et al.* (2003) on *moment matching methods*; Pflug (2001) and Pflug et Pichler (2015) on *optimal quantization methods*; Dupačová *et al.* (2003), Heitsch et Römisch (2003) and Growe-Kuska *et al.* (2003) on *scenario reduction methods*. These methods have their own theoretical or practical justifications. For instance, Monte Carlo and quasi-Monte Carlo methods for two-stage problems are justified by several results about the asymptotic consistency of the methods, which prove that the optimal-value error decreases to zero as the number of scenarios increases to infinity; see, e.g., Shapiro et Homem-de Mello (2000), Homem-de Mello (2008), Mak *et al.* (1999) and Bastin *et al.* (2006). The asymptotic consistency of discretization methods in the multistage setting has been studied first by Olsen (1976a), and more recently by Pennanen et Koivu (2005) who provides conditions under

which the optimal value and solutions of the approximate problem converge to those of the original problem (property known as *epi-convergence*). The optimal-value error has also been extensively studied using probability metrics, which measure the distance between the true probability distribution of the random parameters and its scenario-tree approximation sitting on finitely many scenarios; see Pflug et Pichler (2011) for a review on probability metrics. Bounds on the optimal-value error by means of probability metrics, also known as *stability results*, have been derived for instance in Heitsch et Römisch (2009) and Pflug et Pichler (2012); see also Römisch (2003) for a detailed analysis on stability. As for the assessment of scenario-tree generation methods through the quality of the decisions rather than the optimal-value error, little work has been done in that direction, as it is difficult to quantify the error related to the implementation of the optimal decisions of the approximate and original problems; a general approach to do so is proposed in Keutchayan *et al.* (2017).

As of today, the use of scenario-tree generation methods for problems with many stages or random parameters is limited by the fast growth of the size of the scenario tree. We believe this limitation arises rapidly because current methods are essentially *distribution-driven*. This means that they build scenario trees with the goal to meet some criteria on the distribution of the stochastic process, and they do that fairly independently of the underlying structure of the problem which also depends on its objective function and constraints. In some particular cases, however, the choice of the method to use can be driven by the problem itself. For example, moment matching methods can be used when the optimal solution of the problem is known to be only sensitive to certain moments of the distribution (e.g., the optimal solution of the portfolio selection problem of Markowitz (1952) depends only on the first two moments of the distribution). Also, importance sampling methods can be used to take into account the shape of the objective function in a Monte Carlo sampling framework. However, for general problems such approaches are essentially heuristic. To broaden the set of solvable multistage problems, there is therefore a need to develop approaches that are problem-driven in essence. A *problem-driven* approach should build scenario trees in a way that is guided by the problem's structure and not only by the underlying stochastic process. An approach is problem-driven *in essence* if its theory is established independently of any particular problem. We can also require that its theory is not established within a specific numerical integration method, but rather spans over a large set of methods. Indeed, no numerical integration method is arguably the best for all integrals, hence developing a scenario-tree generation approach within a specific method would necessary close the door on the solution of some multistage problems. The contribution of this paper makes a big step in the direction of developing a problem-driven approach. We provide the ground and demonstrate the applicability of a new scenario-tree generation approach that incorporates knowledge on the

structure of the problem and can be used alongside many numerical integration methods to design more suitable scenario trees.

To this end, we study the optimal-value error that results from approximately solving a general multistage stochastic programming problem with a scenario tree. We derive specifically two main results: an optimal-value error *decomposition* and *upper bound*. Both are written as a sum of lower-level errors made at each node of the scenario tree. The latter errors are called *low-level* because they concern each individual node, as opposed to the optimal-value error which is *high-level* because it concerns the whole scenario tree. Our two main results show how the high-level error emerges as a combination of the low-level errors and, therefore, provide insight into the way the former can be kept small by acting appropriately on the latter. In particular, the upper bound can be written as a sum of worst-case integration errors in some specific function spaces. This feature makes it attractive for scenario-tree generation, as worst-case errors are largely studied in the context of numerical integration, where they are used to develop efficient approximation methods for high-dimensional integrals. The new scenario-tree generation framework is then based on the idea of minimizing the error upper bound. We note that two other approaches based on minimizing an error upper bound have been developed in Heitsch *et al.* (2006) and Pflug et Pichler (2012), however, their bounds are given by a distance between probability distributions (*filtration distance* for the former and *nested distance* for the latter), which makes these two approaches distribution-driven.

The results derived in this paper hold for any multistage stochastic programming problem whose recourse functions (5.3)-(5.4) are well-defined and finite over their effective domain of definition. To make this condition more concrete, we have divided it into a set of five more easily verifiable conditions on the constraints, revenue function and stochastic process, which are introduced along the lines of the problem description. These conditions are kept as general as possible to include as many problems as possible. For instance, no assumption is made on the linearity, convexity or differentiability of the revenue function, although in practice most problems will require some of these properties to hold in order to be effectively able to compute the optimal solutions. Moreover, these conditions are *sufficient* to ensure the well-definedness and finiteness of the recourse functions, but they are by no means *necessary*. For example, for conciseness of the proof we have assumed that the feasible sets are bounded, but problems with unbounded feasible sets can also fall into the framework of this paper as long as this does not result in ill-defined recourse functions.

The remainder of this paper is organized as follows: Section 5.2 contains the preliminaries of the two main results. In particular, Section 5.2.1 formulates the multistage stochastic programming problem and Section 5.2.2 formulates the scenario tree and the approximate

problem. Section 5.2.3 introduces a more concise notation for the quantities described in Sections 5.2.1-5.2.2, which will simplify the following mathematical developments. Sections 5.3 and 5.4 contain the error decomposition and the error bound, respectively. Section 5.5 shows how these results can be used to improve scenario-tree generation. Finally, Section 5.6 concludes the paper.

5.2 Preliminaries

We consider a *stochastic programming problem* where decisions are made at discrete time stages $t = 0, 1, \ldots, T \in \mathbb{N}_+$, where $\mathbb{N}_+$ stands for the positive integers. *Multistage* problems correspond to the case $T \geq 2$, while *two-stage* problems correspond to $T = 1$. For the sake of conciseness, all results in this paper are formulated for $T \geq 2$, but the reader can easily deduce the corresponding results for two-stage problems.

5.2.1 Multistage stochastic programming problem formulation

Multistage stochastic programming problems deal with random parameters whose dynamic is represented by a discrete-time stochastic process of the form $\boldsymbol{\xi} := (\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Each $\mathbb{R}^d$-valued random vector $\boldsymbol{\xi}_t$, for $t > 0$, contains the random parameters revealed during the period $(t-1, t)$. The probability distribution of $\boldsymbol{\xi}_t$ can be arbitrary (hence it can be discrete, continuous, or a combination of both) and its support (possibly unbounded) is denoted by $\Xi_t \subseteq \mathbb{R}^{d_t}$ with $d_t \in \mathbb{N}_+$. For convenience we have added an artificial random vector $\boldsymbol{\xi}_0$ that takes only one value ξ_0 with probability one. Throughout this paper, random vectors are distinguished from their realizations by writing the former in bold font. We denote by Ξ the support of $\boldsymbol{\xi}$, by $\Xi_{..t}$ the support of $\boldsymbol{\xi}_{..t} := (\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_t)$, and by $\Xi_t(\xi_{..t-1})$ the conditional support of $\boldsymbol{\xi}_t$ given the event $\{\omega \in \Omega : \boldsymbol{\xi}_{..t-1}(\omega) = \xi_{..t-1}\} \in \mathcal{A}$.

The decision vector y_t at stage $t \in \{0, \ldots, T\}$ satisfies the constraints of the problem if it belongs to the *feasible set* $Y_t \subseteq \mathbb{R}^{s_t}$ with $s_t \in \mathbb{N}_+$. (For the sake of clarity, and without loss of generality, we assume throughout this paper that $s_t = s$ and $d_t = d$ for all t.) When $t \geq 1$, we also denote the feasible set by $Y_t(y_{..t-1}; \xi_{..t})$ to emphasize that it possibly depends on the decisions $y_{..t-1} := (y_0, \ldots, y_{t-1}) \in \mathbb{R}^{st}$ and on the realization $\xi_{..t} := (\xi_0, \ldots, \xi_t) \in \Xi_{..t}$. We consider feasible sets that can include continuous or discrete variables and that can be represented as the solutions of (finitely or) countably many equality and inequality constraints. In the following condition, let $A_t \subseteq \mathbb{R}^s$ be a closed set for $t \in \{0, \ldots, T\}$ (typically $A_t = \mathbb{R}^s$ for continuous variables and $A_t = \mathbb{Z}^s$ or $A_t = \{0, 1\}^s$ for integer or binary variables) and let $\Pi_{i=0}^t$ denote the $(t+1)$-fold Cartesian product:

Condition 5.2.1. Define $Z_0 := Y_0$ and $Z_t(\xi_{..t}) := \{y_{..t} \in \mathbb{R}^{s(t+1)} : y_{..t-1} \in Z_{t-1}(\xi_{..t-1}),\ y_t \in Y_t(y_{..t-1}; \xi_{..t})\}$ for each $t \in \{1, \dots, T\}$ and $\xi_{..t} \in \Xi_{..t}$. The feasible sets are such that:

(i) At stage 0:

$$y_0 \in Z_0 \quad \Longleftrightarrow \quad y_0 \in A_0 \quad \text{and} \quad \begin{cases} g_{0,i}(y_0) = 0 & \text{for } i \in I_0; \\ g_{0,i}(y_0) \geq 0 & \text{for } i \in J_0; \end{cases} \tag{5.1}$$

with $g_{0,i} : \mathbb{R}^s \to \mathbb{R}$ continuous for $i \in J_0$ and upper semi-continuous for $i \in J_0$ and $I_0, J_0 \subseteq \mathbb{N}$.

(ii) At stage $t \in \{1, \dots, T\}$ for every $\xi_{..t} \in \Xi_{..t}$:

$$y_{..t} \in Z_t(\xi_{..t}) \quad \Longleftrightarrow \quad y_{..t} \in \Pi_{i=0}^{t} A_i \quad \text{and} \quad \begin{cases} g_{t,i}(y_{..t}; \xi_{..t}) = 0 & \text{for } i \in I_t; \\ g_{t,i}(y_{..t}; \xi_{..t}) \geq 0 & \text{for } i \in J_t; \end{cases} \tag{5.2}$$

with $g_{t,i} : \mathbb{R}^{s(t+1)} \times \Xi_{..t} \to \mathbb{R}$ a Carathéodory integrand for $i \in I_t$ and a normal integrand for $i \in J_t$ and $I_t, J_t \subseteq \mathbb{N}$.

The above functions $g_{t,i}$ model the constraints of the stochastic programming problem. For the definitions of *normal* and *Carathéodory* integrands, we refer to Rockafellar et Wets (2009), Definition 14.27 and Example 14.29, respectively. Note that, since in this paper we consider a maximization problem, normal integrands are *upper* semi-continuous by convention. The motivation for Condition 5.2.1 is the fact that it implies that the set Z_0 is closed and that the set-valued mapping $\xi_{..t} \rightrightarrows Z_t(\xi_{..t})$ is closed-valued and measurable; see (Rockafellar et Wets, 2009, Theorem 14.36) and more generally (Rockafellar et Wets, 2009, Chapter 5) for an introduction on set-valued mappings.

We also require a boundedness condition on Z_0 and $Z_t(\xi_{..t})$, which, together with Condition 5.2.1, imply that Z_0 and $Z_t(\xi_{..t})$ are compact sets for every $\xi_{..t}$.

Condition 5.2.2. The set Z_0 is bounded in $\mathbb{R}^s$, and so is $Z_t(\xi_{..t})$ in $\mathbb{R}^{s(t+1)}$ for every $t \in \{1, \dots, T\}$ and $\xi_{..t} \in \Xi_{..t}$.

We restrict our attention to stochastic programming problems that have a non-empty feasible set and a relative complete recourse at every stage, as expressed in Condition 5.2.3:

Condition 5.2.3. The set Y_0 is non-empty, and so is $Y_t(y_{..t-1}; \xi_{..t})$ for every $t \in \{1, \dots, T\}$, $\xi_{..t} \in \Xi_{..t}$, and $y_{..t-1} \in Z_{t-1}(\xi_{..t-1})$.

We introduce a *revenue function* $q : \mathbb{R}^{s(T+1)} \times \Xi \to \mathbb{R}$. The value $q(y; \xi)$ represents all the revenues obtained from stage 0 to T by implementing the decisions $y = (y_0, \dots, y_T)$ in the realization $\xi = (\xi_0, \dots, \xi_T)$.

Given the stochastic process, the feasible sets and the revenue function, we can now define the stage-t *expected recourse function* Q_t and *optimal recourse function* Q_t^*, which characterize the stochastic programming problem. These functions are defined recursively by the following stochastic dynamic programming equations:

$$Q_t^*(y_{..t-1};\xi_{..t}) := \sup_{\{y_t\in\mathbb{R}^s:(y_{..t-1},y_t)\in Z_t(\xi_{..t})\}} Q_t(y_{..t-1},y_t;\xi_{..t}), \qquad \forall t\in\{0,\dots,T\}, \tag{5.3}$$

$$Q_t(y_{..t};\xi_{..t}) := \mathbb{E}[Q_{t+1}^*(y_{..t};\xi_{..t},\boldsymbol{\xi}_{t+1}) \,|\, \boldsymbol{\xi}_{..t}=\xi_{..t}], \qquad \forall t\in\{0,\dots,T-1\}, \tag{5.4}$$

where for $t=T$ the equation (5.3) is initialized by setting $Q_T(y;\xi) := q(y;\xi)$, and for $t=0$ the left-hand side of (5.3) is written Q_0^*.

To ensure that the above recourse functions are well-defined and finite over their domain of definition, we add the following two conditions on the revenue function.

Condition 5.2.4. The function $q(\cdot;\xi)$ is upper semi-continuous for every $\xi\in\Xi$ and $q(\cdot;\cdot)$ is $\mathcal{B}(\mathbb{R}^{s(T+1)})\otimes\Sigma$-measurable, where $\mathcal{B}(\mathbb{R}^{s(T+1)})$ is the Borel σ-algebra of $\mathbb{R}^{s(T+1)}$ and (Ξ,Σ) is a complete measurable space with respect to the distribution of $\boldsymbol{\xi}$.

Condition 5.2.4 directly implies that the revenue function is a normal integrand by (Rockafellar et Wets, 2009, Corollary 14.34).

Condition 5.2.5. There exists a measurable function $h:\Xi\to\mathbb{R}$ such that $|q(y(\xi);\xi)|\le h(\xi)$ for all $\xi\in\Xi$ and all measurable selections $y(\xi)\in Z_T(\xi)$, where h satisfies $\mathbb{E}[|h(\boldsymbol{\xi})|]<\infty$ and $\mathbb{E}[|h(\boldsymbol{\xi})|\,|\,\boldsymbol{\xi}_{..t}=\xi_{..t}]<\infty$ for all $t\in\{1,\dots,T-1\}$ and $\xi_{..t}\in\Xi_{..t}$.

Note that Condition 5.2.5 requires that the conditional integrability of $h(\boldsymbol{\xi})$ given $\boldsymbol{\xi}_{..t}=\xi_{..t}$ holds for any $\xi_{..t}\in\Xi_{..t}$, and not merely for almost every $\xi_{..t}\in\Xi_{..t}$. The reason is that we want the stage-t recourse functions (5.3)-(5.4) to be defined everywhere on $\Xi_{..t}$, which will guarantee that the *node errors* and the *subtree errors*, introduced in Section 5.3, are well-defined even if the scenarios are chosen in a non-random fashion.

The quantity Q_0^* is the *optimal value* of the multistage problem. The *optimal decision policy* of the problem is $x^*=(x_0^*,\dots,x_T^*)$ where x_0^* is a maximizer of $Q_0(\cdot)$ and, for each $t>0$, x_t^* is a function defined on the graph of the set-valued mapping $\xi_{..t}\rightrightarrows Z_{t-1}(\xi_{..t-1})$ such that $x_t^*(y_{..t-1};\xi_{..t})$ is a maximizer of $Q_t(y_{..t-1},\cdot;\xi_{..t})$ at the right-hand side of (5.3). It follows from the five conditions given above that an optimal decision policy exists (it is not necessarily unique) and that the recourse functions are well-defined and finite-valued for all $\xi_{..t}\in\Xi_{..t}$ and $y_{..t}\in Z_t(\xi_{..t})$ (cf. proof in Appendix).

Remark 5.2.1. The revenue function $q(\cdot;\cdot)$ is considered to be path-dependent and no assumption about linearity, convexity or differentiability is made to keep it as general as possible. Most revenue functions encountered in practice will follow from this general form as a particular case and the equations (5.3)-(5.4) will provide the corresponding multistage problem. For example, if one works with linear stage-additive revenues of the form $q(y;\xi) = c_0^\top y_0 + \sum_{t=1}^T c_t(\xi_t)^\top y_t$, then it follows directly from the backward recursion (5.3)-(5.4) that

$$Q_0^* = \sup_{y_0} \mathbb{E}_{\boldsymbol{\xi}_1}\Big[\sup_{y_1} \mathbb{E}_{\boldsymbol{\xi}_2|\boldsymbol{\xi}_1}\Big[\cdots \mathbb{E}_{\boldsymbol{\xi}_T|\boldsymbol{\xi}_{..T-1}}\Big[\sup_{y_T}\big(c_0^\top y_0 + c_1(\boldsymbol{\xi}_1)^\top y_1 + \cdots + c_T(\boldsymbol{\xi}_T)^\top y_T\big)\Big]\Big]\Big]. \quad (5.5)$$

The terms of the inner sum up to $c_{t-1}(\boldsymbol{\xi}_{t-1})^\top y_{t-1}$ can exit the supremums over $y_t, \ldots, y_T$, the terms up to $c_t(\boldsymbol{\xi}_t)^\top y_t$ can exit the expectations with respect to $\boldsymbol{\xi}_{t+1}, \ldots, \boldsymbol{\xi}_T$, which then yields the so-called *nested formulation* of linear multistage problems:

$$Q_0^* = \sup_{y_0} c_0^\top y_0 + \mathbb{E}_{\boldsymbol{\xi}_1}\Big[\sup_{y_1} c_1(\boldsymbol{\xi}_1)^\top y_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\boldsymbol{\xi}_1}\Big[\cdots + \mathbb{E}_{\boldsymbol{\xi}_T|\boldsymbol{\xi}_{..T-1}}\Big[\sup_{y_T} c_T(\boldsymbol{\xi}_T)^\top y_T\Big]\Big]\Big]; \quad (5.6)$$

see, e.g., Ruszczyński et Shapiro (2003c) for a detailed analysis of such problems.

5.2.2 Scenario-tree and approximate problem formulations

The optimal value Q_0^* and the optimal policy x^* of the multistage stochastic programming problem are not readily available in general. The scenario-tree approach to estimate Q_t consists in approximating the right-hand side of (5.4) by a weighted average of the values of Q_{t+1} for a selection of realizations of $\boldsymbol{\xi}_{t+1}$. In turn, Q_{t+1} is approximated in terms of Q_{t+2}, and this recursive discretization scheme is carried out through stage T, where the values of Q_T are computed directly from the revenue function q. A tree structure naturally arises from this scheme, in which sibling nodes at stage $t+1$ represent the discrete values of Q_{t+1} whose weighted average approximates the value of Q_t, represented by their common parent node at stage t. The remainder of this section formalizes the scenario tree and the corresponding approximate problem.

We define a scenario tree as a triplet $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ where $\mathcal{T} = (\mathcal{N}, \mathcal{E}, n_0)$ is a *rooted tree structure*, with $\mathcal{N}$ the (finite) node set, $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ the edge set and $n_0 \in \mathcal{N}$ the root node; $\mathcal{P} = \{\zeta^n : n \in \mathcal{N}\} \subset \mathbb{R}^d$ is the set of *discretization points*; and $\mathcal{W} = \{w^n : n \in \mathcal{N}\} \subset (0, \infty)$ is the set of *discretization weights*.

The rooted tree structure $\mathcal{T}$ is such that T edges separate the root from any tree leaf, where a *leaf* is any node $n \neq n_0$ incident to only one edge. To describe conveniently the structure, we introduce the following notations: $\mathcal{N}^* := \mathcal{N} \setminus \{n_0\}$; $C(n)$ is the set of *child nodes* of n

(i.e., the nodes linked to n at the next stage); $p(n)$ is the *parent node* of n (i.e., the node linked to n at the previous stage); $t(n)$ is the *stage* or *depth* of n (i.e., the number of edges between n_0 and n); $\mathcal{N}_t := \{n \in \mathcal{N} : t(n) = t\}$ is the set of all stage-t nodes; $[n_0, n]$ is the unique sequence of nodes from n_0 to n; and $\mathcal{T}(n) := (\mathcal{N}(n), \mathcal{E}(n), n)$ is the *subtree rooted at node* $n \in \mathcal{N}$ obtained by setting n as the root node and by considering only the nodes that are the descendants of n, $\mathcal{N}(n)$, and the edges connecting them, $\mathcal{E}(n)$.

The discretization points $\mathcal{P}$ are drawn (deterministically or randomly) in the support of the stochastic process. More specifically, we have that $\zeta^{n_0} = \xi_0$ and, for each $n \in \mathcal{N}^*$, ζ^n is a discretization point of $\boldsymbol{\xi}_{t(n)}$ that satisfies $\zeta^n \in \Xi_{t(n)}(\zeta^{\cdot p(n)})$, where $\zeta^{\cdot\cdot n} := (\zeta^m)_{m \in [n_0, n]}$ is the discretization sequence from n_0 to $n \in \mathcal{N}$.

The discretization weights $\mathcal{W}$ are considered to be positive and not necessarily summing to one within each subset $\{w^m : m \in C(n)\}$. They will sum to one if they are seen as the probabilities of the points $\{\zeta^m : m \in C(n)\}$. We have that $w^{n_0} = 1$ and, for each $n \in \mathcal{N}^*$, the value w^n represents the weight of node n with respect to its sibling nodes, $C(p(n))$. The weight of n with respect to all nodes $\mathcal{N}_{t(n)}$ is given by the *product weight* W^n defined as

$$W^n = \prod_{m \in [n_0, n]} w^n. \tag{5.7}$$

If the discretization weights sum to one within each subset $\{w^m : m \in C(n)\}$, then the product weight W^n can be interpreted as the probability of the scenario $\zeta^{\cdot\cdot n}$ and the product weights will sum to one within each subset $\{W^n : n \in \mathcal{N}_t\}$.

Now that we have formalized the scenario tree, we can define the approximate problem that corresponds to it. It is defined in the same way as the original problem but restricted to a subset of scenarios. The recourse functions $\widehat{Q}_t^*$ and $\widehat{Q}_t$ of the approximate problem, which are the scenario-tree estimators of Q_t^* and Q_t, are defined recursively as

$$\widehat{Q}_t^*(y_{\cdot\cdot t-1}; \zeta^{\cdot\cdot n}) := \sup_{y_t \in Y_t(y_{\cdot\cdot t-1}; \zeta^{\cdot\cdot n})} \widehat{Q}_t(y_{\cdot\cdot t-1}, y_t; \zeta^{\cdot\cdot n}), \qquad \forall t \in \{0, \dots, T\},\ \forall n \in \mathcal{N}_t, \tag{5.8}$$

$$\widehat{Q}_t(y_{\cdot\cdot t}; \zeta^{\cdot\cdot n}) := \sum_{m \in C(n)} w^m\, \widehat{Q}_{t+1}^*(y_{\cdot\cdot t}; \zeta^{\cdot\cdot n}, \zeta^m), \qquad \forall t \in \{0, \dots, T-1\},\ \forall n \in \mathcal{N}_t, \tag{5.9}$$

where for $t = T$ the equation (5.8) is initialized by setting $\widehat{Q}_T(y; \zeta^{\cdot\cdot n}) := q(y; \zeta^{\cdot\cdot n})$ for all $n \in \mathcal{N}_T$, and for $t = 0$ the left-hand side of (5.8) is written $\widehat{Q}_0^*$.

The quantity $\widehat{Q}_0^*$ is the *optimal value* of the approximate problem. It is the scenario-tree estimator of the optimal value Q_0^* of the original problem and the quantity $Q_0^* - \widehat{Q}_0^*$ is what we refer to as the *optimal-value error*. The *optimal decision policy* of the approximate

problem is $\widehat{x}^* = (\widehat{x}_0^*, \ldots, \widehat{x}_T^*)$ where $\widehat{x}_0^*$ is a maximizer of $\widehat{Q}_0(\cdot; \zeta^{n_0})$ and, for each $t > 0$, $\widehat{x}_t^*$ is a function defined on the support of the set-valued mapping $\zeta^{\cdot\cdot n} \rightrightarrows Z_{t-1}(\zeta^{\cdot\cdot p(n)})$ such that $\widehat{x}_t^*(y_{\cdot\cdot t-1}; \zeta^{\cdot\cdot n})$ is a maximizer of $\widehat{Q}_t(y_{\cdot\cdot t-1}, \cdot; \zeta^{\cdot\cdot n})$ at the right-hand side of (5.8). It follows from the five conditions in Section 5.2.1 that an optimal decision policy exists for the approximate problem and that the recourse functions are well-defined and finite-valued for all $n \in \mathcal{N}_t$ and $y_{\cdot\cdot t} \in Z_t(\zeta^{\cdot\cdot n})$.

We emphasize that our formulation of the scenario tree estimators is general, since we do not assume a specific form for the tree structure and for the discretization points and weights. This generality allows us to consider essentially all ways to generate scenario trees. For instance, the fact that the weights w^m in (5.9) need not sum to one allows to consider the use of *sparse grid methods* (see, e.g., Holtz (2010)) or *importance sampling methods*; in the latter case w^m is defined through the likelihood ratio function and hence may not sum to one. A well-known particular case of the approximate problem is the so-called *sample average approximation*, which is obtained from (5.8)-(5.9) by setting $w^m = |C(n)|^{-1}$ and by getting the points $\mathcal{P}$ through Monte Carlo methods; see, e.g, Shapiro (2003).

Remark 5.2.2. Following Remark 5.2.1, consider the case of a linear stage-additive revenue function $q(y; \xi) = c_0^\top y_0 + \sum_{t=1}^T c_t(\xi_t)^\top y_t$. It follows directly from the backward recursion (5.8)-(5.9) that the approximate problem can be written as (take $T = 2$ for simplicity):

$$\widehat{Q}_0^* = \sup_{y_0} \sum_{n \in C(n_0)} w^n \left\{ \sup_{y_1} \sum_{m \in C(n)} w^m \left\{ \sup_{y_2} \left(c_0^\top y_0 + c_1(\zeta^n)^\top y_1 + c_2(\zeta^m)^\top y_2 \right) \right\} \right\}. \tag{5.10}$$

Denoting by $\widehat{y}^n$ the decision vector at node $n \in \mathcal{N}$, we can gather all supremums on the left, which yields:

$$\widehat{Q}_0^* = \sup_{(\widehat{y}^n)_{n \in \mathcal{N}}} \sum_{n \in C(n_0)} w^n \sum_{m \in C(n)} w^m \left(c_0^\top \widehat{y}^{n_0} + c_1(\zeta^n)^\top \widehat{y}^n + c_2(\zeta^m)^\top \widehat{y}^m \right), \tag{5.11}$$

or equivalently (using the product weights):

$$\widehat{Q}_0^* = \sup_{(\widehat{y}^n)_{n \in \mathcal{N}}} \sum_{n \in \mathcal{N}_2} W^n \left(c_0^\top \widehat{y}^{n_0} + c_1(\zeta^{p(n)})^\top \widehat{y}^{p(n)} + c_2(\zeta^n)^\top \widehat{y}^n \right). \tag{5.12}$$

We refer to Ruszczyński (2003) for a detailed review of the decomposition methods that can be used to solve the approximate problem in the specific setting of convex revenue functions.

5.2.3 Policy-based formulations

It is more convenient for future developments in this paper to express the original and approximate problems in a pure *policy-based* formulation, where the recourse functions now take as first argument a *decision policy* instead of a *decision vector*. This allows, for instance, to have a single notation for the expected and optimal recourse functions of each problem. In Section 5.2.1, we have introduced the optimal decision policy $(x_0^*, \ldots, x_T^*)$ of the original problem where $x_t^*(y_{..t-1}; \xi_{..t})$ provides the optimal decision vector at stage t given as a function of the decisions $y_{..t-1}$ and the realization $\xi_{..t}$. The optimal decision policy $(\widehat{x}_0^*, \ldots, \widehat{x}_T^*)$ was introduced similarly in Section 5.2.2 for the approximate problem. We can generalize these notations to represent *any* feasible decision at stage t for the original and approximate problems as a function of $(y_{..t-1}; \xi_{..t})$. The development below formalizes this approach and shows the link with the previous definitions.

We define a *feasible decision policy of the original problem* as a sequence $x := (x_0, \ldots, x_T)$ where x_0 is a vector such that $x_0 \in Z_0$ and x_t, for $t \in \{1, \ldots, T\}$, is a function defined on the graph of the set-valued mapping $\xi_{..t} \rightrightarrows Z_{t-1}(\xi_{..t-1})$ such that

$$x_t(y_{..t-1}; \xi_{..t}) \in Y_t(y_{..t-1}; \xi_{..t}), \quad \forall \xi_{..t} \in \Xi_{..t}, \forall y_{..t-1} \in Z_{t-1}(\xi_{..t-1}). \tag{5.13}$$

The set of all such functions x_t is denoted by $\mathcal{X}_t$. At stage 0 we define $\mathcal{X}_0 = Z_0$. The (policy-based) stage-t recourse function of the original problem is $Q_t : \Pi_{i=0}^t \mathcal{X}_i \times \Xi_{..t} \to \mathbb{R}$ defined as

$$Q_t(x_{..t}; \xi_{..t}) := Q_t(x_{..t}(\xi_{..t}); \xi_{..t}), \quad \forall x_{..t} \in \Pi_{i=0}^t \mathcal{X}_i, \tag{5.14}$$

where the right-hand side is the (vector-based) recourse function (5.4) (abusing notation slightly we use the same Q_t-notation) and $x_{..t}(\xi_{..t})$ denotes the vectors of decisions given by the policy $x_{..t}$ in the realization $\xi_{..t}$:

$$x_{..t}(\xi_{..t}) := (y_0, \ldots, y_t) \quad \text{where} \quad y_i = \begin{cases} x_0 & \text{if } i = 0; \\ x_i(y_{..i-1}; \xi_{..i}) & \text{if } i \in \{1, \ldots, t\}. \end{cases} \tag{5.15}$$

Note that the policy-based recourse function does gather the two recourse functions (5.3)-(5.4) in a single notation, as the recourse function (5.3) can now be written $Q_t(x_{..t-1}, x_t^*; \xi_{..t})$ by definition of x_t^*.

A *feasible decision policy of the approximate problem* is a sequence $\widehat{x} := (\widehat{x}_0, \ldots, \widehat{x}_T)$ where $\widehat{x}_0 \in Z_0$ and $\widehat{x}_t$, for $t \in \{1, \ldots, T\}$, is a function defined on the graph of the set-valued

mapping $\zeta^{\cdot\cdot n} \rightrightarrows Z_{t-1}(\zeta^{\cdot\cdot p(n)})$ such that

$$\widehat{x}_t(y_{\cdot\cdot t-1};\zeta^{\cdot\cdot n}) \in Y_t(y_{\cdot\cdot t-1};\zeta^{\cdot\cdot n}), \quad \forall n \in \mathcal{N}_t,\, \forall y_{\cdot\cdot t-1} \in Z_{t-1}(\zeta^{\cdot\cdot p(n)}). \tag{5.16}$$

The set of all such functions $\widehat{x}_t$ is denoted by $\widehat{\mathcal{X}}_t$. At stage 0 we define $\widehat{\mathcal{X}}_0 = Z_0$ (hence $\widehat{\mathcal{X}}_0 = \mathcal{X}_0$). The (policy-based) recourse function of the approximate problem at node $n \in \mathcal{N}_t$ is $\widehat{Q}^n : \Pi_{i=0}^t \widehat{\mathcal{X}}_i \to \mathbb{R}$ defined as

$$\widehat{Q}^n(\widehat{x}_{\cdot\cdot t}) := \widehat{Q}_t(\widehat{x}_{\cdot\cdot t}(\zeta^{\cdot\cdot n});\zeta^{\cdot\cdot n}), \quad \forall \widehat{x}_{\cdot\cdot t} \in \Pi_{i=0}^t \widehat{\mathcal{X}}_i, \tag{5.17}$$

where the right-hand side is the (vector-based) recourse function (5.9) and $\widehat{x}_{\cdot\cdot t}(\zeta^{\cdot\cdot n})$ denotes the vectors of decisions given by the policy $\widehat{x}_{\cdot\cdot t}$ in the scenario $\zeta^{\cdot\cdot n}$ as defined in (5.15). The policy-based recourse function gathers the two recourse functions (5.8)-(5.9) in a single notation, as the recourse function (5.8) can now be written $\widehat{Q}^n(\widehat{x}_{\cdot\cdot t-1}, \widehat{x}_t^*)$ by definition of $\widehat{x}_t^*$.

It should be noted that in a general setting there is no inclusion relation between $\mathcal{X}_t$ and $\widehat{\mathcal{X}}_t$ for $t > 0$, because $\mathcal{X}_t$ contains functions defined on $\Xi_{\cdot\cdot t}$, whereas $\widehat{\mathcal{X}}_t$ contains functions defined on $\{\zeta^{\cdot\cdot n} : n \in \mathcal{N}_t\}$ which is proper subset of $\Xi_{\cdot\cdot t}$. Despite the lack of inclusion, a feasible policy x of the original problem can be used to make decisions in the approximate problem as well, hence it makes sense to extend the definition (5.17) to all policies $\widehat{x}_{\cdot\cdot t} \in \Pi_{i=0}^t(\widehat{\mathcal{X}}_i \cup \mathcal{X}_i)$. Moreover, although a feasible policy $\widehat{x}$ of the approximate problem cannot be used to make decisions in the original problem in general, a subtlety arises when the stage-t realization $\xi_{\cdot\cdot t}$ coincides with a discretization sequence $\zeta^{\cdot\cdot n}$ for some $n \in \mathcal{N}_t$. In this case, any policy $x_{\cdot\cdot t} \in \Pi_{i=0}^t(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ can be used to make decisions from stage 0 through t in the approximate *and* original problems. In this special case, it makes sense to extend the definition (5.14) to all $x_{\cdot\cdot t} \in \Pi_{i=0}^t(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ provided that $\xi_{\cdot\cdot t} = \zeta^{\cdot\cdot n}$ for some $n \in \mathcal{N}_t$.

We end this section by a remark on two cases of equality between the recourse functions of the original and approximate problems.

Remark 5.2.3. Since the stage-T recourse function Q_T and its estimator $\widehat{Q}^n$ at any node $n \in \mathcal{N}_T$ are both computed directly from the revenue function q, we have that

$$\widehat{Q}^n(x) = Q_T(x;\zeta^{\cdot\cdot n}), \quad \forall n \in \mathcal{N}_T,\, \forall x \in \Pi_{t=0}^T(\mathcal{X}_t \cup \widehat{\mathcal{X}}_t). \tag{5.18}$$

Another case of equality is obtained by noticing that the optimization problem at the right-hand side of (5.3) applied at $\xi = \zeta^{\cdot\cdot n}$, for any $n \in \mathcal{N}_T$, is the same problem as (5.8). Consequently, the corresponding recourse functions on the left-hand side are equal:

$$\widehat{Q}^n(x_{\cdot\cdot T-1}, \widehat{x}_T^*) = Q_T(x_{\cdot\cdot T-1}, x_T^*;\zeta^{\cdot\cdot n}), \quad \forall n \in \mathcal{N}_T,\, \forall x_{\cdot\cdot T-1} \in \Pi_{t=0}^{T-1}(\mathcal{X}_t \cup \widehat{\mathcal{X}}_t). \tag{5.19}$$

5.3 Node-by-node decomposition of the optimal-value error

The main result of this section is Theorem 5.3.5, which provides a node-by-node decomposition of the optimal-value error $Q_0^* - \widehat{Q}_0^*$. In the following the recourse functions are expressed using the policy-based formulation of Section 5.2.3.

We start by introducing the concepts of (low-level) *node errors* and (high-level) *subtree errors*. It appears from the stochastic dynamic programming equations (5.3)-(5.4) that the optimal-value error results from successive errors made by approximating alternatively the right-hand side of (5.3) and (5.4). We call specifically *node optimization error* the error made by approximating (5.3) and *node discretization error* the error made by approximating (5.4), at a particular node in the scenario-tree. Their explicit definitions below are expressed by means of the decision policy formulation of Section 5.2.3:

Definition 5.3.1 (Node optimization error)**.** For each stage $t \in \{1, \ldots, T-1\}$, we define the *optimization error* $\mathrm{E}^n_{\mathrm{opt}}(x_{..t-1})$ at node $n \in \mathcal{N}_t$ and for a decision policy $x_{..t-1} \in \Pi_{i=0}^{t-1}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ as

$$\mathrm{E}^n_{\mathrm{opt}}(x_{..t-1}) = Q_t(x_{..t-1}, x_t^*; \zeta^{..n}) - Q_t(x_{..t-1}, \widehat{x}_t^*; \zeta^{..n}). \tag{5.20}$$

At the root node, the *optimization error* is

$$\mathrm{E}^{n_0}_{\mathrm{opt}} = Q_0(x_0^*; \zeta^{n_0}) - Q_0(\widehat{x}_0^*; \zeta^{n_0}). \tag{5.21}$$

The optimization errors are always non-negative since $\widehat{x}_t^*$ cannot provide higher revenues than x_t^* in the original problem. The optimization error at node $n \in \mathcal{N}_t$ measures the error made by using in the original problem the optimal decision function $\widehat{x}_t^*$ of the approximate problem instead of x_t^*. The optimization error is not defined at stage T because no such error is made at nodes $n \in \mathcal{N}_T$ (cf. Remark 5.2.3).

Definition 5.3.2 (Node discretization error)**.** For each stage $t \in \{0, \ldots, T-1\}$, we define the *discretization error* $\mathrm{E}^n_{\mathrm{disc}}(x_{..t})$ at node $n \in \mathcal{N}_t$ and for a decision policy $x_{..t} \in \Pi_{i=0}^{t}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ as

$$\mathrm{E}^n_{\mathrm{disc}}(x_{..t}) = Q_t(x_{..t}; \zeta^{..n}) - \sum_{m \in C(n)} w^m \, Q_{t+1}(x_{..t}, x_{t+1}^*; \zeta^{..m}). \tag{5.22}$$

The node-n discretization error measures the error made by substituting the conditional expectation by a finite sum over the child nodes of n.

We define now the concept of *subtree errors*. Recall that the subtree rooted at node $n \in \mathcal{N}$ is obtained by setting n as the root node and by considering only the nodes that are the descendants of n and the edges connecting them. (The subtree rooted at n_0 is the whole

scenario tree.) We distinguish between two subtree errors: the *optimal subtree error* measures the error between the optimal recourse function (5.3) and its scenario-tree estimator, whereas the *suboptimal subtree error* measures the error between the expected recourse function (5.4) and its scenario-tree estimator. Their explicit definitions below are expressed by means of the decision policy formulation of Section 5.2.3:

Definition 5.3.3 (Subtree errors)**.** (a) For each stage $t \in \{1, \dots, T\}$, we define the *optimal subtree error* $\Delta Q^n(x_{..t-1})$ at node $n \in \mathcal{N}_t$ and for a decision policy $x_{..t-1} \in \Pi_{i=0}^{t-1}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ as

$$\Delta Q^n(x_{..t-1}) = Q_t(x_{..t-1}, x_t^*; \zeta^{..n}) - \widehat{Q}^n(x_{..t-1}, \widehat{x}_t^*). \tag{5.23}$$

At the root node, the *optimal subtree error* is

$$\Delta Q^{n_0} = Q_0(x_0^*; \zeta^{n_0}) - \widehat{Q}^{n_0}(\widehat{x}_0^*). \tag{5.24}$$

(b) For each stage $t \in \{0, \dots, T\}$, we define the *suboptimal subtree error* $\Delta Q^n_{\text{sub}}(x_{..t})$ at node $n \in \mathcal{N}_t$ and for a decision policy $x_{..t} \in \Pi_{i=0}^{t}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ as

$$\Delta Q^n_{\text{sub}}(x_{..t}) = Q_t(x_{..t}; \zeta^{..n}) - \widehat{Q}^n(x_{..t}). \tag{5.25}$$

The subtree errors ΔQ^n_{sub} and ΔQ^n are identically zero for every $n \in \mathcal{N}_T$ (cf. Remark 5.2.3). In the general setting of the scenario-tree formulation of Section 5.2.2, we do not know whether the subtree errors have positive or negative values. Note that the node-n_0 optimal subtree error (5.24) is the optimal-value error that we want to decompose and bound.

The optimal and suboptimal subtree errors at node n gather implicitly all the node errors (optimization and discretization) made at each node $m \in \mathcal{N}(n)$ of the subtree rooted at n. To find an explicit relation between the subtree errors and the node errors, we need to be able to derive a closed-form representation of a quantity at node n from a recursive representation of this quantity over the nodes in the subtree rooted at n. This is the purpose of the following lemma:

Lemma 5.3.4. *Let a real value γ^n be assigned to every node $n \in \mathcal{N} \setminus \mathcal{N}_T$ of the scenario tree.*

(a) The sequence $\{\alpha^n : n \in \mathcal{N}\}$ satisfies the recurrence relation

$$\alpha^n = \begin{cases} \gamma^n + \sum_{m \in C(n)} w^m \, \alpha^m & \text{if } n \in \mathcal{N} \setminus \mathcal{N}_T; \quad (5.26) \\ 0 & \text{if } n \in \mathcal{N}_T, \quad (5.27) \end{cases}$$

if and only if α^n has a closed-form representation at each node given by

$$\alpha^n = 0 \quad \forall n \in \mathcal{N}_T \quad \textit{and} \quad \alpha^n = \frac{1}{W^n} \sum_{m \in \mathcal{N}(n) \setminus \mathcal{N}_T} W^m \, \gamma^m \quad \forall n \in \mathcal{N} \setminus \mathcal{N}_T, \tag{5.28}$$

where $\mathcal{N}(n)$ is the node set of the subtree rooted at n.

(b) If a sequence $\{\beta^n : n \in \mathcal{N}\}$ satisfies the recurrence relation

$$\beta^n \leq \begin{cases} \gamma^n + \sum_{m \in C(n)} w^m \, \beta^m & \textit{if } n \in \mathcal{N} \setminus \mathcal{N}_T; \quad (5.29) \\ 0 & \textit{if } n \in \mathcal{N}_T, \quad (5.30) \end{cases}$$

then β^n has an upper bound at each node $n \in \mathcal{N} \setminus \mathcal{N}_T$ given by

$$\beta^n \leq \frac{1}{W^n} \sum_{m \in \mathcal{N}(n) \setminus \mathcal{N}_T} W^m \, \gamma^m. \tag{5.31}$$

Parts (a) and (b) will be used in deriving the error decomposition theorem and the error bound theorem, respectively.

Proof. (a) Let $\{u^n\}$ and $\{v^n\}$ denote two sequences satisfying the recurrence relation (5.26)-(5.27) and the closed-form (5.28), respectively. Let us show by backward induction that $u^n = v^n$ holds for every node $n \in \mathcal{N} \setminus \mathcal{N}_T$.

Basis. Take an arbitrary $n \in \mathcal{N}_{T-1}$. We have that $\mathcal{N}(n) \setminus \mathcal{N}_T = \{n\}$, hence it follows from (5.28) that

$$v^n = \frac{1}{W^n} W^n \, \gamma^n = \gamma^n = u^n. \tag{5.32}$$

Inductive step. Suppose that $u^m = v^m$ holds for every $m \in \mathcal{N}_t$ for a given $t \in \{1, \ldots, T-1\}$, and take an arbitrary $n \in \mathcal{N}_{t-1}$. Using the following decomposition of $\mathcal{N}(n) \setminus \mathcal{N}_T$:

$$\mathcal{N}(n) \setminus \mathcal{N}_T = \{n\} \cup \Big(\bigcup_{m \in C(n)} \mathcal{N}(m) \setminus \mathcal{N}_T \Big), \tag{5.33}$$

it follows from (5.28) that

$$v^n = \gamma^n + \frac{1}{W^n} \sum_{m \in C(n)} \sum_{l \in \mathcal{N}(m) \setminus \mathcal{N}_T} W^l \, \gamma^l \tag{5.34}$$

$$= \gamma^n + \frac{1}{W^n} \sum_{m \in C(n)} W^m \left[\frac{1}{W^m} \sum_{l \in \mathcal{N}(m) \setminus \mathcal{N}_T} W^l \, \gamma^l \right] \tag{5.35}$$

$$= \gamma^n + \frac{1}{W^n} \sum_{m \in C(n)} W^m \, v^m \tag{5.36}$$

$$= \gamma^n + \sum_{m \in C(n)} w^m \, u^m \tag{5.37}$$

$$= u^n, \tag{5.38}$$

where the equality (5.37) holds by the induction hypothesis and by the relation $W^m = W^n w^m$ for every $m \in C(n)$ (cf. (5.7)). This proves the inductive step and therefore the final result.

(b) Let $\{\alpha^n\}$ and $\{\beta^n\}$ denote two sequences satisfying the recurrence relation (5.26)-(5.27) and (5.29)-(5.30), respectively. Let us show by induction that $\beta^n \leq \alpha^n$ holds for every node $n \in \mathcal{N} \setminus \mathcal{N}_T$.

Basis. For every $n \in \mathcal{N}_{T-1}$, it follows from (5.29)-(5.30) that $\beta^n \leq \gamma^n = \alpha^n$.

Inductive step. Suppose that $\beta^m \leq \alpha^m$ holds for every node $m \in \mathcal{N}_t$ for a given $t \in \{1, \ldots, T-1\}$, and take an arbitrary $n \in \mathcal{N}_{t-1}$. It follows from (5.29) and the induction hypothesis that

$$\beta^n \leq \gamma^n + \sum_{m \in C(n)} w^m \, \alpha^m = \alpha^n. \tag{5.39}$$

This proves the inductive step. The inequality (5.31) follows immediately using part (a) of this lemma. □

We can now state the main theorem of this section:

Theorem 5.3.5. *The scenario-tree optimal-value error can be decomposed into a weighted sum of node discretization and optimization errors as follows:*

$$\Delta Q^{n_0} = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, [\mathrm{E}^n_{\mathrm{opt}}(\widehat{x}^*_{..t(n)-1}) + \mathrm{E}^n_{\mathrm{disc}}(\widehat{x}^*_{..t(n)})], \tag{5.40}$$

where for $n = n_0$ the term $\mathrm{E}^n_{\mathrm{opt}}(\widehat{x}^*_{..t(n)-1})$ corresponds to $\mathrm{E}^{n_0}_{\mathrm{opt}}$.

Proof. We prove this result by deriving a recurrence relation for the optimal subtree error at node $n \in \mathcal{N} \setminus \mathcal{N}_T$.

Take an arbitrary $t \in \{1, \ldots, T-1\}$ and $n \in \mathcal{N}_t$. Using successively (5.9), (5.22) and (5.20), we can write $\Delta Q^n(\widehat{x}^*_{..t-1})$ as follows:

$$\begin{aligned}
\Delta Q^n(\widehat{x}^*_{..t-1}) &= Q_t(\widehat{x}^*_{..t-1}, x^*_t; \zeta^{..n}) - \widehat{Q}^n(\widehat{x}^*_{..t}) && (5.41)\\
&= Q_t(\widehat{x}^*_{..t-1}, x^*_t; \zeta^{..n}) - \sum_{m \in C(n)} w^m\, \widehat{Q}^m(\widehat{x}^*_{..t}, \widehat{x}^*_{t+1}) && (5.42)\\
&= Q_t(\widehat{x}^*_{..t-1}, x^*_t; \zeta^{..n}) - Q_t(\widehat{x}^*_{..t}; \zeta^{..n}) && (5.43)\\
&\quad + Q_t(\widehat{x}^*_{..t}; \zeta^{..n}) - \sum_{m \in C(n)} w^m\, Q_{t+1}(\widehat{x}^*_{..t}, x^*_{t+1}; \zeta^{..m}) && (5.44)\\
&\quad + \sum_{m \in C(n)} w^m \Big[Q_{t+1}(\widehat{x}^*_{..t}, x^*_{t+1}; \zeta^{..m}) - \widehat{Q}^m(\widehat{x}^*_{..t}, \widehat{x}^*_{t+1}) \Big] && (5.45)\\
&= \mathrm{E}^n_{\mathrm{opt}}(\widehat{x}^*_{..t-1}) + \mathrm{E}^n_{\mathrm{disc}}(\widehat{x}^*_{..t}) + \sum_{m \in C(n)} w^m\, \Delta Q^m(\widehat{x}^*_{..t}).
\end{aligned}$$

Doing the same derivation at the root node yields:

$$\Delta Q^{n_0} = \mathrm{E}^{n_0}_{\mathrm{opt}} + \mathrm{E}^{n_0}_{\mathrm{disc}}(\widehat{x}^*_0) + \sum_{m \in C(n_0)} w^m\, \Delta Q^m(\widehat{x}^*_0). \tag{5.46}$$

By defining

$$\gamma^n = \begin{cases} \mathrm{E}^{n_0}_{\mathrm{opt}} + \mathrm{E}^{n_0}_{\mathrm{disc}}(\widehat{x}^*_0) & \text{if} n = n_0; \\ \mathrm{E}^n_{\mathrm{opt}}(\widehat{x}^*_{..t(n)-1}) + \mathrm{E}^n_{\mathrm{disc}}(\widehat{x}^*_{..t(n)}) & \text{if} n \in \mathcal{N}^* \setminus \mathcal{N}_T, \end{cases} \tag{5.47}$$

we see that the sequence $\{\Delta Q^n : n \in \mathcal{N}\}$ satisfies the recurrence relation (5.26)-(5.27) of Lemma 5.3.4(a) (recall that $\Delta Q^n = 0$ for every $n \in \mathcal{N}_T$; cf. Remark 5.2.3). Thus, the decomposition (5.40) follows directly from (5.28) applied at the root node. □

5.4 Node-by-node upper bound on the optimal-value error

Although the error decomposition of Theorem 5.3.5 is useful to enlighten the contributions of two types of errors in the optimal-value error, it cannot be directly used to guide the generation of scenario trees. The reason is that it features node optimization errors, which are difficult to quantity since they depend on the scenario tree solely via the optimal policy $\widehat{x}^*$. Node discretization errors, conversely, depend directly on the characteristics of a scenario tree, namely, the tree structure $\mathcal{T}$, the discretization points $\mathcal{P}$ and the discretization weights $\mathcal{W}$. Moreover, discretization errors have been largely studied in the field of numerical integration methods, where they are also referred to as integration errors. (In this paper, we use the term "discretization error"when the integrand is the recourse functions (cf. Definition 5.3.2) and "integration error" when the integrand is any integrable function (cf. Definition 5.4.4).)

For all these reasons we want to remove the node optimization errors from the right-hand side of (5.40).

The main result of this section is Theorem 5.4.3, which provides an upper bound on the optimal-value that features only discretization errors. Its derivation does not rely on the decomposition of Theorem 5.3.5 but is based on the following two lemmas.

