UNIVERSITÉ DU QUÉBEC À MONTRÉAL

OPTIMISATION STOCHASTIQUE DE PORTEFEUILLES SOUS DIVERSES MESURES DE RISQUE

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

ANTHONY COACHE

FÉVRIER 2020

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

J'aimerais tout d'abord remercier mon directeur François Watier et mon co-directeur René Ferland pour leur appui et m'avoir transmis leur passion pour leurs recherches. Vous m'avez initié à la recherche et j'ai adoré collaborer en votre compagnie.

Je m'en voudrais de ne pas aussi remercier les autres professeurs du département qui ont entretenu ma passion de la statistique et qui m'ont permis de m'épanouir. Un merci tout particulier à Sorana Froda, Jean-François Coeurjolly et Fabrice Larribe avec qui j'ai eu le plaisir de collaborer et aux membres de l'EMoStA.

Un merci à Renaud, Olivier, Patrick, Florence et Joëlle, pour les discussions, les échanges, les collaborations et pour avoir égayé mon parcours académique.

J'aimerais aussi remercier mes parents et ma soeur pour leur soutien inconditionnel tout au long de mes études universitaires.

Enfin, je tiens à remercier le CRSNG et le FRQNT pour leur appui financier lors de ma maîtrise.

TABLE DES MATIÈRES

LISTE DES TABLEAUX v
LISTE DES FIGURES vi
RÉSUMÉ vii
INTRODUCTION 1
CHAPITRE I MESURES DE RISQUE ET NOTIONS PRÉLIMINAIRES 3
1.1 Quelques notions de probabilités 3
1.2 Quelques mesures de risque... 6
1.2.1 ... et leurs propriétés 9
1.3 La notion de cohérence... 11
1.3.1 ... de la variance 12
1.3.2 ... de la valeur-à-risque 12
1.3.3 ... de l'espérance conditionnelle de la queue 14
1.3.4 ... de l'*expected-shortfall* 15
CHAPITRE II PROBLÈME MOYENNE-SHORTFALL À UNE PÉRIODE 19
2.1 Moyenne-shortfall... 19
2.1.1 ... et son équivalence avec le moyenne-variance 21
2.1.2 ... et sa solution analytique 24
2.1.3 ... et ses propriétés 27
2.2 Validation empirique de la solution et de ces propriétés 29
CHAPITRE III PROBLÈME MOYENNE-SHORTFALL MULTIPÉRIODIQUE 31
3.1 Problème moyenne-shortfall multipériodique... 31
3.1.1 ... avec un modèle binomial 33
3.1.2 ... où chercher le minimum ? 41

3.2 Algorithme de recherche stochastique... 45

3.2.1 ... uniforme discret avec remise 47

3.2.2 ... uniforme discret sans remise 48

3.2.3 ... *Markovian change-when-improve* 50

CHAPITRE IV COMPARAISON DES ALGORITHMES STOCHASTIQUES 53

4.1 Paramètres des études de simulation... 53

4.1.1 ... pour la distribution du nombre d'itérations avant l'obtention du minimum global 54

4.1.2 ... pour l'efficacité en un nombre fixé d'itérations 55

CONCLUSION 62

APPENDICE A ANNEXES 64

A.1 Script Matlab pour la comparaison moyenne-variance et moyenne-shortfall à une période 64

A.2 Fonction Matlab pour calculer la mesure de risque d'un candidat . . . 69

A.3 Scripts Matlab pour comparer la méthode uniforme sans remise et le *Markovian change-when-improve* 71

LISTE DES TABLEAUX

Tableau Page

4.1 Comparaison des distributions du nombre d'itérations pour trouver le minimum global pour les deux méthodes 55

LISTE DES FIGURES

Figure		Page
2.1	Illustration du problème moyenne-shortfall à une période	20
3.1	Illustration du problème moyenne-shortfall multipériodique	32
3.2	Illustration des $\pi_i^{(j)}$ correspondants aux différents scénarios de la valeur de l'actif risqué	37
3.3	Algorithme pour générer les valeurs de la richesse terminale et leur probabilité associée	42
3.4	Algorithme de sélection uniforme d'un partage	46
3.5	Algorithme de recherche uniforme avec remise	48
3.6	Algorithme de recherche uniforme sans remise	49
3.7	Algorithme *Markovian change-when-improve*	52
4.1	Distribution du nombre d'itérations pour trouver le minimum global avec la méthode *MCWI*	56
4.2	Distribution de la valeur minimale obtenu après 4 itérations pour le problème à deux périodes	58
4.3	Distribution de la valeur minimale obtenu après 65 itérations pour le problème à trois périodes	58
4.4	Distribution de la valeur minimale obtenu après 150 itérations pour le problème à quatre périodes	59
4.5	Proportion des 10000 réplications ayant trouvé le minimum global	61

RÉSUMÉ

Ce mémoire porte sur l'étude d'un problème de gestion de portefeuilles où un investisseur potentiel souhaite déterminer une stratégie pour obtenir en moyenne une certaine richesse terminale tout en minimisant un risque dit cohérent. Nous nous intéressons tout d'abord à la mesure de risque à privilégier, ce qui nous incitera à utiliser une généralisation discrète de l'*expected-shortfall*.

Nous analysons ensuite le problème *moyenne-shortfall* à une période, un problème d'optimisation résolu par Bertsimas *et al.* (2004), et vérifions certaines de ces propriétés empiriquement.

Nous cherchons finalement à généraliser ce problème dans un contexte plus réaliste. En effet, on peut penser que l'investisseur devrait pouvoir réagir aux fluctuations du marché et adapter sa stratégie financière en conséquence pendant la période d'investissement. Nous nous intéressons donc au problème moyenne-shortfall multipériodique où le prix des actifs financiers suit un modèle d'arbre binomial. Nous explorons diverses approches stochastiques pour trouver une stratégie optimale dans ce contexte particulier et comparons leur performance et leur efficacité par des études de simulation.

MOTS-CLÉS : gestion de portefeuilles, optimisation stochastique, mesure de risque, multipériodique.

INTRODUCTION

En théorie de la finance mathématique, un problème répandu consiste à trouver une stratégie financière minimisant une mesure de risque sous la contrainte d'atteindre une richesse espérée à une date terminale prédéterminée. Si nous modélisons la dynamique des prix des titres du marché où un individu peut investir à l'aide de processus stochastiques, alors ce problème se situe dans la théorie mathématique du contrôle stochastique.

Historiquement, ce problème a été résolu la première fois lorsque la mesure de risque choisie était la variance (Markowitz, 1952, 1968). Cela a d'ailleurs valu au chercheur Harry Markowitz le prix Nobel d'économie en 1990. Cependant, une des principales critiques émises pour le choix de la variance comme mesure de risque est sa symétrie autour de la richesse espérée, si bien que Artzner *et al.* (1999) ont proposé la notion de mesure de risque cohérente. Depuis, plusieurs chercheurs se sont penchés sur le problème de gestion de portefeuilles sous différentes mesures de risque cohérentes, mais leurs résultats ont l'hypothèse restrictive commune que l'individu est limité à une seule période d'investissement.

Ce mémoire porte donc sur l'étude du problème de gestion optimale de portefeuilles moyenne-shortfall dans un contexte multipériodique par différentes approches d'optimisation stochastique. Tout d'abord, au chapitre 1, nous nous intéressons aux mesures de risque et à la notion de cohérence, ce qui nous incite à préconiser une généralisation discrète de l'*expected-shortfall* pour quantifier le risque. Au chapitre 2, nous enchaînons avec l'analyse du problème moyenne-shortfall à une période, un problème résolu par Bertsimas *et al.* (2004). Nous généralisons

ensuite au chapitre 3 ce problème dans un cas multipériodique où le prix des actifs risqués suit un modèle d'arbre binomial. Nous proposons ainsi diverses approches stochastiques pour explorer la résolution de ce problème ouvert. Finalement, le chapitre 4 met en lumière la performance des différentes méthodes proposées.

CHAPITRE I

MESURES DE RISQUE ET NOTIONS PRÉLIMINAIRES

En gestion de portefeuilles, l'investisseur cherche toujours à contrôler le risque. Pour ce faire, il peut soit essayer de minimiser le risque possible sur ces investissements ou encore maximiser son profit tout en n'excédant pas un seuil de risque considéré critique. C'est dans ce contexte que certains définissent des mesures de risque pour quantifier cette incertitude. Or, certaines mesures possèdent des propriétés qui reflètent davantage le comportement d'un investisseur, par exemple privilégier la diversification des actifs, et qui sont plus attrayantes à préconiser.

Dans ce chapitre, nous allons énumérer certaines des mesures de risque les plus connues dans la littérature et rappeler la notion de cohérence, ce qui justifiera notre choix de mesure de risque dans notre problème de gestion de portefeuilles.

1.1 Quelques notions de probabilités

L'objectif de cette section est de définir formellement ce qu'est une mesure de risque. Pour y arriver, commençons par introduire un espace de probabilité et le vocabulaire associé. Le contenu se base sur le chapitre 1 du livre de Casella et Berger (2002) et les chapitres 2 et 3 de l'ouvrage de Rosenthal (2006).

Définition 1.1.1. Soit Ω un ensemble non-vide et $\mathcal{F}$ une famille de sous-ensembles de Ω. On dit que $\mathcal{F}$ est une **tribu** (ou **σ-algèbre**) si les trois conditions suivantes

sont respectées :

1. $\emptyset \in \mathcal{F}$.
2. Si $A \in \mathcal{F}$, alors $A^c \in \mathcal{F}$.
3. Si $\{A_i\ ,\ i = 1, 2, \ldots\}$ où $A_n \in \mathcal{F}\ \forall n$, alors

$$\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}.$$

Si Ω est un ensemble non-vide et $\mathcal{F}$ est une tribu, alors $(\Omega, \mathcal{F})$ est un espace mesurable.

Définition 1.1.2. Soit $(\Omega, \mathcal{F})$ un espace mesurable. Une **mesure** est une application $\mu : \mathcal{F} \to [0, \infty]$ telle que

1. $\mu(\emptyset) = 0$.
2. Si $\{A_i\ ,\ i = 1, 2, \ldots\}$ où $A_n \in \mathcal{F}\ \forall n$ et les A_n sont deux à deux disjoints, alors

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

Si $\mu(\Omega) = 1$, alors μ est une **mesure de probabilité.**

On dit que $(\Omega, \mathcal{F}, P)$ est un espace de probabilité si $(\Omega, \mathcal{F})$ est un espace mesurable et P est une mesure de probabilité. Nous pouvons maintenant définir des fonctions sur cet espace.

Définition 1.1.3. Soit $(\Omega, \mathcal{F}, P)$ un espace de probabilité. Une **variable aléatoire** est une application $X : \Omega \to \mathbb{R}$ telle que pour tout sous-ensemble de la tribu de Borel B, nous avons $\{w \in \Omega \mid X(w) \in B\} \in \mathcal{F}$.

Plutôt que d'appeler la fonction X une variable aléatoire, certains préfèrent dire que X est une fonction $\mathcal{F}$-mesurable. Par ailleurs, nous remarquons que la définition de variable aléatoire ne dépend pas de la mesure de probabilité utilisée. Il ne

reste qu'à introduire l'espérance mathématique pour pouvoir ainsi décrire l'espace des variables aléatoires que nous utiliserons.

Définition 1.1.4. Soit $(\Omega, \mathcal{F}, P)$ un espace de probabilité et une variable aléatoire X. L'**espérance** de X est définie par

$$\mathrm{E}[X] = \int_{\Omega} X \, \mathrm{d}P,$$

où l'intégrale est ici comprise au sens de Lebesgue (Rosenthal, 2006).

Cette espérance est bien définie si $X \geq 0$ ou si X est intégrable, c'est-à-dire si $\mathrm{E}[|X|] < \infty$. Une variable aléatoire peut être qualifiée de p-intégrable, ce qui signifie que la variable aléatoire $|X|^p$ est intégrable.

Note. **L'espace des variables aléatoires p-intégrables** est noté $L^p(\Omega, \mathcal{F}, P)$. L'espace des variables aléatoires est noté $L^0(\Omega, \mathcal{F}, P)$, tandis que l'espace des variables aléatoires bornées est noté par $L^\infty(\Omega, \mathcal{F}, P)$.

Ce travail préliminaire est nécessaire pour définir formellement une mesure de risque dans un cas général. Voici la définition en question.

Définition 1.1.5. Une **mesure de risque** (Artzner *et al.*, 1999) est une application de la forme

$$\rho : L^\infty(\Omega, \mathcal{F}, P) \longrightarrow \mathbb{R}.$$

En résumé, une mesure de risque est une fonction qui associe pour une certaine variable aléatoire, disons la valeur de la richesse à un moment donné, un réel indiquant le degré de risque de cette variable. Pour le moment, nous ne donnons aucune restriction à la mesure de risque, mais nous évoquerons leurs propriétés pour les comparer dans les sections suivantes.