Lemma 5.4.1. *For each stage $t \in \{1, \dots, T\}$, node $n \in \mathcal{N}_t$ and decision policy $x_{..t-1} \in \Pi_{i=0}^{t-1}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$, the following holds:*

$$|\Delta Q^n(x_{..t-1})| \leq \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\Delta Q^n_{\text{sub}}(x_{..t-1}, u)|, \tag{5.48}$$

and at the root node:

$$|\Delta Q^{n_0}| \leq \max_{u \in \{\widehat{x}_0^*, x_0^*\}} |\Delta Q^{n_0}_{\text{sub}}(u)|. \tag{5.49}$$

Proof. Take an arbitrary $t \in \{1, \dots, T-1\}$, $n \in \mathcal{N}_t$ and $x_{..t-1} \in \Pi_{i=0}^{t}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$.
If $\Delta Q^n(x_{..t-1}) \geq 0$, then it follows from the sub-optimality of x_t^* in the approximate problem that

$$\begin{aligned} |\Delta Q^n(x_{..t-1})| &= Q_t(x_{..t-1}, x_t^*; \zeta^{..n}) - \widehat{Q}^n(x_{..t-1}, \widehat{x}_t^*) && (5.50)\\ &\leq Q_t(x_{..t-1}, x_t^*; \zeta^{..n}) - \widehat{Q}^n(x_{..t-1}, x_t^*) && (5.51)\\ &= \Delta Q^n_{\text{sub}}(x_{..t-1}, x_t^*). && (5.52) \end{aligned}$$

If $\Delta Q^n(x_{..t-1}) < 0$, then it follows from the sub-optimality of $\widehat{x}_t^*$ in the original problem that

$$\begin{aligned} |\Delta \widehat{Q}^n(x_{..t-1})| &= -Q_t(x_{..t-1}, x_t^*; \zeta^{..n}) + \widehat{Q}^n(x_{..t-1}, \widehat{x}_t^*) && (5.53)\\ &\leq -Q_t(x_{..t-1}, \widehat{x}_t^*; \zeta^{..n}) + \widehat{Q}^n(x_{..t-1}, \widehat{x}_t^*) && (5.54)\\ &= -\Delta Q^n_{\text{sub}}(x_{..t-1}, \widehat{x}_t^*). && (5.55) \end{aligned}$$

This proves (5.48) for $t \in \{1, \dots, T-1\}$. The inequality holds trivially for $t = T$ since $\Delta Q^n(x_{..T-1}) = \Delta Q^n_{\text{sub}}(x_{..T-1}, x_T) = 0$ for all $n \in \mathcal{N}_T$ and $(x_{..T-1}, x_T) \in \Pi_{i=0}^{T}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$ (cf. Remark 5.2.3). The inequality (5.49) at the root node can be proved similarly. □

Lemma 5.4.2. *For each stage $t \in \{1, \dots, T-1\}$, node $n \in \mathcal{N}_t$ and decision policy $x_{..t-1} \in \Pi_{i=0}^{t-1}(\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$, the following holds:*

$$|\Delta Q^n(x_{..t-1})| \leq \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\mathrm{E}^n_{\text{disc}}(x_{..t-1}, u)| + \sum_{m \in C(n)} w^m \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\Delta Q^m(x_{..t-1}, u)|, \tag{5.56}$$

and at the root node:

$$|\Delta Q^{n_0}| \leq \max_{u \in \{\widehat{x}_0^*, x_0^*\}} |\mathrm{E}_{\mathrm{disc}}^{n_0}(u)| + \sum_{m \in C(n_0)} w^m \max_{u \in \{\widehat{x}_0^*, x_0^*\}} |\Delta Q^m(u)|. \tag{5.57}$$

Proof. Take an arbitrary $t \in \{1, \ldots, T-1\}$, $n \in \mathcal{N}_t$ and $x_{..t} \in \Pi_{i=0}^t (\mathcal{X}_i \cup \widehat{\mathcal{X}}_i)$. Using successively (5.9), (5.23) and (5.22), we can write $\Delta Q_{\mathrm{sub}}^n(x_{..t})$ as

$$\begin{aligned}
\Delta Q_{\mathrm{sub}}^n(x_{..t}) &= Q_t(x_{..t}; \zeta^{..n}) - \widehat{Q}^n(x_{..t}) && (5.58)\\
&= Q_t(x_{..t}; \zeta^{..n}) - \sum_{m \in C(n)} w^m \, \widehat{Q}^m(x_{..t}, \widehat{x}_{t+1}^*) && (5.59)\\
&= Q_t(x_{..t}; \zeta^{..n}) - \sum_{m \in C(n)} w^m \Big[Q_{t+1}(x_{..t}, x_{t+1}^*; \zeta^{..m}) - \Delta Q^m(x_{..t}) \Big] && (5.60)\\
&= \mathrm{E}_{\mathrm{disc}}^n(x_{..t}) + \sum_{m \in C(n)} w^m \Delta Q^m(x_{..t}). && (5.61)
\end{aligned}$$

From the triangle inequality it follows that

$$|\Delta Q_{\mathrm{sub}}^n(x_{..t})| \leq |\mathrm{E}_{\mathrm{disc}}^n(x_{..t})| + \sum_{m \in C(n)} w^m |\Delta Q^m(x_{..t})|. \tag{5.62}$$

In the particular case for which $x_{..t} = (x_{..t-1}, u)$, with $u \in \{\widehat{x}_t^*, x_t^*\}$, we combine the above inequality with the inequality (5.48) to obtain

$$\begin{aligned}
|\Delta Q^n(x_{..t-1})| &\leq \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\Delta Q_{\mathrm{sub}}^n(x_{..t-1}, u)| && (5.63)\\
&\leq \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\mathrm{E}_{\mathrm{disc}}^n(x_{..t-1}, u)| + \sum_{m \in C(n)} w^m \max_{u \in \{\widehat{x}_t^*, x_t^*\}} |\Delta Q^m(x_{..t-1}, u)|. && (5.64)
\end{aligned}$$

This proves (5.56) for $t \in \{1, \ldots, T-1\}$. The result (5.57) at the root node can be proved similarly. □

Theorem 5.4.3. *The scenario-tree optimal-value error is bounded by a weighted sum of node discretization errors as follows:*

$$|\Delta Q^{n_0}| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \max_{u \in \Pi_{i=0}^{t(n)} \{\widehat{x}_i^*, x_i^*\}} |\mathrm{E}_{\mathrm{disc}}^n(u)|. \tag{5.65}$$

Proof. Take an arbitrary $t \in \{1, \ldots, T-1\}$ and $n \in \mathcal{N}_t$. Using the inequality (5.56) in the

particular case for which $x_{..t-1} \in \Pi_{i=0}^{t-1}\{\widehat{x}_i^*, x_i^*\}$ yields

$$\max_{v\in\Pi_{i=0}^{t-1}\{\widehat{x}_i^*,x_i^*\}} |\Delta Q^n(v)| \leq \max_{v\in\Pi_{i=0}^{t-1}\{\widehat{x}_i^*,x_i^*\}} \Big(\max_{u\in\{\widehat{x}_t^*,x_t^*\}} |\mathrm{E}^n_{\mathrm{disc}}(v,u)|\Big) \tag{5.66}$$

$$+ \sum_{m\in C(n)} w^m \max_{v\in\Pi_{i=0}^{t-1}\{\widehat{x}_i^*,x_i^*\}} \Big(\max_{u\in\{\widehat{x}_t^*,x_t^*\}} |\Delta Q^m(v,u)|\Big)$$

$$= \max_{(v,u)\in\Pi_{i=0}^{t}\{\widehat{x}_i^*,x_i^*\}} |\mathrm{E}^n_{\mathrm{disc}}(v,u)| \tag{5.67}$$

$$+ \sum_{m\in C(n)} w^m \max_{(v,u)\in\Pi_{i=0}^{t}\{\widehat{x}_i^*,x_i^*\}} |\Delta Q^m(v,u)|.$$

At all nodes $n \in \mathcal{N}_T$ the following holds trivially (cf. Remark 5.2.3):

$$\max_{v\in\Pi_{i=0}^{T-1}\{\widehat{x}_i^*,x_i^*\}} |\Delta Q^n(v)| = 0. \tag{5.68}$$

By defining

$$\beta^n = \begin{cases} \max\limits_{v\in\Pi_{i=0}^{t(n)-1}\{\widehat{x}_i^*,x_i^*\}} |\Delta Q^n(v)| & \text{if } n \in \mathcal{N}^*; \quad (5.69) \\ |\Delta Q^{n_0}| & \text{if } n = n_0, \quad (5.70) \end{cases}$$

and

$$\gamma^n = \max_{w\in\Pi_{i=0}^{t(n)}\{\widehat{x}_i^*,x_i^*\}} |\mathrm{E}^n_{\mathrm{disc}}(w)|, \quad \text{for } n \in \mathcal{N} \setminus \mathcal{N}_T, \tag{5.71}$$

we see that the sequence $\{\beta^n : n \in \mathcal{N}\}$ satisfies the recurrence relation (5.29)-(5.30) of Lemma 5.3.4(b). Thus, the bound (5.65) follows directly from (5.31) applied at the root node. □

Bound in terms of worst-case integration errors:

We want now to express the bound (5.65) as a weighted sum of worst-case integration errors in some function sets. To this end, we first introduce the notion of *integration error* $\mathcal{E}^n(f)$ at node $n \in \mathcal{N} \setminus \mathcal{N}_T$, which represents the error made by using a scenario tree to approximate numerically the conditional expectation of $f(\boldsymbol{\xi}_{t(n)+1})$ given $\boldsymbol{\xi}_{..t(n)} = \zeta^{..n}$, where f is an appropriately integrable function. The node integration error generalizes the node discretization error of Definition 5.3.2 to the class of all integrable functions.

In the following, $\mathcal{L}^1(\Xi_{t+1}(\xi_{..t}); \mathbb{R})$ denotes the set of all functions $f : \Xi_{t+1}(\xi_{..t}) \to \mathbb{R}$ integrable with respect to the conditional distribution of $\boldsymbol{\xi}_{t+1}$ given $\boldsymbol{\xi}_{..t} = \xi_{..t}$.

Definition 5.4.4 (Node integration error)**.** For every $t \in \{0, \ldots, T-1\}$, we define the

integration error $\mathcal{E}^n(\cdot)$ at node $n \in \mathcal{N}_t$ as

$$\mathcal{E}^n(f) = \mathbb{E}[f(\boldsymbol{\xi}_{t+1}) \,|\, \boldsymbol{\xi}_{..t} = \zeta^{..n}] - \sum_{m \in C(n)} w^m f(\zeta^m), \quad \forall f \in \mathcal{L}^1(\Xi_{t+1}(\zeta^{..n}); \mathbb{R}). \tag{5.72}$$

The concept of integration error naturally leads to the concept of *worst-case integration error* $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G})$ which measures the largest integration error at node $n \in \mathcal{N} \setminus \mathcal{N}_T$ for a non-empty function set $\mathcal{G} \subset \mathcal{L}^1(\Xi_{t(n)+1}(\zeta^{..n}); \mathbb{R})$:

$$\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}) := \sup_{f \in \mathcal{G}} |\mathcal{E}^n(f)|. \tag{5.73}$$

The following sets of recourse functions defined for every $t \in \{1, \ldots, T-1\}$ and $n \in \mathcal{N}_t$ are of particular interest to express the bound:

$$\mathcal{Q}^n = \Big\{ Q_{t+1}(x_{..t}, x^*_{t+1}; \zeta^{..n}, \cdot) : x_{..t} \in \Pi_{i=0}^t \{\widehat{x}^*_i, x^*_i\} \Big\}. \tag{5.74}$$

Corollary 5.4.5 below expresses the bound of Theorem 5.4.3 by means of worst-case integration errors.

Corollaire 5.4.5. *Let $\mathcal{G}^n$, for every $n \in \mathcal{N} \setminus \mathcal{N}_T$, be any function sets satisfying $\mathcal{Q}^n \subseteq \mathcal{G}^n \subseteq \mathcal{L}^1(\Xi_{t(n)+1}(\zeta^{..n}); \mathbb{R})$. The scenario-tree optimal-value error is bounded by a weighted sum of worst-case integration errors as follows:*

$$|\Delta Q^{n_0}| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, \mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}^n). \tag{5.75}$$

Proof. The worst-case integration error (5.73) and the node discretization error of Definition 5.3.2 are linked as follows:

$$\mathrm{E}^n_{\mathrm{wc}}(\mathcal{Q}^n) = \max_{u \in \Pi_{i=0}^{t(n)} \{\widehat{x}^*_i, x^*_i\}} |\mathrm{E}^n_{\mathrm{disc}}(u)|, \quad \forall n \in \mathcal{N} \setminus \mathcal{N}_T. \tag{5.76}$$

Thus, Theorem 5.4.3 directly yields the right-hand side of (5.75) with $\mathcal{Q}^n$ in place of $\mathcal{G}^n$. Moreover, by definition of the worst-case integration error, we have that $\mathcal{Q}^n \subseteq \mathcal{G}^n$ implies that $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{Q}^n) \leq \mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}^n)$ for every $n \in \mathcal{N} \setminus \mathcal{N}_T$, which completes the proof. □

5.5 Scenario-tree generation

We want now to highlight why the bound (5.75) carries relevant information about the structure of the problem, namely, the recourse functions, the constraints and the stochastic

process, and how it can be used to smartly design scenario trees.

To this end, let us first consider another problem that can be seen as a particular case of the stochastic programming problem. This problem is the one of numerically approximating the expectation of a function by a finite sum. It can indeed be seen as a particular case for which the constraints of Condition 5.2.1 have a *unique* solution, i.e., Y_0 and $Y_t(y_{..t-1};\xi_{..t})$ are singletons. To facilitate an intuitive interpretation of the decomposition (5.40) and of the bound (5.65), we examine the special case of a stochastic programming problem for which the constraints of Condition 5.2.1 have a unique solution, i.e., Z_0 and $Z_t(\xi_{..t})$ are singletons. This case is equivalent to a numerical integration problem, where the expectation of a function is approximated by a finite sum, since the equation (5.3) is no longer relevant, as the supremum is trivial, and only equation (5.4) remains. The latter computes recursively the expectation of $q(z(\boldsymbol{\xi}_{..T});\boldsymbol{\xi}_{..T})$ where $z(\xi_{..T})$ denotes the only element in $Z_T(\xi_{..T})$. In this setting, the decomposition (5.40) and the bound (5.65) are written respectively as

$$\Delta Q^{n_0} = \sum_{n\in\mathcal{N}\setminus\mathcal{N}_T} W^n \, \mathrm{E}^n_{\mathrm{disc}}(x^*_{..t(n)}) \quad \text{and} \quad |\Delta Q^{n_0}| \leq \sum_{n\in\mathcal{N}\setminus\mathcal{N}_T} W^n \, |\mathrm{E}^n_{\mathrm{disc}}(x^*_{..t(n)})|, \tag{5.77}$$

because the optimization errors (5.20)-(5.21) equal zero for all nodes and the set $\Pi_{i=0}^t\{\widehat{x}_i^*, x_i^*\}$ over which the maximum is computed in (5.65) is the singleton $\{x^*_{..t}\}$.

From this we draw the following conclusion for the case of a numerical integration problem solved by scenario trees:

(i) What matters for the control of ΔQ^{n_0} is the control of each node-n integration error $\mathcal{E}^n(f)$ for $f \in \mathcal{Q}^n$ and where $\mathcal{Q}^n$ contains a *unique* function for each $n \in \mathcal{N} \setminus \mathcal{N}_T$.

The statement (i) provides insight into the way scenario trees should be built for an efficient numerical integration. Before explaining this, we remind the reader that fundamental results in numerical integration state that the magnitude of the integration error $\mathcal{E}^n(f)$ is linked to the discretization quality of the points and weights as well as to the variability of the integrand f with respect to the probability distribution. This variability is measured, for instance, by the *standard deviation* in the Central Limit Theorem for Monte Carlo sampling or the *Hardy-Krause variation* in the Koksma-Hlawka inequality for quasi-Monte Carlo methods. In any case, an integrand with large variability with respect to the probability distribution typically leads to large integration error, whereas an integrand with low (or no) variability leads to low (or no) integration error. Thus, when building a scenario tree, one should identify nodes n where the variability of the integrand $f \in \mathcal{Q}^n$ is large and assign to them more child nodes $C(n)$ to reduce the integration error, and conversely, assign fewer child nodes $C(m)$

to the nodes m for which $f \in \mathcal{Q}^m$ has low variability. In the limit where $f \in \mathcal{Q}^m$ has no variability, only one node is necessary in $C(m)$ to integrate *exactly* the function numerically. We illustrate qualitatively this in Figure 5.1(b). The structures of the recourse functions and stochastic process are what determine the variability of each integrand $f \in \mathcal{Q}^n$; indeed, it is the variation of f as a map from $\Xi_{t+1}(\zeta^{\cdot\cdot n})$ to $\mathbb{R}$ as well as the conditional variability of $\boldsymbol{\xi}_{t+1}$ given $\boldsymbol{\xi}_{\cdot\cdot t} = \zeta^{\cdot\cdot n}$ that make the expectation in (5.72) difficult or easy to approximate numerically.

Now that we have seen that both the stochastic process and the recourse functions play a role in the approximation quality of a scenario tree in the context of a numerical integration problem, let us return to the original stochastic programming problem where both equations (5.3) and (5.4) need to be approximated by the scenario tree.

Having established that both the stochastic process and the recourse functions should influence the construction of a scenario tree under the above simplified framework, we consider again the general setting of stochastic programming problems where both equations (5.3) and (5.4) need to be approximated by the scenario tree. The difference with the previous one is the fact that the set $\Pi_{i=0}^{t}\{\widehat{x}_i^*, x_i^*\}$ is no longer reduced to a singleton, hence the point (i) above is now stated as:

(i)' What matters for the control of ΔQ^{n_0} is the control of each node-n integration error $\mathcal{E}^n(f)$ for $f \in \mathcal{Q}^n$ and where $\mathcal{Q}^n$ is a *class* of several functions for each $n \in \mathcal{N} \setminus \mathcal{N}_T$.

That is, in the above simplified setting where the feasible sets are singletons, we had to focus on the integration error for a *unique* integrand at each node, whereas in the general setting the focus is on the integration error for a *class* of several integrands at each node. Each class $\mathcal{Q}^n$ includes $2^{t(n)+1}$ integrands as it is the cardinality of the set $\Pi_{i=0}^{t(n)}\{\widehat{x}_i^*, x_i^*\}$. Since x^* is not known in practice and $\widehat{x}^*$ depends on the scenario tree, these classes cannot be determined exactly and, for this reason, we have to consider larger classes $\mathcal{G}^n$ that include $\mathcal{Q}^n$ as in Corollary 5.4.5. Each class $\mathcal{G}^n$ should include $\mathcal{Q}^n$ as tightly as possible to ensure that $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}^n)$ is not much larger than $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{Q}^n)$ so that the bound remains tight. How far apart the integrands in $\mathcal{Q}^n$ are is what makes the inclusion tight or loose and this depends on the *structure of the constraints*. Indeed, if the set $Z_{t(n)}(\zeta^{\cdot\cdot n})$ is narrow for some node n, then the decisions obtained with $\widehat{x}^*$ in the realization $\zeta^{\cdot\cdot n}$ cannot fall too far from those obtained with x^*. In this case, the integrands in $\mathcal{Q}^n$ are not far apart and the set $\mathcal{G}^n$ can be chosen so that it includes tightly $\mathcal{Q}^n$. Conversely, if the set $Z_{t(m)}(\zeta^{\cdot\cdot m})$ is large for some other node m, then the elements in $\mathcal{Q}^m$ may be far apart and $\mathcal{G}^m$ can only be chosen in a way that includes loosely $\mathcal{Q}^m$. This is qualitatively illustrated in Figure 5.1(c), which should be analyzed in comparison with the previous situation of Figure 5.1(b).

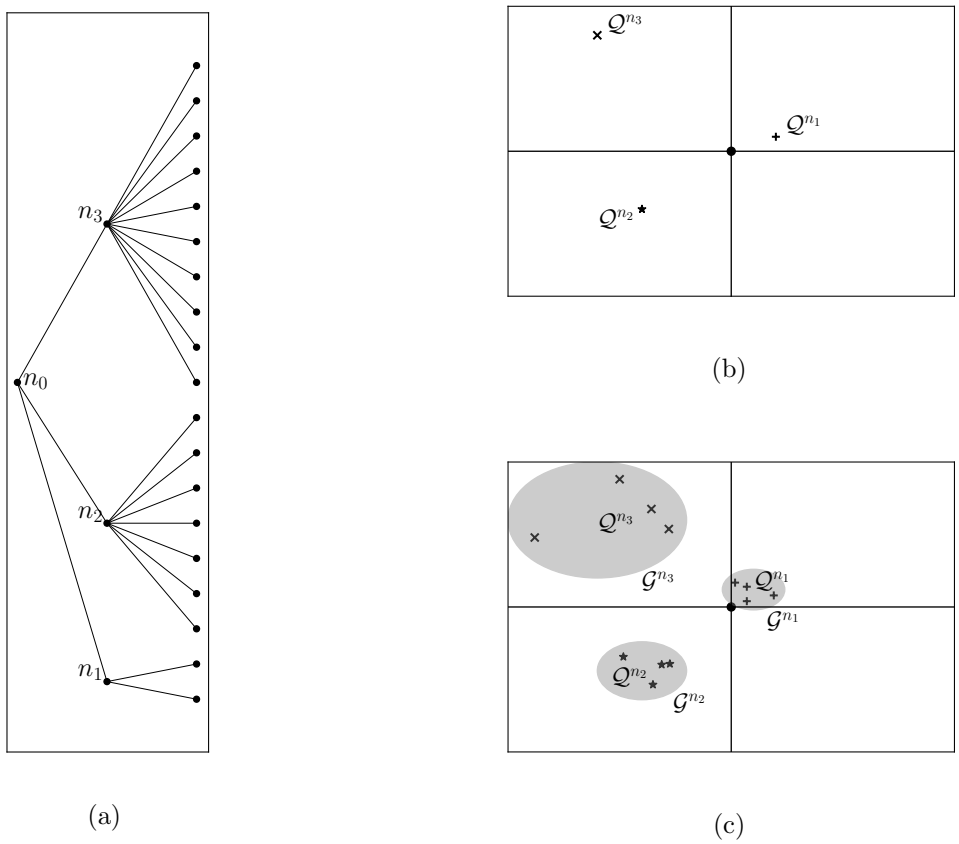


Figure 5.1 Subfigure (a): Representation of a 3-stage scenario tree with three stage-1 nodes $\mathcal{N}_1 = \{n_1, n_2, n_3\}$ and heterogeneous branching at stage 2. Subfigure (b)-(c): Representation of the integrands in each class $\mathcal{Q}^n$ for $n \in \mathcal{N}_1$ by markers ($\star$, $\times$, $+$) in a plane that represents abstractly a function space of integrable functions. The origin $\bullet$ of the plane is the function with no variability and the distance from the origin to some function f is proportional to the variability of f. (b) describes specifically the setting of *numerical integration problems* for which each $\mathcal{Q}^n$ includes a unique integrand and the node with the most variable integrand (n_3) has the most child nodes whereas the node with the least variable integrand (n_1) has the fewest child nodes. (c) describes the setting of *stochastic programming problems* for which each $\mathcal{Q}^n$ includes 4 integrands and each $\mathcal{G}^n$ (the gray area) should include $\mathcal{Q}^n$ as tightly as possible. Integrands in $\mathcal{Q}^{n_3}$ are far apart hence the inclusion $\mathcal{Q}^{n_3} \subseteq \mathcal{G}^{n_3}$ is loose, whereas integrands in $\mathcal{Q}^{n_1}$ are close to each other so the inclusion $\mathcal{Q}^{n_1} \subseteq \mathcal{G}^{n_1}$ is tight. The most variable integrand is inside the class $\mathcal{G}^{n_3}$ hence n_3 has the most child node, whereas the most variable integrand inside $\mathcal{G}^{n_1}$ is the least variable of the three classes so n_1 is the node with the fewest child nodes.

We illustrate in the toy example below how the bound can be calculated and used to generate a scenario tree in an ideal problem where we can compute analytically all quantities of interest. Although such calculation cannot be replicated in a more realistic problem, this example provides insight into the way the bound can be used in practice to generate scenario trees suitable to problems. In light of this example, we then consider again the scenario-tree generation question in full generality.

Example 5.5.1. Consider the following 3-stage stochastic programming problem:

$$\max_{(y_0,y_1,y_2)\in[0,1]^3} \{-ay_0 + \mathbb{E}[-by_1 + \boldsymbol{\xi}_2 y_2] : y_1 \leq y_0,\ y_1 \leq \boldsymbol{\xi}_1,\ y_2 \leq y_1\}, \tag{5.78}$$

where $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ are random variables (possibly correlated) taking values in $[0,\infty)$ and a, b are positive constant. This problem can be interpreted as follows: at stage 0 a volume y_0 of storage space is reserved at cost a; then at stage 1 a volume y_1 of a commodity is purchased at cost b from a wholesaler and stored in the space reserved previously; the wholesale supply available is random and given by $\boldsymbol{\xi}_1$; finally at stage 2 the commodity is sold to customers at the random price $\boldsymbol{\xi}_2$.

Suppose that the scenario tree contains N stage-1 nodes $\mathcal{N}_1 = \{n_1, \ldots, n_N\}$ indexed such that $\zeta^{n_1} \leq \cdots \leq \zeta^{n_N}$. Let us compute the optimal number of child nodes $|C(n_i)|$ for each $i \in \{1, \ldots, N\}$ given by minimizing the bound (5.75).

The stage-2 optimal recourse function Q_2^* and the set $Z_1(\xi_1)$ of all feasible decisions up to stage 1 are given by

$$Q_2^*(y_0, y_1; \xi_1, \xi_2) = -ay_0 - by_1 + \xi_2 y_1, \tag{5.79}$$

$$Z_1(\xi_1) = \{(y_0, y_1) \in [0,1]^2 : 0 \leq y_1 \leq \min(y_0, \xi_1)\}. \tag{5.80}$$

Since the scenario-tree optimal decisions at nodes (n_0, n_i) necessarily belong to $Z_1(\zeta^{n_i})$, we can define the class of functions $\mathcal{G}^{n_i}$ as

$$\mathcal{G}^{n_i} := \{Q_2^*(y_0, y_1; \zeta^{n_i}, \cdot) : (y_0, y_1) \in Z_1(\zeta^{n_i})\}, \tag{5.81}$$

which ensures that $\mathcal{Q}^{n_i} \subseteq \mathcal{G}^{n_i}$ for all $i \in \{1, \ldots, N\}$. Defined this way, these classes also satisfy $\mathcal{G}^{n_1} \subseteq \mathcal{G}^{n_2} \subseteq \cdots \subseteq \mathcal{G}^{n_N}$, which in turn implies that $\mathrm{E}_{\mathrm{wc}}^{n_1}(\mathcal{G}^{n_1}) \leq \mathrm{E}_{\mathrm{wc}}^{n_2}(\mathcal{G}^{n_2}) \leq \cdots \leq \mathrm{E}_{\mathrm{wc}}^{n_N}(\mathcal{G}^{n_N})$, i.e., the structure of the constraints is such that the worst-case integration error at node n_i increases with the index i.

Let us derive the closed-form formula for $\mathrm{E}_{\mathrm{wc}}^{n_i}(\mathcal{G}^{n_i})$. The integration error at node n_i is

$$\mathcal{E}^{n_i}\Big(Q_2^*(y_0, y_1; \zeta^{n_i}, \cdot)\Big) = \mathbb{E}[Q_2^*(y_0, y_1; \zeta^{n_i}, \boldsymbol{\xi}_2)|\boldsymbol{\xi}_1 = \zeta^{n_i}] - \sum_{m \in C(n_i)} w^m \, Q_2^*(y_0, y_1; \zeta^{n_i}, \zeta^m) \quad (5.82)$$

$$= y_1\Big(\mathbb{E}[\boldsymbol{\xi}_2|\boldsymbol{\xi}_1 = \zeta^{n_i}] - \sum_{m \in C(n_i)} w^m \zeta^m\Big), \quad (5.83)$$

where at the second equality we consider that the weights $\{w^m : m \in C(n_i)\}$ are *normalized*, i.e., $\sum_{m \in C(n_i)} w^m = 1$, which allows us to remove the constant terms of Q_2^*. Suppose that we know a discretization method that generates points and weights $\{(\zeta^m, w^m) : m \in C(n_i)\}$ for numerically integrating the functions in $\mathcal{G}^{n_i}$ such that the integration error takes the form:

$$\Big|\mathbb{E}[\boldsymbol{\xi}_2|\boldsymbol{\xi}_1 = \zeta^{n_i}] - \sum_{m \in C(n_i)} w^m \zeta^m\Big| = \frac{\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})}{|C(n_i)|^\alpha}, \quad (5.84)$$

where $\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})$ measures the *conditional variability* of $\boldsymbol{\xi}_2$ given ζ^{n_i} and $\alpha > 0$ is the *rate of convergence* of the method (typically $\alpha = 1/2$ for Monte Carlo sampling and $\alpha \simeq 1$ in some settings of quasi-Monte Carlo methods; see, e.g., Lemieux (2009)). The worst-case integration error $\mathrm{E}_{\mathrm{wc}}^{n_i}(\mathcal{G}^{n_i})$ is given by

$$\mathrm{E}_{\mathrm{wc}}^{n_i}(\mathcal{G}^{n_i}) = \sup_{(y_0, y_1) \in Z_1(\zeta^{n_i})} \mathcal{E}^{n_i}\Big(Q_2^*(y_0, y_1; \zeta^{n_i}, \cdot)\Big) = \frac{\zeta^{n_i}\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})}{|C(n_i)|^\alpha}. \quad (5.85)$$

Thus, the optimal number of child nodes $M_i := |C(n_i)|$ for a total number of K scenarios given by minimizing the bound (5.75) is the optimal solution of

$$\min_{(M_1, \ldots, M_N) \in \mathbb{N}_+^N} \quad \sum_{i=1}^N w^{n_i} \frac{\zeta^{n_i}\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})}{M_i^\alpha} \quad \text{subject to} \quad \sum_{i=1}^N M_i \leq K. \quad (5.86)$$

The optimal M_i is therefore determined by the value of the product $w^{n_i}\zeta^{n_i}\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})$. As this product decreases, node n_i is considered less important in the scenario tree (because the integration error $\mathcal{E}^{n_i}(f)$ is small regardless of $f \in \mathcal{G}^{n_i}$) and hence less child nodes are given to n_i so that other nodes with higher values of the product can have more child nodes. The three situations for which the product is close to zero are intuitive: if $w^{n_i} \simeq 0$ (i.e., the occurrence of ζ^{n_i} is negligible as compared to other ζ^n's), if $\zeta^{n_i} \simeq 0$ (i.e., the wholesale supply available at node n_i is close to zero, hence almost no commodity can be sold at stage 2), or if $\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i}) \simeq 0$ (i.e., the value of the price $\boldsymbol{\xi}_2$ conditional to ζ^{n_i} is almost *not* uncertain). It makes sense that in any of these three situations the discretization at n_i should be done with few scenarios.

The quantity $\mathcal{V}_1(\zeta^{n_i}) := \zeta^{n_i}\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})$ in the above example represents the *worst-case conditional variability* of the integrand set $\mathcal{G}^{n_i}$ defined in (5.81). We see in (5.85) that the worst-case is taken with respect to the set of decisions at stage 1, $Z_1(\zeta^{n_i})$, and the conditional variability is with respect to the stage-2 random parameter given the stage-1 discretization point ζ^{n_i}. This worst-case conditional variability is a product of two terms: the first, ζ^{n_i}, is a feature of the problem's constraints; the second, $\mathbb{V}(\boldsymbol{\xi}_2|\zeta^{n_i})$, is a feature of the stochastic process. A scenario-tree generation method that would focus exclusively on approximating the stochastic process would probably take into account the latter term, but certainly not the former, as it is hidden in the specific structure of this problem. Of course, in this example we were able to derive the exact variability of Q_2^* because the problem was simple enough to admit analytically derivable recourse functions. In a more realistic problem, the quantity $\mathcal{V}_1(\xi_1)$ will have to be inferred for any value of ξ_1 from the decision-maker's expertise or from numerical investigation, and then the integer program (5.86) will be solved to find the suitable number of child nodes for each stage-1 nodes n_i. The decision-maker can indeed bring valuable knowledge about the problem: in the above example one could directly infer from the problem description that the recourse function has no variability if the available amount of wholesale supply is zero, i.e., $\mathcal{V}_1(0) = 0$, and that $\mathcal{V}_1(\xi_1)$ increases with ξ_1. In a more realistic problem of pricing American options, a put option that is deep out of the money at stage t has almost no value (hence almost no variability), therefore, one could directly infer that $\mathcal{V}_t(\xi_t) \simeq 0$ if $\xi_t \gg K$ where K is the strike price and ξ_t is the price of the underlying asset at stage t; more generally, the value of $\mathcal{V}_t(\cdot)$ could be approximated by looking at the conditional variability of the corresponding European put option for which a closed-form solution exists. In both cases, however, it is difficult to infer the exact value of $\mathcal{V}_t(\xi_t)$ without actually solving the problem or at least proceeding to numerical investigation, but the question of whether knowing the exact value is necessary to achieve significant improvement with respect to methods that do not explicitly take into account the structure of the problem is open and requires further investigation. In that respect, what we learn from the program (5.86) is that the exact values of $\mathcal{V}_1(\cdot)$ are irrelevant in *absolute terms*, because the optimal solution of the program is unchanged if $\mathcal{V}_1(\cdot)$ is multiplied by any positive constant. Thus, what only matters are the values of $\mathcal{V}_1(\cdot)$ *relative to each other*, i.e., the variation of $\mathcal{V}_1(\cdot)$ itself. For instance, if we know that $\mathcal{V}_1(\xi_1)$ increases with ξ_1, then we are solely interested in finding an increasing function $f(\cdot)$ such that $\mathcal{V}_1(\xi_1) \propto f(\xi_1)$ where the symbol $\propto$ refers to a proportionality with positive constant. As a result, when one infers the worst-case conditional variability functions of the problem at each stage, one is not interested in finding the actual value of $\mathcal{V}_t(\cdot)$ but rather how it varies over the support of the random parameters; the goal being to differentiate the parts of the support with high or low worst-case conditional

variability.

Assuming that we are given the (worst-case conditional) variability functions:

$$\mathcal{V}_0, \mathcal{V}_1(\xi_1), \mathcal{V}_2(\xi_1, \xi_2), \ldots, \mathcal{V}_{T-1}(\xi_{..T-1}), \tag{5.87}$$

corresponding to the function sets $\mathcal{G}^n$, $n \in \mathcal{N} \setminus \mathcal{N}_T$, of Corollary 5.4.5 for an arbitrary multistage problem that falls into the framework of this paper, let us present a practical way to generate scenario trees that keep the bound (5.75) close to its minimum value. Suppose that these functions are such that the worst-case error $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}^n)$ satisfies

$$\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}^n) \simeq \frac{\mathcal{V}_{t(n)}(\zeta^{..n})}{|C(n)|^{\alpha_{t(n)}}}, \quad \forall n \in \mathcal{N} \setminus \mathcal{N}_T, \tag{5.88}$$

where $\alpha_{t(n)} > 0$ is the rate of convergence of the numerical integration method used to generate the discretization points and weights at stage $t(n)$. Relation (5.88) holds approximately for finite $|C(n)|$ and typically holds at equality asymptotically as $|C(n)| \to \infty$ ($\simeq$ becomes $\sim$). Depending on the numerical integration method considered, the variability function $\mathcal{V}_{t(n)}(\cdot)$ and the rate of convergence $\alpha_{t(n)}$ can take different forms:

- In a quasi-Monte Carlo framework: for an arbitrary *reproducing kernel Hilbert space* $\mathcal{H}$ of integrable functions $f : [0,1]^d \to \mathbb{R}$ equipped with the norm $\|\cdot\|_{\mathcal{H}}$ and for an arbitrary subset of integrands $\mathcal{G} := \{g(y;\cdot) : y \in \mathbb{R}^s\} \subseteq \mathcal{H}$, we have a relation of the form:

  $$\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}) \leq \|k^n\|_{\mathcal{H}} \sup_{y \in \mathbb{R}^s} \|g(y;\cdot)\|_{\mathcal{H}}, \tag{5.89}$$

  where $k^n \in \mathcal{H}$ is the *representer* of the integration error $\mathcal{E}^n(\cdot)$ in the Hilbert space given by Riesz representation theorem; see, e.g., Dick et Pillichshammer (2010, Chapter 2). This inequality becomes tighter as the size of $\mathcal{G}$ increases, and in the case where $\mathcal{G}$ contains all integrands $g(y;\cdot) \in \mathcal{H}$ such that $\|g(y;\cdot)\|_{\mathcal{H}} \leq C$, for $C > 0$, it holds with equality: $\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}) = C\|k^n\|_{\mathcal{H}}$. The first term in the right-hand side of (5.89) depends only on the discretization points and weights at $C(n)$ and will typically achieve a rate of convergence $O(|C(n)|^{-1+\epsilon})$ for any $\epsilon > 0$. A crucial point here is that ϵ hides a dependence on d; we will return to this point below. The second term depends only on the integrands in $\mathcal{G}$ and plays the role of the worst-case conditional variability function for $\mathcal{G}$. The above result also holds in the case where the integrands are not defined over the hyper-cube $[0,1]^d$ but over more general domain $D \subseteq \mathbb{R}^d$, we refer to Hickernell *et al.* (2004a), Novak et Woźniakowski (2010) and Novak *et al.* (2017).

- In an optimal quantization framework: denoting by $\mathrm{Lip}(\mathbb{R}^d)$ the space of all real-valued

1-Lipschitz functions on $\mathbb{R}^d$ and taking the dual representation of the *Wasserstein distance* of order 1 (also called the *Kantorovich-Rubinstein distance*), we can write the worst-case error of a subset of integrands $\mathcal{G}' := \{g(y;\cdot) : y \in \mathbb{R}^s\} \subseteq \mathrm{Lip}(\mathbb{R}^d)$ as follows:

$$\mathrm{E}^n_{\mathrm{wc}}(\mathcal{G}') \leq W_1(\mathbb{P}_{\boldsymbol{\xi}_{t(n)+1}|\zeta^{\cdot\cdot n}}, \sum_{m\in C(n)} w^m \delta_{\zeta^m}) \sup_{y\in\mathbb{R}^s} L_1(g(y;\cdot)), \tag{5.90}$$

where $W_1(\cdot,\cdot)$ is the Wasserstein distance of order 1, $L_1(g(y;\cdot))$ is the 1-Lipschitz constant of $g(y;\cdot)$, the measure $\mathbb{P}_{\boldsymbol{\xi}_{t(n)+1}|\zeta^{\cdot\cdot n}}$ is the conditional distribution of $\boldsymbol{\xi}_{t(n)+1}$ given $\boldsymbol{\xi}_{..t(n)} = \zeta^{\cdot\cdot n}$, and the measure $\sum_{m\in C(n)} w^m \delta_{\zeta^m}$, with δ_{ζ^m} the Dirac measure at ζ^m, is the scenario-tree approximate distribution at nodes $C(n)$; see, e.g., Villani (2008, Chapter 6). Similarly to (5.89), (5.90) holds with equality if $\mathcal{G}'$ contains all integrands $g(y;\cdot) \in \mathrm{Lip}(\mathbb{R}^d)$ such that $L_1(g(y;\cdot)) \leq C$, for $C > 0$. The first term in the right-hand side of (5.90) depends only on the discretization points and weights at $C(n)$ and achieves a rate of convergence $O(|C(n)|^{-1/d})$; see, e.g., Pagès *et al.* (2004) and Pflug et Pichler (2016, Chapter 4). The second term depends only on the integrands in $\mathcal{G}'$ and can serve as the worst-case conditional variability function for $\mathcal{G}'$.

The fact that the rates of convergence in the quasi-Monte Carlo and the optimal quantization frameworks depend on the dimension d is a typical feature of deterministic numerical integration methods. As opposed, the Monte Carlo rate of convergence in $O(|C(n)|^{-1/2})$ is independent of the dimension, although it is typically much slower than deterministic methods in low dimensions. Research has been very active to avoid this so-called "curse of dimensionality" (Bellman (1961)). Since the introduction of *weighted* reproducing kernel Hilbert spaces by Sloan et Woźniakowski (1998) to explain the efficiency of quasi-Monte Carlo methods for some high-dimensional integrals, a lot of articles have been published to extend the advantageous rate of convergence $O(|C(n)|^{-1})$, or higher, to integrands with many variables (see, e.g., Kuo *et al.* (2011) and references therein) as well as to integrands that are non-smooth (see, e.g., Griebel *et al.* (2010, 2013)).

As far as the generation of scenario trees is concerned, we are interested in choosing the numerical integration method with the highest rate of convergence $\alpha_{t(n)}$ in (5.88) for the recourse functions in $\mathcal{G}^n$. Since recourse functions may depend on many random parameters and are generally non-smooth, this choice is not straightforward. Recent work by Heitsch *et al.* (2016) has shown that quasi-Monte Carlo methods for two-stage linear problems with many random parameters ($d = 100$ in their problem) achieve a rate of convergence close to $\alpha_0 = 1$ when the methods are combined with some procedures for reducing the effective dimensions of the integrands in $\mathcal{G}^{n_0}$. The question of whether similar results would hold in a

multistage setting for the integrands in $\mathcal{G}^n$ for every $n \in \mathcal{N} \setminus \mathcal{N}_T$ remains open.

To develop a versatile method of generating scenario trees that can incorporate any current or future improvement of numerical integration methods, we want to minimize the bound (5.75) when the worst-case errors hold in the form (5.88) for an *arbitrary* $\alpha_{t(n)} > 0$. We do this in the case where the number of nodes of the scenario trees is fixed at each stage. That is to say, we are given a *width vector* $(N_1, \ldots, N_T) \in \mathbb{N}_+^T$ and we want to minimize the bound under the constraints that $|\mathcal{N}_1| = N_1$, $|\mathcal{N}_2| = N_2$, etc.

Using (5.88), the bound (5.75) can be separated by stage as follows:

$$\sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, \mathrm{E}_{\mathrm{wc}}^n(\mathcal{G}^n) \simeq \frac{\mathcal{V}_0}{N_1^{\alpha_0}} + \sum_{t=1}^{T-1} \sum_{n \in \mathcal{N}_t} W^n \, \frac{\mathcal{V}_t(\zeta^{\cdot\cdot n})}{|C(n)|^{\alpha_t}}. \tag{5.91}$$

This form shows that a computationally effective way to keep the bound close to its minimum value consists in minimizing the inner sum over $n \in \mathcal{N}_t$ recursively for $t = 1, \ldots, T$. Taking into account the constraints introduced by the width vector $(N_1, \ldots, N_T)$, this leads to the following *forward bound-minimizing* heuristic algorithm that computes recursively:

$$\min_{(M_1, \ldots, M_{N_t}) \in \mathbb{N}_+^{N_t}} \quad \sum_{i=1}^{N_t} W^{n_i} \frac{\mathcal{V}_t(\zeta^{\cdot\cdot n_i})}{M_i^{\alpha_t}} \quad \text{subject to} \quad \sum_{i=1}^{N_t} M_i = N_{t+1}, \tag{5.92}$$

for $t = 1, \ldots, T-1$. At each iteration $t = 1, \ldots, T-1$, the nodes in $\mathcal{N}_t$ are indexed such that $\mathcal{N}_t = \{n_1, \ldots, n_{N_t}\}$ and the optimal integer solution M_i^*, $i = 1, \ldots, N_t$, represents the optimal number of child nodes to assign at n_i. The complete algorithm is described in Figure 5.2.

The program (5.92) is restricted to positive integer solutions. This makes it harder to solve than its corresponding continuous relaxation where the constraint $(M_1, \ldots, M_{N_t}) \in \mathbb{N}_+^{N_t}$ is substituted with $(M_1, \ldots, M_{N_t}) \in [0, \infty)^{N_t}$, since a closed-form optimal solution exists for the latter by the theory of non-linear programming. In practice, we observe that solving the continuous relaxation and rounding off the optimal solution to the nearest positive integer provides a solution close to the optimal integer one. Thus, unless one has at hand a computationally effective way to solve (5.92) to integer optimality, in practice the optimal ratio M_i^*/N_{t+1} can be computed using the following closed-form solution stemming from the continuous relaxation:

$$\frac{M_i^*}{N_{t+1}} = \frac{\left(W^{n_i} \mathcal{V}_t(\zeta^{\cdot\cdot n_i})\right)^{1/(\alpha_t+1)}}{\sum_{i=1}^{N_t} \left(W^{n_i} \mathcal{V}_t(\zeta^{\cdot\cdot n_i})\right)^{1/(\alpha_t+1)}}, \qquad i = 1, \ldots, N_t. \tag{5.93}$$

- **Inputs:**
  - worst-case conditional variability functions (5.87);
  - numerical integration method with rates of convergence $\alpha_0, \ldots, \alpha_{T-1} > 0$;
  - width vector $(N_1, \ldots, N_T) \in \mathbb{N}_+^T$.
- **Iteration $t = 0$:**
  - assign N_1 child nodes to the root node n_0;
  - generate the discretization point and weight (ζ^n, W^n) of $\boldsymbol{\xi}_1$ at each node $n \in \mathcal{N}_1$;
  - index the set $\mathcal{N}_1$ such that $\mathcal{N}_1 = \{n_1, \ldots, n_{N_1}\}$.
- **Iterations $t = 1, \ldots, T-1$:**
  - solve the program (5.92) and retrieve the optimal solution $(M_1^*, \ldots, M_{N_t}^*)$ (you may use (5.93));
  - for each $i = 1, \ldots, N_t$:
    - assign M_i^* child nodes to n_i;
    - generate the discretization point and weight (ζ^m, W^m) of $\boldsymbol{\xi}_{t+1}$ given $\zeta^{\cdot n_i}$ at each node $m \in C(n_i)$;
  - index the set $\mathcal{N}_{t+1}$ such that $\mathcal{N}_{t+1} = \{n_1, \ldots, n_{N_{t+1}}\}$.

Figure 5.2 Forward bound-minimizing heuristic algorithm.

For two nodes n_i and n_j in $\mathcal{N}_t$, the ratio of their numbers of child nodes is

$$\frac{M_i^*}{M_j^*} = \left(\frac{W^{n_i}\mathcal{V}_t(\zeta^{..n_i})}{W^{n_j}\mathcal{V}_t(\zeta^{..n_j})}\right)^{1/(\alpha_t+1)}. \tag{5.94}$$

This ratio indicates two things: first, more child nodes will be given to n_i at the expense of n_j if the worst-case variability and the product weight are greater at n_i than at n_j; second, the ratio converges to one as α_t increases regardless of the value of the inner ratio in the right-hand side. It follows from the first point that the benefit of the forward bound-minimizing algorithm over a generic algorithm that generates symmetrical scenario trees is greatest for problems where $\mathcal{V}_t(\cdot)$ varies widely, and from the second point that this benefit is greater as the number of random parameters per stage increases (as this results in decreasing α_t).

We now illustrate and analyze the scenario trees obtained by the forward bound-minimizing algorithm in the context of the discretization of a Brownian motion. Although we consider a specific setting of problems to illustrate the applicability of our approach, most conclusions drawn in this example would hold true for other choices of stochastic processes and variability functions, as long as $\mathcal{V}_t(\cdot)$ exhibits variations over the set of realizations.

Example 5.5.2. We consider a class of multistage problems for which the stochastic process $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ models a discrete-time standard Brownian motion starting from $\xi_0 = 0$ and the variability functions (5.87) are given by:

$$\mathcal{V}_0 = 1 \quad \text{and} \quad \mathcal{V}_t(\xi_{..t}) = \sqrt{\frac{\pi}{2t}}|\xi_t|, \qquad t = 1, \ldots, T-1. \tag{5.95}$$

It is important to understand that we now consider a *class* of problems rather than a specific problem. This class is defined as all multistage problems with a structure (constraints and revenue function) characterized by the collection (5.95) of variability functions. Abstractly, these functions are therefore the representative of the problem's structure in the set of all multistage problems. Different problems having similar variability functions will be seen as identical as far as scenario-tree generation is concerned.

Our goal in this example is to generate scenario trees suitable to the class characterized by (5.95) and compare them with generic symmetrical scenario trees. Two important features are collected in the form (5.95). The first one is the fact that $\mathcal{V}_t(\xi_{..t})$ varies proportionally to $|\xi_t|$, which means that we expect the conditional variability of the recourse functions to increase linearly as the trajectory of the Brownian motion moves away (positively or negatively) from its marginal mean. The second one concerns the choice of $\mathcal{V}_0$ and the constant $C_t := \sqrt{\pi/2t}$. Although the value of C_t does not change the optimal distribution

of child nodes *at a given stage*, as the optimal solution of (5.92) is invariant under the transformation $\mathcal{V}_t(\xi_{..t}) \to C\mathcal{V}_t(\xi_{..t})$ for any $C > 0$, it does change the optimal distribution of child nodes *across the different stages*. Since this will hold true for any setting of problems, we make a general remark to explain it.

Remark 5.5.1. To see the role of the constants of proportionality in the definition of the variability functions, consider a problem for which $\mathcal{V}_0 = C_0$ and $\mathcal{V}_t(\xi_{..t}) = C_t f_t(\xi_{..t})$, $t = 1, \ldots, T-1$, for arbitrary $C_t > 0$, $f_t(\cdot) \geq 0$ and stochastic process $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ (we assume that $f_t(\boldsymbol{\xi}_{..t})$ has finite mean), and consider a symmetrical scenario tree with a branching coefficient $b_t \in \mathbb{N}_+$ at each stage $t = 0, \ldots, T-1$, i.e., $b_{t(n)} = |C(n)|$ for all $n \in \mathcal{N} \setminus \mathcal{N}_T$. For such problem and scenario tree, the right-hand side of (5.91) takes the form

$$\frac{C_0}{b_0^{\alpha_0}} + \sum_{t=1}^{T-1} \frac{C_t}{b_t^{\alpha_t}} \sum_{n \in \mathcal{N}_t} W^n f_t(\zeta^{..n}). \tag{5.96}$$

The inner sum over $n \in \mathcal{N}_t$ is the scenario-tree estimator of $\mathbb{E}[f_t(\boldsymbol{\xi}_{..t})]$, hence if the numerical integration method used to build the scenario tree provides consistent estimators, then as the number of nodes per stage increases we have that

$$\frac{C_0}{b_0^{\alpha_0}} + \sum_{t=1}^{T-1} \frac{C_t}{b_t^{\alpha_t}} \sum_{n \in \mathcal{N}_t} W^n f(\zeta^{..n}) \simeq \frac{C_0}{b_0^{\alpha_0}} + \sum_{t=1}^{T-1} \frac{C_t\, \mathbb{E}[f_t(\boldsymbol{\xi}_{..t})]}{b_t^{\alpha_t}}. \tag{5.97}$$

This shows that the optimal branching coefficients $b_0, \ldots, b_{T-1}$ of a symmetrical scenario tree depend on the relative values of C_0 and $C_t\, \mathbb{E}[f_t(\boldsymbol{\xi}_{..t})]$. In particular, if the rates of convergence are equal, then taking $C_0 = 1$ and $C_t = 1/\mathbb{E}[f_t(\boldsymbol{\xi}_{..t})]$ ensures that the optimal branching coefficients of a symmetrical scenario tree are such that $b_0 = \cdots = b_{T-1}$.

Coming back to the standard Brownian motion, we have that $\boldsymbol{\xi}_t$ follows a normal distribution with mean zero and standard deviation $\sqrt{t}$, hence $\mathbb{E}[|\boldsymbol{\xi}_t|] = \sqrt{2t/\pi}$. Thus, by choosing $C_0 = 1$ and $C_t = 1/\mathbb{E}[f_t(\boldsymbol{\xi}_{..t})] = \sqrt{\pi/2t}$ we do not change the output scenario trees generated by the forward bound-minimizing algorithm but we guarantee that the best *symmetrical* scenario trees are those with constant branching, b, across stages. This choice is therefore purely motivated by simplicity purposes, it facilitates the comparison between the bound-minimizing and the symmetrical scenario trees, as we know now that we only have to consider width vectors of the form $(b, b^2, \ldots, b^T)$ (all other width vectors of the form $(b_0, b_0 b_1, \ldots, \Pi_{t=0}^{T-1} b_t)$ are necessarily sub-optimal for symmetrical trees).

The bound-minimizing and symmetrical scenario trees are built using two numerical integration methods: a quasi-Monte Carlo (QMC) method that uses rank-1 lattice rules and

inversion of the cumulative distribution function, and an optimal quantization (OQ) method that minimizes the Wasserstein distance of order 1 (see Pflug et Pichler (2015, Algorithm 2)). Since the discretization is that of a one-dimensional normal distribution, we set the rate of convergence to $\alpha = 1$ for both methods and for each stage.