1.2 Quelques mesures de risque...

Maintenant qu'une mesure de risque nous est familière sous sa forme générale, nous allons donner quelques exemples classiques de mesures de risque, à commencer par la variance.

La variance est considérée comme une mesure de risque. En effet, les premiers développements en théorie de la finance mathématique ont débuté en utilisant la variance. Historiquement, le problème moyenne-variance, correspondant à minimiser la variance du rendement du portefeuille tout en fixant son espérance, a été résolu la première fois par le chercheur Harry Markowitz (1952, 1968), ce qui lui a d'ailleurs valu le prix Nobel d'économie en 1990.

Définition 1.2.1. La **variance** de la variable aléatoire X est donnée par

$$\mathrm{Var}[X] = \mathrm{E}[(X - \mathrm{E}[X])^2].$$

Cette mesure de risque, bien que simple d'utilisation, a reçu son lot de critiques. Une des principales critiques émises pour le choix de la variance comme mesure de risque est qu'elle ne prend pas en compte l'asymétrie des distributions. Ainsi, en minimisant la variance, nous pouvons pénaliser aussi bien les rendements financiers supérieurs que les rendements inférieurs à notre objectif, ce qui est complètement absurde du point de vue d'un investisseur. C'est pourquoi plusieurs auteurs se sont penchés sur des développements de différentes mesures de risque ayant des propriétés reflétant davantage le comportement d'un investisseur type. Continuons notre tour d'horizon des différentes mesures de risque en introduisant la valeur-à-risque.

Définition 1.2.2. Soit $\alpha \in (0,1)$. Le **α-quantile** de la variable aléatoire X est donné par

$$q_\alpha(X) = \inf\{x \mid \mathrm{P}[X \leq x] \geq \alpha\}.$$

Définition 1.2.3. Soit un niveau de confiance $\alpha \in (0,1)$. La **valeur-à-risque**, en anglais *value-at-risk* (Artzner *et al.*, 1999, p. 216), au niveau α de la variable aléatoire X, communément notée $\mathrm{VaR}_\alpha(X)$, est définie par

$$\mathrm{VaR}_\alpha(X) = -q_\alpha(X) = -\inf\{x \mid \mathrm{P}[X \leq x] \geq \alpha\}.$$

En d'autres termes, la valeur-à-risque est la valeur négative du α-quantile de la variable aléatoire X. Nous préférons utiliser la valeur négative seulement à des fins pratiques, plus particulièrement puisque nous formulons en termes de pertes plutôt que de gains. L'introduction d'un quantile dans la mesure de risque permet de quantifier l'incertitude seulement dans les pires scénarios, une composante qui est inexistante dans la variance par exemple. Voici une autre mesure de risque couramment mentionnée dans la littérature basée sur un quantile.

Définition 1.2.4. Soit un niveau de confiance $\alpha \in (0,1)$. L'**espérance conditionnelle de la queue**, en anglais *tail conditional expectation* (Landsman et Valdez, 2003), au niveau α de la variable aléatoire X, communément notée $\mathrm{TCE}_\alpha(X)$, est définie par

$$\mathrm{TCE}_\alpha(X) = \mathrm{E}[-X \mid X \leq q_\alpha(X)].$$

Cette mesure de risque est souvent préconisée lorsque la variable X est continue. Or, si la distribution de la variable aléatoire est discrète ou indéterminée, certains préfèrent parfois utiliser la mesure de risque suivante.

Définition 1.2.5. Soit un niveau de confiance $\alpha \in (0,1)$. La **perte moyenne outre-seuil**, en anglais *expected-shortfall* (Acerbi et Tasche, 2002a), au niveau α de la variable aléatoire X, communément notée $\mathrm{ES}_\alpha(X)$, est définie par

$$\mathrm{ES}^1_\alpha(X) = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma.$$

Cette expression est équivalente à la reformulation suivante donnée par Acerbi et Tasche (2002b) :

$$\mathrm{ES}_\alpha^2(X) = -\frac{1}{\alpha}\left(\mathrm{E}\left[X \cdot \mathbb{1}(X \leq q_\alpha(X))\right] + q_\alpha(X)(\alpha - P[X \leq q_\alpha(X)])\right),$$

où $\mathbb{1}(\cdot)$ est la fonction indicatrice.

Tout d'abord, vérifions que les deux expressions de l'*expected-shortfall* dans la définition 1.2.5 sont équivalentes. En effet, comme nous savons que X a la même distribution que $q_U(X)$, où $U \sim \mathcal{U}(0,1)$, nous obtenons

$$\begin{aligned}
\mathrm{ES}_\alpha^1(X) &= \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma \\
&= -\frac{1}{\alpha}\int_0^\alpha q_\gamma(X)\,\mathrm{d}\gamma \\
&= -\frac{1}{\alpha}\mathrm{E}\left[\, q_U(X)\,\mathbb{1}(U \leq \alpha)\right], \quad \text{où } U \sim \mathcal{U}(0,1), \\
&= -\frac{1}{\alpha}\left(\mathrm{E}\left[\, q_U(X)\,\mathbb{1}(q_U(X) \leq q_\alpha(X))\right]\right. \\
&\qquad \left. - \mathrm{E}\left[q_U(X)\,\mathbb{1}(\{U > \alpha\} \cap \{q_U(X) \leq q_\alpha(X)\})\right]\right) \\
&= -\frac{1}{\alpha}\left(\mathrm{E}\left[\, X\,\mathbb{1}(X \leq q_\alpha(X))\right] - q_\alpha(X)\,\mathrm{P}\left[\{U > \alpha\} \cap \{q_U(X) \leq q_\alpha(X)\}\right]\right) \\
&= -\frac{1}{\alpha}\left(\mathrm{E}\left[\, X\,\mathbb{1}(X \leq q_\alpha(X))\right] - q_\alpha(X)\left(\mathrm{P}[X \leq q_\alpha(X)] - \alpha\right)\right) \\
&= \mathrm{ES}_\alpha^2(X).
\end{aligned}$$

Dans la littérature, l'*expected-shortfall* est parfois aussi appelé valeur-à-risque conditionnelle ou *conditional value-at-risk* (Rockafellar et Uryasev, 2002), valeur-à-risque moyenne ou *average value-at-risk* (Föllmer et Schied, 2008) ou bien perte moyenne de la queue ou *expected tail loss.*

Il existe un lien entre l'espérance conditionnelle de la queue et l'*expected-shortfall*, ce qui fait l'objet de la proposition 1.2.6.

Proposition 1.2.6. *Si X est une variable aléatoire continue, alors les définitions 1.2.4 et 1.2.5 sont équivalentes.*

Démonstration. Effectivement, puisque $\mathrm{P}[X \leq q_\alpha(X)] = \alpha$ lorsque X est continue, nous obtenons

$$\begin{aligned}
\mathrm{TCE}_\alpha(X) &= \mathrm{E}[-X \mid X \leq q_\alpha(X)] \\
&= -\frac{1}{\mathrm{P}[X \leq q_\alpha(X)]} \left(\mathrm{E}\left[X \cdot \mathbb{1}(X \leq q_\alpha(X))\right]\right) \\
&= -\frac{1}{\alpha} \left(\mathrm{E}\left[X \cdot \mathbb{1}(X \leq q_\alpha(X))\right] + q_\alpha(X)(\alpha - P[X \leq q_\alpha(X)])\right) \\
&= \mathrm{ES}^2_\alpha(X).
\end{aligned}$$

■

La proposition 1.2.6 nous permet donc d'affirmer que l'*expected-shortfall* est une généralisation de l'espérance conditionnelle de la queue.

Nous remarquons qu'autant la variance, la valeur-à-risque, l'espérance conditionnelle de la queue que la perte moyenne outre-seuil sont des mesures de risque au sens de la définition 1.1.5. Effectivement, toutes les mesures de risque sont des fonctions qui prennent une variable aléatoire comme entrée et donnent un réel.

Il existe aussi d'autres formulations pour les mesures de risque énoncées précédemment. Par exemple, nous pouvons les définir avec une fonction de distorsion, appelée en anglais *distortion risk measure.* Ce type de mesure de risque fait intervenir la fonction de répartition de la variable aléatoire en question. Plus de renseignements sur ces mesures de risque particulières sont disponibles dans l'article de Balbás *et al.* (2009).

1.2.1 ... et leurs propriétés

Avant de continuer et de déterminer quelles mesures de risque devraient être privilégiées, nous pouvons d'ores et déjà remarquer certaines propriétés de celles-ci. Nous pourrons montrer les avantages de chacune des mesures en utilisant les ré-

sultats suivants.

Proposition 1.2.7. $\mathrm{VaR}_\alpha(X)$ *est une fonction décroissante par rapport à* α.

Démonstration. Comme le α-quantile $q_\alpha(X)$ est une fonction croissante par rapport à α, alors $\mathrm{VaR}_\alpha(X) = -q_\alpha(X)$ est une fonction décroissante par rapport à α. ■

Proposition 1.2.8. *Pour toute variable aléatoire* X *et pour tout seuil* $\alpha \in (0,1)$, *on a* $\mathrm{ES}_\alpha(X) \geq \mathrm{VaR}_\alpha(X)$.

Démonstration. Soit $\alpha \in (0,1)$ et X une variable aléatoire. En utilisant la proposition 1.2.7, nous obtenons

$$\begin{aligned}
\mathrm{ES}_\alpha(X) &= \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma \\
&\geq \frac{1}{\alpha}\mathrm{VaR}_\alpha(X)\int_0^\alpha \mathrm{d}\gamma \\
&= \mathrm{VaR}_\alpha(X).
\end{aligned}$$

■

Proposition 1.2.9. $\mathrm{ES}_\alpha(X)$ *est une fonction décroissante par rapport à* α.

Démonstration. Soit $0 < \alpha_1 \leq \alpha_2 < 1$ et X une variable aléatoire. Nous avons

$$\begin{aligned}
\mathrm{ES}_{\alpha_2}(X) &= \frac{1}{\alpha_2}\int_0^{\alpha_1} \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma + \frac{1}{\alpha_2}\int_{\alpha_1}^{\alpha_2} \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma \\
&= \frac{\alpha_1}{\alpha_2}\mathrm{ES}_{\alpha_1}(X) + \frac{1}{\alpha_2}\int_{\alpha_1}^{\alpha_2} \mathrm{VaR}_\gamma(X)\,\mathrm{d}\gamma \\
&\leq \frac{\alpha_1}{\alpha_2}\mathrm{ES}_{\alpha_1}(X) + \frac{\alpha_2-\alpha_1}{\alpha_2}\mathrm{VaR}_{\alpha_1}(X), \quad \text{par proposition 1.2.7,} \\
&\leq \frac{\alpha_1}{\alpha_2}\mathrm{ES}_{\alpha_1}(X) + \frac{\alpha_2-\alpha_1}{\alpha_2}\mathrm{ES}_{\alpha_1}(X), \quad \text{par proposition 1.2.8,} \\
&= \mathrm{ES}_{\alpha_1}(X).
\end{aligned}$$

Alors $\mathrm{ES}_\alpha(X)$ est bel et bien une fonction décroissante par rapport à α. ■

1.3 La notion de cohérence...

Certaines mesures de risque précédemment mentionnées ont des propriétés qui expriment davantage le comportement d'un investisseur, par exemple en ce qui a trait à la diversification des actifs. Ainsi, nous allons définir la notion de cohérence pour ensuite justifier quelles mesures nous devrions préconiser. La définition de cohérence utilisée ici est celle donnée par Delbaen (2002), qui se base sur les axiomes de Artzner *et al.* (1999).

Définition 1.3.1. Soit deux variables aléatoires X et Y. Une mesure de risque ρ est dite **cohérente** (Delbaen, 2002) si les propriétés suivantes sont respectées :

1. (Invariance par translation) Pour tout $a \in \mathbb{R}$, $\rho(X + a) = \rho(X) - a$.
2. (Sous-additivité) $\rho(X + Y) \leq \rho(X) + \rho(Y)$.
3. (Homogénéité positive) Pour tout $t \geq 0$, $\rho(tX) = t\rho(X)$.
4. (Positivité) Si $X \geq 0$, alors $\rho(X) \leq 0$.

Ces quatre propriétés reflètent des comportements qu'un investisseur pourrait souhaiter. En effet, nous pouvons les traduire en termes de mots ainsi :

1. (Invariance par translation) L'ajout d'un capital certain réduit le risque du même montant.
2. (Sous-additivité) La diversification est privilégiée par la mesure de risque.
3. (Homogénéité positive) Le risque est proportionnel à la taille du portefeuille.
4. (Positivité) Une gain certain implique une mesure de risque négative.