We illustrate in Figure 5.3 the first three iterations of the forward bound-minimizing algorithm for a width vector $(3, 3^2, \dots)$. As expected, the nodes with points ζ^n farther from zero are getting more child nodes. Nodes with $\zeta^n = 0$ are getting the minimum number of nodes, 1, as the variability functions (5.95) state that there is no variability in this case. The tree structures generated by QMC and OQ are identical up to the iteration $t = 2$, but they differ (slightly) from the iteration $t = 3$ onward, as we can see in Figure 5.4. A common feature of these tree structures is that they sample more densely the parts of the distribution support where the conditional variability is higher. To see what the bound-minimizing scenario trees mean in terms of the Brownian motion paths that they represent, we plot in Figure 5.5 these paths and compare them with those of a symmetrical scenario tree. To facilitate the visual comparison, the four subplots are displayed with the same y-axis upper and lower limits $[-5.5, 5.5]$. An important observable feature of the bound-minimizing scenario trees is that they cover a wider part of the sample space. Indeed, for the OQ method, all paths of the bound-minimizing scenario trees lie at stage 4 tightly within $[-5, 5]$, whereas those of the symmetrical trees lie within $[-4.1, 4.1]$. Since the marginal distribution of $\boldsymbol{\xi}_4$ has standard deviation two, this means that the bound-minimizing paths cover 98.8% of the sample space, whereas the symmetrical paths cover only 96%. Thus, the bound-minimizing scenario trees are naturally inclined to sample the so-called *extreme events* located here at the tails of the distributions, which quite logically are those for which the worst-case variability functions take large values.

We now turn to a quantitative analysis of the scenario trees. We compute the upper bound (5.75) for different time horizons $T \in \{6, 10, 14, 20\}$ and different width vectors $(b, b^2, \dots, b^T)$ for $b \in \{2, \dots, 10\}$. For computational reasons, all scenario trees are limited to a maximum of 2×10^6 scenarios, hence in practice b varies from 2 to $\min(10, \lfloor (2 \times 10^6)^{1/T} \rfloor)$. We build the scenario trees using the OQ, QMC and Monte Carlo (MC) methods; for MC the results are averaged over ten replications and the rate of convergence is set to $\alpha = 1/2$. The results are displayed in Table A.1 (Appendix A). For each instance of (T, b), we display the bound values of the bound-minimizing scenario trees generated by QMC and OQ (columns "abs." and the percentage improvement relative to the bound values of symmetrical scenario trees generated by QMC, OQ and MC (columns "rel."). The percentage improvement is between 0% and 100%: 0% means the bound values of the bound-minimizing and symmetrical scenario trees are equal and 100% means the improvement is such that the bound value of the bound-

minimizing scenario trees reaches zero. We also display the computational time (in seconds) required to generate the scenario tree and compute the bound value (columns "sec."). The time for MC is averaged over the ten replications. It is greater than that of QMC and OQ because their deterministic data points and weights are computed beforehand and stored. The conclusions that we draw from the results in Table A.1 are the following:

(1) The bound-minimizing scenario trees generated by QMC and OQ have roughly the same bound values for each fixed instance (T, b); this is also true for the corresponding symmetrical scenario trees. Thus, for each fixed instance (T, b) the relative improvement is about the same between QMC and OQ.

(2) The relative improvement over symmetrical OQ (or QMC) scenario trees ranges from 10% to 45%. It increases as T increases and it seems to slightly decrease as b increases. Thus, at equal rates of convergence, the benefit of bound-minimizing scenario trees over symmetrical scenario trees is greatest for problems with many stages and for small scenario sizes.

(3) The relative improvement over symmetrical MC scenario trees ranges from 55% to 75%. It increases as T increases (similarly to (2)) and as b increases (unlike (2)) because MC has a smaller rate of convergence than OQ and QMC in dimension one. Thus, for a problem with many random parameters per stage, two situations can arise depending on whether there exists an efficient numerical integration method for the recourse functions: if such a method exists, its rate of convergence beats MC and hence we can expect the same conclusion as above to hold; otherwise, we can expect that the relative improvement still increases as T increases and b is constant, however, it will rapidly decrease as b increases and T is constant.

(4) The bound values of symmetrical scenario trees decrease monotonically from $b = 2$ onward, whereas the bound values of the bound-minimizing scenario trees have an upward jump when b moves from 3 to 4, after which the decrease starts to be monotonic. This phenomenon is observed for both QMC and OQ and for other values of T not displayed in the table. This can be explained by the fact that we have considered width vectors of the form $(b, b^2, \ldots, b^T)$, which are optimal for the symmetrical trees (cf. (5.97) and the discussion that follows) but not for the bound-minimizing trees. In other words, when we enforce the number of nodes per stage in the forward bound-minimizing algorithm, we require the scenario tree to sample where it may not be optimal to do so. To avoid such behavior, it may be necessary to consider different

forms of width vectors, or to develop a bound-minimizing algorithm that does not enforce the number of nodes per stage.

(5) The bound-minimizing scenario trees have only slightly larger computational times. Thus, the additional computational cost of running the forward bound-minimizing algorithm is essentially insignificant as compared with the improvement of the output scenario trees.

5.6 Conclusion

Scenario-tree generation have proved to be a useful approach to solve multistage stochastic programming problems with few stages. However, problems with many stages currently remain out of reach of scenario-tree generation methods. We believe this limitation occurs because current methods are essentially distribution-based, which means that they focus on approximating the stochastic process with little or no regard to the specific structure of the optimization problem. This paper aims at showing that scenario-tree generation could be improved by also taking into account the features of each problem, which appear in its objective function and constraints, and then by tailoring the method to some classes of problems sharing similar features. The two theorems on the optimal-value error derived in this paper pave the way to designing such methods.

The first theorem is an exact decomposition of the optimal-value error as a weighted sum of discretization and optimization errors made at each node of the scenario tree. It shows that an inappropriate discretization at a node where the recourse function is ill-behaved (e.g., with large variability) can contribute to most of the total optimal-value error. The second theorem is an upper bound on the optimal-value error that features only node discretization errors. It shows that the optimal-value error can be controlled by designing scenario trees suitable for numerically integrating classes of functions determined by the structure of the problem.

Based on these results, we develop the premise of a new scenario-tree generation approach and demonstrate its potential in the case of the discretization of a discrete-time Brownian motion with up to 20 stages. Qualitatively, we observe that the scenario trees generated have heterogeneous branching, which is denser at nodes where the conditional variability of the recourse functions is higher. In the case of the Brownian motion example, the recourse functions are expected to have more variability on the tails of the distribution, which results in scenario trees that expand their paths faster than symmetrical scenario trees in order to cover more suitably the tails. A quantitative comparison with symmetrical scenario trees

shows that the benefit of taking into account the problem's structure lies between 10% and 45% in terms of bound reduction and is greatest for problems with many stages and for scenario trees with small average branching. Computational times are only slightly larger.

Overall, the fact that these results hold for two different discretization methods (a quasi-Monte Carlo and an optimal quantization method) show that our approach is versatile and has a great range of potential applications. To go further into its development, several issues still need to be addressed. For instance, a systematic way to infer the variability functions should be developed to make the approach applicable to realistic problems. On the numerical integration side, efficient methods should be developed to integrate efficiently the high-dimensional non-smooth recourse functions of multistage problems. On the algorithmic side, more algorithms are already developed in Keutchayan *et al.* (2018b) to minimize the error bound in a general setting where only the number of scenarios is fixed and no width vector is required.

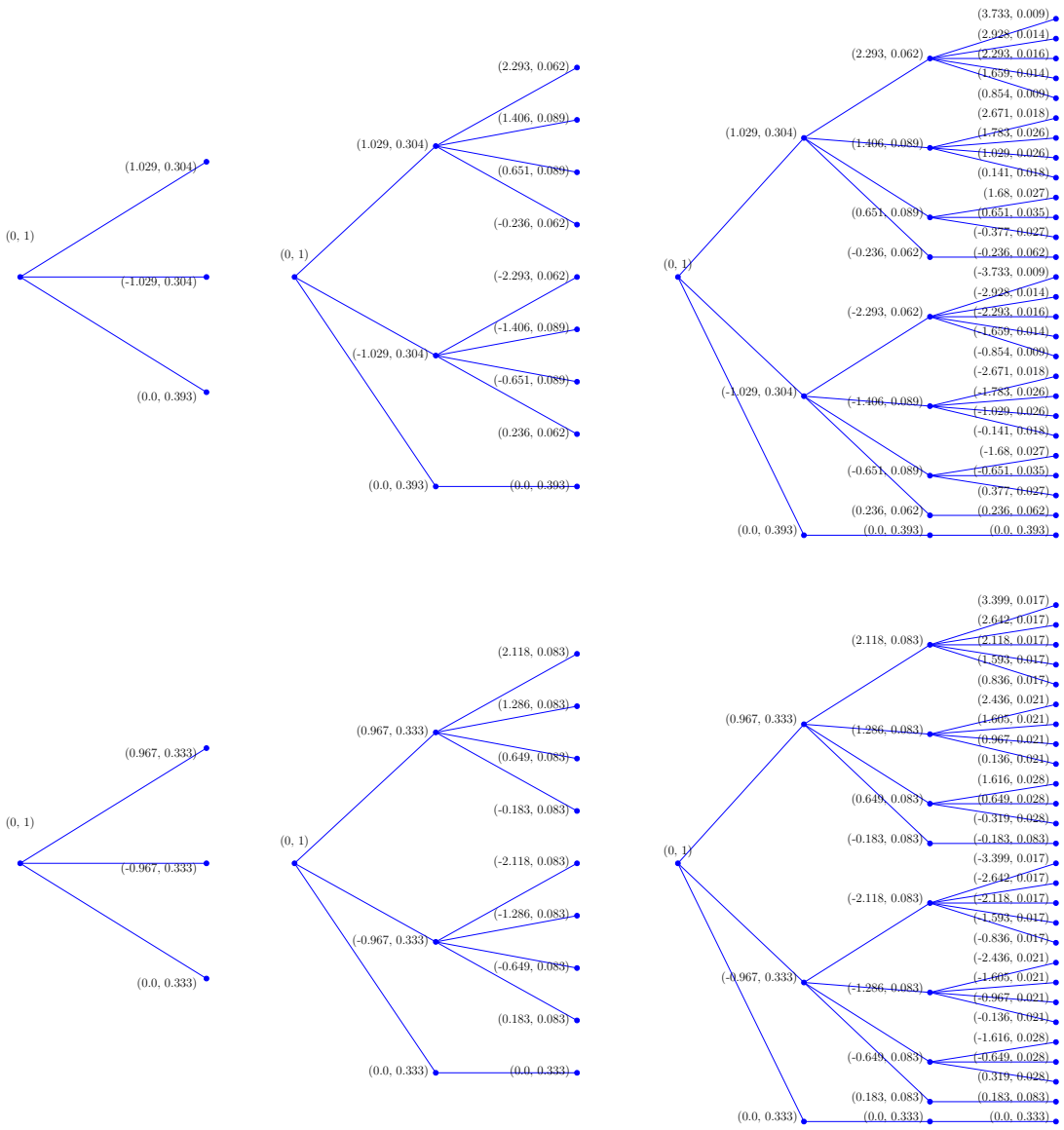


Figure 5.3 Iterations $t = 0, 1, 2$ (left to right) of Algorithm 5.2 for the optimal quantization method (top) and the quasi-Monte Carlo method (bottom). The width vector is $(3, 3^2, \dots)$. The point and weight (ζ^n, W^n) are displayed next to the correspond node.

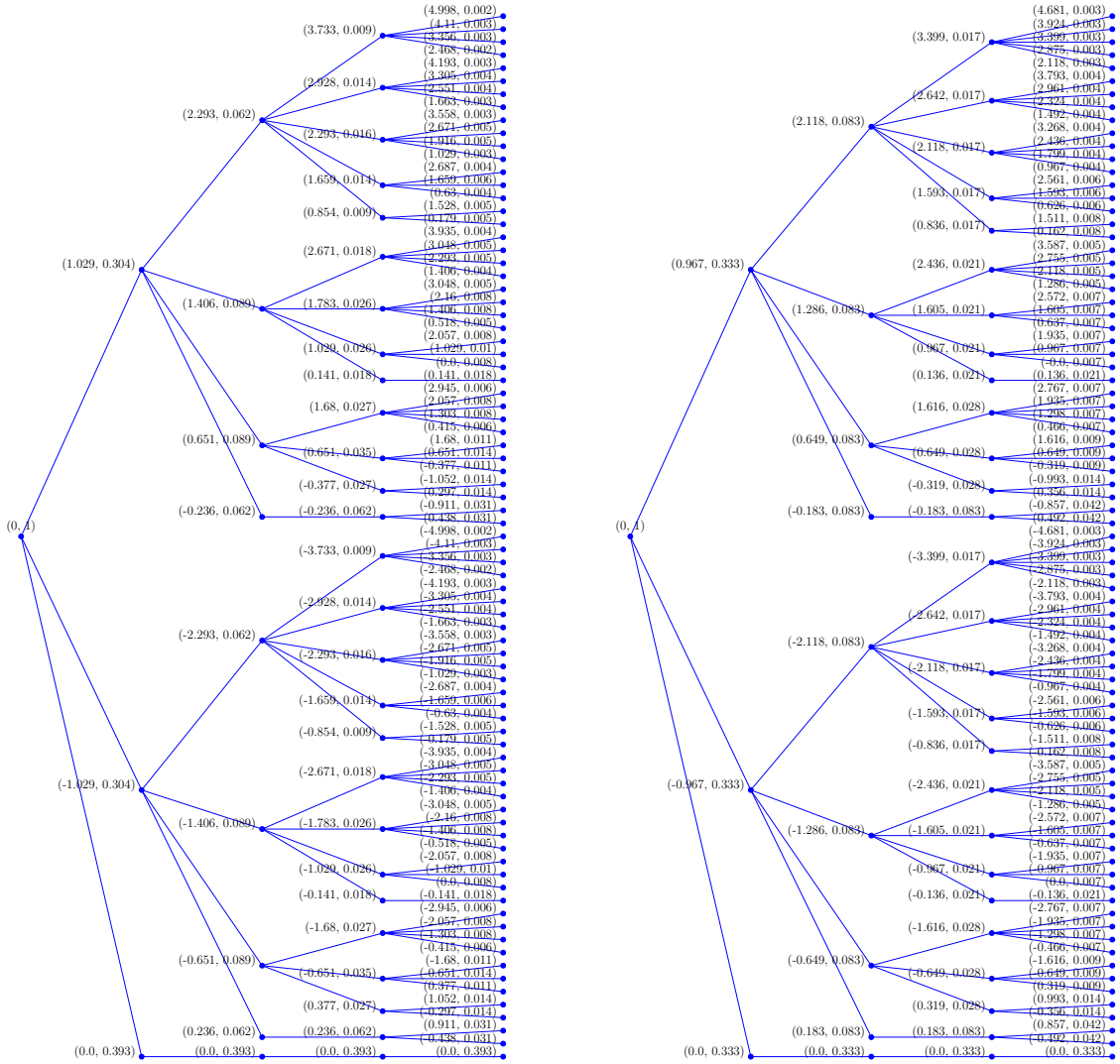


Figure 5.4 Iteration $t = 3$ of Algorithm 5.2 for the OQ method (left) and QMC method (right). The width vector is $(3, 3^2, \dots)$. (The two tree structures are similar but not perfectly identical: e.g., the topmost node at stage 3 has 4 child nodes on the left and 5 child nodes on the right.)

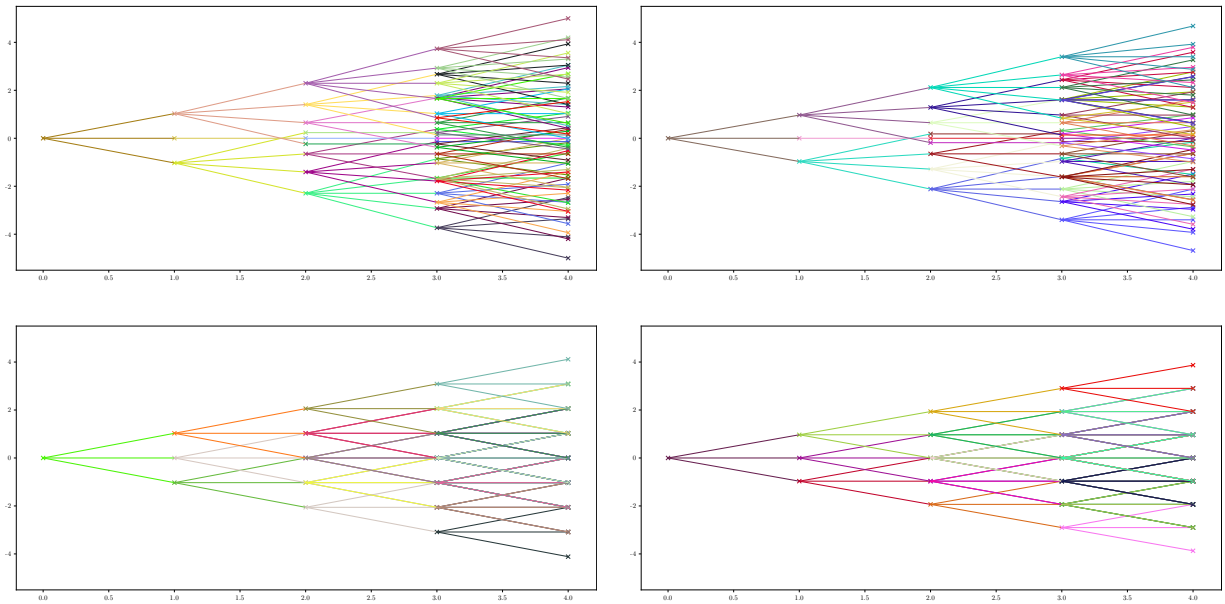


Figure 5.5 81 Brownian motion paths for the following scenario trees: bound-minimizing OQ (top left), symmetrical OQ (bottom left), bound-minimizing QMC (top right) and symmetrical QMC (bottom right). Less paths appear for the symmetrical scenario trees because many of them are overlapping.

CHAPITRE 6 ARTICLE 3: THE FIGURE OF DEMERIT: A QUALITY MEASURE FOR THE DISCRETIZATION OF PROBABILITY DISTRIBUTIONS IN MULTISTAGE STOCHASTIC OPTIMIZATION

Julien Keutchayan[1,2], David Munger[1,2], Michel Gendreau[1,2], Fabian Bastin[3]

[1]Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Canada.

[2]Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Canada.

[3]Département d'Informatique et de Recherche Opérationnelle, Montréal, Canada

Article soumis à la revue *Operations Research* le 27 avril 2018.

Abstract. An important step in solving a stochastic optimization problem is the search for an efficient method for discretizing the probability distribution of the random parameters. This step becomes even more critical when one works with a *multistage* problem, where the discretization of the underlying stochastic process may lead to a scenario tree of large size. This scenario tree, in turn, generates an approximate optimization problem that may be intractable due to too many decision variables and constraints. Finding good multistage discretization schemes leading to reasonable-size scenario trees is therefore essential to broaden the class of solvable multistage problems. In this paper we introduce a new approach for discretizing the probability distributions underlying multistage problems. This approach is based on a quality measure called the *figure of demerit*. This measure leverages on knowledge about the structure of the problem, acquired from expertise or numerical investigation, to design a suitable discretization scheme. The approach developed is versatile as it can be applied to essentially any problem, regardless of linearity, convexity, differentiability, etc., and combined with a great deal of discretization methods used in numerical integration.

6.1 Introduction

Multistage stochastic optimization provides a mathematical framework for modeling and solving decision-making problems that include uncertain parameters being revealed between each decision stage. There are numerous sources of uncertainty in real-world optimization problems: the prices of assets in a *portfolio optimization problem* (see, e.g., Ziemba (2003) and Yu *et al.* (2003)); the demand and price of electricity and the natural inflows in water

reservoirs in a *hydroelectricity production problem* (see, e.g., Wallace et Fleten (2003) and Kovacevic *et al.* (2013)); the number of customers on each route of a transportation network in a *network design problem* (see, e.g., Louveaux (1998) and Powell et Topaloglu (2003)); and the list goes on. Provided a probability distribution is available for the underlying stochastic process modeling the stagewise evolution of information –most often inferred from available data– a *multistage stochastic optimization problem* (MSOP) can be formulated and addressed using different approaches; see, e.g., Ruszczyński et Shapiro (2003a), Birge et Louveaux (2011), and King et Wallace (2012).

Multistage stochastic optimization problems are impossible to solve exactly except in some unrealistic cases of low practical interest. This has to do with the fact that the optimal decisions of MSOPs are functions of the random parameters' realizations, and that the latter are generally of infinite cardinality, which results in an *infinite-dimensional* optimization problems. The *scenario-tree generation* approach provides a way to estimate the optimal solutions of MSOPs (the optimal value and decisions) by discretizing the probability distribution of the random parameters, hence reducing the original infinite-dimensional problem to an approximate *finite-dimensional* problem. Scenario trees are appealing as they allow to consider any type of underlying stochastic processes. Thus, both Markovian or non-Markovian processes can be used to model the stagewise evolution of the random parameters. Non-Markovian processes, for instance, are believed to provide a more accurate description of the uncertainty of the asset returns and the weather; see, e.g., Calvet et Fisher (2002) and Lovejoy *et al.* (2018). However, this advantage of scenario trees over Markov decision processes comes at the cost of a tree structure that grows exponentially with increase of the number of stages, hence limiting the use of scenario trees to multistage problems with a small number of stages. There is active research that aims at reducing the size of scenario trees while keeping good approximation quality. We now review the main approaches.

When dealing with a real-world application, the scenario tree must be carefully chosen to provide a good approximation of the original problem. What constitutes a "good approximation" is a broad question whose answer will generally depend on the considered problem; see, e.g., Kaut et Wallace (2007) and Keutchayan *et al.* (2017). In this paper, we follow the common guideline in the scenario-tree generation approach that consists in constructing the scenario tree with the goal to keep the optimal-value error (i.e., the absolute difference between the optimal values of the original and approximate problems) as small as possible for a given finite computational cost. We will define formally the scenario tree in the next section, but for now we can see it as a triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ where $\mathcal{T}$ refers to the *tree structure*, $\mathcal{P}$ to the set of *discretization points*, and $\mathcal{W}$ to the set of *discretization weights*. We emphasize that by scenario tree we mean the whole triple and not only the couple $(\mathcal{P}, \mathcal{W})$.

The reason we emphasize this is because many scenario-tree approaches that employ discretization procedures from numerical integration, such as Monte Carlo, quasi-Monte Carlo, optimal quantization, and quadrature rules, do not feature a systematic way to compute a tree structure $\mathcal{T}$. This structure is typically taken as symmetrical in the absence of reliable alternatives. Symmetrical structures are considered for example by Defourny *et al.* (2013); Hilli et Pennanen (2008); Høyland et Wallace (2001); Pflug (2001); Pflug et Pichler (2015), and Shapiro (2006). The first goal of our approach is therefore to improve the efficiency of scenario trees by leveraging on a suitable choice for the tree structure $\mathcal{T}$. This means that we want to search among *all* tree structures, including those that are not symmetrical. The second goal is to do so in a way that takes into account the whole properties of the problem (stochastic process, revenue function, and constraints), as all of them influence the scenario tree's approximation quality (Keutchayan *et al.*, 2018a). Thus, our approach aims at being *problem-driven*, as opposed to *distribution-driven* approaches that focus solely on approximating the underlying stochastic process.

The research carried out by Keutchayan *et al.* (2018a) describes general principles to compute scenario trees better suited to problems without providing a systematic way to implement them in practice. The present paper builds on these principles, in particular the optimal-value error upper bound (Keutchayan *et al.*, 2018a, Corollary 4.5), to introduce a practical quality measure for scenario trees called the *figure of demerit*. This figure depends on the triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and on a set of *guidance functions* Γ defined on the support of the probability distributions and used as input of the method. The goal of these guidance functions is to weigh the importance of different regions inside the distribution support regarding their respective impact on the variability of the recourse functions. Regions leading to high variability of the recourse functions are given a large weight, and conversely, regions leading to low variability are given a small weight. Since the recourse functions of a real-world problem are not known exactly (as they are part of the optimization problem), an analytic derivation of the guidance functions will typically be impossible. In practice they will therefore be inferred from the properties of the problem, from expertise, or from numerical investigation.

The conceptual idea of weighing the distribution support to guide the multistage discretization scheme is inspired by the high-dimensional numerical integration techniques used in the quasi-Monte Carlo approach. These techniques associate a weight to each subset of coordinates to guide the generation of low-discrepancy points sets mostly towards the subsets of greater importance, which are typically those where the integrand has large variability; see, e.g., Dick *et al.* (2013). In our case, the weights are functions defined over the distribution supports, whose goal is to guide the multistage discretization scheme mostly towards the regions of the distribution supports that lead to high variability of the recourse functions

(i.e., high variability of future gains or losses).

The remainder of this paper is organized as follows: In Section 6.2 we introduce formally the multistage stochastic optimization problems, the scenario trees, and the scenario-tree approximate problems. In Section 6.3 we introduce the figure of demerit and explain the general procedure to compute scenario trees of minimum demerit. In Section 6.4 we show how the general framework can be implemented in different settings of problems. Finally, in Section 6.5 we conclude the paper and discuss the outlook. Along the lines of our development, we illustrate the approach using several examples.

6.2 Preliminaries

6.2.1 Multistage stochastic optimization problems

We consider a general MSOP with discrete stages $t = 0, 1, \ldots, T$, with $T \in \mathbb{N}_{>0}$, that correspond to different time points (not necessarily evenly spaced) when decisions are made. The uncertain parameters whose realizations become known between the stages $t-1$ and t are modeled by a $\mathbb{R}^{d_t}$-valued random vector $\boldsymbol{\xi}_t$ of arbitrary dimension and probability distribution. Although we consider that there are no random parameters before stage 0, for notational convenience we add an artificial $\mathbb{R}^{d_0}$-random vector $\boldsymbol{\xi}_0$ that takes only one possible value ξ_0 with probability one. The value ξ_0 can be seen as an aggregate of all information known by the decision-maker at the time of the stage-0 decisions. The decisions $y_t \in \mathbb{R}^{s_t}$ are made at each stage $t = 0, \ldots, T$ with the goal to maximize the objective function of the problem while satisfying its constraints. (For the sake of clarity and without loss of generality, we assume throughout the paper that $s_t = s$ and $d_t = d$ for all t.) The objective function of the MSOP is given by the expectation of a revenue function $q : \mathbb{R}^{(T+1)s} \times \mathbb{R}^{(T+1)d} \to \mathbb{R}$ that depends both on the decisions $(y_0, \ldots, y_T)$ and the uncertain parameters $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$.

The MSOP considered has therefore the following underlying dynamic of actions from stage 0 to T:

$$y_0 = x_0(\xi_0) \xrightarrow{\xi_1} y_1 = x_1(y_0; \xi_0, \xi_1) \xrightarrow{\xi_2} \cdots \xrightarrow{\xi_T} y_T = x_T(y_{..T-1}; \xi_{..T}). \tag{6.1}$$

This dynamic means that the decisions y_0 are made when only the non-random information ξ_0 is available. Then, a realization ξ_1 of the uncertain parameters $\boldsymbol{\xi}_1$ becomes known and, based on it, the decisions y_1 are made at stage 1. The decisions y_1 are therefore functions of y_0 and (ξ_0, ξ_1), which we denote by $y_1 = x_1(y_0; \xi_0, \xi_1)$. (Note that we use the notation "y" for decision *vectors* and the notation "x" for decision *functions*.) At each stage $t \geq 2$, the decisions $y_t = x_t(y_{..t-1}; \xi_{..t})$ are functions of the decisions $y_{..t-1} := (y_0, y_1, \ldots, y_{t-1})$ made prior to stage t and of the random information $\xi_{..t} := (\xi_0, \xi_1, \ldots, \xi_t)$ available at stage t.

This dependence of the decisions on past information guarantees that the decisions are non-anticipative (i.e., adapted to the natural filtration of the stochastic process).

The MSOP can be formulated recursively by introducing the stage-t *recourse function* Q_t^* and the stage-t *expected recourse function* Q_t for each $t = 0, \dots, T$. The former represents the optimal conditional expected revenues given the realizations $\xi_{..t}$ and the (non-necessarily optimal) decisions $y_{..t-1}$ made up to stage t, while the latter represents the same quantity expect that the decisions $y_{..t}$ are made up to *and including* stage t. Since at stage T the realization $\xi_{..T}$ of the whole stochastic process is known and all the decisions $y_{..T}$ have already been made, the expected recourse function Q_T is computed directly from the value of the revenue function, i.e., $Q_T(y_{..T}; \xi_{..T}) := q(y_{..T}; \xi_{..T})$. Starting from the values of Q_T, the other recourse functions are given by the following backward recursion that stems from the stochastic dynamic programming equations:

$$Q_t^*(y_{..t-1}; \xi_{..t}) := \sup_{y_t \in Y_t(y_{..t-1}; \xi_{..t})} Q_t(y_{..t-1}, y_t; \xi_{..t}), \quad t = 0, \dots, T, \tag{6.2}$$

$$Q_t(y_{..t}; \xi_{..t}) := \mathbb{E}[Q_{t+1}^*(y_{..t}; \boldsymbol{\xi}_{..t+1}) \,|\, \boldsymbol{\xi}_{..t} = \xi_{..t}], \quad t = 0, \dots, T-1, \tag{6.3}$$

where the set-valued mapping $(y_{..t-1}, \xi_{..t}) \rightrightarrows Y_t(y_{..t-1}; \xi_{..t}) \subseteq \mathbb{R}^s$ models the constraints (or *feasible set*) of the problem at stage t. For notational consistency, at stage 0 the argument $y_{..t-1}$ of the feasible set and the recourse function are removed, i.e., $Y_0(y_{..0-1}; \xi_{..0})$ and $Q_0^*(y_{..0-1}; \xi_{..0})$ become $Y_0(\xi_0)$ and $Q_0^*(\xi_0)$, respectively.

The quantity $Q_0^*(\xi_0)$, which is no longer a function as ξ_0 takes only one value, is the *optimal-value* of the MSOP. The *optimal decision policy* is $(x_0^*, x_1^*, \dots, x_T^*)$ 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 (6.2). The conditions required to ensure the existence of an optimal policy can be found in Keutchayan *et al.* (2018a).

Example 6.2.1. The general MSOP above has a particular case, called the *polyhedral* MSOP, extensively considered in applications. For this problem the following simplifications occur:

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

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 ξ_t and hence are realizations of

random quantities. In particular, $q_0(\cdot\,;\xi_0)$ is a polyhedral function and $q_1, \ldots, q_T$ are random polyhedral functions. In the case of a *linear* MSOP the revenue function is further simplified as $q_t(y_t;\xi_t) = c_t(\xi_t)^\top y_t$. For a theoretical analysis of such problems we refer to Ruszczyński et Shapiro (2003b).

6.2.2 Scenario-tree approximate problems

Real-world problems in the form (6.2)-(6.3) are generally impossible to solve exactly. This is also true for problems that are in the simplified polyhedral or linear forms of Example 6.2.1. One reason that makes such problems so difficult to solve is the fact that the expectation (6.3) cannot be computed exactly in most applications involving continuous distributions. But this is not the only reason. Even when the distribution sits naturally on finitely many points, the size of the resulting problem (measured in terms of the number of decision variables and constraints) is generally very large, and hence the problem cannot be solved exactly in a reasonable time by current optimization algorithms. For this reason, the support of the probability distribution needs to be discretized and restricted to a manageable size to solve the problem at least approximately. The approach that we develop in this paper is consistent with this observation. It consists in building a discretized representation of the stochastic process that involves only a limited number of its scenarios. This representation is usually called *scenario tree* in the literature.

At this point it is important to mention that we consider a concept of scenario trees broader than what is usually considered in the literature. This concept covers the tree structures as they are usually considered in multistage stochastic programming, in which a node is reachable from the root by only one path. Two examples of such structures are given in Figure 6.1. Contrary to the traditional, strong-sense, definition, our view on scenario trees also covers the discretized representations that are not strictly speaking in a tree-like structure, in which case the structure is such that several paths may lead to the same node. An example of the latter representation, referred to as *mesh*, is displayed in Figure 6.2. The recombining structures are important to consider as they are extensively used, for instance, in financial engineering for the pricing of derivatives. (The problem of pricing a derivative with early exercise opportunities is formulated as a MSOP in the form (6.2)-(6.3) with $q(\cdot;\cdot)$ the derivative payoff.) In this paper we call specifically *recombining scenario trees* the latter discretized representations. In our view, recombining scenario trees are a specific type of scenario trees where some nodes, and their corresponding subtrees, have been combined into a single one. To be able to assess the quality of recombining scenario trees in the same framework as (strong-sense) scenario trees, we will undo the recombination. But this procedure is only

artificial, i.e., once the quality is measured and the optimal parameters of the structure are found (such as the numbers of nodes at each stage of the mesh in Figure 6.2), the structure is combined again and used in the way originally intended. The way we switch from a recombining scenario tree to an equivalent (strong-sense) scenario tree, and vice versa, will be explained in Section 6.2.3.

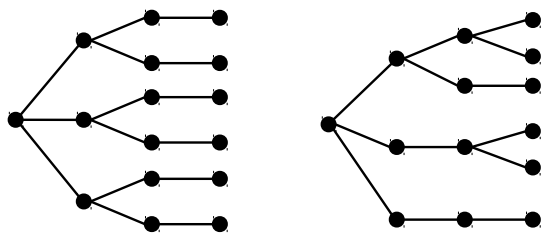


Figure 6.1 Strong-sense tree structures are such that a single path joins every node to the root (the leftmost node). The left structure is symmetrical with bushiness $(3, 2, 1)$; the right structure is not symmetrical (cf. (N8)).

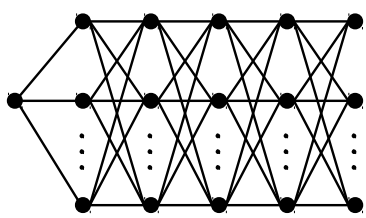


Figure 6.2 Recombining tree structures are such that several paths join some nodes to the root (the leftmost node). The particular recombining structure displayed here is referred to as *mesh*.

We now formalize the above discussion. We call *scenario tree* (in the strong sense) a triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ 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 ζ^n for each node and at the root $\zeta^{n_0} = \xi_0$ by definition);

(D3) $\mathcal{W} = \{w^e \in (0, \infty) : e \in \mathcal{E}\}$ is the set of *discretization weights* (one weight w^e for each edge).

On top of the above definition, we introduce the following additional notations and terminology that will help us describe conveniently a scenario tree (some of them are illustrated in Figure 6.3):

(N1) $t(n)$ is the *stage* (or *depth*) of n (the number of edges between n_0 and n) and $\mathcal{N}_t := \{n \in \mathcal{N} : t(n) = t\}$ is the set of all stage-t nodes;

(N2) $C(n)$ is the set of *child* nodes of n (those connected to n at stage $t(n)+1$) and $p(n)$ is the *parent* node of n (that connected to n at stage $t(n)-1$);

(N3) $e(n)$ is the edge connecting n and $p(n)$; to lighten the notation, we will write from now on w^n instead of $w^{e(n)}$ and consider, when there is no ambiguity, that weights are associated with nodes instead of edges (this is equivalent since the map $\mathcal{N} \setminus \{n_0\} \ni n \mapsto e(n) \in \mathcal{E}$ is a bijection);

(N4) $\mathcal{N}(n)$ is the set of *descendant* nodes of n, which includes n itself, and $\mathcal{E}(n)$ is the subset of $\mathcal{E}$ that contains all edges connecting the nodes in $\mathcal{N}(n)$;

(N5) $[n, m]$ is the (unique) sequence of nodes from n to m, which includes both ends, where m is a descendant node of n; we write $(n, m]$ when n is excluding from the sequence;

(N6) $\zeta^{\cdot n} := (\zeta^m)_{m \in [n_0, n]}$ is the *discretization sequence* from the root node to n and $W^n := \prod_{m \in (n_0, n]} w^n$ is the *product weight* of n;

(N7) $\mathcal{P}^{C(n)}$ and $\mathcal{W}^{C(n)}$ are the sets of discretization points and weights at $C(n)$;

(N8) $\mathcal{T}$ is said to be *symmetrical* if it satisfies $|C(n)| = |C(m)|$ whenever $t(n) = t(m)$; a symmetrical structure is characterized by its *bushiness* $b := (b_0, \ldots, b_{T-1})$ where $b_{t(n)} := |C(n)|$ is the *branching coefficient* at stage $t(n)$;

(N9) $\mathcal{W}$ is said to be *normalized* if it satisfies $\sum_{m \in C(n)} w^m = 1$ 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 $t = 0, \ldots, T-1$), and $\mathcal{W}$ is *standardized* if it is normalized and moreover 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 $n \in \mathcal{N} \setminus \{n_0\}$);

(N10) $\mathcal{P}(n)$ is the sets of discretization points at $\mathcal{N}(n)$, $\mathcal{W}(n)$ is the set of discretization weights at $\mathcal{E}(n)$, and the triple $(\mathcal{T}(n), \mathcal{P}(n), \mathcal{W}(n))$, with $\mathcal{T}(n) := (\mathcal{N}(n), \mathcal{E}(n), n)$, is the *sub-scenario tree* rooted at n.

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

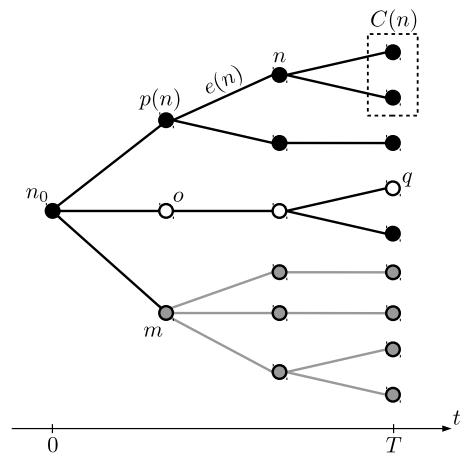


Figure 6.3 Rooted tree structure $\mathcal{T} = (\mathcal{N}, \mathcal{E}, n_0)$. The node n is at stage 2 ($t(n) = 2$). The white nodes constitute the sequence $[o, q]$. The gray nodes and edges constitute the sub-tree structure $\mathcal{T}(m) = (\mathcal{N}(m), \mathcal{E}(m), m)$ of the sub-scenario tree rooted at m.

(D4) each scenario-tree leaf is at depth T (a *leaf* is any node $n \neq n_0$ incident to one edge only);

(D5) each discretization sequence $\zeta^{\cdot\cdot n}$ for $n \in \mathcal{N}_T$ is a possible realization of the stochastic process.

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

$$\widehat{Q}_t^*(y_{..t-1}; \zeta^{\cdot\cdot n}) := \sup_{y_t \in Y_t(y_{..t-1}; \zeta^{\cdot\cdot n})} \widehat{Q}_t(y_{..t-1}, y_t; \zeta^{\cdot\cdot n}), \qquad t = 0, \ldots, T,\ n \in \mathcal{N}_t, \tag{6.4}$$

$$\widehat{Q}_t(y_{..t}; \zeta^{\cdot\cdot n}) := \sum_{m \in C(n)} w^m\, \widehat{Q}_{t+1}^*(y_{..t}; \zeta^{\cdot\cdot n}, \zeta^m), \qquad t = 0, \ldots, T-1,\ n \in \mathcal{N}_t. \tag{6.5}$$

The functions $\widehat{Q}_t^*$ and $\widehat{Q}_t$ are the scenario-tree estimators of Q_t^* and Q_t, respectively. Similarly to the original problem, at the final stage the relation (6.4) is initialized by setting $\widehat{Q}_T(y_{..T}; \zeta^{\cdot\cdot n}) := q(y_{..T}; \zeta^{\cdot\cdot n})$, for every $n \in \mathcal{N}_T$, and at the initial stage $Y_0(y_{..0-1}; \zeta^{\cdot\cdot n_0})$ and $\widehat{Q}_0^*(y_{..0-1}; \zeta^{\cdot\cdot n_0})$ become $Y_0(\zeta^{n_0})$ and $\widehat{Q}_0^*(\zeta^{n_0})$, respectively.

The quantity $\widehat{Q}_0^*(\zeta^{n_0})$ is the *optimal-value* of the STAP. It is the scenario-tree estimator of $Q_0^*(\xi_0)$ and the difference $|\widehat{Q}_0^*(\zeta^{n_0}) - Q_0^*(\xi_0)|$ (abbreviated as $|\widehat{Q}_0^* - Q_0^*|$ from now on) is called the *optimal-value error*. The *optimal decision policy* of the STAP is $(\widehat{x}_0^*, \widehat{x}_1^*, \ldots, \widehat{x}_T^*)$ where $\widehat{x}_0^*(\zeta^{n_0})$ is a maximizer of $\widehat{Q}_0(\cdot\,; \zeta^{n_0})$ and $\widehat{x}_t^*(y_{..t-1}; \zeta^{\cdot\cdot n})$ is a maximizer of $\widehat{Q}_t(y_{..t-1}, \cdot\,; \zeta^{\cdot\cdot n})$ in (6.4).

6.2.3 Recombining scenario trees

In the scenario-tree framework described above, a *recombining scenario tree* is a scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ for which some restrictions exist on $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{W}$. These restrictions are specific to the type of recombining tree considered and they guarantee the equivalence between the two forms. We illustrate in Figure 6.4 this equivalence for the mesh. We see that the mesh is a scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ for which:

(i) $\mathcal{T}$ is symmetrical with bushiness $b = (b_0, \ldots, b_{T-1})$ with $b_0, \ldots, b_{T-1} \geq 1$;

(ii) $\mathcal{P}$ and $\mathcal{W}$ are such that there are always exactly b_{t-1} different sub-scenario trees with roots at depth t (out of a total number of $\Pi_{i=0}^{t-1} b_i$ possible sub-scenario trees).

The rule for recombining a scenario tree is that two sub-scenario trees, say with roots n and m, can be combined into a single one if they are identical, i.e., if $(\mathcal{T}(n), \mathcal{P}(n), \mathcal{W}(n)) = (\mathcal{T}(m), \mathcal{P}(m), \mathcal{W}(m))$. If this equality holds, the nodes n and m have identical present and future and only differ by their respective past. It is therefore relevant to merge these two sub-scenario trees if the problem considered is such that the optimal decisions are not path-dependent. This is for instance the case for the problem of finding the optimal exercise time of an American put option; see, e.g., Hull (2017). Problems that exhibit the path-independence property are however the exception rather than the rule. In a more general MSOP framework, recombining scenarios trees are used as a mean to improve the efficiency of the solution method. When the STAP corresponding to the polyhedral MSOP of Example 6.2.1 is solved using a nested Benders decomposition method, the equality of the two sub-scenario trees ensures that the Benders cuts at node n are also valid cuts at node m. This principle, called *cut sharing*, increases the convergence speed of the method because it reduces the number of cuts that need to be computed at each iteration. Cut sharing is used for instance in the backward pass of the stochastic dual dynamic programming method. The latter was introduced by Pereira et Pinto (1991) to solve long-horizon problems with stagewise independent random parameters. Cut sharing is discussed in a general framework in Ruszczyński (2003). Recombining scenario trees are also considered in Küchler (2009).

As stated above, we now call scenario tree any triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ satisfying (D1)-(D5). This includes as a particular case (after transformation) the recombining scenario trees.

6.3 Derivation of the figure of demerit

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

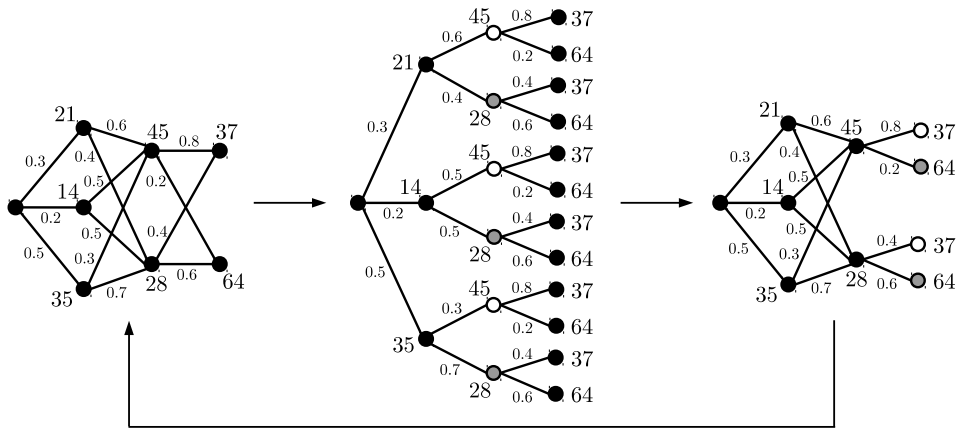


Figure 6.4 Equivalence between the recombining and strong-sense scenario-tree forms for a mesh with bushiness $(3, 2, 2)$. Nodes that are the roots of identical sub-scenario trees are represented in gray or white.

Proposition 6.3.1 (Keutchayan *et al.* (2018a))**.** *The optimal-value error between the MSOP and the STAP for any scenario tree* $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ *is bounded as follows:*

$$|Q_0^* - \widehat{Q}_0^*| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \sup_{f \in \mathcal{G}^n} |\mathcal{E}^n(f)|, \tag{6.6}$$

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

$$\mathcal{E}^n(f) = \mathbb{E}[f(\boldsymbol{\xi}_{t+1}) \,|\, \boldsymbol{\xi}_{..t} = \zeta^{\cdot\cdot n}] - \sum_{m \in C(n)} w^m f(\zeta^m), \tag{6.7}$$

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

$$\mathcal{Q}^n = \Big\{ Q_{t+1}^*(y_{..t}; \zeta^{\cdot\cdot n}, \cdot\,) : y_{..t} = x_{..t}(\zeta^{\cdot\cdot n}) \text{ with } x_{..t} \in \prod_{i=0}^{t} \{x_i^*, \widehat{x}_i^*\} \Big\}. \tag{6.8}$$

The symbol $\prod_{i=0}^{t}$ in (6.8) denotes the $(t+1)$-fold Cartesian product. Thus, $x_{..t}$ is any $(t+1)$-dimensional decision function having each of its i-th component equal to either x_i^* or $\widehat{x}_i^*$, and $x_{..t}(\zeta^{\cdot\cdot n})$ denotes the decisions made with $x_{..t}$ when the realization is $\zeta^{\cdot\cdot n}$ in the dynamic of actions (6.1).

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 the recourse functions in $\mathcal{Q}^n$. Functions in $\mathcal{Q}^n$ are difficult to specify exactly as they depend on the two unknowns x^* and $\widehat{x}$. For this reason a larger set $\mathcal{G}^n$ that includes $\mathcal{Q}^n$ should be considered and, since

the supremum will increase as $\mathcal{G}^n$ becomes larger, the inclusion of $\mathcal{Q}^n$ into $\mathcal{G}^n$ should be as tight as possible. The above result provides the general principle for building scenario trees suitable to problems: Given two nodes n and m, both in $\mathcal{N} \setminus \mathcal{N}_T$, it may happen that the functions in $\mathcal{G}^n$ are easier to integrate numerically than those in $\mathcal{G}^m$. That is to say, for an equal number of child nodes, the integration error $\mathcal{E}^n(f)$ for all $f \in \mathcal{G}^n$ is much smaller than the error $\mathcal{E}^m(f)$ for all $f \in \mathcal{G}^m$. This may occur when the scenario $\zeta^{\cdot\cdot n}$ leading to node n entails a lower variability of the recourse functions than the scenario $\zeta^{\cdot\cdot m}$ leading to m. In this case, the tree structure should reflect this property to increase the accuracy of the scenario tree, i.e., the structure should be such that there are more nodes in $C(m)$ at the expense of $C(n)$ to minimize the overall sum in (6.6).

This theoretical guideline, however, says little about how to select each set $\mathcal{G}^n$ appropriately and how to measure the difficulty of numerically integrating each $f \in \mathcal{G}^n$. It also says little about how the upper bound can be minimized in practice to generate a complete triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ that characterizes the scenario tree. We answer these questions in the remainder of this section.

A first step in the development of our approach is 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

$$\mathcal{G}_t(\xi_{..t}) = \{Q^*_{t+1}(y_{..t}; \xi_{..t}, \cdot\,) : y_{..t} \in Z_t(\xi_{..t})\}, \quad \text{for every } t = 0, \ldots, T-1 \text{ and } \xi_{..t}. \tag{6.9}$$

The set $Z_t(\xi_{..t})$ in (6.9) is the set of all feasible decision sequences up to and including stage t for the realization $\xi_{..t}$, which is defined recursively as $Z_0(\xi_0) = Y_0(\xi_0)$ and

$$Z_t(\xi_{..t}) = \{y_{..t} \in \mathbb{R}^{s(t+1)} : y_{..t-1} \in Z_{t-1}(\xi_{..t-1}),\, y_t \in Y_t(y_{..t-1}; \xi_{..t})\}. \tag{6.10}$$

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

Condition 6.3.1. For any scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and node $n \in \mathcal{N} \setminus \mathcal{N}_T$, the set $\mathcal{G}_{t(n)}(\zeta^{\cdot\cdot n})$ is included in a function space $\mathcal{F}$ for which it holds that

$$|\mathcal{E}^n(f)| \leq \mathcal{V}_{\mathcal{F}}(f)\, \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)}), \quad \text{for every } f \in \mathcal{F}, \tag{6.11}$$

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 $(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$.

Remark 6.3.1. It is possible to assume a weaker form of Condition 6.3.1, where the bound

(6.11) does not hold for *any* scenario tree but only for points and weights $(\mathcal{P}, \mathcal{W})$ that are restricted to some specific form. These forms include, for instance, the normalized or standardized weights defined in (N9), or the points sets generated by specific procedures, such as the lattice point sets of quasi-Monte Carlo methods. If the weaker form is assumed, then all 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 6.3.1 holds in its strong form.

Function spaces for which (6.11) may hold (in strong or weak form) are for instance:

(S1) $\mathcal{F} = W_{2,\gamma,\mathrm{mix}}^{(1,\ldots,1)}([0,1]^d)$ is the *weighted tensor product* Sobolev space: $\mathcal{V}_{\mathcal{F}}(f)$ is the norm of f and $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ is the *weighted discrepancy*;

(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 $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ is the norm of the representer of $\mathcal{E}^n(\cdot)$ given by Riesz representation theorem;

(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 $\mathcal{D}_{\mathcal{F}_q}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ is the order-q *Fortet-Mourier distance* between the conditional distribution of $\boldsymbol{\xi}_{t(n)+1}$ given $\boldsymbol{\xi}_{..t(n)} = \zeta^{\cdot\cdot n}$ and the distribution $\sum_{m \in C(n)} w^m \delta_{\zeta^m}$, where δ_{ζ^m} is the Dirac measure at ζ^m and $\mathcal{W}^{C(n)}$ is normalized;

(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 $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ is the *worst-case error* of $(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ in $\mathcal{F}$.

Multidimensional recourse functions do not belong to settings (S1) or (S2) as they are not smooth enough. However, Heitsch *et al.* (2016) showed that the recourse functions of two-stage linear problems can be smoothed out by the ANOVA decomposition. Thus, the integration error may have a rate of convergence similar to that of (S1) if the effective dimension of the functions is small. This smoothing effect of the ANOVA decomposition for functions with "kinks" is also studied by Griebel *et al.* (2010, 2013). There is a rich literature studying the above settings in the context of numerical integration; we refer for instance to Dick et Pillichshammer (2010); Dick *et al.* (2013); Hickernell *et al.* (2004b); Kuo *et al.* (2011); Novak et Woźniakowski (2010); Pflug et Pichler (2011), and Sloan et Woźniakowski (1998).