Maintenant que nous pouvons juger les mesures de risque sous ce principe de cohérence, vérifions chacune des propriétés des mesures énumérées précédemment, soit

la variance, la valeur-à-risque, l'espérance conditionnelle de la queue et l'*expected-shortfall*, pour déterminer lesquelles peuvent être qualifiées de cohérentes.

1.3.1 ... de la variance

En ce qui a trait à la variance, les quatre propriétés de cohérence ne sont pas respectées. En effet, nous avons que :

1. $\mathrm{Var}[X+a] = \mathrm{Var}[X], \forall a \in \mathbb{R}$.
2. Si la covariance entre X et Y est positive, alors

$$\begin{aligned}\mathrm{Var}[X+Y] &= \mathrm{Var}[X] + \mathrm{Var}[Y] + \mathrm{Cov}[X,Y] \\ &\geq \mathrm{Var}[X] + \mathrm{Var}[Y].\end{aligned}$$

3. $\mathrm{Var}[tX] = t^2 \mathrm{Var}[X], \forall t \geq 0$.
4. $\mathrm{Var}[X] \geq 0$, pour toute variable aléatoire X.

La plupart des critiques émises au sujet du problème de gestion de portefeuille moyenne-variance proviennent de la non cohérence de cette mesure de risque. C'est pourquoi plusieurs ont proposé différentes mesures de risque pour remédier à cette problématique.

1.3.2 ... de la valeur-à-risque

Pour la valeur-à-risque, seulement la propriété 2 de sous-additivité n'est pas respectée. Effectivement, vérifions les quatre propriétés :

1. Pour tout $a \in \mathbb{R}$, nous avons que

$$\begin{aligned}\mathrm{VaR}_\alpha(X+a) &= -\inf\{x \mid \mathrm{P}[X \leq x-a] \geq \alpha\} \\ &= -\inf\{y+a \mid \mathrm{P}[X \leq y] \geq \alpha\} \\ &= -(\inf\{y \mid \mathrm{P}[X \leq y] \geq \alpha\} + a) \\ &= \mathrm{VaR}_\alpha(X) - a.\end{aligned}$$

2. Montrons qu'elle n'est pas respectée avec un contre-exemple. Soit X et Y, deux variables aléatoires indépendantes et identiquement distribuées selon la distribution

$$\begin{cases}\mathrm{P}[X=0] = 0.05 \\ \mathrm{P}[X=10] = 0.95\end{cases}$$

Alors la distribution pour la variable aléatoire $X+Y$ est

$$\begin{cases}\mathrm{P}[X+Y=0] = 0.0025 \\ \mathrm{P}[X+Y=10] = 0.095 \\ \mathrm{P}[X+Y=20] = 0.9025\end{cases}$$

Avec ces variables aléatoires, nous obtenons que pour un seuil $\alpha = 0.07$, les trois valeurs-à-risque sont égales, plus précisément

$$\mathrm{VaR}_{0.07}(X) = \mathrm{VaR}_{0.07}(Y) = \mathrm{VaR}_{0.07}(X+Y) = -10.$$

Ceci implique alors que

$$\mathrm{VaR}_{0.07}(X+Y) \geq \mathrm{VaR}_{0.07}(X) + \mathrm{VaR}_{0.07}(Y).$$

3. Pour tout $t \geq 0$, nous avons que

$$\begin{aligned}\mathrm{VaR}_\alpha(tX) &= -\inf\left\{x \,\middle|\, \mathrm{P}\left[X \leq \frac{x}{t}\right] \geq \alpha\right\} \\ &= -\inf\{ty \mid \mathrm{P}[X \leq y] \geq \alpha\} \\ &= t\,\mathrm{VaR}_\alpha(X).\end{aligned}$$

4. Si $X \geq 0$, alors nous obtenons que

$$\inf\{x \mid \mathrm{P}[X \leq x] \geq \alpha\} \geq 0 \implies \mathrm{VaR}_\alpha(X) \leq 0.$$

Il s'agit d'une amélioration comparativement à la variance, mais il reste tout de même une propriété essentielle qui n'est pas satisfaite avec la valeur-à-risque.

1.3.3 ... de l'espérance conditionnelle de la queue

En ce qui a trait à l'espérance conditionnelle de la queue, comme pour la valeur-à-risque, seulement la propriété 2 de sous-additivité n'est pas respectée. En effet, faisons l'examen des quatre propriétés de la cohérence :

1. Pour tout $a \in \mathbb{R}$, nous avons que

$$\begin{aligned}\mathrm{TCE}_\alpha(X+a) &= \mathrm{E}[-(X+a) \mid X+a \leq q_\alpha(X+a)] \\ &= \mathrm{E}[-(X+a) \mid X \leq q_\alpha(X)] \\ &= \mathrm{TCE}_\alpha(X) - a.\end{aligned}$$

2. Montrons encore une fois qu'elle n'est pas respectée avec un contre-exemple. Soit X et Y, deux variables aléatoires indépendantes et identiquement distribuées selon la distribution

$$\begin{cases}\mathrm{P}[X=0] = 0.05 \\ \mathrm{P}[X=10] = 0.95\end{cases}$$

Alors la distribution pour la variable aléatoire $X+Y$ est

$$\begin{cases}\mathrm{P}[X+Y=0] = 0.0025 \\ \mathrm{P}[X+Y=10] = 0.095 \\ \mathrm{P}[X+Y=20] = 0.9025\end{cases}$$

Avec ces variables aléatoires, nous obtenons que pour un seuil $\alpha = 0.07$, les trois quantiles sont égaux, plus précisément

$$q_{0.07}(X) = q_{0.07}(Y) = q_{0.07}(X+Y) = 10.$$

On peut alors facilement calculer que $\mathrm{TCE}_{0.07}(X) = -9.5$ et $\mathrm{TCE}_{0.07}(X + Y) = -0.95/0.0975$, ce qui implique alors que

$$\mathrm{TCE}_{0.07}(X+Y) \geq \mathrm{TCE}_{0.07}(X) + \mathrm{TCE}_{0.07}(Y).$$

3. Pour tout $t \geq 0$, nous avons que

$$\begin{aligned}\mathrm{TCE}_{\alpha}(tX) &= \mathrm{E}[-tX \mid tX \leq q_{\alpha}(tX)] \\ &= \mathrm{E}[-tX \mid X \leq q_{\alpha}(X)] \\ &= t\,\mathrm{TCE}_{\alpha}(X).\end{aligned}$$

4. Si $X \geq 0$, alors nous obtenons que

$$q_{\alpha}(X) \geq 0 \implies \mathrm{TCE}_{\alpha}(X) \leq 0.$$

Encore une fois, la propriété de diversification des actifs n'est pas satisfaite. Voyons voir si la généralisation de cette mesure de risque est cohérente.

1.3.4 ... de l'*expected-shortfall*

La mesure de risque centrale au problème étudié dans ce mémoire, soit l'*expected-shortfall*, est cohérente, puisqu'elle respecte les quatre propriétés :

1. Pour tout $a \in \mathbb{R}$, nous avons que

$$\begin{aligned}\mathrm{ES}_{\alpha}(X+a) &= \frac{1}{\alpha}\int_0^{\alpha} \mathrm{VaR}_{\gamma}[X+a]\ \mathrm{d}\gamma \\ &= \frac{1}{\alpha}\int_0^{\alpha} (\mathrm{VaR}_{\gamma}[X] - a)\ \mathrm{d}\gamma \\ &= \left(\frac{1}{\alpha}\int_0^{\alpha} \mathrm{VaR}_{\gamma}[X]\ \mathrm{d}\gamma\right) - a \\ &= \mathrm{ES}_{\alpha}(X) - a.\end{aligned}$$

2. La preuve s'inspire de la démarche donnée en appendice de l'article de Acerbi et Tasche (2002b). Passons par la deuxième version de la définition

1.2.5 de l'*expected-shortfall* :

$$\begin{aligned}\mathrm{ES}_\alpha(X) &= -\frac{1}{\alpha}\left(E\left[X \cdot \mathbb{1}(X \leq q_\alpha(X))\right] + q_\alpha(X)(\alpha - P[X \leq q_\alpha(X)])\right) \\ &= -\frac{1}{\alpha}\,\mathrm{E}\left[X \cdot \mathbb{1}^*(X)\right],\end{aligned}$$

où $\mathbb{1}^*(Z) = \mathbb{1}(Z \leq q_\alpha(Z)) + \mathbb{1}(Z = q_\alpha(Z))\frac{\alpha - \mathrm{P}[Z \leq q_\alpha(Z)]}{\mathrm{P}[Z = q_\alpha(Z)]}$ pour une certaine variable aléatoire Z. Nous pouvons constater que

$$\begin{aligned}\mathrm{E}[\mathbb{1}^*(Z)] &= \mathrm{P}(Z \leq q_\alpha(Z)) + \mathrm{P}(Z = q_\alpha(Z))\frac{\alpha - \mathrm{P}[Z \leq q_\alpha(Z)]}{\mathrm{P}[Z = q_\alpha(Z)]} \\ &= \alpha.\end{aligned}$$

De plus, nous avons que $0 \leq \mathbb{1}^*(Z) \leq 1$ pour toute variable aléatoire Z. Lorsque $Z > q_\alpha(Z)$ ou $Z < q_\alpha(Z)$, le résultat est immédiat, tandis que lorsque $Z = q_\alpha(Z)$, on a

$$\begin{aligned}&\mathrm{P}[Z < q_\alpha(Z)] \leq \alpha \leq \mathrm{P}[Z \leq q_\alpha(Z)] \\ \iff\ & 0 \leq \alpha - \mathrm{P}[Z < q_\alpha(Z)] \leq \mathrm{P}[Z \leq q_\alpha(Z)] - \mathrm{P}[Z < q_\alpha(Z)] \\ \iff\ & 0 \leq \frac{\alpha - \mathrm{P}[Z < q_\alpha(Z)]}{\mathrm{P}[Z \leq q_\alpha(Z)] - \mathrm{P}[Z < q_\alpha(Z)]} \leq 1 \\ \iff\ & 0 \leq \frac{\mathrm{P}[Z = q_\alpha(Z)] + \alpha - \mathrm{P}[Z \leq q_\alpha(Z)]}{\mathrm{P}[Z = q_\alpha(Z)]} \leq 1 \\ \iff\ & 0 \leq 1 + \frac{\alpha - \mathrm{P}[Z \leq q_\alpha(Z)]}{\mathrm{P}[Z = q_\alpha(Z)]} \leq 1.\end{aligned}$$

Maintenant, nous allons montrer que l'*expected-shortfall* respecte la sous-

additivité en ajoutant 0 de façon astucieuse. Posons $W = X + Y$. Alors,

$$\begin{aligned}
&- \alpha\left(\mathrm{ES}_\alpha(W) - \mathrm{ES}_\alpha(X) - \mathrm{ES}_\alpha(Y)\right) \\
&= \mathrm{E}\left[W \cdot \mathbb{1}^*(W)\right] - \mathrm{E}\left[X \cdot \mathbb{1}^*(X)\right] - \mathrm{E}\left[Y \cdot \mathbb{1}^*(Y)\right] \\
&= \mathrm{E}\left[X \cdot (\mathbb{1}^*(W) - \mathbb{1}^*(X))\right] + \mathrm{E}\left[Y \cdot (\mathbb{1}^*(W) - \mathbb{1}^*(Y))\right] \\
&= \mathrm{E}\left[X \cdot (\mathbb{1}^*(W) - \mathbb{1}^*(X))\right] + \mathrm{E}\left[Y \cdot (\mathbb{1}^*(W) - \mathbb{1}^*(Y))\right] \\
&\quad - q_\alpha(X)\,\mathrm{E}\left[(\mathbb{1}^*(W) - \mathbb{1}^*(X))\right] - q_\alpha(Y)\,\mathrm{E}\left[(\mathbb{1}^*(W) - \mathbb{1}^*(Y))\right] \\
&= \mathrm{E}\left[(X - q_\alpha(X))\,(\mathbb{1}^*(W) - \mathbb{1}^*(X))\right] + \mathrm{E}\left[(Y - q_\alpha(Y))\,(\mathbb{1}^*(W) - \mathbb{1}^*(Y))\right] \\
&\geq 0.
\end{aligned}$$

L'inégalité provient du fait que si $X > q_\alpha(X)$, alors $\mathbb{1}^*(X) = 0$ et donc la différence d'indicatrices est nécessairement positive. Inversement, si $X < q_\alpha(X)$, alors $\mathbb{1}^*(X) = 1$ et la différence d'indicatrices est nécessairement négative. Nous avons alors que

$$(X - q_\alpha(X))\,(\mathbb{1}^*(W) - \mathbb{1}^*(X)) \geq 0.$$

Nous utilisons le même raisonnement pour la variable aléatoire Y et cela nous permet d'obtenir

$$\mathrm{ES}_\alpha(X + Y) \leq \mathrm{ES}_\alpha(X) + \mathrm{ES}_\alpha(Y).$$

3. Pour tout $t \geq 0$, nous avons que

$$\begin{aligned}
\mathrm{ES}_\alpha\left(tX\right) &= \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma\left[tX\right]\,\mathrm{d}\gamma \\
&= \frac{t}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma\left[X\right]\,\mathrm{d}\gamma \\
&= t\,\mathrm{ES}_\alpha\left(X\right).
\end{aligned}$$

4. Si $X \geq 0$, alors nous obtenons que

$$\mathrm{VaR}_\alpha\left(X\right) \leq 0 \implies \mathrm{ES}_\alpha\left(X\right) \leq 0.$$

Tel que voulu, les quatre propriétés sont respectées. C'est pourquoi nous privilégions cette mesure de risque plutôt que la valeur-à-risque ou toute autre mesure de risque non cohérente. Remarquons que ceci implique que la mesure de risque ES_α est cohérente pour toute distribution et que la mesure de risque TCE_α est cohérente dans le cas où X est continue.