What makes Condition 6.3.1 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 $\mathcal{V}_{\mathcal{F}}(f)$ is large and the bound is tight, then $\mathcal{E}^n(f)$ is large and this means that f is difficult to integrate numerically. In the latter case, it makes sense to allocate more nodes in $C(n)$ and use more computational resources to search for the points and weights that minimize $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$. We call $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ the *demerit of node* n as it measures the quality of the discretization points and weights at $C(n)$ (or rather the absence of quality, as high values mean low quality), and we call $\mathcal{V}_{\mathcal{F}}(f)$ the *$\mathcal{V}_{\mathcal{F}}$-variation* of f as it typically measures the variation of f over its definition domain (high values mean large variations).

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

Condition 6.3.2. For each $t = 0, \ldots, T-1$ and realization $\xi_{\cdot\cdot t}$, the $\mathcal{V}_{\mathcal{F}}$-variation of the recourse function $Q^*_{t+1}(y_{\cdot\cdot t}; \xi_{\cdot\cdot t}, \cdot\,)$ is uniformly bounded in $y_{\cdot\cdot t} \in Z_t(\xi_{\cdot\cdot t})$.

When the set $Z_t(\xi_{\cdot\cdot t})$ is a compact subset of $\mathbb{R}^{s(t+1)}$ (as holds, e.g., under the conditions given in Keutchayan *et al.* (2018a)), the above condition is satisfied in particular if the mapping $Z_t(\xi_{\cdot\cdot t}) \ni y_{\cdot\cdot t} \mapsto \mathcal{V}_{\mathcal{F}}(Q^*_{t+1}(y_{\cdot\cdot t}; \xi_{\cdot\cdot t}, \cdot\,))$ is upper semi-continuous.

We are now ready to define the concepts of *guidance functions* and of *figure of demerit*, which are at the core of our new scenario-tree generation approach.

Definition 6.3.2. We call *set of guidance functions* the set $\Gamma = \{\gamma_t(\cdot) : t = 0, \ldots, T-1\}$ where each $\gamma_t(\cdot)$ is a $\mathbb{R}_{\geq 0}$-valued function defined on the set of all realizations $\xi_{\cdot\cdot t}$. We call *figure of demerit* of the scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ for the set of guidance functions Γ the quantity

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma) := \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, \gamma_{t(n)}(\zeta^{\cdot\cdot n}) \, \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)}). \tag{6.12}$$

The guidance functions $\gamma_0(\cdot), \ldots, \gamma_{T-1}(\cdot)$ are the inputs of our approach. Their goal is to match as closely as possible the $\mathcal{V}_{\mathcal{F}}$-variability of the recourse functions at each stage and for each value of the random parameters. The following result is a corollary of Proposition 6.3.1 that shows that for a suitable choice of Γ the figure of demerit plays the role of a measure of quality for scenario trees.

Corollaire 6.3.3. *Under Conditions 6.3.1 and 6.3.2, there exists a set of guidance functions* $\widetilde{\Gamma} = \{\widetilde{\gamma}_t(\cdot) : t = 0, \ldots, T-1\}$ *such that for every scenario tree* $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ *the figure of demerit* $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \widetilde{\Gamma})$ *is an upper bound on the optimal-value error, i.e,*

$$|Q^*_0 - \widehat{Q}^*_0| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \widetilde{\gamma}_{t(n)}(\zeta^{\cdot\cdot n}) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)}). \tag{6.13}$$

Proof. We show that the guidance functions defined as

$$\widetilde{\gamma}_t(\xi_{..t}) := \sup_{f \in \mathcal{G}_t(\xi_{..t})} \mathcal{V}_{\mathcal{F}}(f), \quad \text{for every } t = 0, \ldots, T-1 \text{ and } \xi_{..t}, \tag{6.14}$$

are a possible choice for (6.13) to hold. These guidance functions are well-defined since the supremum is positive and finite by Condition 6.3.2. Let $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ be any scenario tree. Taking $\mathcal{G}^n := \mathcal{G}_{t(n)}(\zeta^{\cdot n})$ for every $n \in \mathcal{N} \setminus \mathcal{N}_T$ in Proposition 6.3.1 ensures that $\mathcal{Q}^n \subseteq \mathcal{G}^n$ and allows to write by Condition 6.3.1:

$$|Q_0^* - \widehat{Q}_0^*| \leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \sup_{f \in \mathcal{G}^n} |\mathcal{E}^n(f)| \tag{6.15}$$

$$\leq \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)}) \sup_{f \in \mathcal{G}^n} \mathcal{V}_{\mathcal{F}}(f). \tag{6.16}$$

The last term is precisely the right-hand side of (6.13) for this choice of guidance functions. □

It follows from the relation (6.14) that $\widetilde{\Gamma}$ records quantitatively the relevant information about the structure of the MSOP and that the figure of demerit $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \widetilde{\Gamma})$ measures the suitability between the scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and the MSOP. Indeed, the quantity $\mathcal{V}_{\mathcal{F}}(f)$ for a given $f \in \mathcal{G}_t(\xi_{..t})$ depends on the variability of the recourse function f, which in turn depends on the revenue function, the constraints, and the stochastic process that altogether characterize the problem.

The above corollary provides the theoretical justification of our approach to generate scenario trees. This approach is in two steps: (i) a set of guidance functions Γ is selected with the goal to match the $\mathcal{V}_{\mathcal{F}}$-variability of the recourse functions with respect to the random parameters; (ii) the scenario tree with the lowest figure of demerit $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$ is search for 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 guidance functions larger than the ones used in the proof will keep the inequality valid, but more importantly, more suitable guidance functions can be chosen by considering sets of recourse functions that are strictly smaller than $\mathcal{G}_t(\xi_{..t})$ but still include $\mathcal{Q}^n$ for each possible scenario tree. Finding such sets is difficult since we face the problem that we do not know x^* and that $\widehat{x}^*$ depends on the scenario tree. However, if we knew that the decisions yielded by x^* and $\widehat{x}^*$ in the scenario $\xi_{..t}$ always lie in some strict subset $\widehat{Z}_t(\xi_{..t})$ of $Z_t(\xi_{..t})$, we could define accordingly a strict subset $\widehat{\mathcal{G}}_t(\xi_{..t})$ of $\mathcal{G}_t(\xi_{..t})$ that

still includes all the recourse functions of interest. Then, the guidance function defined with $\widehat{\mathcal{G}}_t(\xi_{..t})$ instead of $\mathcal{G}_t(\xi_{..t})$ in (6.14) would entail a tighter upper bound (6.13).

On the practical side, the choice of guidance functions is more flexible than one may think in view of the corollary. Indeed, even if the figure of demerit is not a valid bound on the optimal-value error for the chosen guidance functions, we argue that minimizing the figure of demerit still provides a suitable scenario tree if the guidance functions single out the "quiet"scenarios (those leading to low variability) from the "turbulent" ones (those leading to high variability). In practice, we therefore consider that only the relative values of the guidance functions matter. The mathematical reason behind this is that multiplying all functions by the same constant only results in scaling up or down the figure of demerit, which does not change the scenario tree of lowest demerit. In this sense, the guidance functions are said to be *scale-free*.

We also make several remarks about the second step of the approach. Ideally, we wish to find the scenario tree that minimizes the demerit $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$ over all scenario trees $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ of a given size. However, minimizing the demerit in the general form (6.12) turns out to be a difficult task in itself. It requires to minimize 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., enumerate all (or some) structures and minimize in $(\mathcal{P}, \mathcal{W})$ for each fixed $\mathcal{T}$. The minimization in $(\mathcal{P}, \mathcal{W})$ at fixed $\mathcal{T}$ remains difficult as it is highly non-linear due to the product $W^n \gamma_{t(n)} \mathcal{D}_{\mathcal{F}}$. A way to go around this final difficulty is to restrict the attention to points and weights that are designed to keep each individual node-n demerit $\mathcal{D}_{\mathcal{F}}$ as small as possible. This can be done by using the algorithmic tools provided by some numerical integration methods, such as quasi-Monte Carlo and optimal quantization, which compute the points and weights by minimizing their own version of the node demerit. We note that getting $(\mathcal{P}, \mathcal{W})$ by minimizing all node-n demerits independently of each other is an approximation as this does not imply that $(\mathcal{P}, \mathcal{W})$ minimize the overall figure of demerit.

To summarize, by employing the previous approximation, 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 sub-steps:

(i) finding $(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ that minimizes $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ at each node $n \in \mathcal{N} \setminus \mathcal{N}_T$;

(ii) assigning the points and weights in $(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ to the appropriate node in $C(n)$.

The importance of the sub-step (ii) will become clear in the next section.

We end this section by illustrating on a simple example how a tree structure can be computed

by minimizing the figure of demerit:

Example 6.3.1. 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 of the year and, based on it, assigns the equipment to the different facilities accordingly. The risk forecast is represented by a random variable $\boldsymbol{\xi}_1$ taking four different values: ξ_1^1 (low risk) to ξ_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 $\boldsymbol{\xi}_2$ having continuous distribution correlated to $\boldsymbol{\xi}_1$, and the organization responds to them in a way to maximize the rescue efficiency.

To solve this purposely simplified problem, 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 use the figure of demerit to find the optimal distribution of nodes at stage 2. To this end, we compute the number of child nodes $M_i := |C(n_i)|$ for each stage-1 node n_i that minimizes the figure of demerit among all tree structures with less than N scenarios, i.e., such that $\sum_{i=1}^4 M_i \leq N$ for any $N \geq 4$.

Since the scenario-tree discretization scheme at stage 1 is exact, the integration error $\mathcal{E}^{n_0}(f)$ is zero regardless of f and hence we can set $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n_0)}, \mathcal{W}^{C(n_0)}) = 0$. The figure of demerit of the scenario tree is therefore written as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma) = \sum_{i=1}^{4} p^i \gamma_1(\xi_1^i) \mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n_i)}, \mathcal{W}^{C(n_i)}). \tag{6.17}$$

The guidance function $\gamma_1(\cdot)$ has the following interpretation: it measures the *conditional variability* of the rescue efficiency at stage 2 given the risk forecast at stage 1. Thus, this function does not measure the rescue efficiency itself, but rather the *uncertainty* about it when one is given the risk forecast at stage 1. To see how the values of $\gamma_1(\cdot)$ are determined, consider the following situation: It may happen that highly risky years coincide with natural disasters of highly variable intensities and locations. In such case, one would expect the rescue efficiency to be more uncertain when the risk forecast is ξ_1^4 than when it is ξ_1^1, hence it would be relevant to set $\gamma_1(\cdot)$ such that $\gamma_1(\xi_1^1) \leq \gamma_1(\xi_1^2) \leq \gamma_1(\xi_1^3) \leq \gamma_1(\xi_1^4)$. However, if additional funding is systematically released by governments when the risk is categorized as extreme, which allows the organization to increase substantially its workforce and equipment, then it is possible that the rescue efficiency of highly risky years becomes less uncertain than years of lower risk. In this specific case, one will rather set $\gamma_1(\xi_1^4)$ at a lower value than

the other $\gamma_1(\xi_1^i)$'s. In this example, the subsidy policy of the organization is a feature of the MSOP that will typically appear in the constraints of the problem, and the goal of the guidance function is to account for it as well as for other features.

Suppose that the optimal points and weights at $C(n_i)$ that minimize the node-n_i demerit satisfy $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n_i)}, \mathcal{W}^{C(n_i)}) = A/|C(n_i)|^\alpha = A/M_i^\alpha$ for each $i = 1, \ldots, 4$ with $\alpha > 0$ and $A > 0$. Then, the number of child nodes at stage 2 that minimizes the figure of demerit over all scenario trees with at most N scenarios is given by the optimal solution of

$$\min_{M=(M_1,\ldots,M_4)\in\mathbb{N}^4_{>0}} \sum_{i=1}^{4} \frac{p^i \gamma_1(\xi_1^i)}{M_i^\alpha} \quad \text{subject to} \quad \sum_{i=1}^{4} M_i \leq N. \tag{6.18}$$

As a result, the optimal M_i is large if the product $p^i\gamma_1(\xi_1^i)$ is large, and conversely, M_i is low if $p^i\gamma_1(\xi_1^i)$ is low. In particular, if the risk forecast ξ_1^i almost never occurs (i.e., $p^i \simeq 0$) or if the conditional rescue efficiency given ξ_1^i is almost not uncertain (i.e., $\gamma_1(\xi_1^i) \simeq 0$), then the product $p^i\gamma_1(\xi_1^i)$ is close to zero and hence the optimal M_i equals one. In this case, it makes intuitively sense that only one node is necessary in $C(n_i)$ to keep a small discretization error and that the remaining nodes should be used elsewhere in the tree where the discretization error is typically larger. We show in Figure 6.5 the optimal tree structures obtained for $\alpha = 1$ and different choices of p^i and $\gamma_1(\xi_1^i)$.

6.4 Simplifications and implementation

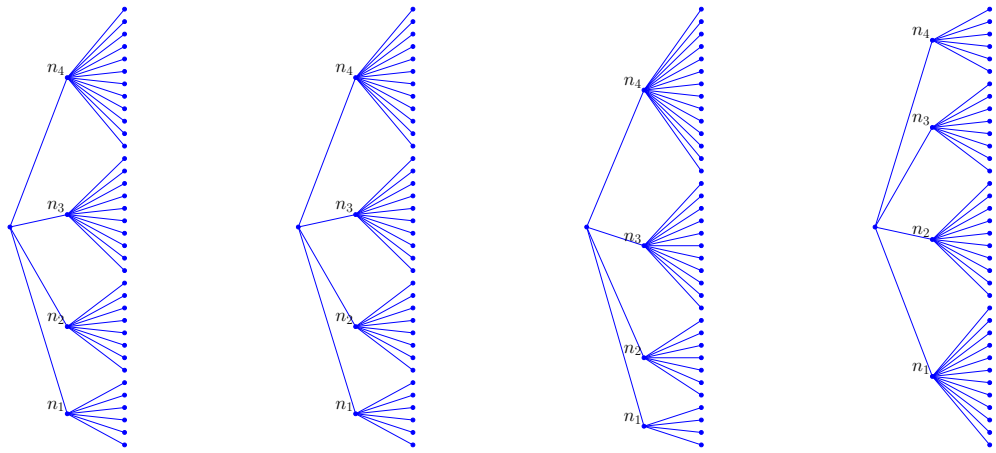
The implementation of our approach is facilitated under the following condition, which we assume to hold throughout this section:

Condition 6.4.1. The stochastic process $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ can be transformed into a *stagewise independent* stochastic process $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$ such that $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ is adapted to the natural filtration of $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$. Moreover, there is a numerical integration method that generates points and weights for the process $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$, denoted specifically by $(\mathcal{P}_*, \mathcal{W}_*)$ throughout this section, that satisfy

$$\mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(n)}, \mathcal{W}_*^{C(n)}) = f_{t(n)}(|C(n)|), \quad \text{for every } n \in \mathcal{N} \setminus \mathcal{N}_T, \tag{6.19}$$

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

The first part of the condition guarantees that there exists a stagewise independent stochastic process $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$ (i.e., $\boldsymbol{\epsilon}_t$ is independent of $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_{t-1})$ for every $t = 1, \ldots, T$) of arbitrary dimension such that $\boldsymbol{\xi}_t = \phi_t(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_t)$ for some measurable mapping ϕ_t for each



(a) $M = (9, 9, 9, 9)$ $(\gamma_1(\xi_1^i) = 1,$ $p^i = \frac{1}{4})$

(b) $M = (6, 8, 10, 12)$ $(\gamma_1(\xi_1^i) = i,$ $p^i = \frac{1}{4})$

(c) $M = (4, 7, 11, 14)$ $(\gamma_1(\xi_1^i) = i^2,$ $p^i = \frac{1}{4})$

(d) $M = (12, 10, 8, 6)$ $(\gamma_1(\xi_1^i) = 1,$ $p^1 = \frac{4}{10},$ $p^2 = \frac{3}{10},$ $p^3 = \frac{2}{10},$ $p^4 = \frac{1}{10})$

Figure 6.5 Tree structures given by (6.18) for 36 scenarios and $\alpha = 1$.

$t = 1, \ldots, T$. This condition is often satisfied in practice as the simulation of a stochastic process usually boils down to the transformation of several independent random variables. For example, a discrete-time Wiener process is obtained from a stagewise independent process by the mapping $\boldsymbol{\xi}_t = \sum_{i=0}^{t} \boldsymbol{\epsilon}_i$ with $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T \sim \mathcal{N}(0, 1)$. This first part is useful for the implementation of our approach as it is typically easier to build a scenario tree for the process $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$, and then map it to a scenario tree for $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ using the non-anticipative mappings $\phi_1, \ldots, \phi_T$.

The second part of the condition guarantees that the node-n demerit decreases at the same rate for all nodes n at the same depth. The fact that the decreasing rate does not depend on the history up to node n makes sense as we are concerned with the discretization of a stagewise independent process. This condition also implies that the decreasing rate is only a function of the number of child nodes at n. This allows us to focus only on the problem of finding the appropriate branching of the tree structure and let the problem of finding the actual values of the points and weights be determined by the numerical integration method. Lastly, the fact that we consider general decreasing functions $f_0, \ldots, f_{T-1}$ ensures that we cover a great deal of numerical integration methods. The interest of a decision-maker who intends to generate efficient scenario trees should obviously go toward methods such that each f_t decreases as fast as possible. A typical family of decreasing functions encountered in

practice is given by $f_t(x) = A_t/x^{\alpha_t}$ where $\alpha_t > 0$ is the *rate of convergence* and $A_t > 0$.

We now illustrate the implementation of our approach on three types of MSOPs. The first type concerns problems where the guidance functions vary across stages but are constant over the distribution support at any given stage. The second type deals with the more general case of guidance functions that depend solely on the realizations at the current stage. The third type deals with guidance functions that depend on the whole history of realizations in a multiplicative fashion. We show in these three examples that the implementation of our approach can be fairly straightforward and that the computational time to compute suitable scenario trees is not prohibitive.

Finally, it is important to keep in mind that the implementation is not restricted to these examples and that other MSOPs with potentially more complex guidance functions can also be addressed.

6.4.1 Constant guidance functions

Consider a set Γ_{ind} of guidance functions that do not depend on past realizations:

$$\Gamma_{\text{ind}} := \{\gamma_0, \gamma_1, \ldots, \gamma_{T-1}\} \subset \mathbb{R}_{\geq 0}. \tag{6.20}$$

The figure of demerit of the scenario tree $(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*)$ with the set Γ_{ind} is then simplified as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{ind}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \, \gamma_{t(n)} \, \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(n)}, \mathcal{W}_*^{C(n)}). \tag{6.21}$$

Since the guidance functions (6.20) give different importance to different stages, but give the same importance to all realizations at the same stage, the minimization of the figure of demerit (6.21) will naturally yield a *symmetrical* tree structures with a suitable *bushiness* $b = (b_0, \ldots, b_{T-1})$. We now show how the values of $\gamma_0, \ldots, \gamma_{T-1}$ determine the optimal bushiness.

Before stating the main result, let us differentiate two ways of generating the discretization points and weights when one is given a symmetrical structure. In the first way, the points $\mathcal{P}_*$ and weights $\mathcal{W}_*$ are designed such that there are exactly b_t different sub-scenario trees with roots at stage t. This condition leads to the recombining scenario tree that we called *mesh* in Section 6.2.3, which has a number of nodes equal to $1 + \sum_{t=0}^{T-1} b_t$ that grows linearly with increase of the number of stages. The way the scenario tree is recombined to yield a mesh was described in Figure 6.4. In the second way, no such condition exists on $\mathcal{P}_*$ and $\mathcal{W}_*$, and hence the scenario tree is almost certainly not recombining. In this case, the number of

scenario is $\prod_{t=0}^{T-1} b_t$, which grows exponentially with increase of the number of stages.

The following proposition show how to compute the bushiness of lowest demerit for the two types of scenario trees described above:

Proposition 6.4.1. *Suppose that $\mathcal{W}_*$ is normalized. Then:*

(a) the symmetrical tree structure of lowest demerit (6.21) *with at most N scenarios is the one with the bushiness given by the optimal solution of*

$$\min_{b=(b_0,\ldots,b_{T-1})\in\mathbb{N}_{>0}^T} \sum_{t=0}^{T-1} f_t(b_t)\gamma_t \quad \textit{subject to} \quad \prod_{t=0}^{T-1} b_t \leq N. \tag{6.22}$$

(b) the mesh of lowest demerit (6.21) *with at most N nodes is the one with the bushiness given by the optimal solution of*

$$\min_{b=(b_0,\ldots,b_{T-1})\in\mathbb{N}_{>0}^T} \sum_{t=0}^{T-1} f_t(b_t)\gamma_t \quad \textit{subject to} \quad \sum_{t=0}^{T-1} b_t \leq N-1. \tag{6.23}$$

Proof. It follows directly from the equality (6.19) and the fact that the structure $\mathcal{T}$ is symmetrical that

$$\mathcal{M}(\mathcal{T},\mathcal{P}_*,\mathcal{W}_*;\Gamma_{\text{ind}}) = \sum_{n\in\mathcal{N}\setminus\mathcal{N}_T} W^n\,\gamma_{t(n)}\,f_{t(n)}(|C(n)|) \tag{6.24}$$

$$= \sum_{t=0}^{T-1} f_t(b_t)\gamma_t \sum_{n\in\mathcal{N}_t} W^n \tag{6.25}$$

$$= \sum_{t=0}^{T-1} f_t(b_t)\gamma_t, \tag{6.26}$$

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 $t = 0,\ldots,T-1$. This proves the objective function in (a) and (b). The two constraints in (6.22) and (6.23) follow from the discussion above about the number of scenarios of the scenario tree and the number of nodes of the mesh. □

We display in Figures 6.6 and 6.7 the tree structures of lowest demerit for different choices of f_t and Γ_{ind}. The integer program (6.22) is solved to optimality by enumerating all possible bushinesses. This solution procedure is computationally feasible because symmetrical tree structures do not generally have many stages due to their exponential growth as the number of stages increases (or if they do have many stages, then most branching coefficients b_t necessarily equal 1). As for the integer program (6.23), since the number of stages of the mesh can be large, the computational cost of enumerating all bushinesses can be prohibitive,

hence we solve it by relaxing the integrity constraint and by rounding-off the optimal solution to the nearest integer solution. As far as the program (6.23) is concerned, we observe that doing so yields a solution close to the optimal integer one.

In the case where each $f_t(\cdot)$ is a strictly convex decreasing function (as holds in particular if $f_t(b_t) = A_t/(b_t)^{\alpha_t}$ with positive α_t and A_t), the objective function in (6.23) is strictly convex in b, and hence the continuous relaxation of (6.23) admits a unique minimizer in the continuous space $[1, \infty)^T$. In particular, if $\alpha_t = \alpha > 0$ and $A_t = 1$ for all $t = 0, \ldots, T-1$, the continuous relaxation of (6.23) have an analytically derivable minimizer given by

$$b_t^* = (N-1)\frac{\gamma_t^{\frac{1}{\alpha+1}}}{\sum_{i=0}^{T-1} \gamma_i^{\frac{1}{\alpha+1}}}, \qquad \text{for all } t = 0, \ldots, T-1. \tag{6.27}$$

(Note that it is always possible to assume that the constant A_t equals 1 by including the original value of A_t into γ_t, i.e., by formally setting $\gamma_t \leftarrow \gamma_t A_t$ and $A_t \leftarrow 1$.)

6.4.2 Current-stage dependent guidance functions

Consider a set Γ_{cs} of guidance functions that depend solely on the current-stage realization:

$$\Gamma_{\text{cs}} := \{\gamma_0(\epsilon_0), \gamma_1(\epsilon_1), \ldots, \gamma_{T-1}(\epsilon_{T-1})\}. \tag{6.28}$$

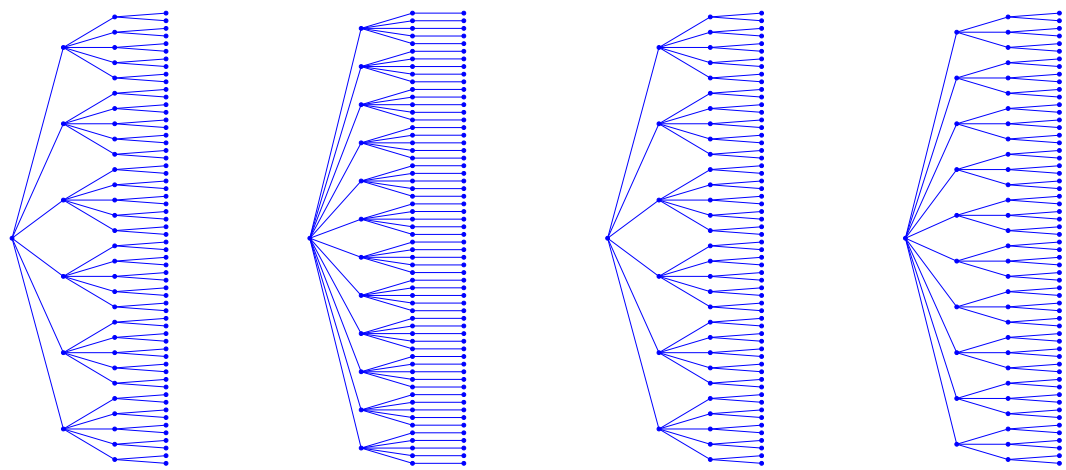
The figure of demerit of the scenario tree $(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*)$ with the set Γ_{cs} takes the simplified form

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\epsilon^n) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(n)}, \mathcal{W}_*^{C(n)}), \tag{6.29}$$

where $\mathcal{P}_* = \{\epsilon^n : n \in \mathcal{N}\}$ and $\mathcal{W}_* = \{w^n : n \in \mathcal{N} \setminus \{n_0\}\}$ are the sets of discretization points and weights generated by the numerical integration method of Condition 6.4.1. Unlike the setting of the previous section, it is not straightforward to find the scenario tree minimizing (6.29) because of the explicit dependence on ϵ^n. As explained at the end of Section 6.3, an algorithmic procedure to the minimization of (6.29) 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 demerit. The following result provides a necessary and sufficient condition for $(\mathcal{P}_*, \mathcal{W}_*)$ to be a minimizer of (6.29) at fixed $\mathcal{T}$:

Proposition 6.4.2. *Suppose that $\mathcal{W}_*$ is standardized. Then the points and weights $(\mathcal{P}_*, \mathcal{W}_*)$ minimize the figure of demerit* (6.29) *at fixed structure $\mathcal{T}$ if and only if:*

$$|C(m)| \leq |C(n)| \iff \gamma_{t(m)}(\epsilon^m) \leq \gamma_{t(n)}(\epsilon^n), \quad \textit{whenever } p(n) = p(m) \notin \mathcal{N}_{T-1}. \tag{6.30}$$



(a) $b = (6, 5, 2)$ ($f_t(x) = x^{-1}$, $\gamma_t = T - t$)

(b) $b = (12, 5, 1)$ ($f_t(x) = x^{-1/2}$, $\gamma_t = T - t$)

(c) $b = (6, 5, 2)$ ($f_t(x) = x^{-1}$, $\gamma_t = (t+1)^{-1}$)

(d) $b = (10, 3, 2)$ ($f_t(x) = x^{-1/2}$, $\gamma_t = (t+1)^{-1}$)

Figure 6.6 Tree structures given by (6.22) for 4 stages and 60 scenarios.

Proof. Using the equality (6.19) in Condition 6.4.1, using also the decomposition

$$\mathcal{N} \setminus \mathcal{N}_T = \{n_0\} \cup \Big(\cup_{t=0}^{T-2} \cup_{n \in \mathcal{N}_t} \cup_{m \in C(n)} \{m\} \Big), \tag{6.31}$$

and the relation $W^m = W^n |C(n)|^{-1}$ that holds for all $m \in C(n)$ (cf. (N6) and (N9), Section 6.2), the figure of demerit (6.29) can be written as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \gamma_0(\epsilon^{n_0})\, f_0(|C(n_0)|) \tag{6.32}$$

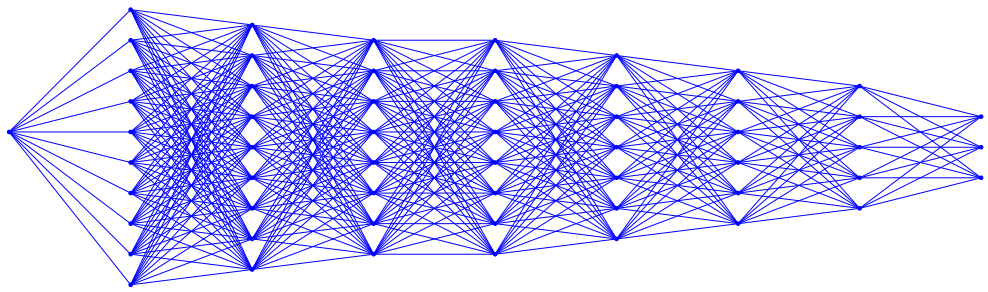
$$+ \sum_{t=0}^{T-2} \sum_{n \in \mathcal{N}_t} \frac{W^n}{|C(n)|} \sum_{m \in C(n)} \gamma_{t(m)}(\epsilon^m) f_{t(m)}(|C(m)|). \tag{6.33}$$

Since (6.19) holds regardless of the way the points $\mathcal{P}_*^{C(n)} = \{\epsilon^m : m \in C(n)\}$ are assigned to the nodes in $C(n)$, it follows that permuting the points inside $C(n)$, i.e., setting

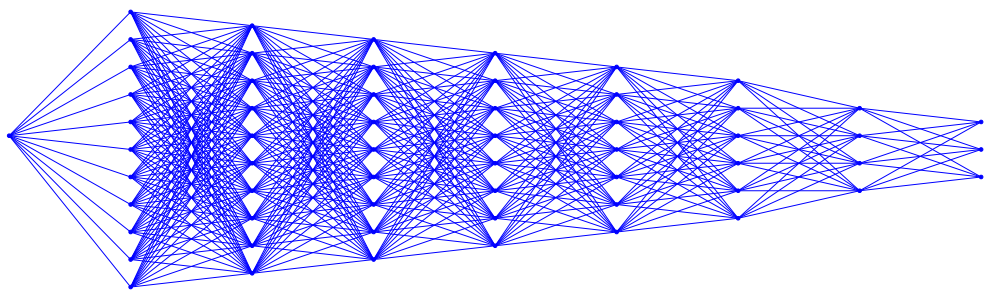
$$\{\epsilon^{\sigma(m)} : m \in C(n)\}, \quad \text{with } \sigma : C(n) \to C(n) \text{ a permutation}, \tag{6.34}$$

provides the only leverage to minimize the figure of demerit.

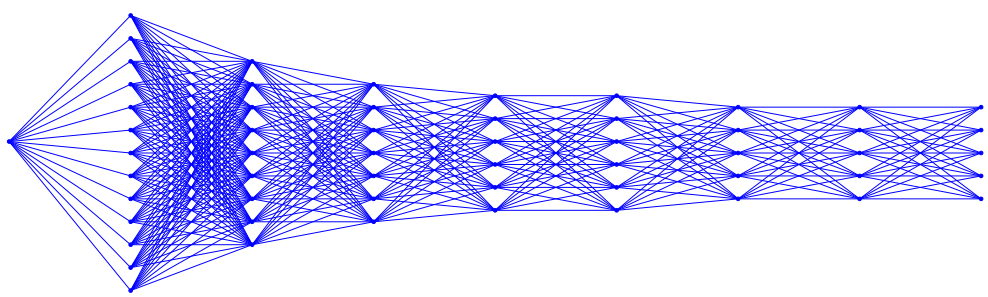
Specifically, it follows from (6.32)-(6.33) that the optimal permutation $\sigma^* : C(n) \to C(n)$ at



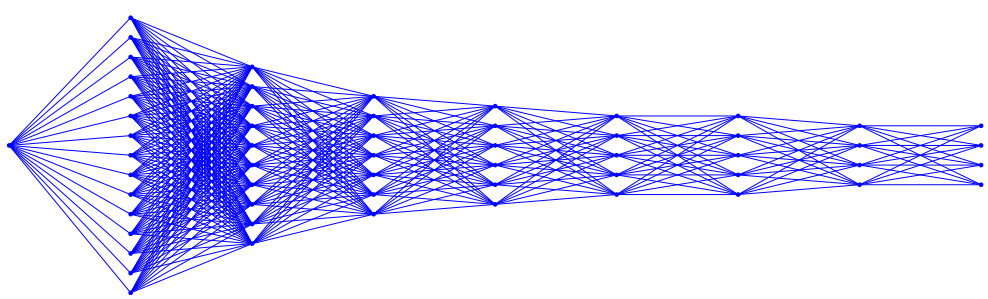
(a) $b = (10, 9, 8, 8, 7, 6, 5, 3)$ ($f_t(x) = x^{-1}$ and $\gamma_t = T - t$)



(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 6.7 Recombining tree structures (meshes) given by (6.23) for 9 stages and 57 nodes.

each node $n \in \mathcal{N} \setminus (\mathcal{N}_T \cup \mathcal{N}_{T-1})$ is the one that minimizes

$$\sigma \mapsto \sum_{m \in C(n)} \gamma_{t(m)}(\epsilon^{\sigma(m)}) f_{t(m)}(|C(m)|). \tag{6.35}$$

It is easy to show that for any $x_i, y_i \geq 0$, $i = 1, \ldots, N$, the function $\sigma \mapsto \sum_{i=1}^{N} x_{\sigma(i)} y_i$ is minimized by the permutation $\sigma^* : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that:

$$x_{\sigma^*(i)} \leq x_{\sigma^*(j)} \Longleftrightarrow y_i \geq y_j, \quad \text{for every } i, j = 1, \ldots, N. \tag{6.36}$$

The assertion (6.30) follows directly from (6.36) and the fact that $f_{t(m)}(\cdot)$ is monotonically decreasing. □

The interpretation of the necessary and sufficient condition (6.30) is quite intuitive: Consider two nodes n and m such that $p(n) = p(m) \notin \mathcal{N}_{T-1}$, i.e., n and m have the same parent node and are at depth $t < T$. If $\gamma_t(\epsilon^n)$ has a larger value than $\gamma_t(\epsilon^m)$, then the variability of the recourse function at stage $t+1$ is expected to be greater at node n than at node m. Thus, n must have more child nodes than m to reduce the integration error at stage $t+1$.

The implementation of Proposition 6.4.2 to find the scenario tree of lowest demerit is described in Figure 6.8

The algorithm starts in Step 1 by iterating over tree structures. An exhaustive iteration is computationally prohibitive unless the MSOP has a small number of stages and the structure is restricted to a fairly small number of scenarios. A more reasonable approach for problems with many stages or scenarios consists in partly exploring the space of tree structures by employing an heuristic approach such as the *variable neighborhood search* (VNS); see Mladenović et Hansen (1997) and Hansen *et al.* (2019). In Step 2, the algorithm assigns the discretization points to the appropriate nodes in accordance with Proposition 6.4.2 in order to minimize the figure of demerit of the current iteration 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 $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ are computed from those of $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$ using the transformation $\boldsymbol{\xi}_t = \phi_t(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_t)$ whose existence is guaranteed under Condition 6.4.1.

Example 6.4.1. Consider the following set of guidance functions:

$$\gamma_0 = 1 \quad \text{and} \quad \gamma_t(\epsilon_t) = 2\epsilon_t, \quad \text{for } t = 1, \ldots, T-1, \tag{6.41}$$

- **Inputs:**
  - set Γ_{cs} of guidance functions satisfying (6.28);
  - numerical integration method satisfying Condition 6.4.1.
- **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 $t = 0, \ldots, T-2$ and node $n \in \mathcal{N}_t$:
  - Set $N := |C(n)|$ and index the nodes $C(n) = \{m_1, \ldots, m_N\}$ such that
  $$|C(m_1)| \leq |C(m_2)| \leq \cdots \leq |C(m_N)|. \tag{6.37}$$
  - Generate N discretization points $\{\epsilon^{m_i} : i = 1, \ldots, N\}$ of $\boldsymbol{\epsilon}_{t+1}$ and index each point such that:
  $$\gamma_{t+1}(\epsilon^{m_1}) \leq \gamma_{t+1}(\epsilon^{m_2}) \leq \cdots \leq \gamma_{t+1}(\epsilon^{m_N}). \tag{6.38}$$
  - Compute (6.35) for the optimal permutation:
  $$v^n := \frac{1}{N} \sum_{i=1}^{N} \gamma_{t+1}(\epsilon^{m_i}) f_{t+1}(|C(m_i)|). \tag{6.39}$$
- **Step 3:** Compute the figure of demerit using (6.32)-(6.33):
$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{cs}}) = \gamma_0(\epsilon^{n_0})\, f_0(|C(n_0)|) + \sum_{t=0}^{T-2} \sum_{n \in \mathcal{N}_t} W^n v^n. \tag{6.40}$$
- **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 $\zeta^m := \phi_{t(n)+1}(\epsilon^{\cdot m})$ for each $m \in C(n)$;
  - otherwise: generate $|C(n)|$ discretization points $\{\epsilon^m : m \in C(n)\}$ of $\boldsymbol{\epsilon}_T$ and set $\zeta^m := \phi_T(\epsilon^{\cdot m})$ for each $m \in C(n)$.

Figure 6.8 Algorithm to compute the lowest demerit scenario tree in the setting of current-stage dependent guidance functions.

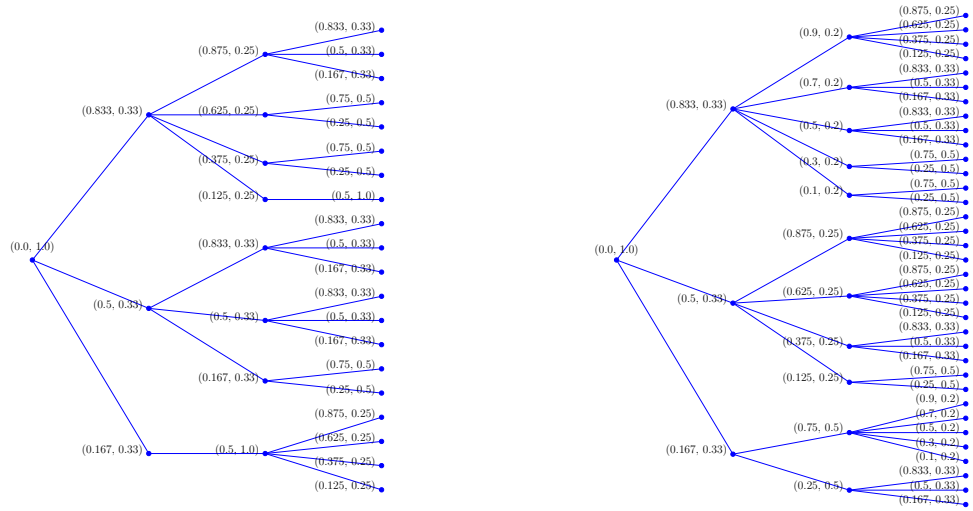
where for simplicity each random variable $\boldsymbol{\epsilon}_t$ follows a uniform distribution $U[0,1]$ and $\epsilon_0 = 0$. The normalization constant of $\gamma_t(\cdot)$ is chosen so that $\mathbb{E}[\gamma_t(\boldsymbol{\epsilon}_t)] = 1 = \gamma_0$ for all $t = 1, \ldots, T$, which means that the guidance functions (6.41) give the same importance (on average) to all stages. We choose this normalization for the sake of the illustration so that the output tree structures will have branching equally dense throughout the stages. At each given stage $t \geq 1$, $\gamma_t(\cdot)$ varies linearly over the distribution support of $\boldsymbol{\epsilon}_t$, hence a larger realization of $\boldsymbol{\epsilon}_t$ will be given a greater importance (as far as the conditional variability of the recourse functions at stage $t+1$ given ϵ_t is concerned).

We compute the tree structures of lowest demerit by implementing the above algorithm in Python 3.6.2 on a Linux machine (Intel Xeon X5472 @ 3.00GHz). We report the results for both the exhaustive search and the heuristic VNS. The exhaustive search is limited to structures with a small number of stages and scenarios due the exponential growth of the search space as either one of these two parameters increases. Specifically, within 10 minutes the algorithm is able to search exhaustively the space of structures with 4 stages and 20 scenarios. Beyond that point the computational cost starts to be prohibitive. For instance, for 25 scenarios the computational time is multiplied by 20, and for 30 scenarios it is multiplied by 400. However, by restricting the search to tree structures that have at least 2 branches emerging from each node, the algorithm is able to extend the number of scenarios to 35 within 10 minutes. The two corresponding scenario trees, with 20 and 35 scenarios, are displayed in Figure 6.9. We note that introducing a lower bound on the number of branches per node not only has the advantage of reducing the search space, it also has a practical interest in portfolio selection problems where *arbitrage-free* scenario trees must have a branching coefficient bounded from below by the number of non-redundant assets; see, e.g., Geyer *et al.* (2010).

As for the heuristic VNS, the neighborhood of a tree structure is defined as all the structures obtained from it by splitting a node or merging two nodes (we only split or merge nodes at depth $t < T$ so that the number of scenarios is constant). For 6 stages and 50 scenarios, the scenario tree found by VNS within 20 minutes is displayed in Figure 6.10. Although the output scenario tree need not be the one of lowest demerit as the VNS approach explores only partly the space, we are confident that it is close to the optimal one since it has a structure that naturally extends that of Figure 6.9(a), which is optimal for 4 stages and 20 scenarios.

All the above scenario trees are computed using evenly-spaced lattice point sets

$$\mathcal{P}_* = \left\{ \frac{i+0.5}{N} : i = 0, \ldots, N-1 \right\} \subset [0,1), \tag{6.42}$$



(a) (b)

Figure 6.9 Lowest demerit scenario trees with 4 stages computed by exhaustive search. The scenario tree (a) has 20 scenarios and no restriction on the branching. The scenario tree (b) has 35 scenarios and at least 2 branches emerging from each node. Each couple (ϵ^n, w^n) is displayed next to the corresponding node.

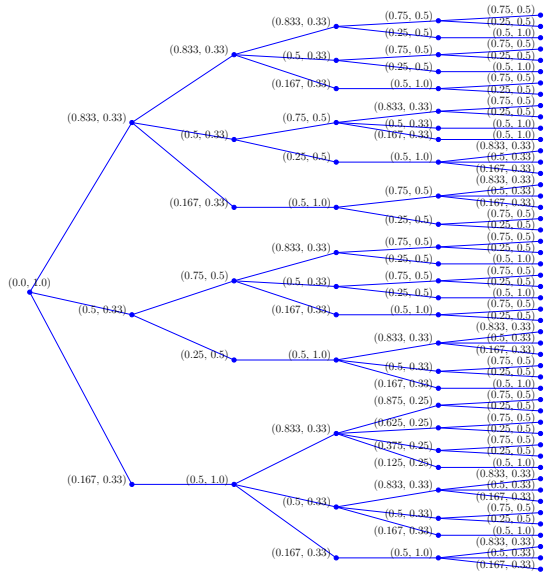


Figure 6.10 Scenario tree computed by VNS for 6 stages and 50 scenarios.

standardized weights $\mathcal{W}_*$ (cf. (N9), Section 6.2.2), and decreasing functions $f_t(x) = x^{-1}$ as lattice point sets in $[0, 1)$ typically achieve a rate of convergence close to one; see, e.g., Dick *et al.* (2013). By randomizing the point sets, we can build several scenario trees from each output tree structure $\mathcal{T}$ found by exhaustive or heuristic search. For example, by randomly shifting the lattice point sets as follows:

$$\mathcal{P}_{*,u} := \left\{ \text{frac}\left(\frac{i+u}{N}\right) : i = 0, \ldots, N-1 \right\}, \quad \text{with } u \sim U[0,1], \tag{6.43}$$

where frac$(\cdot)$ denotes the fractional part function, we can generate as many replications $(\mathcal{T}, \mathcal{P}_{*,u}, \mathcal{W}_*)$ of the scenario tree $(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*)$ as we want, and use them to solve several time the STAP to obtain a statistic of the optimal-value error (e.g., the mean and variance). As the point sets are randomized, we need to permute the points in accordance with Proposition 6.4.2 to ensure that $(\mathcal{P}_{*,u}, \mathcal{W}_*)$ still minimize the demerit at fixed structure $\mathcal{T}$. We display in Figure 6.11 three possible replications of the scenario tree of Figure 6.9(a).

Finally, we consider a more general set of guidance functions of the form:

$$\gamma_0 = \delta_0 \quad \text{and} \quad \gamma_t(\epsilon_t) = \delta_t \epsilon_t^{\beta_t}, \quad \text{for } t = 1, \ldots, T-1, \tag{6.44}$$

with $\delta_t \geq 0$ and $\beta_t \geq 0$ for all t, and we discuss the effect of tuning δ_t and β_t on the lowest demerit scenario tree.

The values of β_t act on the uniformity of the branches emerging from the nodes at depth $t \geq 1$. For instance, if $\beta_t = 0$, then $\gamma_t(\cdot)$ is constant over the support of $\boldsymbol{\epsilon}_t$, and hence all nodes at depth t have the same number of emerging branches regardless of their associated point ϵ^n. Thus, if $\beta_t = 0$ at all stages, then the output tree structure is symmetrical and its bushiness is determined by the values of $\delta_0, \ldots, \delta_{T-1}$ (we then fall into the setting of the previous section). As β_t moves away from zero, the branches emerging from the nodes at depth t are less and less uniform. Indeed, as β_t increases, a node n with realization ϵ^n close to one is given more and more importance as compared with a node m with ϵ^m close to zero (because $\gamma_t(\epsilon^n)/\gamma_t(\epsilon^m) = (\epsilon^n/\epsilon^m)^{\beta_t}$ increases as β_t increases), which results in more and more branches emerging from n at the expense of m.

As for the values of δ_t, they act alongside β_t on the average branching density at depth t. Specifically, they both act on the values of γ_0 and $\mathbb{E}[\gamma_t(\boldsymbol{\epsilon}_t)]$, which determine the average branching density at each stage. Typically, if $\gamma_0 \geq \mathbb{E}[\gamma_1(\boldsymbol{\epsilon}_1)] \geq \cdots \geq \mathbb{E}[\gamma_{T-1}(\boldsymbol{\epsilon}_{T-1})]$, then the average branching is denser as we approach the initial stage.

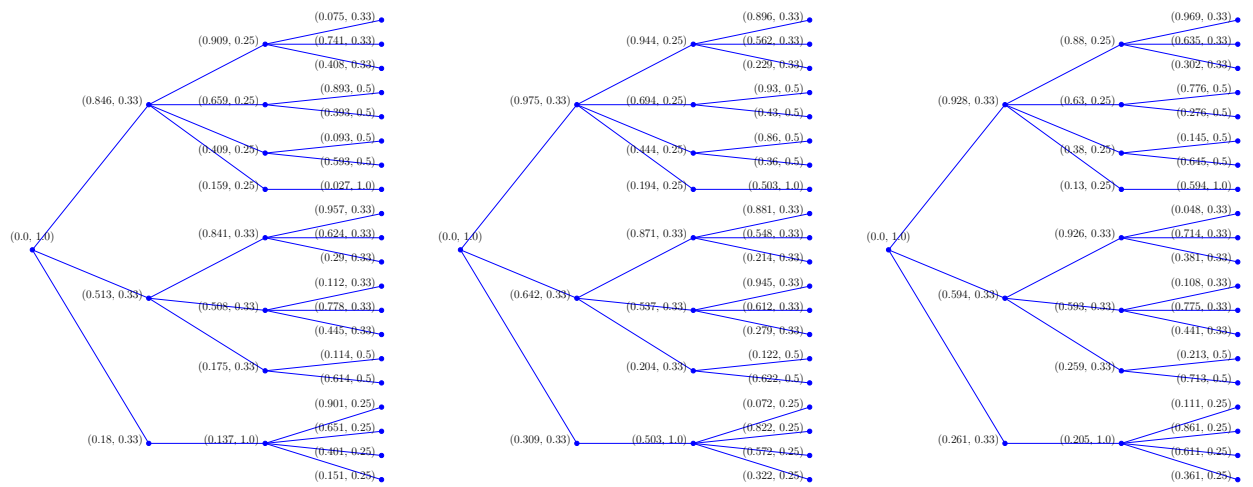


Figure 6.11 Three replications $(\mathcal{T}, \mathcal{P}_{*,u}, \mathcal{W}_*)$ of the scenario tree $(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*)$ of Figure 6.9(a) obtained by randomly shifting the point sets.

6.4.3 All-history dependent guidance functions

Recall that the figure of demerit of a scenario tree $(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*)$ with a general set Γ of guidance functions is given by

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \gamma_{t(n)}(\epsilon^{\cdot \cdot n}) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(n)}, \mathcal{W}_*^{C(n)}), \tag{6.45}$$

where $\mathcal{P}_* = \{\epsilon^n : n \in \mathcal{N}\}$ and $\mathcal{W}_* = \{w^n : n \in \mathcal{N} \setminus \{n_0\}\}$ are the sets of discretization points and weights of $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$ generated by the numerical integration method of Condition 6.4.1. Finding the scenario tree that minimizes (6.45) is computationally cumbersome, even under the simplification (6.19), since each (ϵ^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 (6.45) exists if we consider a set Γ_{prod} of guidance functions of the *product* form:

$$\gamma_0(\epsilon_0) = 1 \quad \text{and} \quad \gamma_t(\epsilon_0, \ldots, \epsilon_t) = \prod_{i=1}^{t} \rho_i(\epsilon_i), \quad \text{for every } t = 1, \ldots, T-1, \tag{6.46}$$

for some functions $\rho_1, \ldots, \rho_{T-1} : \mathbb{R}^d \to [0, \infty)$. With the guidance functions of the product form, the figure of demerit is written as

$$\mathcal{M}(\mathcal{T}, \mathcal{P}_*, \mathcal{W}_*; \Gamma_{\text{prod}}) = \sum_{n \in \mathcal{N} \setminus \mathcal{N}_T} W^n \Big(\prod_{m \in (n_0, n]} \rho_{t(m)}(\epsilon^m) \Big) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(n)}, \mathcal{W}_*^{C(n)}). \tag{6.47}$$

This form can be minimized recursively from the tree leaves to the root node. The following proposition provides a necessary and sufficient condition for $(\mathcal{P}_*, \mathcal{W}_*)$ to be a minimizer of (6.47) at fixed $\mathcal{T}$.