Avec les propriétés de l'*expected-shortfall*, nous pouvons souligner une propriété qui nous sera d'une grande utilité dans la résolution de problèmes d'optimisation avec cette mesure de risque.

Proposition 1.3.2. ES_α *est une fonction convexe.*

Démonstration. Soit $a \in [0,1]$ et X, Y deux variables aléatoires. En utilisant les propriétés de cohérence de l'*expected-shortfall*, plus particulièrement la sous-additivité et l'homogénéité positive, nous obtenons

$$\begin{aligned}\mathrm{ES}_\alpha(aX + (1-a)Y) &\leq \mathrm{ES}_\alpha(aX) + \mathrm{ES}_\alpha((1-a)Y) \\ &= a\,\mathrm{ES}_\alpha(X) + (1-a)\,\mathrm{ES}_\alpha(Y).\end{aligned}$$

■

CHAPITRE II

PROBLÈME MOYENNE-SHORTFALL À UNE PÉRIODE

Après avoir rappelé les définitions et les propriétés des mesures de risque, nous pouvons maintenant aborder le problème moyenne-shortfall à une période. Historiquement, le problème moyenne-variance a été résolu la première fois par le chercheur Harry Markowitz (1952, 1968). Or, suite aux maintes critiques, plusieurs chercheurs ont tenté de modifier la mesure de risque à minimiser dans ce problème d'optimisation. C'est pourquoi nous analysons plus en détail le problème moyenne-shortfall, un problème résolu par Bertsimas *et al.* (2004).

2.1 Moyenne-shortfall...

De façon formelle, mettons-nous dans le contexte d'un investisseur qui désire investir sa richesse $X(t)$ dans plusieurs actifs. Supposons ainsi que l'on désire investir notre richesse au temps initial $t_0 = 0$ pour espérer obtenir une richesse X^* à la date terminale $t_1 = T$ tout en minimisant la mesure de risque ρ. Cette stratégie d'investissement sera notée $\boldsymbol{\pi}$. Le marché est composé de m différents titres risqués et d'un titre sans risque. Nous notons le vecteur aléatoire des rendements des actions risquées par $\boldsymbol{R}$ et le rendement du titre sans risque r en plus de considérer que les rendements suivent une distribution continue. La stratégie $\boldsymbol{\pi}$ est donc un vecteur composé des différents montants investis dans les m titres risqués. La figure 2.1 illustre le problème avec la notation utilisée.

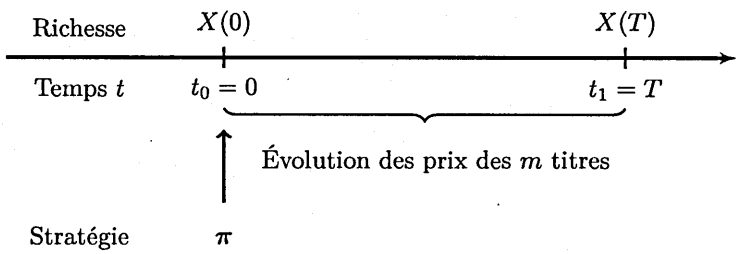


Figure 2.1: Illustration du problème moyenne-shortfall à une période

Le problème d'optimisation auquel nous faisons face en général est le suivant :

$$\begin{aligned} &\min_{\pi} \ \rho(X(T)) \\ &\text{sous } \ \mathrm{E}\left[X(T)\right] = X^*. \end{aligned}$$

Comme nous préconisons l'*expected-shortfall* comme mesure de risque, le problème moyenne-shortfall à une période est de la forme

$$\begin{aligned} &\min_{\pi} \ \mathrm{ES}_\alpha(X(T)) \\ &\text{sous } \ \mathrm{E}\left[X(T)\right] = X^*. \end{aligned}$$

Remarquons ici que nous précisons l'*expected-shortfall*, mais comme les rendements suivent une distribution continue, par la proposition 1.2.6, cela implique que nous pouvons utiliser autant les définitions 1.2.4 que 1.2.5.

Pour le problème d'optimisation, nous proposons une contrainte dite d'autofinancement, ce qui signifie que nous n'injectons pas de richesse dans le portefeuille au fil du temps. Cela se résume ainsi à demander que la richesse s'exprime comme

$$\begin{aligned} X(T) - X(0) &= \sum_{i=1}^{m} \pi_i R_i + \left(X(0) - \sum_{i=1}^{m} \pi_i\right) r \\ \iff X(T) &= X(0) + \boldsymbol{R}^\intercal \boldsymbol{\pi} + \left(X(0) - \boldsymbol{e}^\intercal \boldsymbol{\pi}\right) r, \end{aligned} \tag{2.1}$$

où $\boldsymbol{e}$ est un vecteur colonne rempli de uns. Cette contrainte explicite que nous investissons un montant π_i dans l'action i pour $i = 1, \ldots, m$, tandis que le reste de la richesse est investi dans le titre sans risque.

Nous considérons aussi un marché sans friction, c'est-à-dire que les coûts associés à toutes les transactions sont nuls. La seule préoccupation de l'investisseur est de choisir quels montants investir dans les différents titres risqués disponibles.

2.1.1 ... et son équivalence avec le moyenne-variance

Avant de s'attarder au problème, nous pouvons constater que pour certaines distributions, les problèmes moyenne-variance et moyenne-shortfall à une période sont équivalents. Effectivement, l'*expected-shortfall* et la variance de distributions particulières sont égales à une constante près. Nous allons introduire les distributions elliptiques.

Définition 2.1.1. Un vecteur aléatoire X suit une **distribution elliptique** (Kelker, 1970) de paramètres μ et Σ, respectivement la moyenne et la matrice de covariance, si et seulement si sa fonction caractéristique est de la forme

$$\mathrm{E}[e^{it^\intercal X}] = e^{it^\intercal \mu}\, \psi\left(t^\intercal \Sigma t\right),$$

pour une certaine fonction ψ. Si X admet une densité, alors sa densité est donnée par

$$f_X(\boldsymbol{x}) = \frac{c}{\sqrt{|\Sigma|}}\, g\left((\boldsymbol{x}-\mu)^\intercal \Sigma^{-1}(\boldsymbol{x}-\mu)\right),$$

où c est une constante de normalisation et g est dite une fonction génératrice de densité (*density generator*).

À titre d'exemple, une distribution normale multivariée de dimension k est une distribution elliptique. Sa constante de normalisation est $c = (2\pi)^{-k/2}$ et sa fonction génératrice de densité est $g(x) = \exp(-x/2)$. La proposition qui suit, que

nous retrouvons dans l'article de Bertsimas *et al.* (2004), illustre la correspondance entre l'*expected-shortfall* et la variance pour une distribution elliptique.

Proposition 2.1.2 (Relation entre *expected-shortfall* et variance). *Si $\boldsymbol{R}$ suit une distribution multivariée elliptique de paramètre μ et Σ, alors*

$$\mathrm{ES}_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}) = \xi(\alpha)\,\mathrm{Var}(\boldsymbol{R}^\intercal\boldsymbol{\pi})^{1/2} - \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi},$$

où $\xi(\alpha)$ est une constante spécifique à la distribution elliptique.

Démonstration. Nous avons que $\boldsymbol{R}^\intercal\boldsymbol{\pi}$ suit une distribution elliptique avec moyenne et variance respectivement de $\mu^* = \mu^\intercal\boldsymbol{\pi}$ et $\sigma^2 = \boldsymbol{\pi}^\intercal\Sigma\boldsymbol{\pi}$. En passant par la définition 1.2.4, nous obtenons donc

$$\begin{aligned}
\mathrm{ES}_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}) &= \mathrm{E}[-\boldsymbol{R}^\intercal\boldsymbol{\pi} | \boldsymbol{R}^\intercal\boldsymbol{\pi} \leq q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi})] \\
&= -\frac{c_n}{\alpha\sigma}\int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi})} x\; g\left(\left(\frac{x-\mu^*}{\sigma}\right)^2\right)\,\mathrm{d}x \\
&= -\frac{c_n}{\alpha\sigma}\int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi})} (x-\mu^*)\; g\left(\left(\frac{x-\mu^*}{\sigma}\right)^2\right)\,\mathrm{d}x - \mu^* \\
&= -\sigma\frac{c_n}{\alpha}\int_{-\infty}^{z_\alpha} y\; g(y^2)\,\mathrm{d}y - \mu^* \\
&= -\sigma\xi(\alpha) - \mu^* \\
&= \xi(\alpha)\,\mathrm{Var}(\boldsymbol{R}^\intercal\boldsymbol{\pi})^{1/2} - \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi}.
\end{aligned}$$

■

Nous pouvons ainsi en déduire le corollaire suivant sur les problèmes moyenne-shortfall et moyenne-variance :

Corollaire. *Sous les contraintes de la proposition 2.1.2 et en supposant la contrainte d'auto-financement donnée en* (2.1)

$$X(T) = X(0) + \boldsymbol{R}^\intercal\boldsymbol{\pi} + (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi})\,r,$$

les problèmes d'optimisation

$$\begin{aligned} \min_{\pi} \ & \mathrm{Var}(X(T)) \\ \text{sous } & \mathrm{E}\left[X(T)\right] = X^*. \end{aligned}$$

et

$$\begin{aligned} \min_{\pi} \ & \mathrm{ES}_\alpha(X(T)) \\ \text{sous } & \mathrm{E}\left[X(T)\right] = X^*. \end{aligned}$$

ont la même solution, c'est-à-dire qu'ils sont équivalents.

Démonstration. Par la proposition 2.1.2 et le fait que la contrainte des problèmes d'optimisation peut être exprimée comme

$$X^* = X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi} + (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi})\, r,$$

nous avons que

$$\begin{aligned} \mathrm{ES}_\alpha(X(T)) &= \mathrm{ES}_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}) - X(0) - (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi})\, r \\ &= \xi(\alpha)\,\mathrm{Var}(\boldsymbol{R}^\intercal\boldsymbol{\pi})^{1/2} - \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi} - X(0) - (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi})\, r \\ &= \xi(\alpha)\,\mathrm{Var}(\boldsymbol{R}^\intercal\boldsymbol{\pi})^{1/2} - X^* \\ &= \xi(\alpha)\,\mathrm{Var}(X(T))^{1/2} - X^*. \end{aligned}$$

■

Pour ce type de distribution bien particulière, résoudre le problème moyenne-variance à une période est équivalent à résoudre le problème moyenne-shortfall. Par contre, sous différentes hypothèses du prix des actifs risqués, les solutions peuvent potentiellement différer, ce qui fait l'objet de la prochaine section.

2.1.2 ... et sa solution analytique

Maintenant, attardons-nous à l'analyse du problème moyenne-shortfall pour des distributions continues en général. Tout d'abord, comme nous travaillerons avec des lagrangiens, nous allons expliciter la dérivée de l'*expected-shortfall*. Nous retrouvons cette proposition et une esquisse de démonstration dans l'article de Scaillet (2004).