Proposition 6.4.3. *The discretization points and weights $(\mathcal{P}_*, \mathcal{W}_*)$ minimize the figure of demerit* (6.47) *at fixed structure $\mathcal{T}$ if and only if:*

$$\mathcal{M}(m) \geq \mathcal{M}(n) \Longleftrightarrow w^m \rho_{t(m)}(\epsilon^m) \leq w^n \rho_{t(n)}(\epsilon^n), \quad \textit{whenever } p(n) = p(m) \notin \mathcal{N}_{T-1}, \tag{6.48}$$

where $\mathcal{M}(n)$ is the figure of demerit of the sub-scenario tree $(\mathcal{T}(n), \mathcal{P}_*(n), \mathcal{W}_*(n))$ with Γ_{prod} defined as

$$\mathcal{M}(n) = \frac{1}{W^n} \sum_{m \in \mathcal{N}(n) \setminus \mathcal{N}_T} W^m \Big(\prod_{o \in (n, m]} \rho_{t(o)}(\epsilon^o) \Big) \mathcal{D}_{\mathcal{F}}(\mathcal{P}_*^{C(m)}, \mathcal{W}_*^{C(m)}). \tag{6.49}$$

Proof. It follows from Lemma 3.4(a) in Keutchayan *et al.* (2018a) that the figures of demerit of the sub-scenario trees can be written recursively (from the tree leaves to the root node) as

$$\mathcal{M}(n) = f_{t(n)}(|C(n)|), \quad \text{for all } n \in \mathcal{N}_{T-1}, \tag{6.50}$$

and

$$\mathcal{M}(n) = f_{t(n)}(|C(n)|) + \sum_{m \in C(n)} w^m \rho_{t(m)}(\epsilon^m) \mathcal{M}(m), \quad \text{for all } n \in \mathcal{N} \setminus (\mathcal{N}_{T-1} \cup \mathcal{N}_T), \tag{6.51}$$

The recursion (6.50)-(6.51) is written using the simplification (6.19). Since (6.19) holds regardless of the way the points and weights $\{(\epsilon^m, w^m) : m \in C(n)\}$ are assigned to the nodes in $C(n)$, it follows from (6.50)-(6.51) that the figure of demerit of the whole scenario tree, $\mathcal{M}(n_0)$, is minimized by finding for each node $n \in \mathcal{N} \setminus (\mathcal{N}_T \cup \mathcal{N}_{T-1})$ the permutation $\sigma^* : C(n) \to C(n)$ that minimizes

$$\sigma \mapsto \sum_{m \in C(n)} w^{\sigma(m)} \rho_{t(m)}(\epsilon^{\sigma(m)}) \mathcal{M}(m). \tag{6.52}$$

Using again the fact that for any $x_i, y_i \geq 0$, $i = 1, \ldots, N$, the function $\sigma \mapsto \sum_{i=1}^N x_{\sigma(i)} y_i$

is minimized by the permutation $\sigma^* : \{1, \ldots, N\} \to \{1, \ldots, N\}$ that satisfies (6.36), we conclude that the optimal permutation of the points and weights is the one that satisfies (6.48). $\square$

The necessary and sufficient condition (6.48) also has an intuitive interpretation: Consider two nodes n and m such that $p(n) = p(m) \notin \mathcal{N}_{T-1}$, i.e., n and m have the same parent node and are at depth $t < T$. If $w^n \rho_t(\epsilon^n)$ has a larger value than $w^m \rho_t(\epsilon^m)$, then the weighted variability of the recourse function at stage $t+1$ is expected to be greater at node n than at node m, as well as the weighted variability of the recourse functions at all stages afterwards as the term $\rho_t(\epsilon^n)$ appears also in the guidance function $\gamma_{t(o)}(\epsilon^{\cdot o})$ for every node o in the sub-scenario tree rooted at n. Thus, n must be associated with the sub-scenario tree of lower demerit to reduce the integration error.

The implementation of Proposition 6.4.3 to compute the scenario tree of lowest demerit is described in Figure 6.12.

Steps 1, 4, and 5 are identical to the corresponding steps in the algorithm of Section 6.4.2. Steps 2 and 3 are different. 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 (6.50). In Step 3, it assigns the discretization points and weights to the appropriate nodes according to Proposition 6.4.3 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 6.4.2. Consider a stochastic process $(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_T)$ that models a *discrete-time geometric Brownian motion* $\mathrm{GBM}(\mu, \sigma, T)$:

$$\boldsymbol{\xi}_0 = 1 \quad \text{and} \quad \boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_t \exp(\mu - \frac{\sigma^2}{2} + \sigma(\boldsymbol{W}_{t+1} - \boldsymbol{W}_t)), \quad \text{for every } t = 0, \ldots, T-1, \quad (6.56)$$

where $(\boldsymbol{W}_t)_{t \geq 0}$ is a Wiener process, and μ (drift) and $\sigma \geq 0$ (volatility) are constant; see, e.g., Glasserman (2003). This process can be written in the form $\boldsymbol{\xi}_t = \phi(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_t) = \prod_{i=0}^{t} \boldsymbol{\epsilon}_i$ where $(\boldsymbol{\epsilon}_0, \ldots, \boldsymbol{\epsilon}_T)$ is a stagewise independent process with $\boldsymbol{\epsilon}_0 = 1$ and $\boldsymbol{\epsilon}_i = \exp(\mu - \frac{\sigma^2}{2} + \sigma(\boldsymbol{W}_i - \boldsymbol{W}_{i-1}))$ for each $i = 1, \ldots, T$.

We now show that the guidance functions of the product form (6.46) naturally arise for the GBM and then compute the scenario trees suitable to its discretization. Although to specify the set of guidance functions we would need to specify a MSOP, in this example we simply assume that the variability of the recourse functions exactly matches the variability of the stochastic process and that the latter is measured using the standard deviation as proxy.

Denote by $h_t(\boldsymbol{\xi}_t) := \mathbb{E}[\boldsymbol{\xi}_T \,|\, \boldsymbol{\xi}_t]$ the random variable whose conditional expectation given

- **Inputs:**
  - set Γ_{prod} of guidance functions satisfying (6.46);
  - numerical integration method satisfying Condition 6.4.1.
- **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 $n \in \mathcal{N}_{T-1}$:
  - Generate $|C(n)|$ discretization points and weights $\{(\epsilon^m, w^m) : m \in C(n)\}$ of $\boldsymbol{\epsilon}_T$.
  - Set $\mathcal{M}(n) := f_{t(n)}(|C(n)|)$.
- **Step 3:** For each stage $t = T-2, \ldots, 0$ (backward iteration) and node $n \in \mathcal{N}_t$:
  - Set $N := |C(n)|$ and index the nodes $C(n) = \{m_1, \ldots, m_N\}$ such that
  $$\mathcal{M}(m_1) \geq \mathcal{M}(m_2) \geq \cdots \geq \mathcal{M}(m_N). \tag{6.53}$$
  - Generate N discretization points and weights $\{(\epsilon^{m_i}, w^{m_i}) : i = 1, \ldots, N\}$ of $\boldsymbol{\epsilon}_{t+1}$ and index the elements such that:
  $$w^{m_1}\rho_{t+1}(\epsilon^{m_1}) \leq w^{m_2}\rho_{t+1}(\epsilon^{m_2}) \leq \cdots \leq w^{m_N}\rho_{t+1}(\epsilon^{m_N}). \tag{6.54}$$
  - Compute the figure of demerit $\mathcal{M}(n)$ for the optimal permutation using (6.51):
  $$\mathcal{M}(n) = \; f_{t(n)}(|C(n)|) + \sum_{i=1}^{N} w^{m_i}\rho_{t+1}(\epsilon^{m_i})\,\mathcal{M}(m_i). \tag{6.55}$$
- **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 $\zeta^n := \phi_{t(n)}(\epsilon^{\cdot n})$ for each node $n \in \mathcal{N} \setminus \{n_0\}$.

Figure 6.12 Algorithm to compute the lowest demerit scenario tree in the setting of all-history dependent guidance functions.

$(\epsilon_0, \ldots, \epsilon_{t-1})$ is numerically computed at each stage of the scenario tree. The difficulty in approximating this conditional expectation by a finite sum is measured by the conditional standard deviation of $h_t(\boldsymbol{\xi}_t)$ given $(\epsilon_0, \ldots, \epsilon_{t-1})$, hence we define the guidance functions as

$$\gamma_{t-1}(\epsilon_0, \ldots, \epsilon_{t-1}) := \mathrm{Var}(h_t(\boldsymbol{\xi}_t) \,|\, \epsilon_0, \ldots, \epsilon_{t-1})^{1/2}, \qquad \text{for every } t = 1, \ldots, T. \tag{6.57}$$

Since the GBM satisfies $\mathbb{E}[\boldsymbol{\xi}_T \,|\, \xi_t] = \xi_t e^{(T-t)\mu}$ and $\mathrm{Var}(\boldsymbol{\epsilon}_t) = e^{2\mu}(e^{\sigma^2} - 1)$ for all $t = 1, \ldots, T$, the right-hand side of (6.57) can be simplified as

$$\gamma_{t-1}(\epsilon_0, \ldots, \epsilon_{t-1}) = e^{(T-t+1)\mu}(e^{\sigma^2} - 1)^{1/2} \prod_{i=0}^{t-1} \epsilon_i, \qquad \text{for every } t = 1, \ldots, T. \tag{6.58}$$

The product form (6.46) requires that $\gamma_0(\epsilon_0) = 1$. Since in practice the guidance functions are scale-free (cf. discussion after Theorem 6.3.3), 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:

$$\gamma_0(\epsilon_0) = 1 \quad \text{and} \quad \gamma_t(\epsilon_0, \ldots, \epsilon_t) = \prod_{i=1}^{t} \frac{\epsilon_i}{e^{\mu}}, \qquad \text{for every } t = 1, \ldots, T-1. \tag{6.59}$$

It is insightful to analyze (6.59) in comparison with the set of guidance functions (6.41) of the previous example. In both cases, the dependence of $\gamma_t(\cdot)$ in ϵ_t is linear. As a result, for any two nodes n and m at depth t that share the same parent, the ratio $\gamma_t(\epsilon^{\cdot\cdot n})/\gamma_t(\epsilon^{\cdot\cdot m})$ equals ϵ^n/ϵ^m in both cases, which means that the distribution of branches emerging from n and m is governed by the same relation. Additionally, since $\mathbb{E}[\boldsymbol{\epsilon}_t] = e^{\mu}$ for all t for the GBM, in both cases it holds that

$$\gamma_0(\epsilon_0) = \mathbb{E}[\gamma_1(\boldsymbol{\epsilon}_1)] = \cdots = \mathbb{E}[\gamma_{T-1}(\boldsymbol{\epsilon}_{..T-1})], \tag{6.60}$$

which means that the average branching density is constant across stages, i.e., branching at the early stages is considered as important as branching at the late stages (this is true for our particular choice of guidance functions motivated by illustration purposes, not a general rule for scenario trees).

Eventually, the difference between the two forms stems from the fact that (6.59) is a product of all $\epsilon_1, \ldots, \epsilon_t$, while (6.41) depends only on ϵ_t, hence the former features a *persistence* across stage, while the latter is *memoryless*. That is to say, if most of the ϵ_i's for $i < t$ are close to zero, then $\gamma_t(\epsilon_0, \ldots, \epsilon_t)$ given by (6.59) is small regardless of ϵ_t (provided, however, that ϵ_t is not that enormous so as to compensate alone all previous values), whereas $\gamma_t(\epsilon_0, \ldots, \epsilon_t)$ given by (6.41) is blind to the values of ϵ_i for $i < t$.

The effect of the persistence property is visible in Figures 6.13(a) and 6.14(a) where two scenario trees of low demerit are displayed. They are computed using the heuristic VNS for the iteration over the tree structures and a *quasi-Monte Carlo method* and an *optimal quantization method*, respectively, for the discretization of the standard normal distribution. Specifically, for the former the points are generated by transforming the lattice point sets (6.42) through the inverse cumulative ϕ^{-1} of the standard normal distribution as follows:

$$\left\{\phi^{-1}\left(\frac{i+0.5}{N}\right) : i = 0, \ldots, N-1\right\}, \tag{6.61}$$

and by setting standardized weights as it is customary in quasi-Monte Carlo methods. For the latter, the points and weights are obtained by minimizing the Wasserstein distance of order 2; see Pflug et Pichler (2015). Both scenario trees are computed by the algorithmic procedure described above. The iteration over the tree structures is done heuristically by the same VNS approach as that used in Example 6.4.1. Since the space of all tree structures is not explored exhaustively, we have no guarantee that the output scenario tree is the one of *lowest* demerit. However, the fact that the output structure exhibits a certain regularity (which we make obvious by sorting the nodes from the bottom to the top of the figure by increasing values of ζ^n) provides a strong confidence that it is close to optimality. 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 numerical integration methods, we see that the output scenario trees have denser branching at nodes where the GBM takes higher values (in the upper part of the figures), as it is where the future values taken by the process have larger conditional variability. This feature appears even more clearly in Figures 6.13(b) and 6.14(b) where all 60 scenarios of each scenario tree are displayed in a plot where the y-axis has logarithmic scale (the x-axis is linear and represents the stages from 0 to 5). In the logarithmic scale, the stage-t component of $\mathrm{GBM}(\mu, \sigma, T)$ has a normal distribution with mean $(\mu - \sigma^2/2)t$ and variance $\sigma^2 t$. In the case of the parameters $\mu = 1$ and $\sigma = \sqrt{2}$ considered in our example, this corresponds to a stage-t component having a normal distribution with mean 0 and variance $2t$. Thus, although some trajectories directly sampled from the original process would be symmetrical around the axis $y = 0$ when plotted in a log-scale, we see on both figures that the 60 trajectories of the scenario tree exhibit an asymmetry for $t > 1$ driven toward the positive values. Since the trajectories have different probabilities, this asymmetry of trajectories does not imply that the resulting empirical distribution is abnormally biased toward positive values. Indeed, the two bottom-most trajectories in Figure 6.13(b) have probabilities 0.125 and 0.0833, whereas the two top-most ones have a much lower probability of 0.000781, hence there are less trajectories

below the axis $y = 0$ but they typically have greater probabilities. Overall, we observe that the multistage discretization scheme achieved by a scenario tree of low demerit aims at preserving the original shape of the distribution while making the discretization finer at the upper tail of the distribution (where the variability is large) and coarser at the lower tail (where the variability is low).

Finally, as in Example 6.4.1, it is easy to build several random replications of the scenario tree generated by the quasi-Monte Carlo method by keeping the same tree structure $\mathcal{T}$ and weights $\mathcal{W}_*$, and by randomly shifting the points $\mathcal{P}_*$. In the case of the standard normal distribution, this can be done by substituting the deterministic point sets (6.61) by their randomized versions:

$$\left\{\phi^{-1}\left(\text{frac}\left(\frac{i+u}{N}\right)\right) : i = 0, \ldots, N-1\right\}, \quad \text{with } u \sim U[0,1]. \tag{6.62}$$

For each replication the points and weights must be arranged in accordance with (6.48) to ensure that they minimize the demerit at fixed structure.

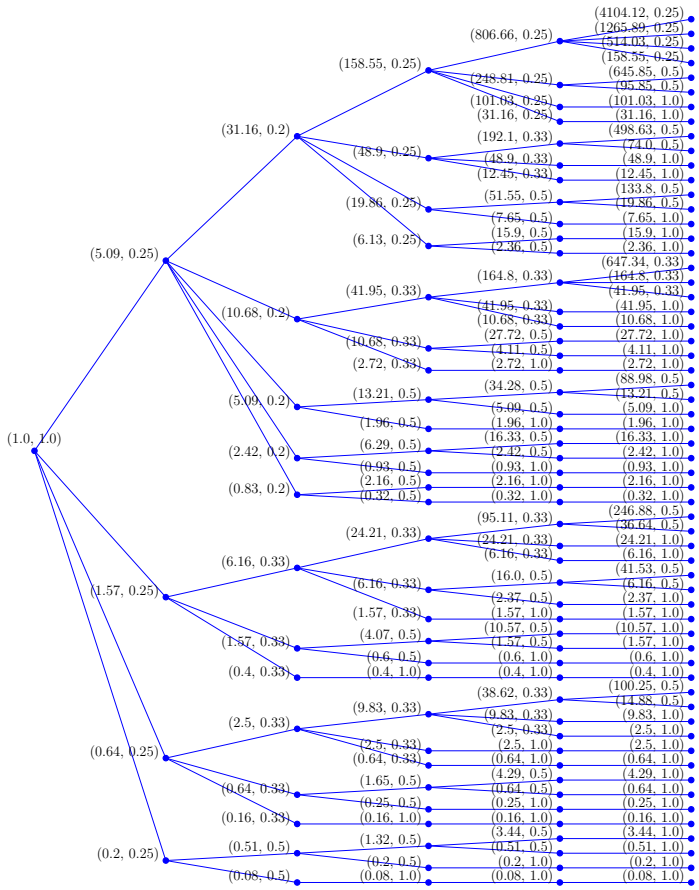
6.5 Conclusion

Building on the results derived in Keutchayan *et al.* (2018a) we have developed a new approach for generating scenario trees. The novelty of this approach is that it takes into account the structure of the MSOP, specifically, the variability of the recourse functions with respect to the random parameters, to design scenario trees better suited to the problem. To account for this variability we have introduced the *guidance functions*, which are defined over the support of the distribution and serve as inputs of the approach. Conceptually, these functions are of great importance as they record quantitatively all relevant information about the problem. Therefore, as far as scenario-tree generation is concerned, two MSOPs can be seen as identical if they have the same set of guidance functions.

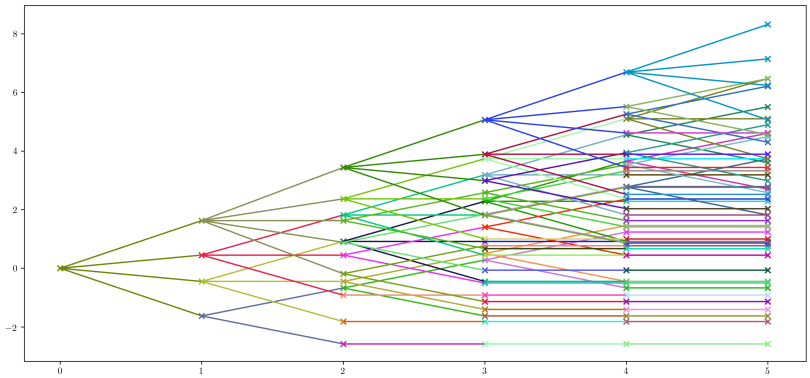
Based on the guidance functions, we have introduced the *figure of demerit* as a measure of suitability between a scenario tree and a problem. The scenario trees considered are general (no assumptions have been made beforehand about the structure, the points, or the weights) and they include as a particular case the *recombining* scenario trees. Recombining scenario trees are particularly relevant for problems with many stages, as their structures enjoy the nice property of not growing exponentially with the number of stages. We have then demonstrated the applicability of the figure of demerit by applying it to different settings of problems. Firstly, in the setting of guidance functions that vary across stages but are constant over the distribution support at each given stage, where it has led to symmetrical

tree structures and meshes of suitable bushinesses. Secondly, in two settings of guidance functions that vary over the distribution supports, where it has led to non-symmetrical tree structures having denser branching where the variability of the recourse functions is higher. The latter was illustrated for example by the discretization of a geometric Brownian motion.

Future works building on the present paper may address, for instance, the following topics: (i) The formalization 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 numerical integration method, randomization in scenario-tree generation allows to derive a confidence interval for the optimality gap; see, e.g., Mak *et al.* (1999) and Shapiro *et al.* (2014). (ii) The search for appropriate numerical integration methods when the MSOP features more than one random parameter per stage. 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 in numerical integration are going toward the generalization of quasi-Monte Carlo methods to non-smooth functions over general domains; see, e.g., Griebel *et al.* (2010, 2013); Hartinger et Kainhofer (2006); Hickernell *et al.* (2004a); Novak *et al.* (2017), and Owen (2006). (iii) The development of exact or heuristic procedures to explore efficiently the space of tree structures in the two algorithms described in Sections 6.4.2 and 6.4.3.

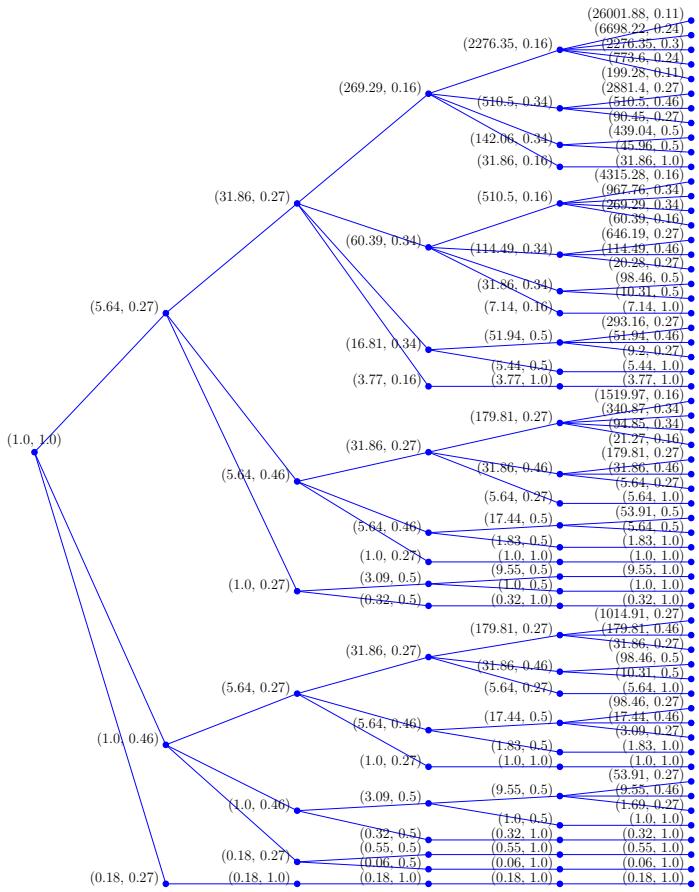


(a) Scenario tree of low figure of demerit for 60 scenarios ($f_t(x) = x^{-1}$)

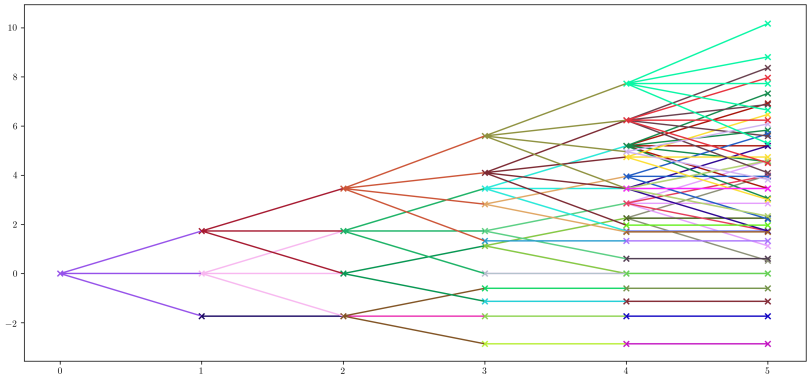


(b) All 60 scenarios of the above scenario tree (y-axis is in log-scale)

Figure 6.13 Scenario tree discretization of GBM($\mu = 1, \sigma = \sqrt{2}, T = 5$) by a *quasi-Monte Carlo* method, specifically, a *rank-1 lattice rule* in one dimension. Each couple (ζ^n, w^n) is displayed next to the corresponding node.



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



(b) All 60 scenarios of the above scenario tree (y-axis is in log-scale)

Figure 6.14 Scenario tree discretization of GBM($\mu = 1, \sigma = \sqrt{2}, T = 5$) by an *optimal quantization method*. Each couple (ζ^n, w^n) is displayed next to the corresponding node.

CHAPITRE 7 ARTICLE 4: PROBLEM-DRIVEN SCENARIO TREES IN MULTISTAGE STOCHASTIC OPTIMIZATION: AN ILLUSTRATION IN OPTION PRICING

Julien Keutchayan[1,2], Michel Gendreau[1,2], Fabian Bastin[3]

[1]Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Canada.

[2]Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Canada.

[3]Département d'Informatique et de Recherche Opérationnelle, Montréal, Canada

Article soumis à la revue *SIAM Journal on Optimization* le 5 novembre 2018.

Abstract. The generation of scenario trees is an important topic in all fields dealing with multistage planning under uncertainty. However, scenario trees tend to grow exponentially as the number of stages increases, which, as of today, limits their application to problems with few stages. Recently, a new framework was developed in Keutchayan *et al.* (2018a,b) to generate more efficient, *problem-driven*, scenario trees. This framework, which is not tied to any particular application, uses a new measure called the *figure of demerit* to assess the suitability between a scenario tree (characterized by its structure, points, and weights) and a multistage problem (characterized by its objective function, constraints, and stochastic process). In the present paper, we demonstrate experimentally the relevance of taking into account the problem's features when building scenario trees, by applying their framework to the problem of pricing Bermudan-Asian call options. We show that problem-driven scenario trees, which have tree structures exploiting the specific payoff's features of Asian options, provide consistently better estimates than scenario trees built the same way but with a structure blind to the problem. Moreover, because their structures match suitably the variability of the recourse functions (the option price) across the stages and scenarios, problem-driven scenario trees appear to reduce the effective number of stages of the problem. On a general standpoint, these results show that the variability of the recourse functions matters a lot when tackling multistage problems using scenario trees. We therefore argue that a significant portion of the work should be done on modeling the distribution of variability across the stages and scenarios, before using a scenario-tree generation approach that will exploit this information.

7.1 Introduction

Problems that involve making optimal decisions sequentially under uncertainty are notoriously hard to solve. Firstly, they generally require a *discretization* (or sampling) step that consists in substituting the original continuous distribution with an approximate one sitting on finitely many points (called *scenarios* in this context). Secondly, the *non-anticipativity* requirement that the decisions must satisfy forces the discretization scheme to describe a sound future out of each scenario, so that decisions are made under uncertainty and do not foresee future events. Because of this requirement, discretization structures must branch out as time moves from the first stage to the last, which typically leads to tree structures growing exponentially in size with the number of stages.

A large class of multistage stochastic optimization problems that has drawn a lot of attention in the literature because of their importance in modern finance are the option pricing problems. An option is a contract that gives the holder the right, but not the obligation, to buy (*call*) or sell (*put*) an underlying asset (stock, currency, commodity, etc.) at a predetermined price (the *strike price*). The exercise decision (whether the holder buys/sells or not) broadly comes in two flavors: for *European-style* options the exercise decision can only be made at the end of the life of the contract (at *maturity*), while for *American-style* options the exercise decision can be made at any time between the origin and maturity. Another exercise style that sits between European and American, hence called Bermudan, can be exercised only at some specific dates. Because numerical methods in multistage stochastic optimization generally see time as discrete, the Bermudan exercise style is the one that we consider in this paper. For a more detailed description about option contracts we refer to Hull (2017).

The problem of pricing options illustrates perfectly the two difficulties mentioned in the first paragraph. Since European options involve making decisions only at maturity, when all information about the underlying asset is known, the difficulty in pricing them lies essentially in finding good discretization or sampling methods to generate a fan of scenarios and compute the option payoffs. However, because of their early exercise feature, American/Bermudan options involve one extra layer of difficulty in that they additionally require the discretization scheme to be non-anticipative. Indeed, the exercise decision must take into account solely the information available at the time of the decision and must not foresee the evolution of the underlying asset. In other words, the decision process must be adapted to the filtration of the stochastic asset price.

Beyond the exercise feature, options are characterized by two other features: the stochastic model used for the asset price and the type of option payoff function. The whole difficulty in

pricing options relies on both features, which also influence the choice of a suitable numerical pricing method. Stochastic models and payoff functions each fall in two broad categories: Markovian or non-Markovian models for the former, and path-dependent or path-independent payoffs for the latter. A Markovian stochastic process enjoy the nice property that the distribution of the future asset prices is independent of the past prices given the present price. A path-independent payoff enjoys the property that the payoff at a given time depends only on the asset price at that time and not on past prices.

Out of the four categories of options that can be created by combining these two features, the one that considers Markovian asset prices and path-independent payoff has received the most attention in the literature. This class of options are referred to as *vanilla* options. Shorty after the seminal works of Black et Scholes (1973) and Merton (1973) in deriving exact formula for pricing European stock options under a geometric Brownian motion model, the *binomial tree* method was introduced by Cox *et al.* (1979) to price the equivalent American-style options. The binomial tree method manages to build a discretization scheme that grows only linearly as the number of exercise dates increases. It avoids the exponential growth because the tree structure is *recombining*, i.e., all nodes that have the same asset price but come from different scenarios are seen as identical and hence are merged into a single node. By doing so, the number of nodes at the m-th decision stage is m instead of 2^{m-1} for a non-recombining structure. The original binomial tree method has undergone many improvements over the years in the calibration of the tree and the extension to multinomial trees; see, e.g., Jarrow et Rudd (1983), Boyle (1986), Amin (1991), Kamrad et Ritchken (1991), and Figlewski et Gao (1999). Another popular approach to price path-independent American-style options on a Markovian process, which also avoids the exponential growth of scenarios by using recombining structures, is the *stochastic mesh* method of Broadie et Glasserman (2004). Unlike the binomial tree, the stochastic mesh method uses randomly sampled scenarios. This makes it attractive for pricing options on several underlying assets. The original method has also undergone a lot of improvements in the choice of the mesh density; also, some authors have proposed to use deterministic methods to generate scenarios instead of Monte Carlo sampling, which led to the *low-discrepancy mesh* method studied in Boyle *et al.* (2013) and Imai (2014), among others.

Another category of options that has received a lot of attention considers Markovian asset prices and path-*dependent* payoff. These options are referred to as *exotic* options by opposition to the non-exotic vanilla options described previously. Path-dependent options do not a priori belong to the framework of Markov decision processes since their payoff does not only depend on the current asset price, but is a function of the whole history of the process. However, they can usually be recast in the framework of Markov decision processes by in-

cluding the path-dependent variables in the state space, i.e., by augmenting the state space dimension. Examples of exotic options include *lookback* options (the payoff is a function of the maximum or minimum asset price) and *Asian* options (the payoff is a function of the average asset price). More examples can be found in Hull (2017, Chapter 26). Numerical methods to price path-dependent options include (to name of few): Hull et White (1993), Barraquand et Pudet (1996), Carriere (1996), Grant *et al.* (1997), Longstaff et Schwartz (2001), and Tsitsiklis et Van Roy (2001).

Pricing problems under *non*-Markovian models have received much less attention despite the now clear evidence that the volatility of stock returns exhibit *long-term memory* and *clustering*. These features were first noted by Mandelbrot (1967) and have been further documented in Ding *et al.* (1993), Granger et Ding (1995), Cont (2005, 2007), and Sun *et al.* (2007), among others. The need to model the volatility persistence that many asset prices exhibit led to the development of various stochastic volatility models with a long-range dependence property. Examples of discrete-time processes include the *fractionally integrated GARCH* of Baillie *et al.* (1996) and the *long memory stochastic volatility* of Breidt *et al.* (1998). Examples of continuous-time processes include the *fractional Brownian motion* of Mandelbrot et Van Ness (1968) and the *multifractal model of asset returns* of Mandelbrot *et al.* (1997) and Calvet et Fisher (2002). Research on pricing options under these models have essentially concerned European-style options. For example, Wang (2010) derived a formula for the vanilla European call option under a multifractal model. As for options with an early exercise feature, the long-lasting path dependency inherent of these models may require the use of tree structures that record the whole history of the process at each time, making the valuation of these options computational intensive in terms of memory requirements and run times. To reduce this computational burden, it is therefore necessary to look for well-designed tree structures that will manage to reduce the number of scenarios to the minimum while keeping high estimation accuracy. This is an issue addressed in our paper.

The use of *non*-recombining tree structures that record the whole stochastic process history at any time is not limited to option pricing problems. The *stochastic programming* framework and the related *scenario-tree generation* topic have been applied to many financial planning problems: from asset-liability management (Consigli et Dempster (1998), Kouwenberg (2001), Oliveira *et al.* (2018)) and risk management (Eichhorn *et al.* (2010), Consigli *et al.* (2012)), to portfolio selection (Gülpınar *et al.* (2004), Topaloglou *et al.* (2008a), Beraldi et Bruni (2014)) and financial engineering (Topaloglou *et al.* (2008b), Consiglio et De Giovanni (2008)). No reference can provide a comprehensive treatment of these topics as the field of stochastic programming applications in finance is so vast. Some books offer however a

detailed treatment of some specific topics, we refer for instance to Ziemba (2003) for asset-liability management topics and Kovacevic *et al.* (2013) for risk-management topics. It is important to mention that the stochastic programming approach is not limited to the realm of finance either. Countless applications exist in energy, logistics, transportation, medicine, etc. The issue of generating efficient scenario trees spans throughout all fields dealing with *multistage planning under uncertainty*.

Due to its large influence, the question of when to use scenario trees and how to build them is of prime importance. The second part of this question ("*how* to build them") has attracted a lot of attention in the stochastic programming community. Many approaches have been developed to generate scenario trees based on different theoretical grounds: Monte Carlo (Shapiro et Homem-de Mello, 1998), quasi-Monte Carlo (Leövey et Römisch, 2015), integration quadrature (Koivu, 2005; Pennanen, 2009), optimal quantization (Pflug, 2001), and moment-matching (Høyland et Wallace, 2001) are examples of popular methods used to generate scenarios. Examples of methods that also compute a tree structure include the sequential importance sampling (Dupačová *et al.*, 2000; Dempster, 2006), the bound-based approximation (Frauendorfer, 1996; Edirisinghe, 1999), and the methods based on the minimization of some probability distances, such as the *filtration distance* of Heitsch *et al.* (2006) and the *nested distance* of Pflug et Pichler (2012).

The first part of the question ("*when* to use scenario trees") has attracted relatively much less attention. From the works of Dyer et Stougie (2006), Shapiro (2006), and Hanasusanto *et al.* (2016) on the complexity of stochastic programming problems and scenario trees, it is clear that the scenario-tree approach cannot be efficient for *all* multistage problems, because of their inherent exponential growth as the number of stages increases. However, this does not mean that the approach cannot approximate efficiently *some* multistage problems that may have attractive features that the scenario trees could build upon to compute accurate estimates with somehow less scenarios than expected. The current belief in this regard, deduced by Keutchayan *et al.* (2018a,b) from their multistage optimal-value error upper bound, is that these so-called "attractive features" are linked to the *variability* of the recourse (cost-to-go) functions of the problem. We will discuss their result in more details in the next sections. In short, it shows that the scenario-tree approach could be most efficient when applied to problems where the recourse functions have heterogeneous variability, either across the stages, or over the distribution support at given stage. In this two situations, the tree structure can take advantage of this heterogeneity to branch out more where the variability is high and less where it is small, in order to limit the exponential expansion while keeping the error under control.

To demonstrate the validity of their belief, we consider in the present paper the problem of pricing a Bermudan-style Asian call option using scenario trees under the standard Black-Scholes setting. We consider specifically the type of Asian option whose payoff is given by the difference between the arithmetic average of the asset prices over a set of discrete monitoring times and the fixed strike price. The averaging feature makes Asian options cheaper than the equivalent vanilla options, because the payoff close to maturity is less volatile than for a vanilla option. This feature is also what makes it attractive to the scenario-tree pricing method: as time gets closer to maturity, the recourse function (in this case: the option price function) has less variability, and therefore the scenario tree need not branch out as much as it did at the beginning of the option's life. The scenario-tree approach introduced in Keutchayan *et al.* (2018a,b) is problem-driven, it will thus take advantage of this payoff feature to generate scenario trees suitable to the Asian option pricing problem.

There are many numerical methods to price Asian options accurately under the Black-Scholes setting (e.g., Chalasani *et al.* (1999); Ben-Ameur *et al.* (2002)), because the problem can be recast in a Markovian framework by considering a two-dimensional state space. We want to make clear that we do not claim that scenario trees outperform these methods in this setting. Our goal is to compare problem-driven and problem-blind scenario trees with each other. The ultimate goal of this is to learn when and how to use the scenario-tree approach in more sophisticated settings where the state-space dimension would be too large and the Markovian-based approaches would typically fail. Such settings could be the long-memory stochastic volatility models described previously. In our paper, we consider the Black-Scholes setting because it allows highly accurate estimation of the true option prices, which is necessary to make the comparison between scenario trees meaningful.

We note that the scenario-tree approach has already been applied to option pricing problems. One of the first attempt to price options using Monte Carlo simulation was done in Broadie et Glasserman (1997) using what they referred to as "random trees", which fall in our broad definition of scenario trees. The authors acknowledge the high computational effort of their method and recommend to explore extensions that would build trees with non-constant branching coefficients and low-discrepancy sequences instead of random sampling. The present paper introduces the appropriate framework to implement these extensions in the case of Asian options.

The remainder of the paper is organized as follows: First, in Section 7.2, we describe the option pricing problem in a general setting. Then, in Section 7.3, we apply the scenario-tree generation framework of Keutchayan *et al.* (2018a,b) to the case of the Asian option pricing problem. Finally, in Section 7.4, we generate scenario trees and perform numerical

experiments to assess their quality and compare them with more standard scenario trees typically used by practitioners. We conclude the paper in Section 7.5. Appendices C and D contain two proofs delayed at the end of the paper to lighten the main text.

7.2 Option pricing problems

We consider an option on an underlying asset whose price dynamic is described by a stochastic process $\{S(t) : t \in [0, T]\}$, where $S(t)$ takes values in $\mathbb{R}^d$. The life of the option starts at the time $t = 0$ and ends at maturity $t = T$. The option can be exercised at $M \geq 1$ discrete times (stages) equally spaced: $t_1 = \Delta t$, $t_2 = 2\Delta t$, $\ldots$, $t_M = T$, with $\Delta t = T/M$.

When the option is exercised at stage m while the underlying security has history $S_{..m} := (S(t_0), \ldots, S(t_m))$, the payoff (expressed in stage-0 dollars) immediately received is $h_m(S_{..m})$. At stage m, the holding value is given by

$$C_m(S_{..m}) = \mathbb{E}[V_{m+1}(S_{..m+1}) \,|\, S_{..m}], \quad m = 0, \ldots, M-1, \tag{7.1}$$

where the expectation is under some risk-adjusted measure and $V_{m+1}(\cdot)$ is the option price function at stage $m+1$ given by

$$V_m(S_{..m}) = \begin{cases} h_M(S_{..M}), & \text{if } m = M; \quad (7.2) \\ \max\{h_m(S_{..m}), C_m(S_{..m})\}, & \text{if } m \in \{1, \ldots, M-1\}; \quad (7.3) \\ C_0(S_0), & \text{if } m = 0. \quad (7.4) \end{cases}$$

The risk-adjusted measure may not be uniquely determined if the market is incomplete. We do not consider the problem of finding such measure, we just assume it is given to us. (From the next section on we will consider the standard Black-Scholes setting, which essentially removes the problem.)

At stage $m = 0$ the option price $V_0(S_0)$ equals the holding value $C_0(S_0)$ because we consider that exercising the option is possible only from stage $m = 1$ onward. The difficulty in estimating the option price $V_0(S_0)$ lies in the recursive calculation of the expectations (7.1). In most cases this calculation cannot be performed analytically and one must turn to some discretization method to compute an approximate value.

Note that if the underlying asset satisfies the Markov property and if the payoff depends only on the value of the underling at the current stage (i.e., $h_m(\cdot)$ is a function of S_m only), then the equation (7.1) simplifies to

$$C_m(S_m) = \mathbb{E}[V_{m+1}(S_{m+1}) \,|\, S_m], \quad m = 0, \ldots, M-1, \tag{7.5}$$

The fact that the holding value $C_m(S_m)$ depends solely on S_m, and no longer on the whole history $S_{..m}$, would allow to consider numerical methods that exploit the Markovian structure of the decision-making process. However, should one of the two above conditions fails (i.e., the underlying is not Markovian or the payoff is path-dependent), the dependence on the history $S_{..m}$ in (7.1) cannot be cut off, and therefore the discretization structures used to approximate recursively the expectations have to record the whole history at any time (or at least some part of it).

In this paper we consider discretization structures in the form of scenario trees. A *scenario tree* is defined as a triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ where

- $\mathcal{T} = (\mathcal{N}, \mathcal{E}, n_0)$ is the *tree structure* with $\mathcal{N}$ the finite node set, $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ the edge set and $n_0 \in \mathcal{N}$ the root node;
- $\mathcal{P} = \{\zeta^n : n \in \mathcal{N}\} \subset \mathbb{R}^d$ is the set of *discretization points* (at the root node: $\zeta^{n_0} = S_0$);
- $\mathcal{W} = \{w^m : n \in \mathcal{N}\} \subset (0, \infty)$ is the set of *discretization weights* (at the root node: $w^{n_0} = 1$).

The tree structure is such that M edges separate the root node from any of the tree leaves.

The scenario-tree estimate of the option price, $\widehat{V}_0(S_0)$, is obtained from the equations (7.1)-(7.4) by substituting $C_m(\cdot)$ with its scenario-tree estimator $\widehat{C}_m(\cdot)$ computed as follows:

$$\widehat{C}_m(\zeta^{..n}) = \sum_{k \in C(n)} w^k \, \widehat{V}_{m+1}(\zeta^{..n}, \zeta^k), \quad \forall n \in \mathcal{N}_m, \ \forall m \in \{0, \ldots, M-1\} \tag{7.6}$$

where:

- $\zeta^{..n}$ is the sequence of discretization points on the path from n_0 to n;
- $C(n)$ is the set of child nodes of n (i.e., the nodes connected to n at the next stage);
- $\mathcal{N}_m$ is the set of all stage-m nodes (i.e., the nodes that are m edges away from n_0).

The advantage of the scenario-tree approach over the Markovian-based approaches is that it can be applied to any underlying stochastic process and payoff function. Its obvious drawback is a direct consequence of this: the tree structure records the whole history of the process at any time and hence its size grows exponentially with the number of stages.

The goal of this paper is to show how scenario trees can be built in an optimal way so as to provide the best estimation quality given a fix computational cost. The estimation quality is

measured by the error between the true option price and its estimate (i.e., $V_0(S_0) - \widehat{V}_0(S_0)$) and the computational cost by the total number of scenarios included in the tree (i.e., $|\mathcal{N}_M|$).

7.3 Scenario-tree construction

The approach developed in Keutchayan *et al.* (2018a,b) to generate problem-driven scenario trees is based on the following two key concepts:

- The concept of *guidance functions*: a stochastic problem with $M+1$ stages is characterized by a set of M guidance functions $\Gamma := \{\gamma_0, \ldots, \gamma_{M-1}\}$, where $\gamma_m(S_{..m}) \geq 0$ represents the conditional variability (given $S_{..m}$) of the recourse functions at stage $m+1$.
- The concept of *figure of demerit*: denoted by $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$, it measures the suitability between a scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and a problem characterized by Γ.

For a suitable choice of Γ, it can be showed (Keutchayan *et al.*, 2018b, Corollary 3.3) that the figure of demerit is an upper bound on the error between the true optimal value and its scenario-tree approximation. Thus, once an appropriate set Γ is defined for the considered problem, the generation of scenario tree is done by minimizing the upper bound, i.e., by selecting the triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ solution of

$$\min_{(\mathcal{T},\mathcal{P},\mathcal{W})} \mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma). \tag{7.7}$$

There are two main difficulties in implementing this approach in a real-world problem. The first one is the computation of Γ. In theory the guidance functions enjoy an analytic expression. In practice, this expression cannot be computed exactly as it involves some unknown quantities. It is therefore necessary to look for approximations and to make sure that these approximations preserve the relevant features of the problem. The second difficulty is the actual minimization of (7.7). The feasible set of the problem is enormous: it is the Cartesian product of the discrete set of all tree structures and the continuous sets of all discretization points and weights, which are subsets of $(\mathbb{R}^d)^{|\mathcal{N}|}$ and $(0, \infty)^{|\mathcal{N}|}$, respectively. In practice, therefore, the minimization problem (7.7) cannot be solved to optimality as it is. Nevertheless, it can be solved approximately, or even optimally under additional constraints. Several algorithmic procedures have been developed in Keutchayan *et al.* (2018a,b) in that respect. We will review these procedures and introduce a new one in Section 7.3.2. In Section 7.3.1, we show how to compute the set Γ for the problem of pricing a Bermudan-Asian call option.

7.3.1 Computation of the guidance functions

Consider a Bermudan-Asian call option with maturity T, fixed strike price K, and M discrete monitoring times for the arithmetic average which coincide with the exercise dates: $t_m = m\Delta t$, with $\Delta t = T/M$ and $m = 1, \ldots, M$. Under the risk-adjusted measure, the underlying asset is described by the stochastic process $\{S(t) : t \in [0, T]\}$ which follows a one-dimensional geometric Brownian motion GBM(r, σ^2) where $r > 0$ is the risk-free rate and $\sigma > 0$ is the volatility; see, e.g., Glasserman (2003). This means that at each time t_m the asset value $S_m := S(t_m)$ is given recursively by

$$S_m = S_{m-1} e^{(r - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}\,\epsilon_m}, \quad m = 1, \ldots, M, \tag{7.8}$$

where $S_0 > 0$ is the fixed initial asset price and $\epsilon_1, \ldots, \epsilon_M$ are independent $N(0, 1)$-distributed random variables.

The Asian call option price is computed by the equations (7.1)-(7.4) where the payoff function at stage m is given by

$$h_m(S_{..m}) = \delta^m \max\left\{\frac{1}{m}\sum_{i=1}^{m} S_i - K, 0\right\}, \tag{7.9}$$

with $\delta = e^{-r\Delta t}$. Note that by convention the average does not include the initial security price S_0 and the option cannot be exercised at stage 0.

The error made when estimating the holding value using (7.6) instead of (7.1) depends on (i) the discretization points and weights used at nodes $C(n)$ and (ii) the conditional variability of the integrand $V_{m+1}(\zeta^{\cdot\cdot n}, \cdot)$. Generally, an integrand with large variability will induce a larger estimation error than an integrand with low (or no) variability. The concept of variability will be formally defined below. Intuitively, it is linked to how much the function $V_{m+1}(\zeta^{\cdot\cdot n}, \cdot)$ varies over the distribution support of S_{m+1} and to how volatile the asset price S_{m+1} is, given $S_{..m} = \zeta^{\cdot\cdot n}$. To see why variability is linked to the estimation error, imagine that one of these two features is reduced to none, i.e., $V_{m+1}(\zeta^{\cdot\cdot n}, \cdot)$ is constant or S_{m+1} is perfectly predictable given $S_{..m} = \zeta^{\cdot\cdot n}$. Then, a unique discretization point ζ^k located in the conditional support of S_{m+1} and a unique weight $w^k = 1$ are sufficient to estimate *exactly* the true holding value $C_m(\zeta^{\cdot\cdot n})$. Of course, this extreme situation will hardly ever happen in reality but it provides guidance for situations where $V_{m+1}(\zeta^{\cdot\cdot n}, \cdot)$ is *almost* constant or S_{m+1} is *almost* perfectly predictable. As a matter of fact, from the geometric Brownian motion assumption we already know from (7.8) that S_{m+1} will be less volatile when S_m is small, since the conditional variance $\mathrm{Var}(S_{m+1}|S_m) = S_m^2 e^{2r\Delta t}(e^{\sigma^2\Delta t} - 1)$ decreases with decrease of

S_m.

Knowing the variability of the integrands (at least approximately) is therefore a prime information when building scenario trees. It allows to balance optimally the number of nodes between some parts of the distribution support that bear more variability and other parts that bear less. The guidance functions Γ are the mathematical representation of this information. To compute the set Γ suitable to the problem of pricing Bermudan-Asian options, we need (i) to define a relevant measure of variability and (ii) to assess the variability of the unknown function $V_{m+1}(S_{..m}, \cdot)$ for each m and each value of $S_{..m}$.

Variability measure:

We consider a measure of variability, $\mathcal{V}(\cdot)$, given by the norm in an appropriate *Sobolev space* of square-integrable functions with square-integrable first-order derivatives. Sobolev spaces have been largely studied in the context of numerical integration over the unit cube $[0,1]^d$, because the integration error of functions that belong to these spaces is connected to the notion of point set discrepancy. This is a highly technical subject matter, hence we refer to the books of Novak et Woźniakowski (2008, 2010) for a detailed treatment of this topic. Broadly speaking, this connection exists because Sobolev spaces are *reproducing kernel Hilbert spaces*, which allows to derive an upper bound on the integration error. The upper bound is the product of two terms: one that depends on the discretization points and weights only, which measures their discrepancy with respect to the original distribution, and one that depends only on the integrand, which measures its variability through the norm of the Hilbert space. In our case we are concerned with numerical integration over the whole real line $\mathbb{R}$, hence we follow the framework of Sobolev spaces described in Griebel *et al.* (2013).

Let $\rho : \mathbb{R} \to \mathbb{R}$ denote the density function of the $N(0,1)$ distribution and let $\mathcal{L}_{2,\rho}(\mathbb{R})$ be the weighted $\mathcal{L}_2(\mathbb{R})$ space equipped with the norm

$$\|f\|_{2,\rho} = \left(\mathbb{E}_{\epsilon\sim\rho}[f(\epsilon)^2]\right)^{1/2} = \left(\int_{\mathbb{R}} f(\epsilon)^2 \rho(\epsilon) d\epsilon\right)^{1/2}. \tag{7.10}$$

The Sobolev space $W^1_{2,\rho}(\mathbb{R})$ is composed of all first-order derivable (in a weak sense) functions $f : \mathbb{R} \to \mathbb{R}$ such that

$$\|f\|_{2,\rho} < \infty \quad \text{and} \quad \|f'\|_{2,\rho} < \infty. \tag{7.11}$$

In this Sobolev space the measure of variability of f, $\mathcal{V}(f)$, is given by the norm of its derivative:

$$\mathcal{V}(f) = \|f'\|_{2,\rho}. \tag{7.12}$$

When dealing with a multivariate function, such as $f(\epsilon_1, \ldots, \epsilon_4)$, the variability of f with respect to ϵ_4 at fixed $(\epsilon_1, \epsilon_2, \epsilon_3)$ is

$$\mathcal{V}(f)(\epsilon_1, \epsilon_2, \epsilon_3) = \left\| \frac{\partial f}{\partial \epsilon_4}(\epsilon_1, \epsilon_2, \epsilon_3, \cdot) \right\|_{2,\rho}. \tag{7.13}$$

Variability of $V_{m+1}(S_{..m}, \cdot)$:

Now that we have defined a specific measure of variability, we can estimate that measure for the option price $V_{m+1}(S_{..m}, \cdot)$. For convenience, we now express every function of the stochastic process $S_{..m}$ as a function of the underlying sequence of independent random variables $\epsilon_1, \ldots, \epsilon_m$, which generate $S_{..m}$ from (7.8). For example, as a function of $(\epsilon_1, \ldots, \epsilon_m)$ the payoff $h_m(\cdot)$ reads

$$h_m(\epsilon_1, \ldots, \epsilon_m) = \delta^m \max\left\{ \frac{S_0}{m} \sum_{i=1}^{m} \left(\prod_{j=1}^{i} e^{Z(\epsilon_j)} \right) - K, 0 \right\}, \tag{7.14}$$

where $Z(\epsilon) = (r - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}\,\epsilon$.

The following proposition provides bounds on the variability of $V_1, \ldots, V_M$ for $M = 4$.

Proposition 7.3.1. *Consider a Bermudan-Asian call option with $M = 4$ and $\delta \leq \frac{4}{3}$. The variability of V_m with respect to ϵ_m in the sense of the Sobolev space $W^1_{2,\rho}(\mathbb{R})$ is bounded as follows:*

(i) $\mathcal{V}(V_4)(\epsilon_1, \epsilon_2, \epsilon_3) \leq \delta^3 \beta S_0 \sigma \sqrt{\Delta t}\, \frac{1}{4} \left(\prod_{i=1}^{3} e^{Z(\epsilon_i)} \right)$ *for all* $(\epsilon_1, \epsilon_2, \epsilon_3) \in \mathbb{R}^3$*;*

(ii) $\mathcal{V}(V_3)(\epsilon_1, \epsilon_2) \leq \delta^2 \beta S_0 \sigma \sqrt{\Delta t} \max\left\{ \frac{1}{3}, \frac{1+\delta}{4} \right\} \left(\prod_{i=1}^{2} e^{Z(\epsilon_i)} \right)$ *for all* $(\epsilon_1, \epsilon_2) \in \mathbb{R}^2$*;*

(iii) $\mathcal{V}(V_2)(\epsilon_1) \leq \delta S_0 \beta \sigma \sqrt{\Delta t} \max\left\{ \frac{1}{2}, \frac{1+\delta}{3}, \frac{\delta}{3} + \frac{(1+\delta)}{4} \right\} e^{Z(\epsilon_1)}$ *for all* $\epsilon_1 \in \mathbb{R}$*;*

(iv) $\mathcal{V}(V_1) \leq S_0 \beta \sigma \sqrt{\Delta t} \max\left\{ 1, \frac{1+\delta}{2}, \frac{\delta}{2} + \frac{1+\delta}{3}, \frac{\delta}{2} + \frac{\delta}{3} + \frac{1+\delta}{4} \right\}$,

where $\beta = e^{\frac{\sigma^2}{2}\Delta t}$.