Proposition 2.1.3 (Dérivée de l'*expected-shortfall*). *Soit $\boldsymbol{R}$ le vecteur aléatoire des rendements des titres risqués, $\boldsymbol{\pi}$ le vecteur des montants investis dans chaque titre risqué et r le rendement du titre sans risque. Si la distribution des rendements est continue, alors pour $k = 1, \ldots, m$, nous avons*

$$\frac{\partial}{\partial \pi_k} \mathrm{ES}_\alpha(X(T)) = -\,\mathrm{E}\left[R_k | \boldsymbol{R}^\intercal \boldsymbol{\pi} \leq q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})\right] + r.$$

Démonstration. Nous utilisons la définition 1.2.4. Tout d'abord, notons que

$$\mathrm{ES}_\alpha(X(T)) = \mathrm{ES}_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - X(0) - (X(0) - \boldsymbol{e}^\intercal \boldsymbol{\pi})\, r.$$

Attardons-nous sur le premier terme de cette expression. Posons $U = \sum_{j \neq k} \pi_j R_j$ et $V = R_k$, ce qui nous donne en intégrant par parties

$$\begin{aligned}
\mathrm{ES}_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) &= -\frac{1}{\alpha}\,\mathrm{E}\left[(U + \pi_k V)\ \mathbb{1}(U + \pi_k V \leq q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})\right] \\
&= -\frac{1}{\alpha} \int_{\mathbb{R}} \int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v} (u + \pi_k v) f(u, v)\, \mathrm{d}u\, \mathrm{d}v \\
&= -\frac{1}{\alpha} \left[\int_{\mathbb{R}} \int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v} u f(u, v)\, \mathrm{d}u\, \mathrm{d}v \right. \\
&\qquad \left. + \int_{\mathbb{R}} \pi_k v F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v)\, \mathrm{d}v \right] \\
&= -\frac{1}{\alpha} \left[\int_{\mathbb{R}} (q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v) F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v)\, \mathrm{d}v \right. \\
&\qquad \left. - \int_{\mathbb{R}} \int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v} F(u, v)\, \mathrm{d}u\, \mathrm{d}v + \int_{\mathbb{R}} \pi_k v F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v)\, \mathrm{d}v \right].
\end{aligned}$$

Ainsi, en prenant la dérivée par rapport à π_k, nous obtenons

$$\begin{aligned}
\frac{\partial}{\partial \pi_k} \mathrm{ES}_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) = -\frac{1}{\alpha} \Bigg[& \int_{\mathbb{R}} (q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v) f(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) \mathrm{d}v \\
& + \int_{\mathbb{R}} \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \\
& - \int_{\mathbb{R}} \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \\
& + \int_{\mathbb{R}} v F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \\
& + \int_{\mathbb{R}} \pi_k v \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) f(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \Bigg] \\
= -\frac{1}{\alpha} \Bigg[& \int_{\mathbb{R}} \int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v} v f(u, v) \, \mathrm{d}u \, \mathrm{d}v \\
& + \int_{\mathbb{R}} q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) f(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \Bigg].
\end{aligned}$$

Nous savons que

$$\begin{aligned}
0 = \frac{\partial}{\partial \pi_k} \alpha &= \frac{\partial}{\partial \pi_k} \int_{\mathbb{R}} \int_{-\infty}^{q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v} f(u, v) \, \mathrm{d}u \, \mathrm{d}v \\
&= \frac{\partial}{\partial \pi_k} \int_{\mathbb{R}} F(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \, \mathrm{d}v \\
&= \int_{\mathbb{R}} f(q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \pi_k v, v) \left(\frac{\partial q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi})}{\partial \pi_k} - v \right) \mathrm{d}v.
\end{aligned}$$

Ainsi, nous obtenons, tel que désiré,

$$\begin{aligned}
\frac{\partial}{\partial \pi_k} \mathrm{ES}_\alpha(X(T)) &= \frac{\partial}{\partial \pi_k} \mathrm{ES}_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) - \frac{\partial}{\partial \pi_k} X(0) - (X(0) - \boldsymbol{e}^\intercal \boldsymbol{\pi}) r \\
&= -\mathrm{E}\left[R_k | \boldsymbol{R}^\intercal \boldsymbol{\pi} \leq q_\alpha(\boldsymbol{R}^\intercal \boldsymbol{\pi}) \right] + r.
\end{aligned}$$

■

Maintenant, nous pouvons nous lancer dans l'analyse du problème de minimisation

qui nous intéresse, c'est-à-dire

$$\min_{\pi} \ \mathrm{ES}_\alpha(X(T))$$
$$\text{sous } \ \mathrm{E}[X(T)] = X^*.$$

Rappelons que nous imposons toujours la contrainte d'auto-financement donnée en (2.1). Le théorème qui suit et la démarche pour sa démonstration se retrouvent dans l'article de Bertsimas *et al.* (2004).

Théorème 2.1.4. *Le solution optimale $\hat{\boldsymbol{\pi}}$ au problème d'optimisation*

$$\min_{\pi} \ \mathrm{ES}_\alpha(X(T))$$
$$\text{sous } \ \mathrm{E}[X(T)] = X^*.$$

satisfait la condition

$$\mathrm{E}[R_j] - r = (X^* - X(0)(1+r))\frac{r - \mathrm{E}\left[R_j | \boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}})\right]}{\boldsymbol{e}^\intercal \hat{\boldsymbol{\pi}} r - \mathrm{E}\left[\boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}} | \boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}})\right]}.$$

pour $j = 1, \ldots, m$.

Démonstration. Le lagrangien pour ce problème est de la forme

$$\mathcal{L}(\boldsymbol{\pi}, \gamma) = \mathrm{ES}_\alpha(X(T)) - \gamma\left(X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi} + (X(0) - \boldsymbol{e}^\intercal \boldsymbol{\pi})\, r - X^*\right).$$

Ainsi, en prenant les dérivées par rapport aux deux variables d'intérêt, nous obtenons

$$\frac{\partial}{\partial \boldsymbol{\pi}} \mathcal{L}(\boldsymbol{\pi}, \gamma)\Big|_{(\hat{\boldsymbol{\pi}}, \hat{\gamma})} = 0$$
$$\iff 0 = -\mathrm{E}\left[\boldsymbol{R} \mid \boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal \hat{\boldsymbol{\pi}})\right] + \boldsymbol{e}r - \hat{\gamma}\left(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r\right) \tag{2.2}$$

et

$$\frac{\partial}{\partial \gamma} \mathcal{L}(\boldsymbol{\pi}, \gamma)\Big|_{(\hat{\boldsymbol{\pi}}, \hat{\gamma})} = 0$$
$$\iff 0 = X^* - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\hat{\boldsymbol{\pi}} + (X(0) - \boldsymbol{e}^\intercal \hat{\boldsymbol{\pi}})r). \tag{2.3}$$

Nous pouvons multiplier (2.2) par $\hat{\boldsymbol{\pi}}$ et isoler $\hat{\gamma}$:

$$\begin{aligned}\hat{\gamma} &= \frac{-\mathrm{E}\left[\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right] + \boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}}r}{\mathrm{E}[\boldsymbol{R}^\intercal]\hat{\boldsymbol{\pi}} - \boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}}r} \\ &= \frac{-\mathrm{E}\left[\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right] + \boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}}r}{X^* - X(0)(1+r)}, \quad \text{par (2.3).}\end{aligned}$$

Ainsi, en reprenant (2.3) et en utilisant le développement précédent, nous avons

$$\begin{aligned}&\hat{\gamma}(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r) = -\mathrm{E}\left[\boldsymbol{R}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right] + \boldsymbol{e}r \\ \iff &\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r = (X^* - X(0)(1+r))\frac{\boldsymbol{e}r - \mathrm{E}\left[\boldsymbol{R}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right]}{\boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}}r - \mathrm{E}\left[\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right]}.\end{aligned}$$

Ceci nous indique que pour les m actions risquées, c'est-à-dire $j = 1, \ldots, m$, la stratégie optimale $\hat{\boldsymbol{\pi}}$ satisfait la condition suivante :

$$\mathrm{E}[R_j] - r = (X^* - X(0)(1+r))\frac{r - \mathrm{E}\left[R_j|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right]}{\boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}}r - \mathrm{E}\left[\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}}|\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}} \leq q_\alpha(\boldsymbol{R}^\intercal\hat{\boldsymbol{\pi}})\right]}.$$

■

2.1.3 ... et ses propriétés

Une des propriétés du problème moyenne-shortfall, remarquée par Tasche (2000), est que ce problème peut en fait être résolu pour toute valeur de richesse terminale espérée en le résolvant pour une certaine valeur fixée X^*. En effet, soit la richesse finale espérée X^* ayant comme solution $(\boldsymbol{\pi}^*, \gamma^*)$ et posons $X(0)(1+r) \leq X^+ \leq X^*$. Soit un scalaire $\lambda \in (0,1)$ tel que $X^+ = \lambda X(0)(1+r) + (1-\lambda)X^*$. Puisque

$$q_\alpha((1-\lambda)\boldsymbol{R}^\intercal\boldsymbol{\pi}) = (1-\lambda)q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}),$$

nous avons, en utilisant les équations (2.2) et (2.3), respectivement

$$\begin{aligned}0 &= -\mathrm{E}\left[\boldsymbol{R}|\boldsymbol{R}^\intercal\boldsymbol{\pi}^* \leq q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}^*)\right] + \boldsymbol{e}r - \gamma^*(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r) \\ &= -\mathrm{E}\left[\boldsymbol{R}|\boldsymbol{R}^\intercal(1-\lambda)\boldsymbol{\pi}^* \leq q_\alpha(\boldsymbol{R}^\intercal(1-\lambda)\boldsymbol{\pi}^*)\right] + \boldsymbol{e}r - \gamma^*(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r)\end{aligned}$$

et

$$\begin{aligned}
0 &= X^* - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\hat{\boldsymbol{\pi}} + (X(0) - \boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}})r) \\
&= (1-\lambda)X^* + \lambda X(0)(1+r) \\
&\quad - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal](1-\lambda)\boldsymbol{\pi}^* + (X(0) - \boldsymbol{e}^\intercal(1-\lambda)\boldsymbol{\pi}^*)r) \\
&= X^+ - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal](1-\lambda)\boldsymbol{\pi}^* + (X(0) - \boldsymbol{e}^\intercal(1-\lambda)\boldsymbol{\pi}^*)r).
\end{aligned}$$

Nous obtenons ainsi que la richesse espérée X^+ a comme solution $((1-\lambda)\boldsymbol{\pi}^*, \gamma^*)$. De façon analogue, si $X^+ > X^*$, soit un scalaire $\lambda \in (0,1)$ tel que

$$X^* = \lambda X(0)(1+r) + (1-\lambda)X^+.$$

Nous obtenons, en utilisant les équations (2.2) et (2.3), respectivement

$$\begin{aligned}
0 &= -\mathrm{E}\left[\boldsymbol{R} | \boldsymbol{R}^\intercal\boldsymbol{\pi}^* \leq q_\alpha(\boldsymbol{R}^\intercal\boldsymbol{\pi}^*)\right] + \boldsymbol{e}r - \gamma^*(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r) \\
&= -\mathrm{E}\left[\boldsymbol{R} | \boldsymbol{R}^\intercal(1-\lambda)^{-1}\boldsymbol{\pi}^* \leq q_\alpha(\boldsymbol{R}^\intercal(1-\lambda)^{-1}\boldsymbol{\pi}^*)\right] + \boldsymbol{e}r - \gamma^*(\mathrm{E}[\boldsymbol{R}] - \boldsymbol{e}r)
\end{aligned}$$

et

$$\begin{aligned}
0 &= X^* - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\hat{\boldsymbol{\pi}} + (X(0) - \boldsymbol{e}^\intercal\hat{\boldsymbol{\pi}})r) \\
&= (1-\lambda)X^+ + \lambda X(0)(1+r) \\
&\quad - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal]\boldsymbol{\pi}^* + (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi}^*)r) \\
&= X^+ + \lambda X(0)(1+r)(1-\lambda)^{-1} \\
&\quad - (X(0)(1-\lambda)^{-1} + \mathrm{E}[\boldsymbol{R}^\intercal](1-\lambda)^{-1}\boldsymbol{\pi}^* + (X(0) - \boldsymbol{e}^\intercal\boldsymbol{\pi}^*)(1-\lambda)^{-1}r) \\
&= X^+ - (X(0) + \mathrm{E}[\boldsymbol{R}^\intercal](1-\lambda)^{-1}\boldsymbol{\pi}^* + (X(0) - \boldsymbol{e}^\intercal(1-\lambda)^{-1}\boldsymbol{\pi}^*)r).
\end{aligned}$$

La richesse espérée X^+ a cette fois-ci comme solution $((1-\lambda)^{-1}\boldsymbol{\pi}^*, \gamma^*)$. Cela signifie que le problème peut être résolu pour une seule valeur de richesse terminale espérée et nous pouvons ensuite multiplier la solution par un scalaire pour toute autre richesse X^+.