Proof. Let us notice that $h_m = \max\{g_m, 0\}$ where g_m is the smooth function given by

$$g_m(\epsilon_1, \ldots, \epsilon_m) = \delta^m \frac{S_0}{m} \sum_{i=1}^{m} \left(\prod_{j=1}^{i} e^{Z(\epsilon_j)} \right) - \delta^m K. \tag{7.15}$$

The partial derivative of h_m with respect to ϵ_i exists pointwise almost everywhere and satisfy

$$0 \leq \frac{\partial h_m}{\partial \epsilon_i} = \frac{\partial g_m}{\partial \epsilon_i} \mathbf{1}_{g_m > 0} \leq \frac{\partial g_m}{\partial \epsilon_i}, \quad i = 1, \ldots, m. \tag{7.16}$$

The bounds can be derived recursively by noticing that the weak derivatives of V_m and C_m satisfy:

$$\begin{cases} \dfrac{\partial V_M}{\partial \epsilon_i} \leq \dfrac{\partial g_M}{\partial \epsilon_i}, & 1 \leq i \leq M; \quad (7.17) \\ \dfrac{\partial V_m}{\partial \epsilon_i} \leq \max\left\{\dfrac{\partial g_m}{\partial \epsilon_i}, \dfrac{\partial C_m}{\partial \epsilon_i}\right\}, & 1 \leq i \leq m,\ 0 < m < M; \quad (7.18) \\ \dfrac{\partial C_m}{\partial \epsilon_i} = \mathbb{E}_{\epsilon_{m+1}}\left[\dfrac{\partial V_{m+1}}{\partial \epsilon_i}\right], & 1 \leq i \leq m,\ 0 < m < M. \quad (7.19) \end{cases}$$

The first inequality follows directly from $V_M = h_M$ and (7.16). At stage $0 < m < M$, since $V_m = \max\{h_m, C_m\}$ the second inequality is obtained as follows:

$$\begin{aligned} \frac{\partial V_m}{\partial \epsilon_i} &= \frac{\partial h_m}{\partial \epsilon_i} \mathbf{1}_{h_m > C_m} + \frac{\partial C_m}{\partial \epsilon_i} \mathbf{1}_{h_m < C_m} && (7.20) \\ &\leq \frac{\partial g_m}{\partial \epsilon_i} \mathbf{1}_{h_m > C_m} + \frac{\partial C_m}{\partial \epsilon_i} \mathbf{1}_{h_m < C_m} && (7.21) \\ &\leq \max\left\{\frac{\partial g_m}{\partial \epsilon_i}, \frac{\partial C_m}{\partial \epsilon_i}\right\} (\mathbf{1}_{h_m > C_m} + \mathbf{1}_{h_m < C_m}) && (7.22) \\ &= \max\left\{\frac{\partial g_m}{\partial \epsilon_i}, \frac{\partial C_m}{\partial \epsilon_i}\right\}. && (7.23) \end{aligned}$$

Finally, the equality (7.19) follows from interchanging derivative and expectation. This is possible because V_{m+1} is convex and its partial derivative $\partial V_{m+1}/\partial \epsilon_i$ exists almost everywhere (see, e.g, Shapiro *et al.* (2014, Theorem 7.46)). Convexity of V_{m+1} is proved recursively from the fact that each h_m is convex and that convexity is preserved through expectation and maximum.

For conciseness the actual computation of the bounds (i)-(iv) is done in Appendix C. □

The above proposition can easily be generalized to any number of monitoring dates $M \geq 1$ to show that for $\delta \leq M/(M-1)$ the variability at stage m is bounded as follows:

$$\mathcal{V}(V_m)(\epsilon_1, \ldots, \epsilon_{m-1}) \leq \delta^{m-1} u_m(\delta) \beta \sigma \sqrt{\Delta t} S_0 \Big(\prod_{i=1}^{m-1} e^{Z(\epsilon_i)} \Big), \quad m = 1, \ldots, M, \tag{7.24}$$

where $\{u_m(\delta)\}$ is the sequence given recursively (backward from M to 1) by

$$\begin{cases} u_M(\delta) = \frac{1}{M}; \\ u_m(\delta) = \max\left\{\frac{1}{m}, \frac{\delta}{m+1} + u_{m+1}(\delta)\right\}, \quad m = 1, \ldots, M-1. \end{cases} \tag{7.25}$$

Note that as a function of the stochastic process the bound value reads

$$\mathcal{V}(V_m)(S_{m-1}) \leq \delta^{m-1} u_m(\delta) \beta\sigma\sqrt{\Delta t} S_{m-1}, \quad m = 1, \ldots, M \tag{7.26}$$

The bound on the variability $\mathcal{V}(V_m)$ exhibits several relevant features for the construction of scenario trees:

- The variability increases with σ and Δt. This can be easily understood because σ is the volatility of the asset price and a large time interval Δt means that the next asset price is more difficult to predict given the present one. Thus, the bound shows that the efficiency of scenario trees scales negatively with the quantity $\beta\sigma\sqrt{\Delta t}$.

- The variability decreases to zero as the stage m increases and everything else being fixed, because the sequence $\{\delta^{m-1} u_m(\delta)\}$ is decreasing as showed in Figure 7.1. This means that the variations of the asset price has less and less impact on the option price as time moves from the first decision stage to the last. This feature was expected, since the payoff of Asian options is given by an average which becomes more and more robust to variations as m increases. As a comparison, it is easy to apply the same bounding scheme to vanilla call options to find out that the sequence bounding their price variability is merely $\{\delta^{m-1}\}$. The decrease of such sequence is much slower than that of $\{\delta^{m-1} u_m(\delta)\}$ (see Figure 7.1).

- The variability at stage m decreases as the asset price S_{m-1} decreases. Again, this was expected since the variations of S_m given S_{m-1} (measured by its conditional variance) have less amplitude as S_{m-1} gets closer to zero.

These three features are highly intuitive and the bound allow to quantify them in order to take advantage of them in the construction of scenario trees. The first one will not have any impact on the shape of the tree structure because it impacts all node equally. The last two features are the ones impacting the tree structure: (i) The fact that $\mathcal{V}(V_m)$ decreases as m gets closer to M will make the average branching factors decrease across stages. Thus, the scenario-tree discretization will be finer at the early stages and coarser at the late stages. (ii) The fact that $\mathcal{V}(V_m)$ decreases with decrease of S_{m-1} will generate an asymmetry in the

branching factors out of the nodes $\mathcal{N}_{m-1}$. Nodes with small asset price S_{m-1} will be given less child nodes that those with large values of S_{m-1} (everything else being equal).

Guidance functions Γ:

For all the reasons expressed above, we use the variability bounds as the guidance functions that will guide the generation of scenario trees. Specifically, we consider a set $\Gamma = \{\gamma_0, \ldots, \gamma_{M-1}\}$ given by

$$\gamma_m(S_m) = \delta^m u_{m+1}(\delta) S_m, \quad m = 0, \ldots, M-1. \tag{7.27}$$

We have dropped the constant $\beta\sigma\sqrt{\Delta t}$ because it would appear in every γ_m with the same magnitude, hence it would not impact the construction of scenario trees.

Setting the guidance functions as bounds on the variability guarantees that the guidance functions will never underestimate the difficulty of approximating the holding value (7.1) by the scenario-tree estimator (7.6). However, the drawback of this choice is that they will typically overestimate it. Overestimating the variability implies that the scenario tree will over-discretize some part of the distribution support that did not require a fine discretization. Over-discretization should be avoided because it increases the size of the scenario tree with no significant gain in terms of error reduction.

The variability bounds are derived from the backward recursion (7.17)-(7.19), which are themselves derived from the inequality (7.16). This inequality consists in ignoring the fact that the derivative of h_m is zero when $g_m < 0$, i.e., that the immediate payoff has no variability when the option is not exercised. Because of this, the scenario tree will tend to over-discretize out of the scenarios where the option is not, and will never be, exercised.

To fix this, we introduce a *cut-off term* in the definition of the guidance functions. This term will cut-off the value of $\gamma_m(S_m)$ and put it at an arbitrary low value if the average up to m is so small that the option is unlikely to be ever exercised. Let κ and ν denote the two cut-off parameters. We consider that the option price $V_{m+1}(S_{..m}, \cdot)$ has a variability lower than ν ($\nu \geq 0$) if

$$\frac{1}{M}\left(\sum_{i=1}^{m} S_i + S_m \sum_{i=m+1}^{M} e^{Z(\kappa)(i-m)}\right) \leq K, \tag{7.28}$$

for some $\kappa \geq 0$ large enough. In other words, the variability is low (possibly zero) if the running average up to stage m is so low that even a process that grows deterministically at rate $e^{Z(\kappa)}$ from $m+1$ to M will not make the average higher that the strike price at maturity. The final definition of Γ is therefore the following:

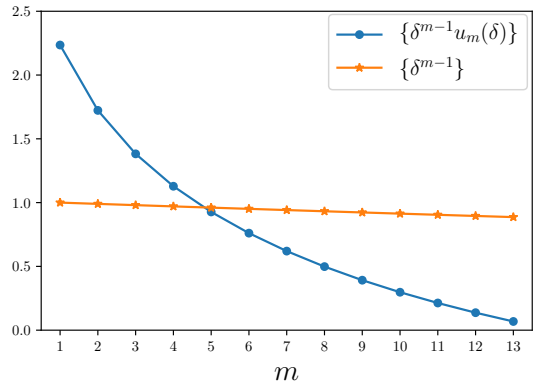


Figure 7.1 Sequence $\{\delta^{m-1}u_m(\delta)\}$ which bounds the stagewise variability of an Asian option and sequence $\{\delta^{m-1}\}$ which bounds that of a vanilla option. ($M = 13$, $\delta = 0.99$)

Definition 7.3.2. Let $\kappa \geq 0$ and $\nu \geq 0$ be some cut-off parameters. The set of guidance functions, $\Gamma^{\kappa,\nu}$, considered for the problem of pricing a Bermudan-Asian call option with $M \geq 1$ and $\delta \leq M/(M-1)$ is defined as $\gamma_0^{\kappa,\nu}(S_0) = S_0 u_1$ and

$$\gamma_m^{\kappa,\nu}(S_{..m}) = \begin{cases} \nu & \text{if } \frac{1}{M}\left(\sum_{i=1}^{m} S_i + S_m \sum_{i=m+1}^{M} e^{Z(\kappa)(i-m)}\right) \leq K \quad (7.29) \\ \delta^m u_{m+1}(\delta) S_m & \text{otherwise,} \quad (7.30) \end{cases}$$

for $m = 1, \ldots, M-1$.

Taking $\kappa = \infty$ ensures that the condition (7.28) is never satisfied, hence for this value we recover the guidance functions (7.27) without cut-off. Figure 7.2 shows the plots of $\gamma_1^{\kappa,\nu}$ and $\gamma_2^{\kappa,\nu}$ for a problem with three monitoring dates and $(\kappa, \nu) = (2, 10)$.

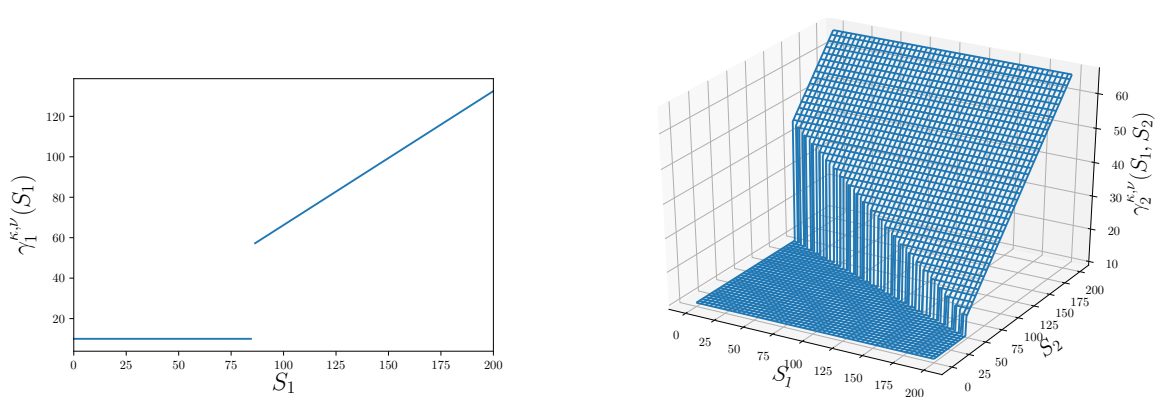


Figure 7.2 $\gamma_1^{\kappa,\nu}(S_1)$ (left) and $\gamma_2^{\kappa,\nu}(S_1, S_2)$ (right) for $(\kappa, \nu) = (2, 10)$ for a call option with $M = 3$, $r = 0.05$, $\sigma = 0.25$, $T = 0.25$, $S_0 = 100$, $K = 100$.

In the following we take the radical view of considering that the variability is simply zero when the condition (7.28) is satisfied. Thus we now set $\nu = 0$ and simply write Γ^κ instead of $\Gamma^{\kappa,0}$.

7.3.2 Minimization of the figure of demerit

We can now express the figure of demerit $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma^\kappa)$ that measures the suitability between the scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and the class of Bermudan-Asian call option pricing problems having guidance functions Γ^κ. Following the simplifications in Keutchayan *et al.* (2018b), the figure of demerit reads

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma^\kappa) = \sum_{m=0}^{M-1} \sum_{n \in \mathcal{N}_m} \frac{W^n \gamma_m^\kappa(\zeta^{\cdot n})}{|C(n)|^\alpha}, \tag{7.31}$$

where :

- $\zeta^{\cdot n}$ is the sequence of discretization points on the path from n_0 to n;
- W^n is the product of all the discretization weights w^k on the path from n_0 to n;
- $|C(n)|$ is the number of child nodes of n;
- $\alpha > 0$ is the rate of convergence of the discretization method.

By *discretization method* we mean any procedure that generates sets of points and weights of arbitrary cardinality to approximate a given probability distribution. We consider specifically *deterministic* discretization methods.

We consider two families of methods for the discretization of the $N(0,1)$ distribution: the *quasi-Monte Carlo* (QMC) methods and the *optimal quantization* (OQ) methods. The QMC point set of size $N \geq 1$ that we consider is generated from a low discrepancy lattice in $(0,1)$ through inversion of the (approximate) normal cumulative distribution function $\phi_{N(0,1)}$ as follows:

$$\left\{ \phi_{N(0,1)}^{-1}\left(\frac{i+0.5}{N}\right) : i = 0, \ldots, N-1 \right\}, \tag{7.32}$$

and the weights are set to $1/N$. We denote it by QMC-Lat. The OQ discretization set is generated directly by minimizing the Wasserstein distance of order 1 and 2, leading to two different sets of points and weights, referred to as OQ-W_1 and OQ-W_2, respectively. The sets of weights of QMC-Lat, OQ-W_1, and OQ-W_2 add up to one, hence these methods integrate exactly constant functions. However, unlike QMC, the weights in OQ are not equal, they

match the probabilities of the normal distribution. We refer to Pflug et Pichler (2015) for a description of the procedures to minimize the Wasserstein distances. Since we consider the discretization of a one-dimensional normal distribution, the rate of convergence of QMC-Lat, OQ-W_1, and OQ-W_2 is set to $\alpha = 1$ in (7.31).

Using the previous discretization methods, we consider three algorithmic procedures to find the scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ that approaches the minimum of the figure of demerit. The first two procedures are developed in Keutchayan *et al.* (2018b) and Keutchayan *et al.* (2018a), respectively. The first one requires merely the specification of the total number of scenarios, $|\mathcal{N}_M|$, and is feasible for scenario trees with few scenarios and stages (instances up to 60 scenarios and 6 stages are easily computed in Keutchayan *et al.* (2018b)). The second one requires the specification of a so-called *width vector* $(N_1, \ldots, N_M)$ that constraints the number of nodes at each stage and is feasible for scenario trees with more scenarios and stages (instances up to 2×10^6 scenarios and 20 stages are easily computed in Keutchayan *et al.* (2018a)). The first procedure will be referred to as $\mathrm{P}_1(N)$, where N refers to the number of scenarios, and the second one as $\mathrm{P}_2(N_1, \ldots, N_M)$. We also introduce a third procedure, $\mathrm{P}_3(N)$, that builds on P_2 and requires only the number of scenarios N.

Procedure $\mathrm{P}_1(N)$ **(Keutchayan *et al.*, 2018b):** Let $\mathcal{T}$ be a given tree structure of size $|\mathcal{N}_M| = N$ and $\mathfrak{D}$ be a given discretization method. The minimum of $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma^\kappa)$ with respect to $(\mathcal{P}, \mathcal{W})$ generated by $\mathfrak{D}$ ($\mathcal{T}$ and $\mathfrak{D}$ being fixed) is achieved by assigning the points and weights in $(\mathcal{P}, \mathcal{W})$ to the tree nodes in an optimal way. Such optimal assignment depends on the properties of the guidance functions, we refer to (Keutchayan *et al.*, 2018b, Propositions 4.2 and 4.3) for a formal description of them. Broadly speaking, it consists in matching nodes with fine outgoing discretization to couples (ζ^n, w^n) with high value of $W^n \gamma_m^\kappa(\zeta^{\cdot n})$, and vice versa, so that the scenarios that induce a high variability of the recourse functions are assigned to the part of the tree structure with dense branching.

The figure of demerit obtained by an optimal assignment of points and weights for a given structure $\mathcal{T}$ and discretization method $\mathfrak{D}$ is

$$\min_{(\mathcal{P},\mathcal{W})} \{\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma^\kappa) : (\mathcal{P}, \mathcal{W}) \text{ generated by } \mathfrak{D}\}. \tag{7.33}$$

Thus, a procedure to get a complete triple $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ consists in enumerating all (or some) tree structures $\mathcal{T}$, then assigning optimally the points and weights to obtain the demerit (7.33) of each one, and finally retaining the one with the lowest demerit. Since the space of all tree structures is finite, enumerating all of them guarantees that the output scenario tree is the one of minimum figure of demerit over all possible scenario trees (of a given size)

generated from the method $\mathfrak{D}$. That is to say this procedure guarantees that the output scenario tree is the minimum of

$$\min_{(\mathcal{T},\mathcal{P},\mathcal{W})} \{\mathcal{M}(\mathcal{T},\mathcal{P},\mathcal{W};\Gamma^\kappa) : |\mathcal{N}_M| = N,\ (\mathcal{P},\mathcal{W}) \text{ generated by } \mathfrak{D}\}. \tag{7.34}$$

However, if the space of tree structures is too large (this will often be the case in practice), an exhaustive search is impossible and an alternative approach is to consider heuristic exploration methods such as those based on *variable neighborhood search* (Hansen *et al.*, 2019).

Note that the scenario tree minimizing (7.34) need not be the optimal solution of the general minimization problem (7.7) since we have restricted ourselves to scenarios generated by a specific discretization method.

Procedure $\mathrm{P}_2(N_1,\ldots,N_M)$ **(Keutchayan *et al.*, 2018a):** An efficient approach to generate scenario trees with more stages and scenarios consists in enforcing the number of nodes per stage and then minimizing the figure of demerit recursively stage-by-stage by moving forward in time. Let $(N_1,\ldots,N_M)$ denote the number of nodes per stage, with $1 \leq N_1 \leq N_2 \leq \cdots \leq N_M$. This approach solves recursively from $m = 1$ to $m = M$ the minimization problem:

$$\min_{(J_1,\ldots,J_{N_m})\in\mathbb{N}_+^{N_m}} \quad \sum_{i=1}^{N_t} \frac{W^{n_i}\gamma_m^\kappa(\zeta^{\cdot n_i})}{J_i^\alpha} \quad \text{subject to} \quad \sum_{i=1}^{N_m} J_i = N_{m+1}. \tag{7.35}$$

At each stage the nodes in $\mathcal{N}_m$ are indexed such that $\mathcal{N}_m = \{n_1,\ldots,n_{N_m}\}$. The minimizing vector $(J_1^*,\ldots,J_{N_m}^*)$ is the number of child nodes to assign at $(n_1,\ldots,n_{N_m})$. Once the child nodes are added to the tree structure, the points and weights are generated by the discretization method $\mathfrak{D}$ and the problem (7.35) is solved anew at stage $m+1$.

Since this procedure builds the scenario tree gradually from the root node to the leaves, and not as a whole, we have no guarantee that the output scenario tree is the one of lowest demerit. However, we can expect that the output is somehow close to the minimum of

$$\min_{(\mathcal{T},\mathcal{P},\mathcal{W})} \{\mathcal{M}(\mathcal{T},\mathcal{P},\mathcal{W};\Gamma^\kappa) : |\mathcal{N}_1| = N_1,\ldots,|\mathcal{N}_M| = N_M,\ (\mathcal{P},\mathcal{W}) \text{ generated by } \mathfrak{D}\}. \tag{7.36}$$

A drawback of this procedure is the fact that it requires the specification of the width vector $(N_1,\ldots,N_M)$, which greatly influences the estimation accuracy of the scenario trees. As opposed, the procedure $\mathrm{P}_1(N)$ only requires the number of scenarios and is able to compute the optimal numbers of nodes at the intermediate stages. However, this comes at the high

cost of trying many different structures, which is unpractical when one deals with many stages or scenarios.

For this reason we now introduce a new procedure, $\mathrm{P}_3(N)$, which combines the advantages of both procedures: it only requires the number of scenarios, N, and it can be used to generate scenario trees with many stages and scenarios.

Procedure $\mathrm{P}_3(N_M)$**:** This procedure consists in finding a way to compute a high quality width vector $(N_1, \ldots, N_M)$, given only the final number of nodes N_M. Finding such width vector is equivalent to finding a *fractional* bushiness $(b_0, \ldots, b_{M-1})$, as both are linked as follows:

$$N_1 = b_0 \in [1, \infty) \quad \text{and} \quad \frac{N_{m+1}}{N_m} = b_m \in [1, \infty), \quad m = 1, \ldots, M-1. \tag{7.37}$$

Thus, the problem can be transformed into finding a fractional bushiness $(b_0, \ldots, b_{M-1})$ such that $\Pi_{m=0}^{M} b_m = N_M$. Introducing the fractional bushiness in the figure of demerit (7.31) yields (take $\kappa = \infty$)

$$\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma) = \sum_{m=0}^{M-1} \frac{1}{(b_m)^\alpha} \sum_{n \in \mathcal{N}_m} W^n \gamma_m(\zeta^{\cdot n}). \tag{7.38}$$

The inner sum over $\mathcal{N}_m$ is the scenario-tree estimator of $\mathbb{E}[\gamma_m(S_m)]$. This shows that, as the number of nodes increases, the optimal bushiness is linked to the hierarchy of expectations: $\mathbb{E}[\gamma_m(S_m)]$ for $m = 0, \ldots, M-1$. The higher $\mathbb{E}[\gamma_m(S_m)]$, the higher the average variability at stage m, therefore the higher b_m to ensure a fine discretization out of the nodes $\mathcal{N}_m$. The fractional bushiness that minimizes the figure of demerit is found by solving

$$\min_{(b_0, \ldots, b_{M-1}) \in [1, \infty)^M} \sum_{m=0}^{M-1} \frac{\mathbb{E}[\gamma_m(S_m)]}{(b_m)^\alpha} \quad \text{subject to} \quad \prod_{m=0}^{M-1} b_m = N_M. \tag{7.39}$$

Since the feasible set

$$\left\{ (b_0, \ldots, b_{M-1}) \in [1, \infty)^M \;\middle|\; \prod_{m=0}^{M-1} b_m = N_M \right\}, \tag{7.40}$$

is compact (closed and bounded) and the objective function is continuous, a global minimum of (7.39) must exist. The standard theory of nonlinear constraint programming allows to finding its closed-form. It is given as follows:

Proposition 7.3.3. *The fractional bushiness minimizing* (7.39) *is given by*

$$b_m^* = (N_M)^{1/M} \frac{\mathbb{E}\left[\gamma_m(S_m)\right]^{1/\alpha}}{\left(\prod_{i=0}^{M-1} \mathbb{E}\left[\gamma_i(S_i)\right]^{1/\alpha}\right)^{1/M}}, \quad m = 0, \ldots, M-1, \tag{7.41}$$

whenever all inequality constraints are inactive at $(b_0^*, \ldots, b_{M-1}^*)$*, i.e.,* $b_m^* > 1$ *for all* m*. If some inequality constraints are active, let* $\mathcal{A} \subseteq \{0, \ldots, M-1\}$ *be the active set and* $\mathcal{I}$ *its complement. Then the set* $\mathcal{A}$ *and the value of* b_m^* *for* $m \in \mathcal{I}$ *satisfy*

$$\exists \mu \in \mathbb{R} \quad \textit{such that} \quad \begin{cases} \dfrac{\mathbb{E}\left[\gamma_m(S_m)\right]}{(b_m^*)^\alpha} = \mu & \textit{for all } m \in \mathcal{I}; \quad (7.42) \\ \mathbb{E}\left[\gamma_m(S_m)\right] \leq \mu & \textit{for all } m \in \mathcal{A}. \quad (7.43) \end{cases}$$

Proof. See Appendix D. □

The formula (7.41) has an intuitive interpretation: If the expectations are all equal, then all branching coefficients b_m^* should have the same value, which is necessarily $(N_M)^{1/M}$. This would happen if the average variability of the recourse function (here, the option price function) is constant across the stages. Conversely, if the average variability varies with the stages, then the value of b_m^* is determined by the importance of the stage-m average variability (the numerator in (7.41)) as compared to the geometric average of all average variabilities (the denominator). This fraction, which is greater than one if stage m has a higher variability than average and lower than one otherwise, determined the fraction of $(N_M)^{1/M}$ that b_m^* will get. As for the role of the rate of convergence α, a value $\alpha \leq 1$ induces more diversity in the branching coefficients than $\alpha \geq 1$. Indeed, the smaller α the more difficult it is to approximate integrals by finite sums, and therefore the more important it becomes to adapt the branching structure to the variability of the problem.

In the case of a Bermudan-Asian option the guidance function's expectation boils down to $\mathbb{E}[\gamma_m(S_m)] = u_{m+1}(\delta)$, because $\{S_m\}$ satisfies $\mathbb{E}[S_m] = \delta^{-m}$ under the risk-neutral measure. Since the sequence $\{u_m(\delta)\}$ is decreasing, the inequality (7.43) will be satisfied first by the highest index, which simplifies the search of the active set. The above proposition then leads to the following simple procedure to compute the optimal fractional bushiness:

(1) Set $\mathcal{I} = \{0, \ldots, M-1\}$ and $\mathcal{A} = \emptyset$;

(2) Compute

$$b_m^* = \begin{cases} (N_M)^{1/|\mathcal{I}|} \dfrac{(u_{m+1}(\delta))^{1/\alpha}}{\left(\prod_{i\in\mathcal{I}} (u_{i+1}(\delta))^{1/\alpha}\right)^{1/|\mathcal{I}|}}, & m \in \mathcal{I}; \quad (7.44) \\ 1 & m \in \mathcal{A}; \quad (7.45) \end{cases}$$

(3) If $b_m^* > 1$ for all $m \in \mathcal{I}$, then STOP: the optimal fractional bushiness is $(b_0^*, \ldots, b_{M-1}^*)$;

(4) Otherwise: $\mathcal{I} \leftarrow \mathcal{I} \setminus \{\max \mathcal{I}\}$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{\max \mathcal{I}\}$, and go to step (2).

At step (2) the formula (7.44) follows directly from (7.42). At step (4) the highest index in $\mathcal{I}$ is added to the active set $\mathcal{A}$. This is specific to the Asian option pricing problem: the sequence of expectations $\{\mathbb{E}[\gamma_m(S_m)]\}$ is decreasing, therefore the first index that satisfies the inequality (7.43) is necessarily the highest index in $\mathcal{I}$. In a general setting of multistage problems, the index added to the active set would be $m^* = \operatorname{argmin}_{m\in\mathcal{I}} \mathbb{E}[\gamma_m(S_m)]$.

Once the optimal fractional bushiness is found, the width vector follows from (7.38) by rounding-off $N_1, \ldots, N_M$ to the closest integers when necessary.

Note that since we consider pricing problems under risk-neutral measure, we do not need to impose a lower bound on the branching coefficients to ensure the absence of arbitrage, as in the case of portfolio management problems under real-world measure (see Klaassen (1998, 2002) and Geyer *et al.* (2010, 2013)). However, when low-demerit scenario trees are built for such problems, a lower bound on J_i and b_m can be added in the constraints of problems (7.35) and (7.39) to meet the non-arbitrage requirement.

7.3.3 Figure of demerit analysis: Symmetrical vs. Low demerit scenario trees

The above Proposition 7.3.3 allows to compare the figure of demerit of symmetrical scenario trees with that of low-demerit trees. This comparison provides insights into the type of situations where the low demerit scenario trees are expected to outperform the symmetrical ones in terms of estimation accuracy, and conversely, the situations where they will essentially perform identically.

By introducing the optimal fractional bushiness into the minimization problem (7.39), we find that the minimum value of (7.39), denoted $\mathcal{M}^{\mathrm{LD}}$ as it is the figure of demerit of low

demerit (LD) scenario trees, simplifies to

$$\mathcal{M}^{\mathrm{LD}} = \sum_{m=0}^{M-1} \frac{\mathbb{E}\left[\gamma_m(S_m)\right]}{(b_m^*)^\alpha} \leq \sum_{m=0}^{M-1} \frac{1}{(N_M)^{\alpha/|\mathcal{I}|}} \left(\prod_{i\in\mathcal{I}} \mathbb{E}\left[\gamma_i(S_i)\right]\right)^{1/|\mathcal{I}|} \tag{7.46}$$

$$= \frac{M}{(N_M)^{\alpha/|\mathcal{I}|}} \left(\prod_{i\in\mathcal{I}} \mathbb{E}\left[\gamma_i(S_i)\right]\right)^{1/|\mathcal{I}|}, \tag{7.47}$$

where the inequality holds from (7.42)-(7.43). Thus,

$$\mathcal{M}^{\mathrm{LD}} \leq \frac{M}{(N_M)^{\alpha/|\mathcal{I}|}} \, \mathrm{G_M}(\{\mathbb{E}\left[\gamma_m(S_m)\right] : m \in \mathcal{I}\}), \tag{7.48}$$

where $\mathrm{G_M}(\cdot)$ denotes the *geometric mean* of a finite set of positive numbers.

By comparison, the figure of demerit of a symmetrical (SYM) scenario tree with constant bushiness $(b, \ldots, b)$, where b necessarily equals $(N_M)^{1/M}$, simplifies to

$$\mathcal{M}^{\mathrm{SYM}} = \sum_{m=0}^{M-1} \frac{\mathbb{E}\left[\gamma_m(S_m)\right]}{b^\alpha} = \frac{1}{(N_M)^{\alpha/M}} \sum_{m=0}^{M-1} \mathbb{E}\left[\gamma_m(S_m)\right], \tag{7.49}$$

hence

$$\mathcal{M}^{\mathrm{SYM}} = \frac{M}{(N_M)^{\alpha/M}} \, \mathrm{A_M}(\{\mathbb{E}\left[\gamma_m(S_m)\right] : m = 0, \ldots, M-1\}), \tag{7.50}$$

where $\mathrm{A_M}(\cdot)$ denotes the *arithmetic mean* of a finite set of positive numbers.

From (7.48) and (7.50) we can now deduce the difference between the figures of demerit of symmetrical and low-demerit scenario trees. The former features the geometric mean of the stagewise expected variability, whereas the latter features the arithmetic mean. For a set of positive numbers, the geometric mean is always lower or equal to the arithmetic mean, and the difference between the two means increases as the numbers are spread apart from their arithmetic mean. This is the concept of *mean-preserving spread*, which keeps the arithmetic mean constant but makes the geometric mean decrease, see, e.g., Mitchell (2004). Thus, in a multistage problem, the difference between $\mathcal{M}^{\mathrm{LD}}$ and $\mathcal{M}^{\mathrm{SYM}}$ will be most significant when the variability of the recourse functions are very different across the stages. This happens, for instance, when the variability decreases over time with the one at the early stages being much larger than the one at the late stages (as in the case of the Bermudan-Asian option pricing problem, see Figure 7.2). But we note that the variability need not decrease over time. The same difference will happen if the variability increases over time, or even fluctuates up and down. In any case, the scenario tree built from minimizing the figure of demerit will adapt its branching structure based on these fluctuations, in order to get the best estimate

for a given number of scenarios.

Another notable difference has to do with the convergence rate as the number of scenarios N_M increases. This rate is $(N_M)^{-\alpha/M}$ for symmetrical trees and $(N_M)^{-\alpha/|\mathcal{I}|}$ for low-demerit trees, where $\mathcal{I}$ is the set of indices $m \in \{0, \ldots, M-1\}$ such that $b_m^* > 1$. When the expectations in the set $\{\mathbb{E}[\gamma_m(S_m)] : m = 0, \ldots, M-1\}$ are spread far apart, meaning that some stages have high variability while others do not, the low-demerit tree does not branch out of the stages in $\mathcal{A}$ (those with low variability) to focus exclusively on the stages in $\mathcal{I}$ (those with high variability). This results in a convergence rate of the form $\alpha/|\mathcal{I}|$, thus faster than the α/M rate of symmetrical trees. We note that the rate holds within a finite (possibly very large) number of scenarios, but not asymptotically per se. Indeed, when N_M becomes large enough (how large will depend on how spread the set $\{\mathbb{E}[\gamma_m(S_m)]\}$ is), all branching coefficients b_m will enter the inactive set $\mathcal{I}$, and the rate of convergence will asymptotically become similar to that of symmetrical trees. However, this asymptotic behavior may appear only beyond a number of scenarios computational unreachable, and hence one may enjoy a faster convergence rate up any instance that can be possibly tested (as in the Bermudan-Asian option pricing problem when $M = 13$; see next section).

In the next section, we will see that the above analysis made about the figures of demerit is extremely close to what is observed experimentally for the estimation errors. The errors are found to decrease at a rate of the form $(N_M)^{-\omega(M)}$ with an exponent $\omega(M) \simeq \alpha/M$ for symmetrical trees and $\omega(M) > \alpha/M$ for low-demerit trees. The fact that $\omega(M)$ is greater than α/M shows that low-demerit trees discretize the multistage problem as if it had less than M stages. This then leads to the idea that the problem has an *effective number of stages* smaller than the actual number of stages M.

7.4 Numerical experiments

In this section we implement the theoretical framework developed in the previous sections and in Keutchayan *et al.* (2018a,b). This framework claims that the *figure of demerit* is a suitable quality measure to generate high quality scenario trees to solve approximately multistage stochastic optimization problems –in our case, the Bermudan-Asian option pricing problem. Throughout this section we will consider twelve instances of this problem, which were first considered and solved by Ben-Ameur *et al.* (2002) using a dynamic programming procedure based on finite-element method. These instances and the corresponding estimates of the option prices are summarized in Table 7.1.

Their method relies heavily on the Markovian property of the asset price dynamic. It relies

Table 7.1 Benchmark option prices. Parameters are $r = 0.05$, $S_0 = 100$ and σ, T, K, and M given in the table.

(σ, T, K)	$M = 2$	$M = 4$	$M = 13$
$(0.25, 0.25, 100)$	4.395	3.920	3.650
$(0.15, 0.25, 100)$	2.842	2.512	2.321
$(0.25, 0.50, 100)$	6.463	5.745	5.332
$(0.25, 0.50, 105)$	4.245	3.475	2.966

also on the fact that Asian options, although being path-dependent, can still be modeled as a Markovian decision process over a low-dimensional state space. Indeed, augmenting the state space by one dimension to include the running averages of the asset price is all it takes to recover the Markov property.

The price estimates obtained by Ben-Ameur *et al.* (2002) are of very high accuracy. Thus, as far as we are concerned, we will consider those estimates to be the *exact* option prices, which will provide the benchmark for our comparison. Let us stress that we want to compare different types of scenario trees *with each other*. We do not compare scenario trees with methods that solve the option pricing problem by exploiting the Markovian property of the decision process. The reason is that these two types of approaches –those exploiting the Markov property and those that work without it– perform well under their own paradigm and should be used in different situations in practice.

We will focus on comparing the *problem-driven* scenario trees obtained by minimizing the figure of demerit with the *problem-blind* scenario trees built using the same discretization methods but with a symmetrical structures of constant integer bushiness.

Overall, the results that we obtain in this section can be summarized as follows: First, in Section 7.4.1, we establish the strong positive correlation between the figure of demerit and the estimation error. Then, in Section 7.4.2, we demonstrate that low demerit scenario trees enjoy faster error convergence than symmetrical trees as the number of scenarios increases and, in Section 7.4.3, we estimate what this faster convergence implies in terms of the effective number of stages. Finally, in Section 7.4.4, we show that the benefit of low demerit trees over symmetrical trees in terms of error reduction and scenario reduction is substantial and typically increases with the number of stages.

7.4.1 Correlation between figure of demerit and estimation error

The goal of this section is to demonstrate experimentally that the figure of demerit (FOD) is a relevant quality measure for scenario trees. To this end, we study the statistical correlation

between the FOD and the estimation error that results from solving the Bermudan-Asian option pricing problem using scenario trees. By demonstrating that the correlation coefficient is close to one, and that scenario trees of lowest demerit and of lowest error are very close (sometimes identical), we will validate the idea of using the FOD as a criterion to select suitable scenario trees.

Correlation for procedure $\mathrm{P}_1(N)$**:**

We first consider the option pricing problem with two monitoring/exercise dates, i.e., $M = 2$. We use the procedure $\mathrm{P}_1(N)$ to generate *all tree structures* $\mathcal{T}$ such that

$$|\mathcal{N}_2| = N = 25 \quad \text{and} \quad |\mathcal{N}_1| \geq 2. \tag{7.51}$$

Each structure $\mathcal{T}$ is then filled with the points and weights $(\mathcal{P}, \mathcal{W})$ of the discretization method $\mathfrak{D}$ using the optimal assignment of nodes, so as to ensure that the scenario-tree demerit is given by (7.33). Each scenario tree $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ is then used to solve the Bermudan-Asian option pricing problem. The error (in percentage of the true value) is recorded alongside the FOD as a two-dimensional point, and the correlation between these two measures is estimated. Table 7.2 displays the correlation coefficients for each instances of the problem and for different values of κ. Figures 7.3 and 7.4 illustrate how these two-dimensional data points spread in the plane.

We observe that the correlation between the FOD and the estimation error is very high for each discretization method, typically ranging between 0.75 and 0.90. Moreover, we can see on Figures 7.3 and 7.4 that, out of the 1957 tree structures that satisfy (7.51), the one of lowest error actually coincides with the one of lowest demerit for two methods, namely QMC-Lat and OQ-W_1. Even when the scenario trees do not coincide (as in the case of OQ-W_2), the tree structure of lowest demerit still provides a relatively low error as compared with all the other possible structures. This error is much smaller than that of the symmetrical scenario tree with constant branching factor, which is also represented on the figures.

The scenario trees of minimum error and of minimum demerit are displayed in Table 7.3 for each discretization method. We see that they discretize more finely the asset price at stage 1 than at stage 2. This matches the theoretical analysis of the previous sections. The fractional bushiness ranges from $(10, 2.5)$ to $(7, 3.6)$, which means that the scenario trees typically discretize two to four times more S_1 than S_2. This is significantly different from the integer bushiness $(5, 5)$, which explains why the symmetrical structure has such poor performance.

As for the branching factor out of the stage-1 nodes, we see that it varies depending on the value of the discretization point ζ^n and weight w^n of S_1. For QMC-Lat (equal-weight rule), the higher ζ^n, the higher $\gamma_1^\kappa(\zeta^n)$, and hence the higher the branching factor out of n. For OQ-W$_1$ and OQ-W$_2$ (not equal-weight rules), nodes with extremely high values of ζ^n have high values of $\gamma_1^\kappa(\zeta^n)$ but small weights w^n, since they are at the tail of the distribution. Since the branching factor is determined by the hierarchy of products $w^n\gamma_1^\kappa(\zeta^n)$, this results in nodes close to the mean of the distribution having larger branching factors than those at the right tail. As for the nodes at the left tail, their variability is cut off to zero due to the cut-off parameter $\kappa < \infty$. This is why there is no more than one branch out of these nodes.

In Figures 7.3 and 7.4 the cut-off parameter was set to 1.2. The strong correlation between the FOD and the error still holds for other values of κ and for different instances of the problem, as shown in Table 7.2. We see that the coefficient correlation is relatively constant across the values of κ. It drops slightly when $\kappa = \infty$ (no cut-off) and when $\kappa \leq 0.8$ (sharp cut-off). This also validates experimentally the idea of introducing a cut-off term in the guidance functions, as we see that it tightens the correlation between the figure of demerit and the estimation error.

Table 7.2 Correlation coefficients $\rho_{\text{FOD,Error}}$ for QMC-Lat (top), OQ-W$_1$ (middle), OQ-W$_2$ (bottom) for the scenario trees generated by procedure P$_1$. ($M = 2$, $N = 25$). (Data in bold font single out correlations $\rho_{\text{FOD,Error}} \geq 0.75$.)

(σ, T, K)	$\kappa = 0.4$	$\kappa = 0.8$	$\kappa = 1.2$	$\kappa = 1.6$	$\kappa = \infty$
$(0.25, 0.25, 100)$	**0.84**	**0.90**	**0.91**	**0.90**	**0.84**
	0.80	**0.79**	**0.76**	0.67	0.52
	0.57	**0.75**	**0.80**	**0.80**	0.73
$(0.15, 0.25, 100)$	**0.82**	**0.88**	**0.89**	**0.88**	**0.84**
	0.74	0.74	0.73	0.64	0.49
	0.57	0.70	**0.79**	**0.78**	0.71
$(0.25, 0.50, 100)$	**0.85**	**0.90**	**0.91**	**0.90**	**0.85**
	0.83	**0.85**	**0.82**	0.73	0.61
	0.65	**0.77**	**0.84**	**0.84**	**0.79**
$(0.25, 0.50, 105)$	**0.82**	**0.91**	**0.93**	**0.89**	0.71
	0.88	**0.90**	**0.85**	**0.75**	0.44
	0.57	0.72	**0.82**	**0.84**	**0.76**

Correlation for procedure P$_2(N_1, \ldots, N_M)$**:**

The previous analysis dealt with the case of two monitoring dates ($M = 2$) and tree structures with 25 scenarios being enumerated exhaustively. To push the analysis further, and to show

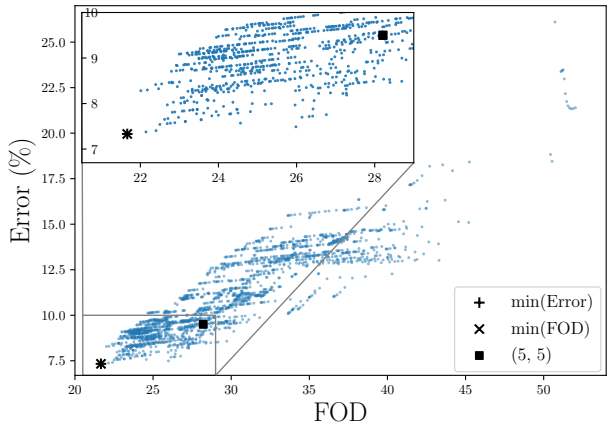


Figure 7.3 Correlation between the FOD and the estimation error for QMC-Lat. Each dot represents a tree structure given by (7.51). The scenario trees of lowest demerit and lowest error coincide. For comparison the symmetrical scenario tree of bushiness $(5, 5)$ is also displayed. ($\kappa = 1.2$ and $(\sigma, T, K) = (0.25, 0.25, 100)$)

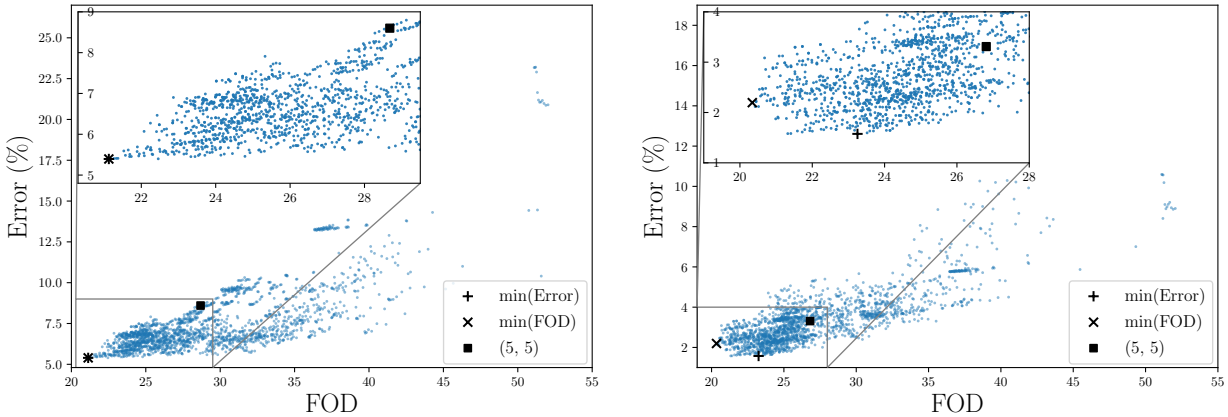
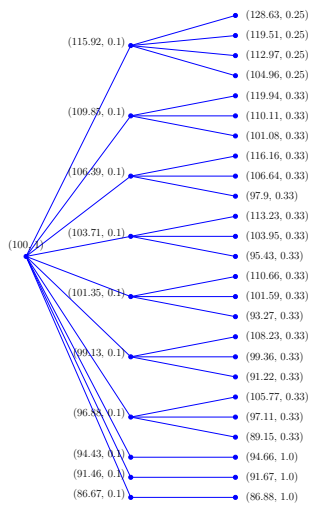
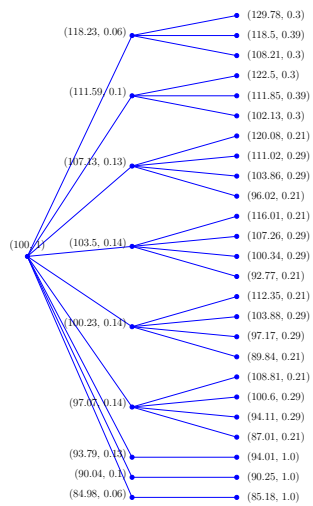
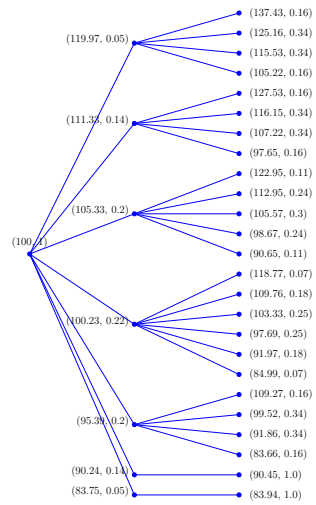
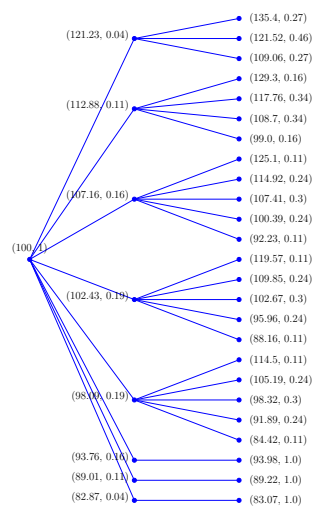


Figure 7.4 Correlation between the FOD and the estimation error for OQ-W_1 (left) and OQ-W_2 (right). In the left figure the scenario trees of lowest demerit and lowest error coincide, in the right figure they are different but still close relatively to all other structures. ($\kappa = 1.2$ and $(\sigma, T, K) = (0.25, 0.25, 100)$)

Table 7.3 Scenario trees of minimum error and minimum FOD for the three discretization methods considered. Each points and weights are displayed next to the corresponding node. The fractional bushinesses (from left to right, top to bottom) are $(10, 2.5)$, $(9, 2.8)$, $(8, 3.1)$, and $(7, 3.6)$.

	QMC-Lat	OQ-W_1	OQ-W_2
min(Error)			
min(FOD)			

that the previous results still hold for more monitoring dates, we now deal with the case $M = 4$. To this end, we now use the procedure $\mathrm{P}_2(N_1, \ldots, N_4)$ to generate all tree structures such that

$$N_1 \leq 40, \quad N_4 = 81, \quad N_1 \geq \frac{N_2}{N_1} \geq \frac{N_3}{N_2} \geq \frac{N_4}{N_3}. \tag{7.52}$$

The first two conditions constraint the number of nodes at stage 1 and 4, respectively. The third condition ensures that the scenario trees will be of decreasing fractional bushiness, i.e., that they will sample more S_1 than S_2, S_2 than S_3, etc.

The option pricing problem is solved using each scenario tree. The estimation error and the figure of demerit are computed and plotted as a point in a two-dimensional plane as before. The results are displayed in Figure 7.5 for QMC-Lat and in Figure 7.6 for OQ-W_1 and OQ-W_2. The correlation appears to be even stronger for $M = 4$ than for $M = 2$, as all points lie very tightly around a straight line of positive slope. As a result, the correlation coefficient is almost one for all methods as we can see in Table 7.5.

The scenario trees of lowest error and lowest FOD are again very close one another, although not strictly identical. Their fractional bushinesses are represented in Table 7.4. For comparison, we also display on the figures the estimation error and FOD of the symmetrical tree structures that satisfy the condition (7.52), which are specifically those with bushiness $(3, 3, 3, 3)$, $(9, 3, 3, 1)$, $(9, 9, 1, 1)$, and $(27, 3, 1, 1)$. Once again, symmetrical structures have very poor performance as compared to low demerit structures.

7.4.2 Error vs. number of scenarios

Now that we have established the strong connection between the figure of demerit (7.31) and the estimation error, we shall analyze the benefit of using low demerit scenario trees over symmetrical scenario trees in terms of the error convergence as the number of scenarios increases. Since computing structures of low demerit requires more work than generating plain symmetrical structures, the goal of this section is to show that getting low demerit structures is worth the extra work as they consistently provide faster convergence of the error.

In this section the scenario trees are generated through the procedure $P_3(N_M)$ for different values of N_M and M. Recall that P_3 extends P_2 by computing a suitable fractional bushiness from the expected values of the guidance functions at each stage. For illustration, we display in Table 7.6 some fractional bushinesses suitable for the Bermudan-Asian option pricing problem with $M = 4$ and $M = 13$. Low demerit scenario trees provide much finer discretiza-

Table 7.4 Fractional bushiness of the scenario trees in Figures 7.5 and 7.6.

	QMC-Lat	OQ-W_1	OQ-W_2
min(Error)	$(7, 5.4, 2.1, 1)$	$(8, 5.6, 1.7, 1.1)$	$(6, 4.2, 2.6, 1.2)$
min(FOD)	$(8, 4.0, 2.5, 1.0)$	$(10, 4.0, 2.0, 1.0)$	$(8, 4.6, 2.0, 1.1)$

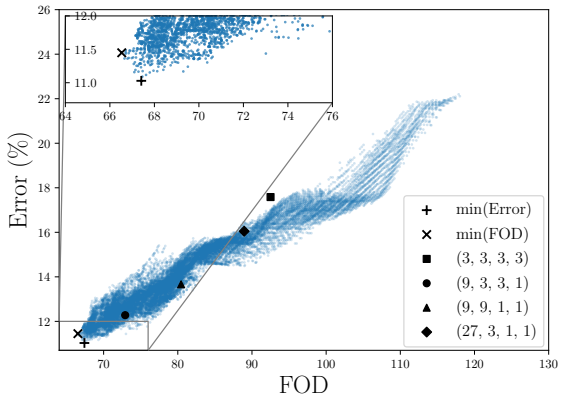


Figure 7.5 Correlation between the FOD and the estimation error for QMC-Lat. Each dot in the plane represents a scenario tree generated by procedure $\mathrm{P}_2(N_1, \ldots, N_4)$ with $(N_1, \ldots, N_4)$ given by (7.52). For comparison we also plot various symmetrical structures. ($\kappa = 1.2$ and $(\sigma, T, K) = (0.25, 0.25, 100)$)

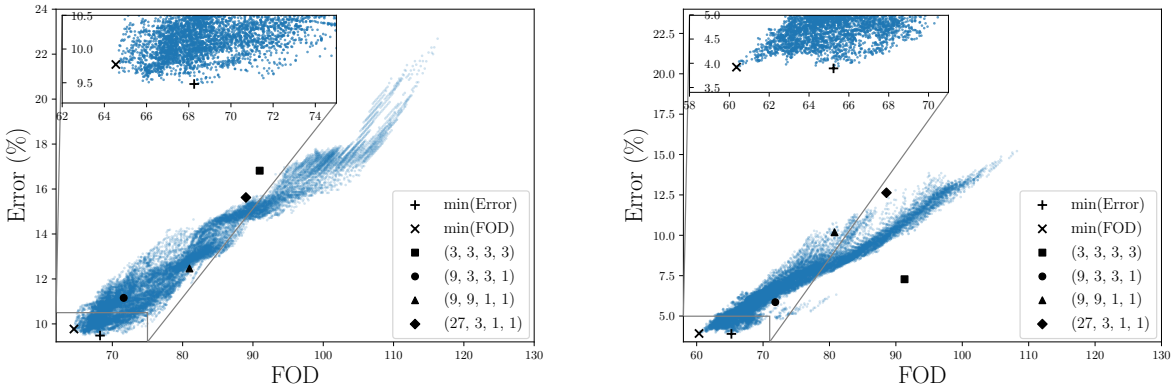


Figure 7.6 Correlation between the FOD and the estimation error for OQ-W_1 (left) and OQ-W_2 (right). ($\kappa = 1.2$ and $(\sigma, T, K) = (0.25, 0.25, 100)$)

Table 7.5 Correlation coefficients $\rho_{\text{FOD,Error}}$ for QMC-Lat (top), OQ-W$_1$ (middle), OQ-W$_2$ (bottom) for the scenario trees generated by procedure P$_2$.($M = 4$, $N_4 = 81$) (Data in bold font single out correlations $\rho_{\text{FOD,Error}} > 0.95$)

(σ, T, K)	$\kappa = 0.4$	$\kappa = 0.8$	$\kappa = 1.2$	$\kappa = 1.6$	$\kappa = \infty$
$(0.25, 0.25, 100)$	0.93	**0.98**	**0.97**	**0.97**	0.94
	0.92	**0.98**	**0.98**	**0.97**	0.95
	0.78	**0.97**	**0.98**	**0.98**	**0.98**
$(0.15, 0.25, 100)$	0.94	**0.98**	**0.98**	**0.97**	0.95
	0.92	**0.97**	**0.97**	**0.97**	0.94
	0.85	**0.97**	**0.98**	**0.98**	**0.98**
$(0.25, 0.50, 100)$	0.95	**0.98**	**0.98**	**0.97**	0.95
	0.92	**0.98**	**0.98**	**0.97**	0.95
	0.83	**0.97**	**0.98**	**0.98**	**0.98**
$(0.25, 0.50, 105)$	0.84	**0.97**	**0.96**	0.95	0.92
	0.77	**0.98**	**0.98**	**0.98**	0.95
	0.41	0.94	**0.96**	**0.96**	**0.98**

tion at the early stages, at the expense of the late stages, because the average variability decreases over time (see Figure 7.1).

Since the procedure P$_3(N_M)$ only requires the number of scenarios, the only parameter that remains to be set to generate scenario trees is the cut-off parameter κ. In the previous section we saw that an appropriate value of κ (generally $0.8 \leq \kappa < \infty$) tightens the correlation between the figure of demerit and the estimation error. This observation was done for a *fixed* number of scenarios. We now want to analyze the role of κ as far as the *asymptotic* behavior of the error is concerned.

Table 7.6 Fractional bushinesses computed by procedure P$_3(N_M)$. ($\delta = 0.99$)

N_M	$M = 4$
81	$(7, 3.9, 2.4, 1.2)$
10^3	$(12, 7.8, 4.6, 2.3)$
10^4	$(22, 13.5, 8.2, 4.1)$
10^5	$(39, 24.1, 14.6, 7.3)$

N_M	$M = 13$
10^3	$(5, 3.8, 3.2, 2.5, 2.1, 1.8, 1.5, 1.2, 1, 1, 1, 1, 1)$
10^4	$(6, 5.3, 4.1, 3.3, 2.8, 2.3, 1.9, 1.5, 1.2, 1, 1, 1, 1)$
10^5	$(8, 6.5, 5.1, 4.2, 3.5, 2.9, 2.4, 2.0, 1.5, 1.2, 1, 1, 1)$
10^6	$(10, 8.1, 6.4, 5.3, 4.4, 3.6, 3.0, 2.4, 1.9, 1.5, 1.1, 1, 1)$

Figure 7.7 shows the error convergence for different values of κ for the discretization method OQ-W_2. We observe that for a sharp cut-off (i.e., $\kappa \leq 1.2$) the error does not seem to converge to zero. For such values, the scenario-tree method no longer provides *consistent* estimators of the option price. The reason is that a sharp cut-off means that the scenario tree is not branching out on some scenarios where it should have done so. This results in the scenario tree not solving the actual option pricing problem but a slightly different one, hence the convergence is toward a different option price. For larger values of κ, the error does seem to converge to zero, with the fastest convergence occurring for $\kappa = 2$ within the numbers of scenarios tested. However, this is true because we can only look up to a finite number of scenarios. As a matter of fact, any $\kappa < \infty$ will see a convergence of the scenario-tree estimator towards a different value of the option price. This value, however, will be only slightly (possibly insignificantly) different from the true option value if κ is large enough. As a result, the cut-off parameter improves the scenario-tree estimation up to any finite number of scenarios, but not asymptotically per se. Since the truly asymptotic rate of convergence is anyway out of reach of any machine with finite computational resources, we want to choose the parameter κ that provides the fastest convergence within a finite, computationally achievable, number of scenarios (say, 10^5). It appears that $\kappa \simeq 2$ is the right value for that.

Having set the cut-off parameter to an appropriate value, we now generate different sizes of scenario trees to solve the option pricing problem. We plot in Figures 7.8 and 7.9 the estimation error against the number of scenarios N_M in a $\log_{10}$-$\log_{10}$ scale for $M = 4$ and $M = 13$. The instance considered is $(\sigma, T, K) = (0.25, 0.25, 100)$. The scenario trees are generated by the procedure P_3 along with the three discretization methods previously considered: QMC-Lat, OQ-W_1, OQ-W_2. The number of scenarios ranges from 16 to 10^5 for $M = 4$. For $M = 13$, the symmetrical trees are computed for the only two tractable sizes: 2^{13} $(= 8192)$ and 3^{13} $(= 1,594,323)$. We also display as a baseline the root-mean-square error of the symmetrical scenario trees generated by a standard Monte Carlo (MC) method, along with the 95% confidence interval. Overall, we see on both figures that the low demerit (LD) scenario trees consistently provide faster error convergence than the symmetrical (SYM) scenario trees of constant bushiness, and this holds independently of the discretization methods used.

Theses plots also reveal that the estimation error behaves approximately as a straight line in a log-log scale. This means that the error, denoted by $E(N_M, M)$ as a function of the number

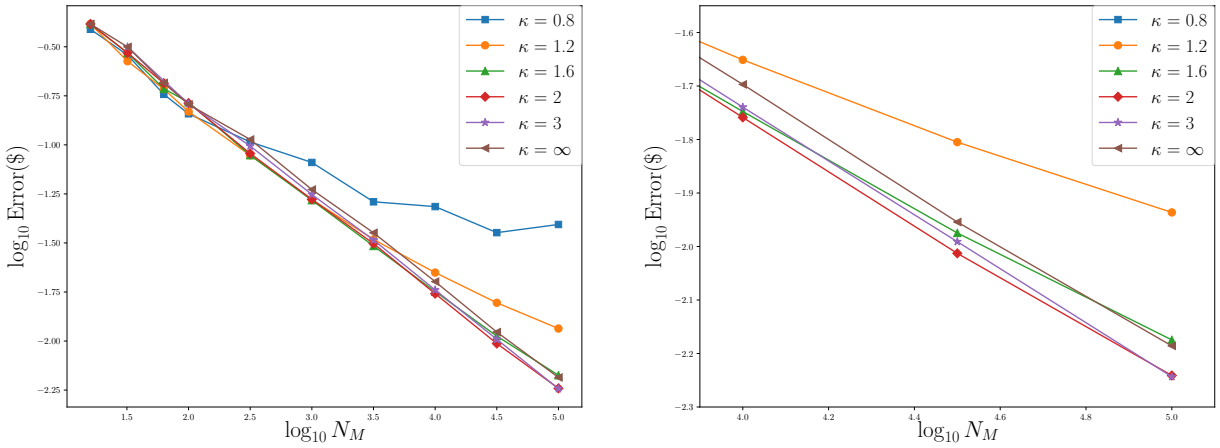


Figure 7.7 Estimation error vs. number of scenarios for different values of κ. (The right plot zooms in on the bottom right corner of the left plot.) ($M = 4$ and $\mathfrak{D} = \text{OQ-W}_2$)

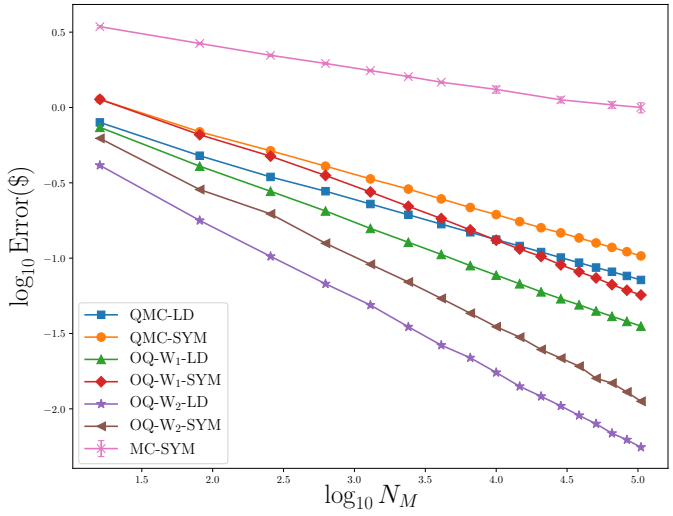


Figure 7.8 Estimation error vs. number of scenarios for $M = 4$.

Table 7.7 Coefficients $\omega(M)$ and $\Lambda(M)$ for the data in Figures 7.8 and 7.9. (Data in bold font single out the best values –largest ω, smallest Λ– between LD and SYM scenario trees.)

	$M=4$				$M=13$			
	$\omega(4)$		$\Lambda(4)$		$\omega(13)$		$\Lambda(13)$	
$\mathfrak{D}$	LD	SYM	LD	SYM	LD	SYM	LD	SYM
MC	–	0.142	–	4.920	–	0.064	–	0.584
QMC-Lat	0.269	0.269	**1.587**	2.311	**0.098**	0.083	**1.875**	2.477
OQ-W_1	**0.346**	0.344	**1.888**	3.120	**0.117**	0.093	**2.037**	2.716
OQ-W_2	**0.488**	0.455	**1.566**	2.299	**0.160**	0.135	**1.811**	2.415

Table 7.8 Coefficients $\omega(M)$ and $\Lambda(M)$ for QMC-Lat (top), OQ-W_1 (middle), OQ-W_2 (bottom). (Data in bold font single out the best values –largest ω, smallest Λ– between LD and SYM scenario trees.)

	$M=4$				$M=13$			
	$\omega(4)$		$\Lambda(4)$		$\omega(13)$		$\Lambda(13)$	
(σ, T, K)	LD	SYM	LD	SYM	LD	SYM	LD	SYM
$(0.15, 0.25, 100)$	**0.276**	0.271	**0.955**	1.331	**0.098**	0.084	**1.086**	1.435
	0.350	0.347	**1.106**	1.792	**0.117**	0.095	**1.194**	1.578
	0.484	0.454	**0.845**	1.290	**0.161**	0.136	**1.057**	1.400
$(0.25, 0.50, 100)$	**0.268**	0.265	**2.429**	3.401	**0.097**	0.083	**2.799**	3.631
	0.341	0.341	**2.826**	4.581	**0.115**	0.093	**3.020**	3.981
	0.475	0.448	**2.268**	3.344	**0.156**	0.134	**2.636**	3.548
$(0.25, 0.50, 105)$	0.265	**0.269**	**2.244**	3.511	**0.096**	0.079	**2.472**	3.141
	0.346	0.344	**2.714**	4.607	**0.114**	0.089	**2.698**	3.443
	0.474	0.462	**1.955**	3.705	**0.158**	0.132	**2.455**	3.177

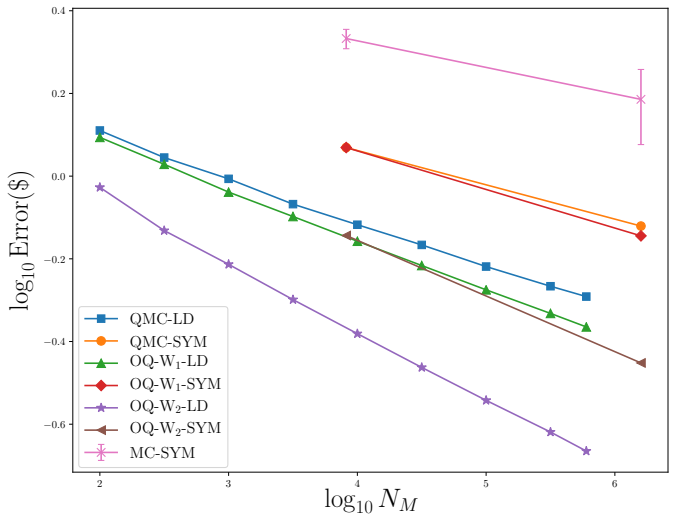


Figure 7.9 Estimation error vs. number of scenarios for $M = 13$.

of stages M and scenarios N_M, can be considered to be of the form

$$\mathrm{E}(N_M, M) \simeq \frac{\Lambda(M)}{(N_M)^{\omega(M)}}, \tag{7.53}$$

for some coefficients $\Lambda(M)$ and $\omega(M)$ that are solely functions of the number of stages. The logarithm of $\Lambda(M)$ corresponds to the y-intercept and $\omega(M)$ corresponds to the slope of the regression line fitting the data in the log-log scale.

To make the comparison between LD and SYM scenario trees more precise, we include in Table 7.7 the estimates of $\Lambda(M)$ and $\omega(M)$ for the four discretization methods considered in Figures 7.8 and 7.9. These results show that the faster error convergence of LD trees is mostly due to a much better $\Lambda(M)$ coefficient, but also to slightly but consistently better $\omega(M)$. This is also observed when solving the three other instances of the option pricing problem that we have considered; see the results in Table 7.8.

As for the quality of the discretization methods themselves, it appears that OQ-W$_2$ clearly outperforms QMC-Lat and OQ-W$_1$. This observation holds throughout all tested instances of the problem. In particular, it should be noticed that for $M = 4$ the method OQ-W$_2$ achieves a convergence rate $\omega(4) = 0.488$, when OQ-W$_1$ (the second best) achieves $\omega(4) = 0.346$ and Monte Carlo sampling (the worst) only achieves $\omega(4) = 0.142$. The fact that a convergence rate close to $1/2$ can be reached in a problem with 4 decision stages reveals how powerful some well-designed deterministic discretization method can be as compared to a crude Monte Carlo

sampling. Let us recall, however, that our pricing problem includes one random variable per stage (one asset), and that the convergence rate of Monte Carlo has the advantage of being insensitive to the number of random variables considered, which is typically not the case for deterministic methods.

7.4.3 Effective number of stages

In Section 7.3.3 we showed that the FOD of symmetrical and low-demerit scenario trees may have different convergence rates ($(N_M)^{-\alpha/M}$ and $(N_M)^{-\alpha/|\mathcal{I}|}$, respectively), if the stagewise expected variability of the recourse functions varies a lot over time, i.e., if the numbers in the set $\{\mathbb{E}[\gamma_m(S_m)]\}_m$ are spread far apart from their (arithmetic) average. In this section we show that this behavior matches precisely what we observe for the estimation errors.

We saw in Section 7.4.2 that the convergence rate of the estimation error is $(N_M)^{-\omega(M)}$, with some exponents $\omega(M)$ displayed in Figure 7.7 and 7.8. To compare the (experimental) exponent $\omega(M)$ with the (theoretical) exponents α/M or $\alpha/|\mathcal{I}|$, we need to estimate first the genuine rate of convergence α of the three discretization methods that we have considered. The most obvious way to do so is to solve numerically the Bermudan-Asian option pricing problem for $M = 1$, and deduce α by plotting the error against the number of scenarios (N_1) in a log-log scale. Since for $M = 1$ the option reduces to a European vanilla option, we can use the Black-Scholes formula to know the true price. However, doing so leads to very noisy errors towards the small values of N_1, which prevents an accurate and doubtless estimation of α. As a result, we rather consider the problem with $M = 2$, because the error plot is much less noisy than $M = 1$ (it behaves almost perfectly as a straight line as in the cases $M = 4$ and $M = 13$ seen previously). By estimating $\omega(2)$ for each discretization method, we find that their respective rate of convergence α is

$$\alpha = 2 \times \omega(2) \simeq \begin{cases} 1.05 & \text{for QMC-Lat;} \quad (7.54) \\ 1.17 & \text{for OQ-W}_1\text{;} \quad (7.55) \\ 1.91 & \text{for OQ-W}_2\text{.} \quad (7.56) \end{cases}$$

Given these estimates of α, we can now define the *effective number of stages* of a multistage problem with M actual stages as the number M_{eff} that solves $\omega(M) = \alpha/M_{\text{eff}}$. According to the analysis made about the figure of demerit in Section 7.3.3, we should have $M_{\text{eff}} \simeq M$ for the symmetrical trees, and $M_{\text{eff}} < M$ for low-demerit trees when the problem has heterogeneous variability across the stages and the number of stages is too large to allow fine branching at all stages.

The estimates of M_{eff} are displayed in Table 7.9. We see that for $M = 4$ the effective number

of stages is roughly equal to the actual number M for both LD ($M_{\text{eff}} = 3.9, 3.4, 3.9$) and SYM ($M_{\text{eff}} = 3.9, 3.4, 4.2$). This matches the theoretical analysis: when $M = 4$, both symmetrical and low-demerit scenario trees can provide a detailed discretization at every stage for a number of stages computational feasible. Thus, in that situation the exponents $\omega(M)$ are almost identical and the difference in convergence speed comes from the numerator $\Lambda(M)$ in (7.53). The $\Lambda(M)$ coefficient is different from LD to SYM because of the different type of averages (geometric or arithmetic) for the stagewise expected variability, as explained in Section 7.3.3.

The situation is however different for $M = 13$. For that many stages, no computationally feasible scenario tree can provide a detailed discretization at every stages. The symmetrical trees discretize equally every stage with a branching factor limited to 2 and 3, which results in a convergence rate very close to α/M. Thus, the effective number of stages is very close to M ($M_{\text{eff}} = 12.7, 12.6, 14.1$). That is, from the perspective of symmetrical trees, the problem does not appear to have less stages than the actual number. On the other hand, low-demerit scenario trees, by prioritizing the discretization toward the stages with more variability, achieve higher convergence rates and have an effective number of stages smaller than M ($M_{\text{eff}} = 10.7, 10, 11.9$). From their perspective, the problem has less stages because most of the problem's total variability is concentrated over the first 10 stages or so. This reduction of about 2-3 stages matches the branching structure of the fractional bushinesses displayed in Table 7.6, which stops branching out towards the end of the option's life and simply uses of fan of scenarios at the last 2 or 3 stages.

Another way to estimate the effective number of stages of low-demerit trees, without having to estimate first α, consists in assuming a priori that symmetrical trees satisfy $M_{\text{eff}} = M$. The effective number of stages is therefore what solves $\omega^{\text{LD}}(M)M_{\text{eff}} = \alpha = \omega^{\text{SYM}}(M)M$, i.e., $M_{\text{eff}} = (\omega^{\text{SYM}}(M)/\omega^{\text{LD}}(M))M$. The corresponding estimates are displayed in Table 7.10. They match those in Table 7.9.

Table 7.9 Effective number of stages: $M_{\text{eff}} = \frac{\alpha}{\omega(M)}$ (from the exponents $\omega(M)$ in Table 7.7).

	$M = 4$		$M = 13$	
$\mathfrak{D}$	LD	SYM	LD	SYM
QMC-Lat	3.9	3.9	10.7	12.7
OQ-W_1	3.4	3.4	10	12.6
OQ-W_2	3.9	4.2	11.9	14.1

Table 7.10 Effective number of stages: $M_{\text{eff}} = \frac{\omega^{\text{SYM}}(M)}{\omega^{\text{LD}}(M)} M$ (from the exponents $\omega(M)$ in Table 7.7).

	$M = 4$		$M = 13$	
$\mathfrak{D}$	LD	SYM	LD	SYM
QMC-Lat	4	4	11.0	13
OQ-W_1	4	4	10.3	13
OQ-W_2	3.7	4	11.0	13

7.4.4 Error vs. number of stages

In the previous section we studied how the estimation error behaves as the number of scenarios increases. We saw that the error is consistently smaller for low demerit scenario trees than for symmetrical scenario trees. We also saw that the error difference varies with the number of stages of the problem. In this section, we want to quantify the benefit of low demerit trees over symmetrical trees as the number of stages increases.

When letting the number of stages M increase, there are two quantities that are relevant to study. The first one is the estimation error $\mathrm{E}(N_M, M)$ as a function of M given a fixed number of scenarios N_M, i.e., given a fixed computational budget. The second one is the reciprocal of the first one: this time we fix the estimation error at a value $\varepsilon > 0$ and we compute the minimum number of scenarios $\mathrm{N}_{\min}(\varepsilon, M)$ required to reach that error as a function of M. The formal definition of the latter is

$$\mathrm{N}_{\min}(\varepsilon, M) = \inf\{N_M : \mathrm{E}(N_M, M) \leq \varepsilon\}. \tag{7.57}$$

The function $\mathrm{N}_{\min}(\cdot, \cdot)$ is notoriously known to increase extremely rapidly with decreasing ε or increasing M. For this reason it is highly important to develop high quality scenario trees which use the fewest number of scenarios to reach a given error value.

From $\mathrm{E}(N_M, M)$ and $\mathrm{N}_{\min}(\varepsilon, M)$ we define two empirical measures to assess the benefit of low demerit (LD) scenario trees over symmetrical (SYM) scenario trees: the *error reduction* and the *scenario reduction*. Their definitions read

$$\text{Error reduction:} \quad \frac{\mathrm{E}^{\text{SYM}}(N_M, M) - \mathrm{E}^{\text{LD}}(N_M, M)}{\mathrm{E}^{\text{SYM}}(N_M, M)} \quad (\times 100\%), \tag{7.58}$$

$$\text{Scenario reduction:} \quad \frac{\mathrm{N}^{\text{SYM}}_{\min}(\varepsilon, M) - \mathrm{N}^{\text{LD}}_{\min}(\varepsilon, M)}{\mathrm{N}^{\text{SYM}}_{\min}(\varepsilon, M)} \quad (\times 100\%). \tag{7.59}$$

Both measures take values between 0% and 100% because LD trees consistently provide smaller error for a given N_M or reach a given error using fewer scenarios, as demonstrated

in the previous sections. An error/scenario reduction close to 0% means that LD trees have essentially no benefit over SYM trees, whereas a reduction close to 100% means that LD trees are significantly better than SYM trees, either in terms of their ability to reduce the error for a given computational cost or to reduce the number of scenarios for a given error tolerance.

We display in Tables 7.11 and7.12 the error and scenario reductions for $N_M = 10^6$ and $\varepsilon = 0.2$ (the latter corresponds to about 5% error on the true option prices).

The error reduction ranges between about 30% and 60% throughout all instances, with the highest reduction achieved by OQ-W_2. It increases with the number of stages for QMC-Lat and OQ-W_1, but decreases for OQ-W_2. The number of stages actually influences only slightly the error reduction, since the variations from $M = 4$ to $M = 13$ are typically less than 10%. What matters the most in this respect is the choice of the discretization method.

Table 7.11 Error reduction (%) for $N_M = 10^6$.

	QMC-Lat		OQ-W_1		OQ-W_2	
(σ, T, K)	$M = 4$	$M = 13$	$M = 4$	$M = 13$	$M = 4$	$M = 13$
$(0.25, 0.25, 100)$	31%	38%	41%	46%	57%	47%
$(0.15, 0.25, 100)$	33%	38%	41%	44%	57%	47%
$(0.25, 0.50, 100)$	32%	36%	38%	44%	53%	45%
$(0.25, 0.50, 105)$	32%	38%	43%	45%	55%	46%

The scenario reduction ranges from about 70% to 80% for $M = 4$ and is only slightly smaller than 100% for $M = 13$. A reduction this close to 100% may look surprisingly high, but it actually reflects the fact that for $M = 13$ the quantities $\mathrm{N}^{\mathrm{LD}}_{\min}$ and $\mathrm{N}^{\mathrm{SYM}}_{\min}$ are no longer of the same order of magnitude. For instance, for OQ-W_2 and $(\sigma, T, K) = (0.25, 0.50, 105)$: $\mathrm{N}^{\mathrm{LD}}_{\min}(0.2, 13)$ is about 7.8 millions of scenarios, whereas $\mathrm{N}^{\mathrm{SYM}}_{\min}(0.2, 13)$ is as much as 1.2 billions scenarios.

The scenario reduction is already quite significant for $M = 4$. For instance, for OQ-W_2 and $(\sigma, T, K) = (0.25, 0.15, 100)$: $\mathrm{N}^{\mathrm{LD}}_{\min}(0.01, 4)$ is about 9,600 scenarios, whereas $\mathrm{N}^{\mathrm{SYM}}_{\min}(0.01, 4)$ is about 44,000 scenarios. Such scenario reduction may not have a tremendous impact on the run times, since option pricing problems involve only one binary decision variable per node (exercise or not). However, if a similar scenario reduction could be achieved in more general planning problems involving hundreds or thousands of decision variables per node, then the gain in run times reduction would be multiplied and would be substantial.

The run times are displayed in Table 7.13 for the best discretization method, OQ-W_2. What we consider as run time is the time required to generate the scenario tree and to solve the

problem. Since LD and SYM trees achieve different errors for a given number of scenarios, the comparison can only be meaningful if we compare the run times of each scenario tree for the same value of the error. Additionally, symmetrical trees can only be generated for some specific number of scenarios N_M, given by $N_4 = b^4$ and $N_{13} = b^{13}$ $(b = 1, 2, ...)$ for $M = 4$ and $M = 13$, respectively. For this reason, for the SYM trees the run times displayed correspond to the largest structures generated in Section 7.4.2, i.e., $N_4 = 18^4 = 104,976$ and $N_{13} = 3^{13} = 1,594,323$. While for LD trees, the run times correspond to tree structures that achieve the same error, which have a number of scenarios given by

$$\mathrm{N}^{\mathrm{LD}}_{\min}(\mathrm{E}^{\mathrm{SYM}}(N_M, M), M), \qquad \text{for } N_M \in \{18^4, 3^{13}\}. \tag{7.60}$$

The gain in run time reduction increases with the number of stages: for $M = 4$ it goes from 21 sec. to 6 sec., while for $M = 13$ the gain is much more substantial since it goes from 16 min to only 1 min. The run time gain does not vary across the instances.

Table 7.12 Scenario reduction (%) for $\varepsilon = 0.2$.

	QMC-Lat		OQ-W$_1$		OQ-W$_2$	
(σ, T, K)	$M = 4$	$M = 13$	$M = 4$	$M = 13$	$M = 4$	$M = 13$
$(0.25, 0.25, 100)$	75.22%	99.94%	77.62%	99.97%	68.37%	99.07%
$(0.15, 0.25, 100)$	73.5%	99.8%	76.05%	99.84%	67.53%	98.11%
$(0.25, 0.50, 100)$	74.91%	99.96%	75.78%	99.98%	69.02%	99.28%
$(0.25, 0.50, 105)$	78.24%	99.98%	79.33%	99.99%	77.92%	99.38%

Table 7.13 Run times for $M = 4$ and $M = 13$ for OQ-W$_2$.

	$M = 4$	$M = 13$
LD	6 sec.	1 min 15 sec.
SYM	21 sec.	16 min 28 sec.

7.5 Conclusion

We implemented the problem-driven multistage discretization approach newly introduced in Keutchayan *et al.* (2018a,b) to generate scenario trees specifically tailored to the problem of pricing Bermudan-Asian call options. This approach, which is not tied to any particular application, is based on the concept of *figure of demerit*. The figure of demerit, $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma)$, measures the suitability between a *scenario tree* $(\mathcal{T}, \mathcal{P}, \mathcal{W})$ and a problem characterized by the *guidance functions* Γ.

By bounding the variability of the option price function at each stage and for each possible history of the asset price, we were able to compute a set of guidance functions appropriate to the problem of pricing Bermudan-Asian call options. These guidance functions contain the relevant features of the problem, such as the quick decrease of payoff's variability as the number of monitoring times increases, that the problem-driven construction method takes advantage of to generate suitable tree structures $\mathcal{T}$. These structures have heterogeneous branching across the stages and across the nodes at a given stage. For this reason, they are characterized by a *fractional* bushiness $(b_0, \ldots, b_{M-1})$, where b_m can take any real value greater or equal to one (as opposed to the standard notion of bushiness where b_m is restricted to integer numbers).

We demonstrated experimentally the relevance of this approach in three steps:

- First, we showed that the figure of demerit is tightly correlated to the estimation error: a scenario tree with a small demerit provides an estimate with a small error, and vice versa. Thus, this validates the core idea of the approach: to select scenario trees by minimizing the figure of demerit.

- Second, we showed that scenario trees of low demerit, by prioritizing the discretization towards the stages that concentrate most payoff's variability, have consistently faster error convergence than symmetrical trees as the number of scenarios increases. Additionally, while the convergence rate $\omega(M)$ of symmetrical trees matches the relation α/M (where α is the convergence rate of the discretization method used to generate the points and weight $(\mathcal{P}, \mathcal{W})$), we observe that the convergence rate of low-demerit trees rather follow a relation α/M_{eff}, where $M_{\text{eff}} < M$ is a number interpreted as the *effective number of stages* of the problem. This number can be thought as the number of stages that effectively matter (for some given finite computational resources), because they concentrate most of the problem's variability.

- Third, we showed that the benefit of low demerit trees over symmetrical trees, in terms of error reduction and scenario reduction, tends to increase as the number of stages increases. This reduction impacts the run times: while run times are divided by 4 for problems with 4 exercise stages, they are divided by 16 for problems with 13 stages.

Finally, we stress that the obtained results should be interpreted on a more general standpoint, beyond the particular application considered in this paper. In principle, all the developments done throughout can be replicated in any problem dealing with multistage planning under uncertainty. The only difficulty in doing so lies in the computation of the guidance

functions, since they are specific to the problem and need to be somehow approximated. This approximation can be done either by bounding the variability as we did in this paper, or purely by expertise or numerical estimation. Provided a suitable set of guidance functions is available, substantial error/scenario reduction can also be expected if the problem exhibits the desirable features: heterogeneous variability of the recourse functions across the stages and/or across the nodes at any given stage. Not to mention that multistage planning problems have much more decision variables than option pricing problems (since the latter has only one binary variable per node). Thus, the extra cost of minimizing the figure of demerit (which does not vary with the number of decision variables) will quickly become insignificant as compared to the gain in run times (which will be multiplied by the number of decision variables).

As an example of possible application, consider a hydropower production planning problem with uncertain water inflows, a planning horizon of two years and two stages per season. The resulting multistage problem has 16 stages, thus branching equally at every stage rapidly becomes unfeasible as only 3 branches out of each node results in more than 43 millions scenarios. In a region like Quebec, which relies almost exclusively on hydroelectricity, water inflows have almost no variability in winter and summer, while all the variability is concentrated in spring (when snow melts) and fall. Such heterogeneous variability *across the seasons* may lead to an effective number of stages smaller than 16, which will then allow to derive solutions using problem-driven scenario trees. Moreover, the efficiency of scenario trees may be further improved by the fact that hydrological time series exhibit long-range dependence (see Hurst (1951)), because this will additionally induce heterogeneous variability *across the scenarios*. Indeed, two nodes at the same stage, with the same current value of inflows but coming from different scenarios, may have different variances for the future inflows under a non-Markovian model.

CHAPITRE 8 DISCUSSION GÉNÉRALE

La méthode développée permet de générer des arbres de scénarios aux facteurs de branchements variables en fonction des étapes et des scénarios. La variabilité en fonction des étapes n'est pas nouvelle en soit, puisque les applications existantes dans la littérature considèrent généralement des arbres dont la *bushiness* (à valeur entière) décroit avec les étapes. Cependant, à la différence de la littérature où celle-ci est généralement choisie de façon empirique, notre approche la détermine analytiquement de sorte à minimiser l'erreur d'estimation à travers la *figure of demerit*. De plus, la *bushiness* pourrait tout au aussi bien croître avec les étapes si le problème s'avérait avoir plus de variabilité à la fin de l'horizon qu'au début. Notons que cela va à l'encontre de la tendance (ou croyance) générale qui veut que l'arbre discrétise toujours plus finement les premières étapes. Cependant, nous pensons que cette tendance néglige le fait qu'une discrétisation trop grossière aux dernières étapes génère une estimation erronée des effets futurs des décisions en cas de grande variabilité, ce qui peut mener à prendre de mauvaises décisions aux premières étapes (même si la discrétisation à celles-ci est détaillée).

L'application considérée dans cette thèse illustre l'intérêt de prendre en compte l'intégralité des propriétés du problème, y compris celles qui découlent d'une forme particulière de la fonction objectif. En effet, en observant uniquement le processus stochastique, qui dans cet exemple suit un mouvement Brownien géométrique *à volatilité constante*, il est impossible de détecter une quelconque asymétrie de variabilité dans le temps. Ce n'est donc qu'en étudiant la fonction de revenus que l'asymétrie de variabilité du problème peut apparaitre, ce qui permet à l'arbre de scénarios de réduire le nombre d'étapes *apparentes* du problème. Cette réduction est d'autant plus grande que la variabilité du problème est concentrée dans un petit sous-ensemble d'étapes. En effet, si toute la variabilité est concentrée dans une seule étape, il est logique d'espérer que l'arbre de scénarios le détecte est branche uniquement à celle-ci pour traiter le problème comme s'il avait seulement deux étapes. Dans l'exemple de l'option asiatique, une plus grande réduction aurait lieu si la moyenne était pondérée avec des poids décroissants.

Le travail réalisé dans cette thèse amène à l'idée nouvelle que la difficulté d'un problème d'optimisation stochastique multi-étape est reliée à la distribution de variabilité de ses fonctions de recours à travers les étapes et les scénarios. On observe qu'une distribution constante suivant ces deux paramètres mène aux problèmes les plus difficiles à résoudre, alors qu'une variabilité cumulée identique mais dispersée inégalement entre les scénarios ou entre les étapes

permet une résolution plus efficace du problème. Le fait de considérer la variabilité des fonctions de recours sous l'action du processus stochastique –et non la variabilité du processus stochastique seul– est la raison pour laquelle la méthode est guidée par le problème et non par la distribution.

CHAPITRE 9 CONCLUSION ET RECOMMANDATIONS

9.1 Synthèse des travaux

Dans cette thèse, nous avons étudié deux des quatre étapes principales menant de la modélisation à la résolution de problèmes d'optimisation stochastique multi-étapes (cf. Figure 2.1) : la génération d'arbres de scénarios et la mise en pratique des décisions de l'arbre. Dans un premier temps (article 1), nous avons montré comment les décisions obtenues avec les arbres de scénarios peuvent être étendues sur l'ensemble des réalisations des paramètres aléatoires afin de produire une politique de décision pour le problème original dont la qualité peut être estimée numériquement à faible coût. Dans un deuxième temps (articles 2 et 3), nous avons développé une nouvelle approche de génération d'arbres de scénarios. Celle-ci à la particularité d'être guidée par le problème, ce qui signifie qu'elle tire avantage des caractéristiques du problème pour construire des structures et des ensembles de scénarios adaptés à celui-ci. Nous avons démontré que la variabilité des fonctions de recours à travers les étapes et les scénarios est une caractéristique qu'il est essentiel de prendre en compte, car elle mesure l'amplitude de l'erreur pouvant être produite à chaque nœud de l'arbre. En considérant cette donnée de variabilité, nous avons développé une mesure d'adéquation entre le problème et l'arbre de scénarios et nous avons montré comment celle-ci est utilisée pour construire des arbres adaptés aux problèmes stochastiques multi-étapes. Dans un troisième et dernier temps (article 4), cette nouvelle méthodologie a été appliquée à un problème d'évaluation d'options asiatiques. Les résultats obtenus démontrent que les arbres générés par notre méthode produisent des erreurs d'estimation significativement inférieures aux arbres de structures symétriques qui ignorent les propriétés du problème. Cet exemple produit la preuve de concept nécessaire à la validation de notre approche.

9.2 Limitations

L'approche développée à plusieurs limitations. Tout d'abord, au niveau de sa portée : elle s'applique à une certaine classe de problèmes multi-étapes où l'optimisation est faite en espérance et les contraintes sont vérifiées presque sûrement. Par conséquent, les problèmes qui considèrent des mesures de risque ou des contraintes en probabilité ou en espérance ne rentrent pas directement dans le cadre de notre méthodologie.

Il y a ensuite plusieurs limitations concernant la mise en pratique de l'approche :
(1) Nous avons considéré tout au long du développement théorique que la variabilité des

fonctions de recours optimal était une donnée du problème. Cependant, cette donnée n'est généralement pas connue explicitement puisque la forme exacte des fonctions de recours est inconnue avant la résolution du problème. En pratique, il est donc nécessaire de s'en remettre à des approximations. Par exemple, dans le quatrième article, la variabilité du prix de l'option est bornée supérieurement à chaque étape et pour chaque scénario, ce qui est possible car l'ensemble des décisions réalisables du problème est restreint à seulement deux valeurs (continue ou arrêt). Cela est cependant un cas particulier.

(2) Les procédures pour construire des structures d'arbres développées dans les articles 2, 3 et 4 ont chacune leur limite. Celle l'article 2 s'applique à des problèmes à grand nombre d'étapes et de scénarios mais nécessite de spécifier à l'avance le nombre de nœuds par étape. Celle de l'article 3 ne requiert aucune spécification au niveau de la structure (à part le nombre de scénarios), mais elle repose sur une stratégie d'exploration de l'espace de toutes les structures. La taille gigantesque de cet espace limite donc l'applicabilité aux problèmes avec peu d'étapes et de scénarios. La procédure de l'article 4 parvient à corriger les limites précédentes, en générant des arbres à grand nombre d'étapes et de scénarios sans contrainte particulière pour la structure, mais cela nécessite de calculer au préalable l'espérance des fonctions $\gamma_t(\xi_{..t})$, ce qui ajoute un coût numérique supplémentaire.

(3) Bien que la méthodologie développée autorise un nombre quelconque de paramètres aléatoires par étape, jusqu'à présent les applications n'ont considéré que des problèmes avec un seul paramètre par étape. Le passage au multi-dimensionnel est délicat car il entraine une réduction du taux de convergence des procédures de discrétisation que nous avons considérées (quasi-Monte Carlo et quantification optimale).

9.3 Améliorations futures

Les améliorations futures visent essentiellement à corriger les limitations exprimées précédemment. En particulier :

(1) Il est nécessaire de développer des procédures systématiques permettant d'inférer la distribution de variabilité des fonctions de recours à travers les étapes et les scénarios. Une piste pour cela consisterait à résoudre itérativement le problème. A chaque itération, une approximation de la distribution de variabilité est mise à jour et celle-ci est utilisée pour générer les arbres de scénarios à l'itération suivante. Cette approche purement numérique a l'avantage d'être générique et de pouvoir être appliquée à n'importe quel problème sans connaissance préalable sur celui-ci. Cependant, son inconvénient est le coût numérique lié au processus itératif. Une autre approche, qui n'est pas systématique mais que nous pensons tout de même prometteuse, consiste à inférer la variabilité d'une façon semi-

empirique/théorique. L'agent décideur a généralement une connaissance (empirique) des paramètres qui influencent la variabilité du problème considéré et celle-ci peut être complétée par une analyse (théorique) de la formulation mathématique du problème. Cette approche a été suivie avec succès dans l'exemple de l'évaluation d'options asiatiques, où des connaissances basiques sur l'évaluation des produits dérivés permettent de savoir à l'avance comment la variabilité du prix de l'option se comporte en fonction du prix de l'actif sous-jacent et de l'étape d'exercice. L'avantage de l'approche semi-empirique/théorique, par rapport à l'approche numérique, réside dans le fait qu'elle calcule analytiquement le représentant $\Gamma(p)$ du problème p, ce qui veut dire que celui-ci peut alors être utilisé par n'importe quel agent décideur qui fait face au même problème (mais une instance différente). A l'inverse, l'approche numérique permet seulement de calculer $\Gamma(p)$ pour une instance particulière et donc la procédure doit être répétée à chaque nouvelle instance rencontrée. Idéalement, nous souhaitons que soit développée une banque de représentants $\Gamma(p)$ pour tous les problèmes p d'intérêt, afin que quiconque souhaite résoudre un de ces problèmes puisse directement utiliser son meilleur représentant connu.

(2) Il est nécessaire d'améliorer les procédures algorithmiques de minimisation de la mesure $\mathcal{M}$. Il y a pour cela deux pistes à explorer. La première consisterait à remplacer la stratégie (extrêmement coûteuse) d'exploration de l'espace des structures d'arbres par une autre l'étant beaucoup moins qui consiste à explorer un ensemble de règles récursives de construction, du type de celles exprimées dans la Section 1.2. Cette réduction de l'espace de recherche est pertinente puisque pour un problème donné, la grande majorité des structures ne sont clairement pas adaptées à celui-ci. La deuxième direction de recherche consiste à ne plus découpler la procédure de recherche de la structure de celle de génération des points et des poids. En d'autres termes, il faudrait pouvoir minimiser la mesure $\mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma(p))$ *simultanément* sur l'espace de tous les triplets $(\mathcal{T}, \mathcal{P}, \mathcal{W})$, et non plus en $\mathcal{T}$ puis en $(\mathcal{P}, \mathcal{W})$. Le but est de se rapprocher encore plus de la résolution du problème général de minimisation :

$$\min_{(\mathcal{T},\mathcal{P},\mathcal{W})} \mathcal{M}(\mathcal{T}, \mathcal{P}, \mathcal{W}; \Gamma(p)). \tag{9.1}$$

Pour réaliser cela, il faut abandonner la Condition 6.4.1 du troisième article et à la place considérer la forme explicite du *demerit* $\mathcal{D}_{\mathcal{F}}(\mathcal{P}^{C(n)}, \mathcal{W}^{C(n)})$ à chaque nœud n.

(3) Il est nécessaire de développer des ensembles de points et de poids qui permettent d'intégrer efficacement les fonctions de recours optimal en grande dimension. Cela n'est pas évident car ces fonctions sont généralement non-régulières et peuvent être définies sur des sous-ensembles non bornés de $\mathbb{R}^d$. La notion de variabilité joue un rôle important à ce niveau là aussi. En effet, une fonction dont la variabilité est concentrée suivant certaines dimensions de taille largement inférieure à d pourrait potentiellement être intégrée effi-

cacement si l'ensemble de points et de poids discrétise l'espace plus finement suivant ces dimensions.

RÉFÉRENCES

Ahmed, Shabbir and Sahinidis, Nikolaos V. (2003). An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research*, *51*(3), 461–471.

Amin, Kaushik I. (1991). On the computation of continuous time option prices using discrete approximations. *Journal of Financial and Quantitative Analysis*, *26*(4), 477–495.

Artzner, Philippe and Delbaen, Freddy and Eber, Jean-Marc and Heath, David (1999). Coherent measures of risk. *Mathematical Finance*, *9*(3), 203–228.

Baillie, Richard T. and Bollerslev, Tim and Mikkelsen, Hans Ole (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *74*(1), 3–30.

Barraquand, Jérôme and Pudet, Thierry (1996). Pricing of American path-dependent contingent claims. *Mathematical Finance*, *6*(1), 17–51.

Bastin, Fabian and Cirillo, Cinzia and Toint, Philippe L. (2006). Convergence theory for nonconvex stochastic programming with an application to mixed logit. *Mathematical Programming*, *108*(2-3), 207–234.

Bayraksan, Güzin and Morton, David P. (2009). Assessing solution quality in stochastic programs via sampling. M. R. Oskoorouchi, P. Gray et H. J. Greenberg, éditeurs, *Tutorials in Operations Research : Decision Technologies and Applications*, Informs. 102–122.

Beale, Evelyn M.L. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, 173–184.

Bellman, Richard (1957a). *Dynamic Programming*. Princeton University Press.

Bellman, Richard (1957b). A Markovian decision process. *Journal of Mathematics and Mechanics*, 679–684.

Bellman, Richard E. (1961). *Adaptive control processes : a guided tour*. Princeton university press.

Ben-Ameur, Hatem and Breton, Michèle and L'Ecuyer, Pierre (2002). A dynamic programming procedure for pricing American-style Asian options. *Management Science*, *48*(5), 625–643.

Ben-Tal, Aharon and El Ghaoui, Laurent and Nemirovski, Arkadi (2009). *Robust optimization*. Princeton University Press.

Benders, Jacques F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*(1), 238–252.

Beraldi, Patrizia and Bruni, Maria Elena (2014). A clustering approach for scenario tree reduction : an application to a stochastic programming portfolio optimization problem. *Top*, *22*(3), 934–949.

Beraldi, Patrizia and Bruni, Maria E. and Conforti, Domenico (2004). Designing robust emergency medical service via stochastic programming. *European Journal of Operational Research*, *158*(1), 183–193.

Bernoulli, Daniel (1738). Specimen teoriae novae de mensura sortis. *Commentarii Academiae Scientarium Imperialis Petropolitanae*, *V*, 175–192. English translation : "Exposition of a New Theory on the Measurement of Risk," Econometrica, 22 (1954), 23–36.

Bertocchi, Marida and Consigli, Giorgio and Dempster, Michael A.H. (2011). *Stochastic optimization methods in finance and energy : new financial products and energy market strategies*, vol. 163. Springer.

Bertsekas, Dimitri P. (2000). *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, seconde édition.

Birge, John R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, *33*(5), 989–1007.

Birge, John R. and Louveaux, Francois (2011). *Introduction to stochastic programming*. Springer Science & Business Media, seconde édition.

Black, Fischer and Scholes, Myron (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654.

Boyle, Phelim P. (1986). Option valuation using a tree-jump process. *International Options Journal*, *3*, 7–12.

Boyle, Phelim P. and Kolkiewicz, Adam W. and Tan, Ken Seng (2013). Pricing bermudan options using low-discrepancy mesh methods. *Quantitative Finance*, *13*(6), 841–860.

Brandimarte, Paolo (2006). Multi-item capacitated lot-sizing with demand uncertainty. *International Journal of Production Research*, *44*(15), 2997–3022.

Brandimarte, Paolo (2013). *Numerical methods in finance and economics : a MATLAB-based introduction*. John Wiley & Sons.

Breidt, Jay F. and Crato, Nuno and De Lima, Pedro (1998). The detection and estimation of long memory in stochastic volatility. *Journal of Econometrics*, *83*(1-2), 325–348.

Broadie, Mark and Glasserman, Paul (1997). Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, *21*(8-9), 1323–1352.

Broadie, Mark and Glasserman, Paul (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, *7*, 35–72.

Calvet, Laurent and Fisher, Adlai (2002). Multifractality in asset returns : theory and evidence. *Review of Economics and Statistics*, *84*(3), 381–406.

Carino, David R. and Kent, Terry and Myers, David H. and Stacy, Celine and Sylvanus, Mike and Turner, Andrew L. and Watanabe, Kouji and Ziemba, William T. (1994). The russell-yasuda kasai model : An asset/liability model for a japanese insurance company using multistage stochastic programming. *Interfaces*, *24*(1), 29–49.

Carriere, Jacques F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance : Mathematics and Economics*, *19*(1), 19–30.

Casey, Michael S. and Sen, Suvrajeet (2005). The scenario generation algorithm for multistage stochastic linear programming. *Mathematics of Operations Research*, *30*(3), 615–631.

Chalasani, Prasad and Jha, Somesh and Egriboyun, Feyzullah and Varikooty, Ashok (1999). A refined binomial lattice for pricing American Asian options. *Review of Derivatives Research*, *3*(1), 85–105.

Charnes, Abraham and Cooper, William W. and Symonds, Gifford H. (1958). Cost horizons and certainty equivalents : an approach to stochastic programming of heating oil. *Management Science*, *4*(3), 235–263.

Chen, Michael and Mehrotra, Sanjay (2008). Epi-convergent scenario generation method for stochastic problems via sparse grid. *Stochastic Programming E-Print Series*, (7).

Chen, Michael and Mehrotra, Sanjay and Papp, Dávid (2015). Scenario generation for stochastic optimization problems via the sparse grid method. *Computational Optimization and Applications*, *62*(3), 669–692.

Chen, Z. and Consigli, G. and Dempster, Michael A.H. and Hicks-Pedrón, N. (1998). Towards sequential sampling algorithms for dynamic portfolio management. *Operational Tools in the Management of Financial Risks*, Springer. 197–211.

Chiralaksanakul, Anukal and Morton, David P. (2004). Assessing policy quality in multistage stochastic programming. *Stochastic Programming E-Print Series*, (12).

Colvin, Matthew and Maravelias, Christos T. (2008). A stochastic programming approach for clinical trial planning in new drug development. *Computers & Chemical Engineering*, *32*(11), 2626–2642.

Consigli, Giorgio and Dempster, Michael A.H. (1998). Dynamic stochastic programming for asset-liability management. *Annals of Operations Research*, *81*, 131–162.

Consigli, Giorgio and Iaquinta, Gaetano and Moriggia, Vittorio (2012). Path-dependent scenario trees for multistage stochastic programmes in finance. *Quantitative Finance*, *12*(8), 1265–1281.

Consigli, Giorgio and Kuhn, Daniel and Brandimarte, Paolo (2017). Optimal financial decision making under uncertainty. *Optimal Financial Decision Making under Uncertainty*, Springer. 255–290.

Consiglio, Andrea and De Giovanni, Domenico (2008). Evaluation of insurance products with guarantee in incomplete markets. *Insurance : Mathematics and Economics*, *42*(1), 332–342.

Cont, Rama (2005). Long range dependence in financial markets. *Fractals in Engineering*, Springer. 159–179.

Cont, Rama (2007). Volatility clustering in financial markets : empirical facts and agent-based models. *Long Memory in Economics*, Springer. 289–309.

Cox, John C. and Ross, Stephen A. and Rubinstein, Mark (1979). Option pricing : A simplified approach. *Journal of financial Economics*, *7*(3), 229–263.

Dantzig, George B. (1955). Linear programming under uncertainty. *Management Science*, *1*(3,4), 197–206.

Defourny, Boris and Ernst, Damien and Wehenkel, Louis (2011). Multistage stochastic programming : A scenario tree based approach. L. E. Sucar, E. F. Morales et J. Hoey, éditeurs, *Decision Theory Models for Applications in Artificial Intelligence : Concepts and Solutions : Concepts and Solutions*, IGI Global. 97–143.

Defourny, Boris and Ernst, Damien and Wehenkel, Louis (2013). Scenario trees and policy selection for multistage stochastic programming using machine learning. *INFORMS Journal on Computing*, *25*(3), 488–501.

Delage, Erick and Ye, Yinyu (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, *58*(3), 595–612.

Dempster, Michael A.H and Thompson, R.T. (1999). Evpi-based importance sampling solution proceduresfor multistage stochastic linear programmeson parallel mimd architectures. *Annals of Operations Research*, *90*, 161–184.

Dempster, Michael Alan Howarth (2006). Sequential importance sampling algorithms for dynamic stochastic programming. *Journal of Mathematical Sciences*, *133*(4), 1422–1444.

Dick, Josef and Kuo, Frances Y. and Sloan, Ian H. (2013). High-dimensional integration : the quasi-Monte Carlo way. *Acta Numerica*, *22*, 133–288.

Dick, Josef and Pillichshammer, Friedrich (2010). *Digital nets and sequences : Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge University Press.

Ding, Zhuanxin and Granger, Clive W.J. and Engle, Robert F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, *1*(1), 83–106.

Drew, Shane S. and Homem-de-Mello, Tito (2006). Quasi-Monte Carlo strategies for stochastic optimization. *Proceedings of the 38th conference on Winter simulation*. Winter Simulation Conference, 774–782.