Une autre des propriétés, remarquée cette fois-ci par Bertsimas *et al.* (2004), est que nous pouvons exprimer le problème moyenne-shortfall comme un problème d'optimisation linéaire. En utilisant la dualité dans les systèmes linéaires (Goh, 2002), nous pouvons reformuler ce problème sous forme linéaire contenant $(m + N + 1)$ variables à minimiser et $(2N + 2)$ contraintes, où N est le nombre de simulations. Formuler le problème ainsi permet d'éviter d'effectuer un tri, une opération coûteuse qui est nécessaire dans le calcul d'un quantile. Cette façon de procéder pour résoudre le problème peut potentiellement être plus efficace d'un point de vue informatique. Pour le développement rigoureux de cette formulation linéaire, nous référons le lecteur à la section 5 de l'article de Bertsimas *et al.* (2004).

2.2 Validation empirique de la solution et de ces propriétés

Nous avons analysé théoriquement les développements du problème moyenne-shortfall et son équivalence avec le moyenne-variance. Maintenant, nous vérifions empiriquement ces résultats avec un programme Matlab. Nous analysons plus particulièrement les problèmes moyenne-shortfall et moyenne-variance à une période avec deux actifs risqués. Ce script Matlab peut facilement être généralisé au cas à m actifs risqués. Le code utilisé est fourni au lecteur (voir Annexe A.1).

Dans le script Matlab, nous initialisons tout d'abord les paramètres du problème d'optimisation, c'est-à-dire la richesse initiale, la richesse espérée, le seuil α de la mesure de risque, le nombre de simulations de richesse terminales noté `Nbr_sim_ratios` et le rendement du titre sans risque. Par la suite, nous générons `Nbr_sim_ratios` paires de rendements des actifs risqués. Pour permettre la comparaison entre les deux problèmes, l'utilisateur peut choisir de générer d'une distribution elliptique, soit une loi normale multivariée, ou d'une distribution elliptique, dans ce cas-ci une loi gamma multivariée (Schmeiser et Lal, 1982). Le

choix se fait en changeant la condition `elliptique`. Finalement, nous construisons `Nbr_sim_ratios` richesses terminales ainsi que la contrainte et les fonctions à minimiser à partir de ces richesses.

La première remarque est que lorsque nous utilisons la loi normale multivariée pour générer les paires de rendements des titres risqués, les montants d'investissements π qui minimisent chacun des problèmes sont identiques. En effet, ceci s'explique par la proposition 2.1.2 et son corollaire, dû à l'utilisation d'une distribution elliptique.

Ensuite, nous nous assurons que la solution du moyenne-shortfall satisfait les conditions du théorème 2.1.4. Tel qu'attendu, nous avons bel et bien que les $m = 2$ conditions sont respectées. Le script affiche les termes de chaque côté de l'égalité pour chacune des conditions.

Finalement, une autre propriété que nous pouvons vérifier avec le programme Matlab est que le problème peut se résoudre pour toute valeur de richesse terminale espérée. Tel que discuté dans la section 2.1.3, choisissons un scalaire $\lambda \in (0, 1)$ et posons une nouvelle richesse terminale espérée

$$X^+ = \lambda X(0)(1 + r) + (1 - \lambda)X^*$$

telle que $X^+ > X(0)(1 + r)$. Ainsi, la solution au problème d'optimisation avec comme contrainte $\mathrm{E}[X(T)] = X^+$ a comme solution un multiple du problème ayant comme contrainte $\mathrm{E}[X(T)] = X^*$. Plus précisément, nous devons multiplier la solution par $(1 - \lambda)$.

CHAPITRE III

PROBLÈME MOYENNE-SHORTFALL MULTIPÉRIODIQUE

Nous nous sommes intéressés au problème moyenne-shortfall à une période, ce qui procure un avantage à l'investisseur comparativement au problème moyenne-variance dû à la cohérence de la mesure de risque utilisée. Par contre, nous pouvons tenter de généraliser ce problème pour le rendre encore plus réaliste. En effet, nous pourrions supposer que sur de longues périodes d'investissements, l'individu devrait pouvoir réagir aux fluctuations du marché et adapter sa stratégie financière en conséquence. Ainsi, dans ce chapitre, nous nous concentrons sur l'analyse du problème moyenne-shortfall multipériodique.

3.1 Problème moyenne-shortfall multipériodique...

Nous allons maintenant introduire de façon générale le problème moyenne-shortfall multipériodique. De façon formelle, remettons-nous dans le contexte d'un investisseur qui désire investir sa richesse dans plusieurs actifs et qui, cette fois-ci, dispose de n temps distincts d'investissement notés $t_0, \ldots, t_{n-1}$. Nous notons la richesse au temps t par $X(t)$. Supposons ainsi que l'on désire investir notre richesse au temps initial $t_0 = 0$ pour espérer obtenir une richesse X^* à la date terminale $t_n = T$ tout en minimisant la mesure de risque ρ. Nous supposons aussi que les temps d'investissements sont équidistants, c'est-à-dire que $t_k = k\Delta t$, $k = 0, 1, \ldots, n$. Pour faciliter le problème, nous posons que le marché est composé de $m = 1$ titre risqué

et d'un titre sans risque. La stratégie d'investissement sera notée $\boldsymbol{\pi}$, où π_i est le montant investi dans l'actif risqué au temps t_{i-1}.

Nous notons le rendement de l'action risquée à l'instant t par R_t et le rendement du titre sans risque à l'instant t par r_t. Nous considérons aussi la tribu $\mathcal{H}_t$ constituant toute l'information disponible jusqu'au temps t en ce qui a trait aux prix du titre risqué et aux signaux du marché. Nous désirons donc que la stratégie π_i se base uniquement sur $\mathcal{H}_{t_{i-1}}$. La figure 3.1 illustre le problème avec la notation utilisée.

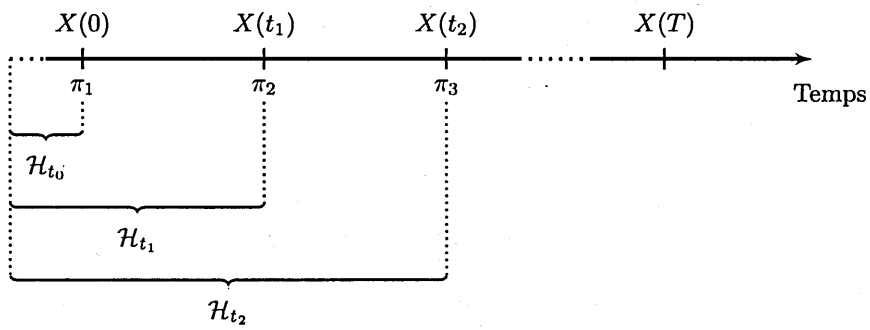


Figure 3.1: Illustration du problème moyenne-shortfall multipériodique

Encore une fois, nous proposons une contrainte d'auto-financement pour empêcher l'ajout ou le retrait de richesse dans le portefeuille au fil du temps. Cela se résume ainsi à demander que la richesse s'exprime, pour $i = 1, \ldots, n-1$, comme

$$\begin{aligned} & X(t_i) - X(t_{i-1}) = \pi_i R_{t_i} + (X(t_{i-1}) - \pi_i)\, r_{t_i} \\ \iff & X(t_i) = X(t_{i-1}) + \pi_i R_{t_i} + (X(t_{i-1}) - \pi_i)\, r_{t_i} \\ \iff & X(t_i) = X(t_{i-1})\,(1 + r_{t_i}) + \pi_i\,(R_{t_i} - r_{t_i})\,. \end{aligned} \tag{3.1}$$

Cette contrainte traduit qu'à chaque période $i = 1, \ldots, n-1$, nous investissons un montant π_i dans l'action risquée et le reste de la richesse à ce moment dans le titre sans risque.

Le problème d'optimisation auquel nous faisons face en général est le suivant :

$$\min_{\pi} \rho(X(T))$$
$$\text{sous } \mathrm{E}[X(T)] = X^*.$$

Le problème moyenne-variance multipériodique tel qu'énoncé ici, c'est-à-dire sous les mêmes hypothèses, a été résolu par Watier (2002) dans sa thèse de doctorat. Comme nous préconisons plutôt l'*expected-shortfall* comme mesure de risque, le problème moyenne-shortfall multipériodique est de la forme

$$\min_{\pi} \mathrm{ES}_{\alpha}(X(T))$$
$$\text{sous } \mathrm{E}[X(T)] = X^*.$$

Ce problème de gestion de portefeuille est un problème ouvert, ce qui nous incite davantage à explorer différentes pistes pour découvrir certaines propriétés de la solution analytique. De plus, des développements dans ce problème permettraient d'ouvrir la voie à une solution dans un problème non-résolu encore plus réaliste, soit le problème moyenne-shortfall à temps continu. Dans ce type de problème, l'investisseur pourrait modifier son portefeuille quand bon lui semble au gré des variations de prix des titres risqués, contrairement à des instants d'investissements précis.

Nous avons d'ores et déjà simplifié les hypothèses de départ en supposant l'existence d'un seul actif risqué. Nous allons renforcer davantage ces hypothèses pour notre analyse du problème en modélisant le rendement du titre risqué avec un modèle simple comportant des propriétés intéressantes.

3.1.1 ... avec un modèle binomial

Pour la modélisation de l'actif risqué, nous préconisons un modèle discret, plus précisément un modèle binomial. Tout d'abord, un modèle discret est privilégié

pour sa facilité à optimiser des variables aléatoires prenant un nombre fini de valeurs, ce qui facilite davantage le problème d'optimisation auquel nous faisons face. Avant de s'attaquer à une discrétisation de ce problème, nous avons réalisé quelques tentatives avec des modèles continus sur le problème à deux périodes, tentatives qui se sont toutefois avérées infructueuses.

Entre autres, notre première tentative a été de proposer des types de variables aléatoires simples pour les rendements des titres, suggérer des formes paramétrées de variables aléatoires pour les stratégies d'investissement et optimiser selon ces paramètres. Comme la solution exacte ne devraient pas dépendre du choix explicite des variables aléatoires utilisées, nous avons tenté de remarquer si une forme proposée menait toujours à une valeur d'*expected-shortfall* inférieure aux autres, en vain.

Nous avons aussi tenté diverses distributions pour que les problèmes moyenne-variance et moyenne-shortfall à deux périodes soient équivalents. Notre inspiration nous est provenue de la proposition 2.1.2 et de son corollaire, plus particulièrement du fait que lorsque les rendements des actifs suivent une distribution elliptique, les problèmes moyenne-variance et moyenne-shortfall à une période sont équivalents. Il suffisait ainsi de trouver une distribution pour laquelle la résolution du problème d'optimisation entraîne des valeurs optimales ne dépendant pas du seuil α. Malheureusement, aucune des distributions tentées n'a été concluante.

Ensuite, le modèle binomial peut facilement se généraliser à des modèles multinomiaux, ce qui le rend attrayant pour construire une généralisation de ce problème. Finalement, une des principales motivations nous poussant à utiliser un modèle binomial est sa possibilité d'approximer des modèles continus sous certaines conditions. En effet, dans l'article de Kim *et al.* (2016), plusieurs modèles d'arbres binomiaux qui convergent faiblement vers un mouvement brownien géo-

métrique sont donnés. Cette approximation discrète est d'un grand intérêt pour analyser le problème d'optimisation de portefeuille en utilisant une modélisation d'actif de type Black-Scholes, un modèle prisé dans la littérature. Voici une des formes d'arbres binomiaux explicitées par Kim *et al.* (2016) dans la proposition suivante.

Proposition 3.1.1. *Soit un intervalle de temps divisé en n périodes égales définies par les instants $t_k = k\Delta t$, $k = 1, \ldots, n$ et un titre risqué ayant un prix S_t à l'instant t. Posons la modélisation de l'actif risqué par un modèle binomial prenant de façon équiprobable les valeurs*

$$S_{t_k+\Delta t} = \begin{cases} S_{t_k+\Delta t,up} = S_{t_k} e^{(r-\sigma^2/2)\Delta t+\sigma\sqrt{\Delta t}} \\ S_{t_k+\Delta t,low} = S_{t_k} e^{(r-\sigma^2/2)\Delta t-\sigma\sqrt{\Delta t}} \end{cases}.$$

Le processus stochastique généré par ce modèle ***a les deux premiers moments égaux à un mouvement brownien géométrique***

$$S_t = e^{(r-\sigma^2/2)t+\sigma B(t)},$$

où $B(t)$ représente un mouvement brownien (Jarrow et Rudd, 1983).