Dror, Moshe (1993). Modeling vehicle routing with uncertain demands as a stochastic program : Properties of the corresponding solution. *European Journal of Operational Research*, *64*(3), 432–441.

Dupačová, Jitka and Consigli, Giorgio and Wallace, Stein W. (2000). Scenarios for multistage stochastic programs. *Annals of Operations Research*, *100*(1-4), 25–53.

Dupačová, Jitka and Gröwe-Kuska, Nicole and Römisch, Werner (2003). Scenario reduction in stochastic programming. *Mathematical Programming*, *95*(3), 493–511.

Dyer, Martin and Stougie, Leen (2006). Computational complexity of stochastic programming problems. *Mathematical Programming*, *106*(3), 423–432.

Edirisinghe, Nalin E. (1999). Bound-based approximations in multistage stochastic programming : Nonanticipativity aggregation. *Annals of Operations Research*, *85*, 103–127.

Eichhorn, Andreas and Heitsch, Holger and Römisch, Werner (2010). Stochastic optimization of electricity portfolios : Scenario tree modeling and risk management. *Handbook of power systems II*, Springer. 405–432.

Ellsberg, Daniel (1961). Risk, ambiguity, and the savage axioms. *The Quarterly Journal of Economics*, 643–669.

Figlewski, Stephen and Gao, Bin (1999). The adaptive mesh model : a new approach to efficient option pricing. *Journal of Financial Economics*, *53*(3), 313–351.

Fleten, Stein-Erik and Wallace, Stein W. and Ziemba, William T. (2002). Hedging electricity portfolios via stochastic programming. *Decision making under uncertainty*, Springer. 71–93.

Frauendorfer, Karl (1996). Barycentric scenario trees in convex multistage stochastic programming. *Mathematical Programming*, *75*(2), 277–293.

Geyer, Alois and Hanke, Michael and Weissensteiner, Alex (2010). No-arbitrage conditions, scenario trees, and multi-asset financial optimization. *European Journal of Operational Research*, *206*(3), 609–613.

Geyer, Alois and Hanke, Michael and Weissensteiner, Alex (2013). Scenario tree generation and multi-asset financial optimization problems. *Operations Research Letters*, *41*(5), 494–498.

Glasserman, Paul (2003). *Monte Carlo methods in financial engineering*, vol. 53. Springer Science & Business Media.

Graf, Siegfried and Luschgy, Harald (2007). *Foundations of quantization for probability distributions*. Springer.

Granger, Clive William John and Ding, Zhuanxin (1995). Some properties of absolute return : An alternative measure of risk. *Annales d'Economie et de Statistique*, 67–91.

Grant, Dwight and Vora, Gautam and Weeks, David (1997). Path-dependent options : Extending the Monte Carlo simulation approach. *Management Science*, *43*(11), 1589–1602.

Griebel, Michael and Kuo, Frances Y. and Sloan, Ian H. (2010). The smoothing effect of the ANOVA decomposition. *Journal of Complexity*, *26*(5), 523–551.

Griebel, Michael and Kuo, Frances Y. and Sloan, Ian H. (2013). The smoothing effect of integration in $\mathbb{R}^d$ and the ANOVA decomposition. *Mathematics of Computation*, *82*(281), 383–400.

Growe-Kuska, Nicole and Heitsch, Holger and Romisch, Werner (2003). Scenario reduction and scenario tree construction for power management problems. *Power Tech Conference Proceedings, 2003 IEEE Bologna*. IEEE, vol. 3, 7–pp.

Gülpınar, Nalan and Rustem, Berç and Settergren, Reuben (2004). Simulation and optimization approaches to scenario tree generation. *Journal of Economic Dynamics and Control*, *28*(7), 1291–1315.

Hanasusanto, Grani A. and Kuhn, Daniel and Wiesemann, Wolfram (2016). A comment on "computational complexity of stochastic programming problems". *Mathematical Programming*, *159*(1), 557–569.

Hansen, Pierre and Mladenović, Nenad and Brimberg, Jack and Pérez, José Moreno A. (2019). Variable neighborhood search. M. Gendreau et J.-Y. Potvin, éditeurs, *Handbook of metaheuristics*, Springer. Troisième édition, 57–97.

Harrison, Michael J. and Kreps, David M. (1979). Martingales and arbitrage in multiperiod securities markets. *Journal of Economic theory*, *20*(3), 381–408.

Hartinger, Jürgen and Kainhofer, Reinhold (2006). Non-uniform low-discrepancy sequence generation and integration of singular integrands. H. Niederreiter et D. Talay, éditeurs, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, Springer. 163–179.

Heitsch, Holger and Leövey, Hernan and Römisch, Werner (2016). Are Quasi-Monte Carlo algorithms efficient for two-stage stochastic programs ? *Computational Optimization and Applications*, *65*(3), 567–603.

Heitsch, Holger and Römisch, Werner (2003). Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, *24*(2-3), 187–206.

Heitsch, Holger and Römisch, Werner (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, *118*(2), 371–406.

Heitsch, Holger and Römisch, Werner and Strugarek, Cyrille (2006). Stability of multistage stochastic programs. *SIAM Journal on Optimization*, *17*(2), 511–525.

Hickernell, Fred and Sloan, Ian H. and Wasilkowski, Grzegorz W. (2004a). On tractability of weighted integration over bounded and unbounded regions in $\mathbb{R}^s$. *Mathematics of Computation*, *73*(248), 1885–1901.

Hickernell, Fred J. and Sloan, Ian H. and Wasilkowski, Grzegorz W. (2004b). On tractability of weighted integration for certain banach spaces of functions. H. Niederreiter, éditeur, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, Springer, Heidelberg. 51–71.

Hilli, Petri and Pennanen, Teemu (2008). Numerical study of discretizations of multistage stochastic programs. *Kybernetika*, *44*(2), 185–204.

Hochreiter, Ronald and Pflug, Georg Ch. (2007). Financial scenario generation for stochastic multi-stage decision processes as facility location problems. *Annals of Operations Research*, *152*(1), 257–272.

Holtz, Markus (2010). *Sparse grid quadrature in high dimensions with applications in finance and insurance*, vol. 77. Springer Science & Business Media.

Homem-de-Mello, Tito (2008). On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling. *SIAM Journal on Optimization*, *19*(2), 524–551.

Howard, Ronald A. (1960). *Dynamic programming and Markov processes*. The MIT Press.

Høyland, Kjetil and Kaut, Michal and Wallace, Stein W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, *24*(2-3), 169–185.

Høyland, Kjetil and Wallace, Stein W. (2001). Generating scenario trees for multistage decision problems. *Management Science*, *47*(2), 295–307.

Huang, Kai and Ahmed, Shabbir (2009). The value of multistage stochastic programming in capacity planning under uncertainty. *Operations Research*, *57*(4), 893–904.

Hull, John C. (2017). *Options, futures, and other derivatives*. Pearson : New York, 10th édition.

Hull, John C. and White, Alan D. (1993). Efficient procedures for valuing European and American path-dependent options. *The Journal of Derivatives*, *1*(1), 21–31.

Hurst, Harold E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, *116*, 770–799.

Hvattum, Lars M. and Løkketangen, Arne and Laporte, Gilbert (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, *40*(4), 421–438.

Imai, Junichi (2014). Comparison of low discrepancy mesh methods for pricing bermudan options under a lévy process. *Mathematics and Computers in Simulation*, *100*, 54–71.

Jarrow, Robert A. and Rudd, Andrew T. (1983). *Option pricing*. Richard D. Irwin, Homewood, IL.

Kall, Peter and Wallace, Stein W. (1994). *Stochastic programming*. John Wiley & Sons, Chichester.

Kamrad, Bardia and Ritchken, Peter (1991). Multinomial approximating models for options with k state variables. *Management Science*, *37*(12), 1640–1652.

Kantorovich, Leonid V. and Rubinstein, Gennady S. (1958). On a space of completely additive functions. *Vestnik Leningrad. Univ*, *13*(7), 52–59.

Kantorovitch, Leonid V. (1958). On the translocation of masses. *Management Science*, *5*(1), 1–4.

Kaut, Michal (2012). Scenario-tree generation. A. J. King et S. W. Wallace, éditeurs, *Modeling with stochastic programming*, Springer. 77–102.

Kaut, Michal and Wallace, Stein W. (2007). Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, *3*(2), 257–271.

Keutchayan, Julien and Gendreau, Michel and Saucier, Antoine (2017). Quality evaluation of scenario-tree generation methods for solving stochastic programming problems. *Computational Management Science*, *14*(3), 333–365.

Keutchayan, Julien and Munger, David and Gendreau, Michel (2018a). On the scenario-tree optimal-value error for stochastic programming problems. *Preprint*.

Keutchayan, Julien and Munger, David and Gendreau, Michel and Bastin, Fabian (2018b). The figure of demerit : a quality measure for the discretization of probability distributions in multistage stochastic optimization. *Preprint*.

King, Alan J. and Wallace, Stein W. (2012). *Modeling with stochastic programming*. Springer Science & Business Media.

Klaassen, Pieter (1997). Discretized reality and spurious profits in stochastic programming models for asset/liability management. *European Journal of Operational Research*, *101*(2), 374–392.

Klaassen, Pieter (1998). Financial asset-pricing theory and stochastic programming models for asset/liability management : A synthesis. *Management Science*, *44*(1), 31–48.

Klaassen, Pieter (2002). Comment on “generating scenario trees for multistage decision problems”. *Management Science*, *48*(11), 1512–1516.

Koivu, Matti (2005). Variance reduction in sample approximations of stochastic programs. *Mathematical Programming*, *103*(3), 463–485.

Kouwenberg, Roy (2001). Scenario generation and stochastic programming models for asset liability management. *European Journal of Operational Research*, *134*(2), 279–292.

Kovacevic, Raimund (2018). Arbitrage conditions for electricity markets with production and storage. *Research Report*.

Kovacevic, Raimund M. and Pflug, Georg Ch. and Vespucci, Maria Teresa (2013). *Handbook of risk management in energy production and trading*. Springer.

Küchler, Christian (2009). *Stability, Approximation, and Decomposition in Two- and Multistage Stochastic Programming*. Springer Vieweg Verlag.

Küchler, Christian and Vigerske, Stefan (2010). Numerical evaluation of approximation methods in stochastic programming. *Optimization*, *59*(3), 401–415.

Kuo, Frances Y. and Schwab, Christoph and Sloan, Ian 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.

L’Ecuyer, Pierre and Lemieux, Christiane (2000). Variance reduction via lattice rules. *Management Science*, *46*(9), 1214–1235.

Lemieux, Christiane (2009). *Monte Carlo and quasi-Monte Carlo sampling*. Springer New York.

Leövey, Hernan and Römisch, Werner (2015). Quasi-Monte Carlo methods for linear two-stage stochastic programming problems. *Mathematical Programming*, *151*(1), 315–345.

Longstaff, Francis A. and Schwartz, Eduardo S. (2001). Valuing American options by simulation : a simple least-squares approach. *The Review of Financial Studies*, *14*(1), 113–147.

Louveaux, François (1998). An introduction to stochastic transportation models. M. Labbé, G. Laporte, K. Tanczos et P. Toint, éditeurs, *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, Springer. 244–263.

Louveaux, Francois V. (1980). A solution method for multistage stochastic programs with recourse with application to an energy investment problem. *Operations Research*, *28*(4), 889–902.

Lovejoy, Shaun and Amador, Lenin D.R. and Hébert, Raphaël (2018). Harnessing butterflies : theory and practice of the stochastic seasonal to interannual prediction system (stocsips). *Advances in Nonlinear Geosciences*, Springer. 305–355.

Mak, Wai-Kei and Morton, David P. and Wood, Kevin R. (1999). Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, *24*(1), 47–56.

Mandelbrot, Benoit (1967). The variation of some other speculative prices. *The Journal of Business*, *40*(4), 393–413.

Mandelbrot, Benoit and Fisher, Adlai and Calvet, Laurent (1997). The multifractal model of asset returns. *Cowles Foundation discussion paper no. 1164*.

Mandelbrot, Benoit and Van Ness, John W. (1968). Fractional brownian motions, fractional noises and applications. *SIAM Review*, *10*(4), 422–437.

Markowitz, Harry (1952). Portfolio selection. *The journal of Finance*, *7*(1), 77–91.

Merton, Robert C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 141–183.

Mitchell, Douglas W. (2004). More on spreads and non-arithmetic means. *The Mathematical Gazette*, *88*(511), 142–144.

Mladenović, Nenad and Hansen, Pierre (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100.

Monge, Gaspard (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Morgenstern, Oskar and Von Neumann, John (1944). *Theory of games and economic behavior*. Princeton university press.

Mulvey, John M. and Vladimirou, Hercules (1991). Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, *31*(1), 399–424.

Nickel, Stefan and Saldanha-da-Gama, Francisco and Ziegler, Hans-Peter (2012). A multistage stochastic supply network design problem with financial decisions and risk management. *Omega*, *40*(5), 511–524.

Novak, Erich and Ullrich, Mario and Woźniakowski, Henryk and Zhang, Shun (2017). Reproducing kernels of Sobolev spaces on $\mathbb{R}^d$ and applications to embedding constants and tractability. *arXiv preprint arXiv :1709.02568*.

Novak, Erich and Woźniakowski, Henryk (2008). *Tractability of multivariate problems - Volume I : Linear Information*. European Mathematical Society, Zürich.

Novak, Erich and Woźniakowski, Henryk (2010). *Tractability of multivariate problems - Volume II : Standard Information for Functionals*. European Mathematical Society, Zürich.

Oliveira, Alan Delgado de and Filomena, Tiago Pascoal and Righi, Marcelo Brutti (2018). Performance comparison of scenario-generation methods applied to a stochastic optimization asset-liability management model. *Pesquisa Operacional*, *38*(1), 53–72.

Olsen, Paul (1976a). Discretizations of multistage stochastic programming problems. R. J.-B. Wets, éditeur, *Stochastic Systems : Modeling, Identification and Optimization, II*, Springer. 111–124.

Olsen, Paul (1976b). Multistage stochastic programming with recourse as mathematical programming in an L_p space. *SIAM Journal on Control and Optimization*, *14*(3), 528–537.

Olsen, Paul (1976c). When is a multistage stochastic programming problem well-defined ? *SIAM Journal on Control and Optimization*, *14*(3), 518–527.

Owen, Art B (2006). Quasi-Monte Carlo for integrands with point singularities at unknown locations. H. Niederreiter et D. Talay, éditeurs, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, Springer. 403–417.

Pagès, Gilles (1998). A space quantization method for numerical integration. *Journal of Computational and Applied Mathematics*, *89*(1), 1–38.

Pagès, Gilles and Pham, Huyên and Printems, Jacques (2004). Optimal quantization methods and applications to numerical problems in finance. S. T. Rachev, éditeur, *Handbook of Computational and Numerical Methods in Finance*, Birkhäuser Boston, Boston, MA. 253–297.

Pennanen, Teemu (2009). Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Mathematical Programming*, *116*(1-2), 461–479.

Pennanen, Teemu and Koivu, Matti (2002). Integration quadratures in discretization of stochastic programs. *Stochastic Programming E-Print Series*, (11).

Pennanen, Teemu and Koivu, Matti (2005). Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische Mathematik*, *100*(1), 141–163.

Pereira, Mario V.F. and Pinto, Leontina M.V.G. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, *52*(1-3), 359–375.

Pflug, Georg Ch. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, *89*(2), 251–271.

Pflug, Georg Ch. and Pichler, Alois (2011). Approximations for probability distributions and stochastic optimization problems. M. Bertocchi, G. Consigli et M. A. H. Dempster, éditeurs, *Stochastic Optimization Methods in Finance and Energy*, Springer. 343–387.

Pflug, Georg Ch. and Pichler, Alois (2012). A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, *22*(1), 1–23.

Pflug, Georg Ch. and Pichler, Alois (2015). Dynamic generation of scenario trees. *Computational Optimization and Applications*, *62*(3), 641–668.

Pflug, Georg Ch. and Pichler, Alois (2016). *Multistage stochastic optimization.* Springer.

Pflug, Georg Ch. and Wozabal, David (2007). Ambiguity in portfolio selection. *Quantitative Finance*, *7*(4), 435–442.

Powell, Warren B. and Topaloglu, Huseyin (2003). Stochastic programming in transportation and logistics. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 555–635.

Prékopa, András (1995). *Stochastic programming.* Kluwer Academic Publishers, Dordrecht, Boston.

Prékopa, András (2003). Probabilistic programming. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 267–351.

Proulx, Simon (2014). *Génération de scénarios par quantification optimale en dimension élevée.* Mémoire de maîtrise, École Polytechnique de Montréal.

Rebennack, Steffen and Pardalos, Panos M. and Pereira, Mario V.F. and Iliadis, Niko A. (2010). *Handbook of power systems II.* Springer.

Rockafellar, R. Tyrrell and Uryasev, Stanislav (2000). Optimization of conditional value-at-risk. *Journal of Risk*, *2*, 21–42.

Rockafellar, R. Tyrrell and Wets, Roger J.-B. (1974). Continuous versus measurable recourse in n-stage stochastic programming. *Journal of Mathematical Analysis and Applications*, *48*(3), 836–859.

Rockafellar, R. Tyrrell and Wets, Roger J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, *16*(1), 119–147.

Rockafellar, R. Tyrrell and Wets, Roger J.-B. (2009). *Variational analysis*, vol. 317. Springer Science & Business Media.

Römisch, Werner (2003). Stability of stochastic programming problems. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 483–554.

Ruszczyński, Andrzej (2003). Decomposition methods. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 141–211.

Ruszczyński, Andrzej (2006). *Nonlinear optimization*, vol. 13. Princeton university press.

RUSZCZYŃSKI, A. et SHAPIRO, A., éditeurs (2003a). *Handbooks in Operations Research and Management Science : Stochastic Programming*, vol. 10. Elsevier.

Ruszczyński, Andrzej and Shapiro, Alexander (2003b). Optimality and duality in stochastic programming. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 65–139.

Ruszczyński, Andrzej and Shapiro, Alexander (2003c). Stochastic programming models. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 1–64.

Schultz, Rüdiger (2003). Stochastic programming with integer variables. *Mathematical Programming*, *97*(1), 285–309.

Schultz, Rüdiger and Nowak, Matthias P. and Nürnberg, Robert and Römisch, Werner and Westphalen, Markus (2003). Stochastic programming for power production and trading under uncertainty. *Mathematics—Key Technology for the Future*, Springer. 623–636.

Shapiro, Alexander (2003). Monte Carlo sampling methods. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 353–425.

Shapiro, Alexander (2006). On complexity of multistage stochastic programs. *Operations Research Letters*, *34*(1), 1–8.

Shapiro, Alexander (2018). Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming.

Shapiro, Alexander and Dentcheva, Darinka and Ruszczyński, Andrzej (2014). *Lectures on stochastic programming : modeling and theory*, vol. 16. SIAM.

Shapiro, Alexander and Homem-de-Mello, Tito (1998). A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, *81*(3), 301–325.

Shapiro, Alexander and Homem-de-Mello, Tito (2000). On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, *11*(1), 70–86.

Sloan, Ian H. and Kuo, Frances Y. and Joe, Stephen (2002). Constructing randomly shifted lattice rules in weighted sobolev spaces. *SIAM Journal on Numerical Analysis*, *40*(5), 1650–1665.

Sloan, Ian H. and Woźniakowski, Henryk (1998). When are quasi-Monte Carlo algorithms efficient for high dimensional integrals ? *Journal of Complexity*, *14*(1), 1–33.

Sun, Wei and Rachev, Svetlozar and Fabozzi, Frank J. (2007). Fractals or iid : evidence of long-range dependence and heavy tailedness from modeling german equity market returns. *Journal of Economics and Business*, *59*(6), 575–595.

Taleb, Nassim N. (2007). *The black swan : The impact of the highly improbable*, vol. 2. Random house.

Topaloglou, Nikolas and Vladimirou, Hercules and Zenios, Stavros A. (2008a). A dynamic stochastic programming model for international portfolio management. *European Journal of Operational Research*, *185*(3), 1501–1524.

Topaloglou, Nikolas and Vladimirou, Hercules and Zenios, Stavros A. (2008b). Pricing options on scenario trees. *Journal of Banking & Finance*, *32*(2), 283–298.

Tsitsiklis, John N. and Van Roy, Benjamin (2001). Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, *12*(4), 694–703.

Van Slyke, Richard M. and Wets, Roger J.-B. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, *17*(4), 638–663.

Varas, Mauricio and Maturana, Sergio and Cholette, Susan and Mac Cawley, Alejandro and Basso, Franco (2018). Assessing the benefits of labelling postponement in an export-focused winery. *International Journal of Production Research*, 1–20.

Villani, Cédric (2008). *Optimal transport : old and new*, vol. 338. Springer Science & Business Media.

Wallace, Stein W. and Fleten, Stein-Erik (2003). Stochastic programming models in energy. A. Ruszczyński et A. Shapiro, éditeurs, *Handbooks in operations research and management science : Stochastic Programming*, Elsevier, vol. 10. 637–677.

Wang, Xiao-Tian (2010). Scaling and long range dependence in option pricing, iv : pricing European options with transaction costs under the multifractional Black–Scholes model. *Physica A : Statistical Mechanics and its Applications*, *389*(4), 789–796.

Wets, Roger J.-B. (1972). Stochastic programs with recourse : A basic theorem for multistage problems. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, *21*(3), 201–206.

Wiesemann, Wolfram and Kuhn, Daniel and Rustem, Berç (2013). Robust Markov decision processes. *Mathematics of Operations Research*, *38*(1), 153–183.

Xie, Fei and Huang, Yongxi (2018). A multistage stochastic programming model for a multi-period strategic expansion of biofuel supply chain under evolving uncertainties. *Transportation Research Part E : Logistics and Transportation Review*, *111*, 130–148.

Xu, Huan and Mannor, Shie (2010). Distributionally robust Markov decision processes. *Advances in Neural Information Processing Systems*. 2505–2513.

Yen, Joyce W. and Birge, John R. (2006). A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, *40*(1), 3–14.

Yu, Li-Yong and Ji, Xiao-Dong and Wang, Shou-Yang (2003). Stochastic programming models in financial optimization : A survey. *AMO - Advanced Modeling and Optimization*, *5*(1).

Žáčková, Jitka (1966). On minimax solutions of stochastic linear programming problems. *Časopis pro pěstování matematiky*, *91*(4), 423–430.

Zeng, Zuo and Cremaschi, Selen (2017). Multistage stochastic programming models for pharmaceutical clinical trial planning. *Processes*, *5*(4), 71.

Ziemba, William T. (2003). *The stochastic programming approach to asset, liability, and wealth management*. AIMR, Charlottesville, VA.

ANNEXE A ADDITIONAL ARTICLE 2: TABLE OF NUMERICAL RESULTS

Table A.1 Bound values for different instances of (T, b). The two columns "abs." contain the bound values of the bound-minimizing (BM) scenario trees generated by QMC and OQ. The four columns "rel." contain the percentage improvement of bound-minimizing scenario trees relative to the symmetrical (SYM) scenario trees given by 100% × (Bound(SYM) − Bound(BM))/Bound(SYM). The columns "sec." contain the computational times in seconds. The dash symbol means that the time is lower than one second. The column N_T contains the total number of scenarios.

			QMC				OQ				MC		
			BM		SYM		BM		SYM		SYM		
T	b	N_T	sec.	abs.	sec.	rel.	sec.	abs.	sec.	rel.	sec.	rel.(QMC)	rel.(OQ)
	2	64	-	1.6909	-	25%	-	1.6745	-	26%	-	55%	55%
	3	729	-	1.1843	-	28%	-	1.1548	-	29%	-	66%	67%
	4	4096	-	1.1984	-	10%	-	1.2147	-	10%	1	61%	61%
	5	15625	1	0.8396	-	23%	-	0.81168	-	26%	7	68%	69%
6	6	46656	2	0.82	2	12%	3	0.84888	2	10%	24	69%	68%
	7	117649	7	0.66121	6	18%	7	0.66071	6	19%	68	70%	70%
	8	262144	15	0.63921	14	10%	16	0.64631	14	10%	168	70%	69%
	9	531441	31	0.52792	28	17%	32	0.53482	29	17%	373	72%	71%
	10	1000000	58	0.51415	54	11%	59	0.5284	53	9%	759	74%	73%
	2	1024	-	2.3764	-	34%	-	2.3568	-	35%	-	66%	66%
10	3	59049	6	1.812	5	32%	6	1.7428	5	35%	23	69%	71%
	4	1048576	104	1.9294	94	12%	106	1.9182	92	13%	451	63%	63%
14	2	16384	5	2.9991	2	40%	3	2.9799	2	40%	6	72%	73%
20	2	1048576	280	3.8714	227	45%	287	3.853	246	45%	513	74%	74%

ANNEXE B ADDITIONAL ARTICLE 2: PROOF EXISTENCE OPTIMAL DECISION POLICY (SECTION 5.2.1)

We shall show that the five conditions given in Section 5.2.1 guarantee the existence of an optimal decision policy x^* and the finiteness of the recourse functions (5.3)-(5.4) for all $\xi_{..t} \in \Xi_{..t}$ and $y_{..t} \in Z_t(\xi_{..t})$. We do so by proving recursively, from stage T to stage 0, the existence of an optimal solution for the optimization problem at the right-hand side of (5.4).

In the following, we denote by $\delta_C(\cdot)$ the function defined as $\delta_C(x) = 0$ if $x \in C$ and $\delta_C(x) = -\infty$ otherwise. Through this notation, we express the fact that the supremum of a real-valued function f over a set $C \subseteq \mathbb{R}^s$ can be written equivalently as the supremum of the *extended*-real-valued function $f + \delta_C$ over $\mathbb{R}^s$; see Rockafellar et Wets (2009, Chapter 1) for detailed developments on extended real analysis.

Take an arbitrary $t \in \{1, \ldots, T\}$ and suppose that the stage-t expected recourse function $(y_{..t}; \xi_{..t}) \mapsto Q_t(y_{..t}; \xi_{..t})$ is a normal integrand and that its effective domain

$$\operatorname{dom} Q_t(\cdot\,; \xi_{..t}) := \{y_{..t} \in \mathbb{R}^{s(t+1)} \mid Q_t(y_{..t}; \xi_{..t}) > -\infty\}, \tag{B.1}$$

includes $Z_t(\xi_{..t})$ for every $\xi_{..t} \in \Xi_{..t}$.

It follows from the above induction hypothesis and the properties of measurability and compactness of $Z_t(\xi_{..t})$ that the function

$$(y_{..t}; \xi_{..t}) \mapsto Q_t(y_{..t}; \xi_{..t}) + \delta_{Z_t(\xi_{..t})}(y_{..t}), \tag{B.2}$$

is a normal integrand too and, moreover, is level-bounded in y_t locally uniformly in $y_{..t-1}$ for each fixed $\xi_{..t} \in \Xi_{..t}$; see (Rockafellar et Wets, 2009, Example 14.32, Definition 1.16). Thus, the stage-t optimal recourse function, defined as

$$Q_t^*(y_{..t-1}; \xi_{..t}) = \sup_{y_t \in \mathbb{R}^s} \{Q_t(y_{..t-1}, y_t; \xi_{..t}) + \delta_{Z_t(\xi_{..t})}(y_{..t-1}, y_t)\}, \tag{B.3}$$

is a normal integrand by (Rockafellar et Wets, 2009, Proposition 14.47). Moreover, take an arbitrary $\xi_{..t} \in \Xi_{..t}$ and consider the following two cases: (i) if $y_{..t-1} \in Z_{t-1}(\xi_{..t-1})$, then $Z_t(\xi_{..t}) \neq \emptyset$ by Condition 5.2.3, and hence the supremum in (B.3) is attained, $Q_t^*(y_{..t-1}; \xi_{..t})$ is finite, and an optimal solution $x_t^*(y_{..t-1}; \xi_{..t})$ exists; (ii) if $y_{..t-1} \notin Z_{t-1}(\xi_{..t-1})$, then the supremum in (B.3) equals $-\infty$, and this value is consistent with the fact that such $y_{..t-1}$ is

not a vector of feasible decisions. Therefore, for every $\xi_{..t} \in \Xi_{..t}$ we have that

$$Q_t^*(y_{..t-1}; \xi_{..t}) \begin{cases} \in \mathbb{R} & \text{if } y_{..t-1} \in Z_{t-1}(\xi_{..t-1}); \\ = -\infty & \text{otherwise.} \end{cases} \tag{B.4}$$

This concludes the analysis of the equation (5.3).

As for the equation (5.4), we shall prove that the stage-$(t-1)$ expected recourse function

$$Q_{t-1}(y_{..t-1}; \xi_{..t-1}) = \mathbb{E}[Q_t^*(y_{..t-1}; \xi_{..t-1}, \boldsymbol{\xi}_t) \,|\, \boldsymbol{\xi}_{..t-1} = \xi_{..t-1}], \tag{B.5}$$

is a normal integrand and that $\operatorname{dom} Q_{t-1}(\cdot\,; \xi_{..t-1})$ includes $Z_{t-1}(\xi_{..t-1})$ for every $\xi_{..t-1} \in \Xi_{..t-1}$. This will allow the above arguments to be repeated at stage $t-1$, and hence will complete the proof since the initial stage-T expected recourse function is a finite-valued normal integrand by Condition 5.2.4. Take an arbitrary $\xi_{..t-1} \in \Xi_{..t-1}$ and consider the following two cases: (i) if $y_{..t-1} \in Z_{t-1}(\xi_{..t-1})$, then it follows from Condition 5.2.5 and an application of Lebesgue's dominated convergence theorem that $Q_{t-1}(\,\cdot\,; \xi_{..t-1})$ is finite-valued and upper semi-continuous at $y_{..t-1}$; (ii) if $y_{..t-1} \notin Z_{t-1}(\xi_{..t-1})$, then we have by (B.4) that $Q_t^*(y_{..t-1}; \xi_{..t-1}, \xi_t) = -\infty$ for all $\xi_t \in \Xi_t(\xi_{..t-1})$, and hence $Q_{t-1}(y_{..t-1}; \xi_{..t-1}) = -\infty$. Since $\xi_{..t-1} \rightrightarrows Z_{t-1}(\xi_{..t-1})$ is closed-valued and Q_{t-1} remains measurable, we deduce from (i)-(ii) that Q_{t-1} is a normal integrand and that its effective domain satisfies $\operatorname{dom} Q_{t-1}(\cdot\,; \xi_{..t-1}) = Z_{t-1}(\xi_{..t-1})$ for every $\xi_{..t-1} \in \Xi_{..t-1}$; see (Rockafellar et Wets, 2009, Corollary 14.34). This completes the proof.

ANNEXE C ADDITIONAL ARTICLE 4: PROOF PROPOSITION 7.3.1

We derive the bound values for $M = 4$. It is useful to notice that

$$\mathbb{E}\left[e^{Z(\epsilon_m)}\right] = 1/\delta \quad \text{and} \quad \mathbb{E}\left[e^{2Z(\epsilon_m)}\right]^{1/2} = \beta/\delta, \quad m = 1, \ldots, M, \tag{C.1}$$

where $\epsilon_1, \ldots, \epsilon_M$ are i.i.d. $N(0,1)$ random variables, $Z(\epsilon) = (r - \sigma^2/2)\Delta t + \epsilon\,\sigma\sqrt{\Delta t}$, $\delta = e^{-r\Delta t}$, and $\beta = e^{\frac{\sigma^2}{2}\Delta t}$.

(i) It follows directly from the inequality (7.17) that

$$\mathcal{V}(V_4)(\epsilon_1, \epsilon_2, \epsilon_3) = \mathbb{E}_{\epsilon_4}\left[\left(\frac{\partial V_4}{\partial \epsilon_4}\right)^2\right]^{1/2} \tag{C.2}$$

$$\leq \mathbb{E}_{\epsilon_4}\left[\left(\frac{\partial g_4}{\partial \epsilon_4}\right)^2\right]^{1/2} \tag{C.3}$$

$$= \delta^4 \frac{S_0}{4}\sigma\sqrt{\Delta t}\,\mathbb{E}\left[e^{2Z(\epsilon_4)}\right]^{1/2}\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right) \tag{C.4}$$

$$= \delta^3 \beta \frac{S_0}{4}\sigma\sqrt{\Delta t}\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right). \tag{C.5}$$

(ii) By the inequality (7.17):

$$\frac{\partial V_4}{\partial \epsilon_3} \leq \frac{\partial g_4}{\partial \epsilon_3} = \delta^4 \frac{S_0}{4}\sigma\sqrt{\Delta t}\left(1 + e^{Z(\epsilon_4)}\right)\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right), \tag{C.6}$$

hence by (7.19):

$$\frac{\partial C_3}{\partial \epsilon_3} = \mathbb{E}_{\epsilon_4}\left[\frac{\partial V_4}{\partial \epsilon_3}\right] \leq \mathbb{E}_{\epsilon_4}\left[\frac{\partial g_4}{\partial \epsilon_3}\right] = \delta^4 \frac{S_0}{4}\sigma\sqrt{\Delta t}\left(1 + \frac{1}{\delta}\right)\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right) \tag{C.7}$$

$$= \delta^3 \frac{S_0}{4}\sigma\sqrt{\Delta t}\,(1 + \delta)\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right), \tag{C.8}$$

and finally the inequality (7.18) yields:

$$\frac{\partial V_3}{\partial \epsilon_3} \leq \max\left\{\frac{\partial g_3}{\partial \epsilon_3}, \frac{\partial C_3}{\partial \epsilon_3}\right\} \leq \delta^3 S_0 \sigma\sqrt{\Delta t}\max\left\{\frac{1}{3}, \frac{1+\delta}{4}\right\}\left(\prod_{i=1}^{3} e^{Z(\epsilon_i)}\right). \tag{C.9}$$

Thus the variability of V_3 with respect to ϵ_3 satisfies:

$$\mathcal{V}(V_3)(\epsilon_1,\epsilon_2) = \mathbb{E}_{\epsilon_3}\left[\left(\frac{\partial V_3}{\partial \epsilon_3}\right)^2\right]^{1/2} \leq \delta^3 S_0 \sigma\sqrt{\Delta t}\max\left\{\frac{1}{3},\frac{1+\delta}{4}\right\}\mathbb{E}\left[e^{2Z(\epsilon_3)}\right]^{1/2}\left(\prod_{i=1}^{2} e^{Z(\epsilon_i)}\right) \tag{C.10}$$

$$= \delta^2 \beta S_0 \sigma\sqrt{\Delta t}\max\left\{\frac{1}{3},\frac{1+\delta}{4}\right\}\left(\prod_{i=1}^{2} e^{Z(\epsilon_i)}\right). \tag{C.11}$$

The bounds (iii) and (iv) are derived similarly:

(iii)

$$\frac{\partial V_4}{\partial \epsilon_2} \leq \frac{\partial g_4}{\partial \epsilon_2} = \delta^4 \frac{S_0}{4}\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)}\left(1+e^{Z(\epsilon_3)}\left(1+e^{Z(\epsilon_4)}\right)\right), \tag{C.12}$$

$$\Rightarrow \frac{\partial C_3}{\partial \epsilon_2} = \mathbb{E}_{\epsilon_4}\left[\frac{\partial V_4}{\partial \epsilon_2}\right] \leq \delta^4 \frac{S_0}{4}\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)}\left(1+e^{Z(\epsilon_3)}\left(1+\mathbb{E}_{\epsilon_4}\left[e^{Z(\epsilon_4)}\right]\right)\right), \tag{C.13}$$

$$= \delta^3 \frac{S_0}{4}\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)}\left(\delta+e^{Z(\epsilon_3)}\left(1+\delta\right)\right), \tag{C.14}$$

$$\Rightarrow \frac{\partial V_3}{\partial \epsilon_2} \leq \max\left\{\frac{\partial g_3}{\partial \epsilon_2},\frac{\partial C_3}{\partial \epsilon_2}\right\} \tag{C.15}$$

$$\leq \delta^3 S_0 \sigma\sqrt{\Delta t}\ \max\left\{\frac{1+e^{Z(\epsilon_3)}}{3},\frac{\delta+e^{Z(\epsilon_3)}(1+\delta)}{4}\right\} e^{Z(\epsilon_1)+Z(\epsilon_2)}. \tag{C.16}$$

For $\delta \leq \frac{4}{3}$, it holds that

$$\max\left\{\frac{1+e^{Z(\epsilon_3)}}{3},\frac{\delta+e^{Z(\epsilon_3)}(1+\delta)}{4}\right\} \leq \max\left\{\frac{1}{3}+\frac{e^{Z(\epsilon_3)}}{3},\frac{1}{3}+\frac{e^{Z(\epsilon_3)}(1+\delta)}{4}\right\} \tag{C.17}$$

$$= \frac{1}{3}+e^{Z(\epsilon_3)}\max\left\{\frac{1}{3},\frac{(1+\delta)}{4}\right\}, \tag{C.18}$$

hence

$$\frac{\partial C_2}{\partial \epsilon_2} = \mathbb{E}_{\epsilon_3}\left[\frac{\partial V_3}{\partial \epsilon_2}\right] \leq \delta^3 S_0 \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)} \left(\frac{1}{3} + \mathbb{E}_{\epsilon_3}\left[e^{Z(\epsilon_3)}\right] \max\left\{\frac{1}{3}, \frac{(1+\delta)}{4}\right\}\right) \tag{C.19}$$

$$= \delta^2 S_0 \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)} \left(\frac{\delta}{3} + \max\left\{\frac{1}{3}, \frac{(1+\delta)}{4}\right\}\right) \tag{C.20}$$

$$= \delta^2 S_0 \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)+Z(\epsilon_2)} \max\left\{\frac{1+\delta}{3}, \frac{\delta}{3} + \frac{(1+\delta)}{4}\right\}, \tag{C.21}$$

$$\Rightarrow \frac{\partial V_2}{\partial \epsilon_2} \leq \max\left\{\frac{\partial g_2}{\partial \epsilon_2}, \frac{\partial C_2}{\partial \epsilon_2}\right\} \leq \delta^2 S_0 \sigma \sqrt{\Delta t}\, \max\left\{\frac{1}{2}, \frac{1+\delta}{3}, \frac{\delta}{3} + \frac{(1+\delta)}{4}\right\} e^{Z(\epsilon_1)+Z(\epsilon_2)}. \tag{C.22}$$

$$\Rightarrow \mathcal{V}(V_2)(\epsilon_1) = \mathbb{E}_{\epsilon_2}\left[\left(\frac{\partial V_2}{\partial \epsilon_2}\right)^2\right]^{1/2} \leq \delta S_0 \beta \sigma \sqrt{\Delta t}\, \max\left\{\frac{1}{2}, \frac{1+\delta}{3}, \frac{\delta}{3} + \frac{(1+\delta)}{4}\right\} e^{Z(\epsilon_1)}. \tag{C.23}$$

(iv)

$$\frac{\partial V_4}{\partial \epsilon_1} \leq \frac{\partial g_4}{\partial \epsilon_1} = \delta^4 \frac{S_0}{4} \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)} \left(1 + e^{Z(\epsilon_2)}\Big(1 + e^{Z(\epsilon_3)}\Big(1 + e^{Z(\epsilon_4)}\Big)\Big)\right), \tag{C.24}$$

$$\Rightarrow \frac{\partial C_3}{\partial \epsilon_1} = \mathbb{E}_{\epsilon_4}\left[\frac{\partial V_4}{\partial \epsilon_1}\right] \leq \delta^4 \frac{S_0}{4} \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)} \left(1 + e^{Z(\epsilon_2)}\Big(1 + e^{Z(\epsilon_3)}\Big(1 + \mathbb{E}_{\epsilon_4}\left[e^{Z(\epsilon_4)}\right]\Big)\Big)\right), \tag{C.25}$$

$$= \delta^3 \frac{S_0}{4} \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)} \left(\delta + e^{Z(\epsilon_2)}\Big(\delta + e^{Z(\epsilon_3)}\Big(1+\delta\Big)\Big)\right), \tag{C.26}$$

$$\Rightarrow \frac{\partial V_3}{\partial \epsilon_1} \leq \max\left\{\frac{\partial g_3}{\partial \epsilon_1}, \frac{\partial C_3}{\partial \epsilon_1}\right\} \tag{C.27}$$

$$\leq \delta^3 S_0 \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)} \max\left\{\frac{1 + e^{Z(\epsilon_2)}(1 + e^{Z(\epsilon_3)})}{3}, \frac{\delta + e^{Z(\epsilon_2)}\Big(\delta + e^{Z(\epsilon_3)}(1+\delta)\Big)}{4}\right\}. \tag{C.28}$$

For $\delta \leq \frac{4}{3}$, it holds that

$$\max\left\{\frac{1+e^{Z(\epsilon_2)}(1+e^{Z(\epsilon_3)})}{3}, \frac{\delta+e^{Z(\epsilon_2)}\Big(\delta+e^{Z(\epsilon_3)}\Big(1+\delta\Big)}{4}\right\} \tag{C.29}$$

$$\leq \frac{1}{3}+e^{Z(\epsilon_2)}\max\left\{\frac{1+e^{Z(\epsilon_3)}}{3}, \frac{\delta+e^{Z(\epsilon_3)}\Big(1+\delta\Big)}{4}\right\} \tag{C.30}$$

$$\leq \frac{1}{3}+e^{Z(\epsilon_2)}\left(\frac{1}{3}+e^{Z(\epsilon_3)}\max\left\{\frac{1}{3}, \frac{1+\delta}{4}\right\}\right), \tag{C.31}$$

hence

$$\frac{\partial C_2}{\partial \epsilon_1} = \mathbb{E}_{\epsilon_3}\left[\frac{\partial V_3}{\partial \epsilon_1}\right] \leq \delta^3 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{1}{3}+e^{Z(\epsilon_2)}\left(\frac{1}{3}+\mathbb{E}_{\epsilon_3}\left[e^{Z(\epsilon_3)}\right]\max\left\{\frac{1}{3}, \frac{1+\delta}{4}\right\}\right)\right) \tag{C.32}$$

$$= \delta^2 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{\delta}{3}+e^{Z(\epsilon_2)}\left(\frac{\delta}{3}+\max\left\{\frac{1}{3}, \frac{1+\delta}{4}\right\}\right)\right) \tag{C.33}$$

$$= \delta^2 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{\delta}{3}+e^{Z(\epsilon_2)}\max\left\{\frac{1+\delta}{3}, \frac{1+\delta}{4}+\frac{\delta}{3}\right\}\right) \tag{C.34}$$

$$\Rightarrow \frac{\partial V_2}{\partial \epsilon_1} \leq \max\left\{\frac{\partial g_2}{\partial \epsilon_1}, \frac{\partial C_2}{\partial \epsilon_1}\right\} \tag{C.35}$$

$$\leq \delta^2 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\max\left\{\frac{1+e^{Z(\epsilon_2)}}{2}, \frac{\delta}{3}+e^{Z(\epsilon_2)}\max\left\{\frac{1+\delta}{3}, \frac{1+\delta}{4}+\frac{\delta}{3}\right\}\right\} \tag{C.36}$$

$$\leq \delta^2 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{1}{2}+e^{Z(\epsilon_2)}\max\left\{\frac{1}{2}, \frac{1+\delta}{3}, \frac{1+\delta}{4}+\frac{\delta}{3}\right\}\right), \tag{C.37}$$

where at the last inequality we use the fact $\frac{\delta}{3} \leq \frac{1}{2}$ whenever $\delta \leq 4/3$.

$$\frac{\partial C_1}{\partial \epsilon_1} = \mathbb{E}_{\epsilon_2}\left[\frac{\partial V_2}{\partial \epsilon_1}\right] \leq \delta^2 S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{1}{2}+\mathbb{E}_{\epsilon_2}\left[e^{Z(\epsilon_2)}\right]\max\left\{\frac{1}{2}, \frac{1+\delta}{3}, \frac{1+\delta}{4}+\frac{\delta}{3}\right\}\right) \tag{C.38}$$

$$= \delta S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\left(\frac{\delta}{2}+\max\left\{\frac{1}{2}, \frac{1+\delta}{3}, \frac{1+\delta}{4}+\frac{\delta}{3}\right\}\right) \tag{C.39}$$

$$= \delta S_0\sigma\sqrt{\Delta t}\, e^{Z(\epsilon_1)}\max\left\{\frac{1+\delta}{2}, \frac{\delta}{2}+\frac{1+\delta}{3}, \frac{\delta}{2}+\frac{\delta}{3}+\frac{1+\delta}{4}\right\}, \tag{C.40}$$

$$\Rightarrow \frac{\partial V_1}{\partial \epsilon_1} \leq \max\left\{\frac{\partial g_1}{\partial \epsilon_1}, \frac{\partial C_1}{\partial \epsilon_1}\right\} \tag{C.41}$$

$$\leq \delta S_0 \sigma \sqrt{\Delta t}\, e^{Z(\epsilon_1)} \max\left\{1, \frac{1+\delta}{2}, \frac{\delta}{2} + \frac{1+\delta}{3}, \frac{\delta}{2} + \frac{\delta}{3} + \frac{1+\delta}{4}\right\}, \tag{C.42}$$

$$\Rightarrow \mathcal{V}(V_1) = \mathbb{E}_{\epsilon_1}\left[\left(\frac{\partial V_1}{\partial \epsilon_1}\right)^2\right]^{1/2} \leq S_0 \beta \sigma \sqrt{\Delta t}\, \max\left\{1, \frac{1+\delta}{2}, \frac{\delta}{2} + \frac{1+\delta}{3}, \frac{\delta}{2} + \frac{\delta}{3} + \frac{1+\delta}{4}\right\}. \tag{C.43}$$

This completes the proof.

ANNEXE D ADDITIONAL ARTICLE 4: PROOF PROPOSITION 7.3.3

To simplify the notations, let c_m be a shorthand for $\mathbb{E}[\gamma_m(S_m)]$ in the following. The minimization problem (7.39) is not convex because of the non-convexity of the feasible set (7.40). It is however easy to transform it into an equivalent convex problem by the change of variables: $\widetilde{b}_m := \log(b_m),\ m = 0, \ldots, M-1$, and $\widetilde{N}_M := \log(N_M)$.

In terms of the new variables, the minimization problem is now written as

$$\min_{(\widetilde{b}_0,\ldots,\widetilde{b}_{M-1})\in\mathbb{R}_+^M} \sum_{m=0}^{M-1} c_m \exp(-\alpha\widetilde{b}_m) \quad \text{subject to} \quad \sum_{m=0}^{M-1} \widetilde{b}_m = \widetilde{N}_M. \tag{D.1}$$

The objective function is smooth convex and the feasible set is convex (the equality constraint are even defining an affine subspace). The sufficient conditions for some point $(\widetilde{b}_0^*, \ldots, \widetilde{b}_{M-1}^*) \in \mathbb{R}_+^M$ to be a global minimum of (D.1) can be found for instance in (Ruszczyński, 2006, Theorem 3.34). We now apply those conditions to our problem. The problem can be recast into the framework of the theorem as follows:

$$\min_{\widetilde{b}\in X_0}\{f(\widetilde{b}) : h(\widetilde{b}) = 0\}, \tag{D.2}$$

with:

- $X_0 = \mathbb{R}_+^M$;
- $f(\widetilde{b}) = \sum_{m=0}^{M-1} c_m \exp(-\alpha\widetilde{b}_m)$;
- $h(\widetilde{b}) = \sum_{m=0}^{M-1} \widetilde{b}_m - \widetilde{N}_M$.

The gradient vectors of f and h are

$$\nabla f(\widetilde{b}) = -\alpha \begin{bmatrix} c_0 \exp(-\alpha\widetilde{b}_0) \\ \vdots \\ c_{M-1}\exp(-\alpha\widetilde{b}_{M-1}) \end{bmatrix} \quad \text{and} \quad \nabla h(\widetilde{b}) = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{D.3}$$

and the normal cone to X_0 at $\widetilde{b}$ is

$$N_{X_0}(\widetilde{b}) = \{\lambda \in \mathbb{R}_-^M : \lambda^\top \widetilde{b} = 0\}. \tag{D.4}$$

To obtain the optimality conditions, let us differentiate two cases: (a) the global minimum

lies in the interior of X_0 (i.e., the active set is empty), (b) the global minimum lies on the boundary of X_0 (the active set is not empty). Recall that the existence of a global mimimum is guaranteed by continuity of f and compactness of the feasible set $\{\widetilde{b} \in X_0 : h(\widetilde{b}) = 0\}$.

(a) If the global minimum $\widetilde{b}^*$ lies in the interior of X_0, then the normal cone at $\widetilde{b}^*$ only contains the zero vector $(0, \ldots, 0)^\top$, and therefore the optimality condition is

$$\nabla f(\widetilde{b}^*) = -\mu \nabla h(\widetilde{b}^*) = -\mu \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad \text{for some } \mu \in \mathbb{R}. \tag{D.5}$$

Taking the product of all the components in the above equality yields

$$\mu^M = \alpha^M \left(\prod_{m=0}^{M-1} c_m \right) \exp\left(-\alpha \sum_{m=0}^{M-1} \widetilde{b}_m^* \right) = \alpha^M \left(\prod_{m=0}^{M-1} c_m \right) \exp(-\alpha \widetilde{N}_M) \tag{D.6}$$

$$\Rightarrow \mu = \alpha \left(\prod_{m=0}^{M-1} c_m \right)^{1/M} \exp(-\frac{\alpha}{M} \widetilde{N}_M). \tag{D.7}$$

By inserting μ into the above optimality condition, we have that $\widetilde{b}^*$ satisfies

$$c_m \exp(-\alpha \widetilde{b}_m^*) = \left(\prod_{m=0}^{M-1} c_m \right)^{1/M} \exp(-\frac{\alpha}{M} \widetilde{N}_M), \quad m = 0, \ldots, M-1. \tag{D.8}$$

By changing the variables back to b and N_M, we have that the global minimum b^* is given by

$$b_m^* = (N_M)^{1/M} \frac{(c_m)^{1/\alpha}}{\left(\prod_{i=0}^{M-1} (c_i)^{1/\alpha} \right)^{1/M}}, \quad m = 0, \ldots, M-1. \tag{D.9}$$

This proves (7.41).

(b) Suppose now that the global minimum $\widetilde{b}^*$ lies on the boundary of X_0. Let $\mathcal{A}$ be the active set, i.e., $\mathcal{A} = \{m \in \{0, \ldots, M-1\} : \widetilde{b}_m^* = 0\}$, and $\mathcal{I}$ its complement. The normal cone at $\widetilde{b}^*$ is

$$N_{X_0}(\widetilde{b}^*) = \{(\lambda_0, \ldots, \lambda_{M-1}) : \lambda_i \leq 0,\ i \in \mathcal{A},\ \lambda_j = 0,\ j \in \mathcal{I}\}. \tag{D.10}$$

Therefore, in that case the optimality condition is

$$-\nabla f(\widetilde{b}^*) - \mu \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in N_{X_0}(\widetilde{b}^*), \qquad \text{for some } \mu \in \mathbb{R}. \tag{D.11}$$

Depending on whether the index m is in $\mathcal{A}$ or $\mathcal{I}$, the above condition splits into two different conditions:

$$\exists \mu \in \mathbb{R} \quad \text{such that} \quad \begin{cases} \alpha c_j \exp(-\alpha \widetilde{b}_j^*) - \mu = 0, & \text{for all } j \in \mathcal{I}; \quad \text{(D.12)} \\ \alpha c_i - \mu \leq 0, & \text{for all } i \in \mathcal{A}. \quad \text{(D.13)} \end{cases}$$

By changing the variables back to b and N_M and by changing $\mu \leftarrow \mu/\alpha$, we have that the global minimum b^* satisfies:

$$\exists \mu \in \mathbb{R} \quad \text{such that} \quad \begin{cases} \dfrac{c_j}{(b_j^*)^\alpha} = \mu, & \text{for all } j \in \mathcal{I}; \quad \text{(D.14)} \\ c_i \leq \mu, & \text{for all } i \in \mathcal{A}. \quad \text{(D.15)} \end{cases}$$

This proves the optimality conditions (7.42)-(7.43).