Comme nos maintes essais avec des modèles continus se sont révélés inefficaces, analyser le problème à temps discret pourra améliorer notre compréhension du problème d'optimisation et de sa solution optimale. Nous pourrions ainsi dans un scénario idéal faire tendre le nombre de périodes vers l'infini dans notre modèle discret et obtenir une approximation plus ou moins précise d'une distribution continue. Il s'agirait d'un développement majeur en théorie mathématique du contrôle stochastique. D'ailleurs, cette piste fera l'objet de futures recherches.

Maintenant que les motivations d'analyser ce problème avec le modèle binomial sont énumérées, nous pouvons définir le problème en incluant cette modélisation

spécifique de l'actif risqué. Nous utiliserons une notation quelque peu différente avec cette modélisation pour alléger l'écriture. Voici les paramètres en question :

- Nombre de périodes : n
- Richesse initiale : X_0
- Richesse à la fin de la période i : X_i
- Stratégie d'investissement au début de la période i : π_i
- Rendement du titre sans risque lors de la période i : $r_i \equiv r$
- Rendement du titre risqué lors de la période i : R_i
- Seuil pour la mesure de risque : α

Nous introduisons aussi quelques hypothèses supplémentaires au marché. Tout d'abord, nous considérons les facteurs actualisés, plus précisément que $r \equiv 0$. Ceci implique que la richesse, toujours sous la contrainte d'auto-financement (3.1), peut être exprimée comme

$$X_n = X_{n-1} + \pi_n R_n = X_0 + \sum_{i=1}^{n} \pi_i R_i.$$

Pour l'actif risqué, nous optons pour un modèle binomial où les rendements sont indépendants entre chaque période. Nous posons

$$R_i = \begin{cases} U & \text{avec probabilité } p \\ L & \text{avec probabilité } 1-p \end{cases}, \quad \forall i = 1, \ldots, n.$$

Ici, U représente un rendement considéré élevé de l'actif («Up») et L un rendement considéré faible («Low»). Pour être réaliste dans la modélisation des rendements, nous choisissons U, L et p tels que $U > 0$, $L < 0$ et $\mathrm{E}[R_i] > 0$.

En ce qui a trait aux stratégies d'investissement, nous souhaitons qu'elles tiennent compte uniquement de l'information disponible, c'est-à-dire des rendements ayant eu lieu dans le passé. Puisque nous sommes en présence d'un modèle binomial, tous les rendements ne peuvent prendre que deux valeurs, soit U et L. Ainsi,

chaque variable aléatoire π_i peut prendre 2^{i-1} valeurs différentes pour chaque combinaison de rendements. Notons que, de cette façon, π_1 est une constante, puisque l'information disponible au départ est toujours la même. Par exemple, les variables aléatoires π_2 et π_3 sont respectivement de la forme

$$\pi_2 = \begin{cases} \pi_2^{(1)} & \text{si } R_1 = U \\ \pi_2^{(2)} & \text{si } R_1 = L \end{cases}$$

et

$$\pi_3 = \begin{cases} \pi_3^{(1)} & \text{si } R_2 = U,\ \pi_2 = \pi_2^{(1)} \\ \pi_3^{(2)} & \text{si } R_2 = L,\ \pi_2 = \pi_2^{(1)} \\ \pi_3^{(3)} & \text{si } R_2 = U,\ \pi_2 = \pi_2^{(2)} \\ \pi_3^{(4)} & \text{si } R_2 = L,\ \pi_2 = \pi_2^{(2)} \end{cases}.$$

Si nous préférons avoir une idée visuelle des investissements, la figure 3.2 montre quelle est la valeur de $\pi_i^{(j)}$ à utiliser en fonction des valeurs du titre risqué.

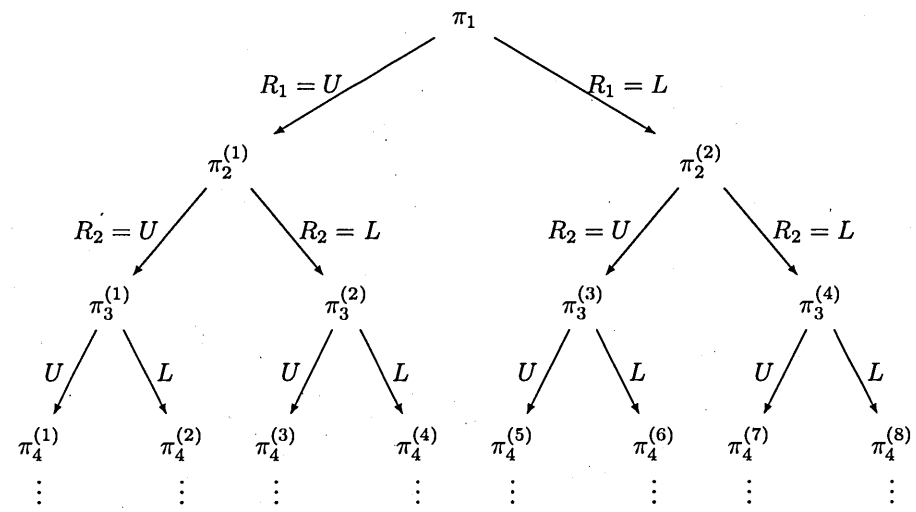


Figure 3.2: Illustration des $\pi_i^{(j)}$ correspondants aux différents scénarios de la valeur de l'actif risqué

En décrivant les stratégies de la sorte et en utilisant le modèle binomial pour

les rendements, nous pouvons exprimer la richesse terminale X_n comme une variable aléatoire prenant 2^n valeurs. Effectivement, les richesses X_1, X_2 et X_3 sont respectivement de la forme

$$X_1 = \begin{cases} X_0 + \pi_1 U & \text{avec probabilité } p \\ X_0 + \pi_1 L & \text{avec probabilité } 1-p \end{cases},$$

$$X_2 = \begin{cases} X_0 + \pi_1 U + \pi_2^{(1)} U & \text{avec probabilité } p^2 \\ X_0 + \pi_1 U + \pi_2^{(1)} L & \text{avec probabilité } p(1-p) \\ X_0 + \pi_1 L + \pi_2^{(2)} U & \text{avec probabilité } (1-p)p \\ X_0 + \pi_1 L + \pi_2^{(2)} L & \text{avec probabilité } (1-p)^2 \end{cases}$$

et

$$X_3 = \begin{cases} X_0 + \pi_1 U + \pi_2^{(1)} U + \pi_3^{(1)} U & \text{avec probabilité } p^3 \\ X_0 + \pi_1 U + \pi_2^{(1)} U + \pi_3^{(1)} L & \text{avec probabilité } p^2(1-p) \\ X_0 + \pi_1 U + \pi_2^{(1)} L + \pi_3^{(2)} U & \text{avec probabilité } p^2(1-p) \\ X_0 + \pi_1 U + \pi_2^{(1)} L + \pi_3^{(2)} L & \text{avec probabilité } (1-p)^2 p \\ X_0 + \pi_1 L + \pi_2^{(2)} U + \pi_3^{(3)} U & \text{avec probabilité } p^2(1-p) \\ X_0 + \pi_1 L + \pi_2^{(2)} U + \pi_3^{(3)} L & \text{avec probabilité } (1-p)^2 p \\ X_0 + \pi_1 L + \pi_2^{(2)} L + \pi_3^{(4)} U & \text{avec probabilité } (1-p)^2 p \\ X_0 + \pi_1 L + \pi_2^{(2)} L + \pi_3^{(4)} L & \text{avec probabilité } (1-p)^3 \end{cases}.$$

Développons maintenant l'espérance de la richesse terminale X_n, qui fait partie intégrante de la contrainte du problème de minimisation. En tenant compte des

hypothèses introduites précédemment, nous obtenons

$$\begin{aligned}
\mathrm{E}[X_n] &= \mathrm{E}\left[X_0 + \sum_{i=1}^{n} \pi_i R_i\right] \\
&= X_0 + \pi_1 \,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}[\pi_i R_i] \\
&= X_0 + \pi_1 \,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}\left[\mathrm{E}[\pi_i R_i | R_{i-1}]\right] \\
&= X_0 + \pi_1 \,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}\left[\pi_i \,\mathrm{E}[R_i | R_{i-1}]\right] \\
&= X_0 + \pi_1 \,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}[\pi_i] \,\mathrm{E}[R_i].
\end{aligned}$$

Sous la contrainte $\mathrm{E}[X_n] = X^*$, la constante π_1 peut être exprimée en termes des autres investissements aléatoires. En effet, il suffit d'isoler π_1 dans le développement précédent pour obtenir

$$\mathrm{E}[X_n] = X^* \quad \Longleftrightarrow \quad \pi_1 = \frac{X^* - X_0 - \sum_{i=2}^{n} \mathrm{E}[\pi_i] \,\mathrm{E}[R_i]}{\mathrm{E}[R_1]}. \tag{3.2}$$

Pour résoudre le problème d'optimisation initial, nous pouvons donc remplacer π_1 dans la richesse par la valeur donnée en (3.2) et résoudre le problème d'optimisation en éliminant la contrainte $\mathrm{E}[X_n] = X^*$. Voici la richesse X_2 obtenue en

remplaçant π_1 par l'expression en (3.2) :

$$X_2 = \begin{cases} X_0 + \pi_1 U + \pi_2^{(1)} U & \text{avec probabilité } p^2 \\ X_0 + \pi_1 U + \pi_2^{(1)} L & \text{avec probabilité } p(1-p) \\ X_0 + \pi_1 L + \pi_2^{(2)} U & \text{avec probabilité } (1-p)p \\ X_0 + \pi_1 L + \pi_2^{(2)} L & \text{avec probabilité } (1-p)^2 \end{cases}$$

$$= \begin{cases} X_0 + \dfrac{X^* - X_0}{\mathrm{E}[R_1]} U + \pi_2^{(1)} U(1-p) - \pi_2^{(2)} U(1-p) & \text{avec prob. } p^2 \\ X_0 + \dfrac{X^* - X_0}{\mathrm{E}[R_1]} U + \pi_2^{(1)} (L - pU) - \pi_2^{(2)} U(1-p) & \text{avec prob. } p(1-p) \\ X_0 + \dfrac{X^* - X_0}{\mathrm{E}[R_1]} L - \pi_2^{(1)} Lp + \pi_2^{(2)} (U - (1-p)L) & \text{avec prob. } (1-p)p \\ X_0 + \dfrac{X^* - X_0}{\mathrm{E}[R_1]} L - \pi_2^{(1)} Lp + \pi_2^{(2)} Lp & \text{avec prob. } (1-p)^2 \end{cases}$$

De façon analogue, pour exprimer la richesse terminale X_n dans le problème à n périodes, chacune des 2^n valeurs de cette variable aléatoire s'exprimera comme une constante suivie d'une combinaison linéaire des variables aléatoires $\pi_2, \ldots, \pi_n$. Pour les coder dans un logiciel, nous aimerions bien pouvoir générer les coefficients devant chacune des variables sous forme de matrice :

$$\begin{bmatrix} 1 & \pi_2^{(1)} & \pi_2^{(2)} & \pi_3^{(1)} & \ldots & \pi_3^{(4)} & \ldots & \pi_n^{(1)} & \ldots & \pi_n^{(2^{n-1})} \end{bmatrix}.$$

Ainsi, chaque ligne représente une des différentes valeurs possibles de la variable aléatoire qu'est la richesse terminale X_n. Nous aimerions du même coup générer les probabilités associées à chacune des valeurs possibles. Pour y parvenir, rappelons brièvement une notion d'algèbre linéaire qui intervient dans un des algorithmes.

Définition 3.1.2. Le **produit de Kronecker**, dénoté $\otimes$, de deux matrices $\boldsymbol{A}$ de

dimension $m \times n$ et $\boldsymbol{B}$ de dimension $p \times q$ est défini comme

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

La figure 3.3 illustre le pseudo-code pour créer la matrice des coefficients et le vecteur ligne des probabilités associées. Dans cet algorithme, l'initialisation des paramètres correspond à fixer les paramètres du problème d'optimisation, tels que X_0, X^*, n, p, U et L. Notons aussi que la fonction `Ones` génère une matrice remplie de uns tandis que la fonction `Eye` génère une matrice identité. Nous remarquons que l'algorithme double le nombre de richesses terminales à chaque itération de la boucle, ce qui est attendu.

3.1.2 ... où chercher le minimum ?

Maintenant que le problème est bien posé, nous aimerions bien savoir comment trouver le minimum global de ce problème d'optimisation. Avant de s'attarder directement au problème d'optimisation et à la fonction à minimiser, regardons l'ensemble d'admissibilité du problème auquel nous faisons face.

Proposition 3.1.3. *L'ensemble d'admissibilité du problème*

$$\begin{aligned} &\min_{\pi} \ \mathrm{ES}_\alpha(X(T)) \\ &\text{sous } \ \mathrm{E}\left[X(T)\right] = X^*. \end{aligned}$$

est convexe.

Démonstration. Soit $\theta \in (0,1)$ et deux éléments respectant l'ensemble d'admissibilité π et π'. Montrons que $\theta\pi + (1-\theta)\pi'$ respecte aussi la condition. En effet,

Entrées : Initialisation des paramètres ;

CoefRichesses $\leftarrow \left[X_0 + \frac{X^* - X_0}{\mathrm{E}[R_1]} U \quad X_0 + \frac{X^* - X_0}{\mathrm{E}[R_1]} L\right]^{\mathsf{T}}$;

Proba $\leftarrow 1$;

pour $j = 2$ *à* n **faire**

  Const $\leftarrow$ Kronecker(CoefRichesses, Ones(2,1)) ;

  Proba $\leftarrow$ [Proba .* p Proba .* $(1-p)$] ;

  Temp1 $\leftarrow$ RepMat(Proba, 2^j, 1) ;

  Temp2 $\leftarrow$ [U .* Ones(2^{j-1}) L .* Ones(2^{j-1})]$^{\mathsf{T}}$;

  Temp3 $\leftarrow$ Kronecker(Eye(2^{j-1}), [U L]$^{\mathsf{T}}$) ;

  CoefRichesses $\leftarrow$ [Const Temp3 - Temp1 .* Temp2] ;

fin

Proba $\leftarrow$ [Proba .* p Proba .* $(1-p)$] ;

Sorties : [CoefRichesses, Proba] ;

Figure 3.3: Algorithme pour générer les valeurs de la richesse terminale et leur probabilité associée

nous avons

$$\begin{aligned}
\mathrm{E}\left[X(T)\right] &= X_0 + (\theta\pi_1 + (1-\theta)\pi_1')\,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}[(\theta\pi_i + (1-\theta)\pi_i')]\,\mathrm{E}[R_i] \\
&= X^* + \theta\left(\pi_1\,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}[\pi_i]\,\mathrm{E}[R_i]\right) - \theta\left(\pi_1'\,\mathrm{E}[R_1] + \sum_{i=2}^{n} \mathrm{E}[\pi_i']\,\mathrm{E}[R_i]\right) \\
&= X^*.
\end{aligned}$$

■

Ainsi, par les propositions 1.3.2 et 3.1.3, nous pouvons en déduire que la solution optimale pour ce problème d'optimisation se situe sur les frontières. Plus particulièrement, tout laisse présager que les candidats potentiels à être minimum global

sont obtenus uniquement lorsque la richesse terminale est une variable aléatoire prenant deux valeurs.

Commençons par tenter de poser toutes les valeurs de la richesse terminale égales. La seule manière d'obtenir une richesse terminale constante est de poser $\pi_i = 0$ pour toutes les périodes $i = 1, 2, \ldots, n$. Or, la contrainte du problème d'optimisation ne sera pas respectée, puisque $\mathrm{E}[X_n] = X_0 \neq X^*$. Le système d'équations linéaires que nous tentons de résoudre n'est donc pas consistant.

Essayons de poser les valeurs de la richesse terminale égales en deux groupes. Nous supposons que les nombres de valeurs de la richesse terminale égales dans chacun des deux groupes sont respectivement

$$2^n - k \text{ et } k, \text{ avec } k = 1, 2, \ldots, 2^{n-1}.$$

Nous devons alors résoudre un système d'équations linéaires de

$$(2^n - k - 1) + (k - 1) = 2^n - 2$$

équations à $2^n - 2$ variables. Pour la preuve formelle de ce qu'on avance par la suite, nous devons prouver deux points essentiels :

1. La matrice du système d'équations est de plein rang et que le système d'équations est consistant.
2. Le minimum global existe sur ces partages particuliers.

Pour le premier point, il semble que la seule façon de ne pas respecter cette condition est que $U = L$, ce qui est contraire aux hypothèses de départ. Or, nous n'avons pas trouvé de preuve rigoureuse pour le cas général à n périodes bien que notre intuition soit vérifiée à l'aide d'essais numériques. En ce qui a trait au second point, l'existence est nécessaire pour s'assurer qu'une énumération exhaustive de tous ces partages nous permettra à tout coup d'obtenir le minimum global. Cette tâche est beaucoup plus ardue que la précédente et après maintes simulations, nous

nous convainquons empiriquement de sa véracité. En fait, ce dernier problème fera l'objet de futures recherches.

De façon analogue au cas à deux groupes, lorsque nous tentons de poser les valeurs de la richesse terminale égales en plus de deux groupes, nous devons résoudre un système à moins de $2^n - 2$ équations avec toujours $2^n - 2$ variables. Si le système d'équations est consistant, il aura nécessairement une infinité de solutions, ce qui signifie que le candidat ne se situe pas sur une frontière.

Chacun des candidats potentiels à être le minimum global se trouve de la façon suivante. Nous séparons les 2^n différentes valeurs de la richesse terminale en deux groupes. Ensuite, nous résolvons un système d'équations linéaires de manière à ce que toutes les richesses terminales dans un même groupe soient égales. Pour chaque candidat, nous pouvons finalement calculer la mesure de risque associée à cette richesse terminale ne prenant que deux valeurs.

Pour être certain de trouver le minimum global, il suffit de faire l'énumération de tous les partages possibles de 2^n individus en deux groupes et d'évaluer les mesures de risque correspondantes. Or, ce travail nécessite beaucoup de ressources en raison du nombre élevé de partages. En effet, le problème à n périodes comporte $2^n - 2$ variables à minimiser et le nombre de partages à évaluer est

$$\frac{1}{2}\binom{2^n}{2^{n-1}} + \sum_{k=1}^{2^{n-1}-1} \binom{2^n}{k} = 2^{2^n-1} - 1.$$

Pour donner une idée de grandeur, le nombre de candidats pour le problème à 6 périodes s'élève à environ 9.2×10^{18}. Un algorithme de recherche stochastique est donc essentiel pour éviter de devoir faire l'énumération exhaustive de la totalité des différentes richesses terminales.

3.2 Algorithme de recherche stochastique...

Dans cette section, nous allons proposer trois méthodes de recherche stochastique comportant chacune des avantages et des inconvénients. Dans l'ordre, nous présenterons deux algorithmes de recherche uniforme discrète, soit avec et sans remise, ainsi qu'une approche novatrice qui prend avantage du modèle binomial intitulée *Markovian change-when-improve*. Les avantages, désavantages, propriétés et algorithmes en pseudo-code seront analysés pour chaque approche.

Rappelons brièvement le contexte du problème d'optimisation multipériodique de portefeuille avec un modèle binomial pour la modélisation de l'actif risqué qui nous préoccupe. Pour chaque méthode proposée, ces parties seront identiques. Lorsque nous mentionnons que nous initialisons les paramètres, il est question des paramètres tels que la richesse initiale X_0, le nombre de périodes n, le seuil de la mesure de risque α et les paramètres de l'action risquée entre autres. La fonction de coût représente la fonction à minimiser, dans ce cas-ci notre mesure de risque cohérente ES_α donnée dans la définition 1.2.5. Nous avons 2^n différentes valeurs possibles de la richesse terminale X_n. Si nous sélectionnons une combinaison de ces 2^n valeurs et que nous posons les valeurs dans les deux groupes égales, nous obtenons ainsi un candidat pouvant être le minimum global.

Le vocabulaire utilisé est général de façon intentionnelle, puisque les algorithmes peuvent potentiellement être applicables à divers problèmes d'optimisation. Nous avons volontairement rédigé les pseudo-codes des algorithmes en langage presque naturel pour permettre une bonne compréhension des méthodes. Le code Matlab pour ces algorithmes se retrouve en Annexe A.3.

Pour sélectionner de façon équiprobable un certain partage, nous allons utiliser la représentation en nombre binaire. Soit 2^n variables aléatoires indépendantes et

identiquement distributées (iid.) suivant une distribution de Bernoulli. Ainsi,

$$Y_i \sim \text{Ber}\,(p = 0.5)\,, \quad i = 1, \ldots, 2^n.$$

Entrées : Initialiser le germe ;

1 Initialiser le nombre de valeurs de richesse terminale à 2^n ;

2 Créer un vecteur `Partage` de longueur 2^n rempli de zéros ;

3 **tant que** *`Partage` est rempli de zéros ou de uns* **faire**

4 | Remplacer `Partage` par 2^n tirages d'une Bernoulli($p = 0.5$) ;

5 **fin**

Sorties : Les indices tels que `Partage` vaut un ;

Figure 3.4: Algorithme de sélection uniforme d'un partage

Le partage est obtenu en prenant les indices des variables aléatoires correspondants aux succès. Il est important ici de noter que nous ne désirons pas obtenir seulement des échecs ou des succès, sans quoi nous aurons un seul groupe. Le partage doit être composé de deux groupes non vides. Chaque combinaison a nécessairement la même probabilité d'être sélectionnée et nous pouvons aussi numéroter de façon unique chaque partage en utilisant la représentation en nombres binaires. En effet, posons

$$D = 1Y_1 + 2Y_2 + \ldots + 2^{n-1}Y_n = \sum_{i=1}^{2^n} 2^{i-1}Y_i.$$

D est ainsi unique pour chaque candidat. La variable D est une variable permettant non seulement d'identifier chaque partage, mais aussi de conserver une trace des combinaisons visitées. La figure 3.4 illustre le pseudo-code d'une des nombreuses façons de sélectionner de manière uniforme un desdits partages.

3.2.1 ... uniforme discret avec remise

Commençons par décrire la méthode la plus naïve de recherche aléatoire. Plutôt que d'énumérer tous les candidats possibles, nous pouvons choisir au hasard une combinaison à chaque itération. La méthode, appelée recherche uniforme discrète avec remise, comporte tous les éléments dans son nom ; à chaque itération, nous choisissons au hasard un partage parmi tous les candidats. Si nous améliorons le minimum global trouvé par la procédure jusqu'à cette itération, nous mettons à jour cette nouvelle valeur. La figure 3.5 montre le pseudo-code de l'algorithme de recherche uniforme avec remise.

Cette première approche est attrayante par sa simplicité d'utilisation et le fait qu'elle ne nécessite pas beaucoup d'allocation de mémoire. Par contre, bien que simple d'utilisation, cet algorithme ne tient pas compte des événements précédents. Ainsi, il y a une probabilité non nulle que nous choisissions plusieurs fois le même candidat lors d'itérations différentes, ce qui n'est pas efficace.

Soit X le nombre d'itérations avant d'obtenir le minimum global. En utilisant une loi géométrique où la probabilité de succès est l'inverse multiplicatif du nombre de candidats et en supposant que le minimum global ne s'obtient qu'avec un seul partage spécifique, nous trouvons facilement que l'espérance de X est

$$\begin{aligned} \mathrm{E}[X] &= \frac{1}{1/(2^{2^n-1}-1)} \\ &= 2^{2^n-1}-1. \end{aligned}$$

Il faut donc en moyenne parcourir tous les candidats pour trouver le minimum global. L'algorithme ne présente donc pas d'avantages majeurs face à l'énumération exhaustive de tous les partages. Nous allons modifier ce présent algorithme pour améliorer son efficacité.

Entrées : Initialisation des paramètres ;

1 Initialiser le nombre d'itérations ;
2 Fixer la fonction de coût minimum à ∞ ;
3 **pour** *ii = 1 au nombre d'itérations* **faire**
4 | Générer un entier k d'une distribution uniforme discrète sur le nombre de partitions ;
5 | Sélectionner le partage numéroté k ;
6 | Calculer la fonction de coût associée au partage k ;
7 | **si** *la fonction de coût minimum est améliorée* **alors**
8 | | Mettre à jour la fonction de coût minimum ;
9 | **fin**
10 **fin**

Sorties : La fonction de coût minimum et son partage associé ;

Figure 3.5: Algorithme de recherche uniforme avec remise

3.2.2 ... uniforme discret sans remise

Reprenons l'approche uniforme discrète, mais cette fois-ci, sans remise. Ceci peut s'illustrer comme si nous conservons en mémoire les candidats déjà pigés. L'algorithme, semblable au précédent, est écrit en pseudo-code dans la figure 3.6.

Les avantages de la méthode sans remise sont nombreux comparativement à la méthode avec remise. Tout d'abord, il est impossible de sélectionner deux fois le même partage lors de différentes itérations. Ceci implique nécessairement que le nombre d'itérations maximal avant de trouver le minimum global est inférieur au nombre de candidats. De plus, la méthode sans remise est en moyenne environ deux fois plus rapide que la méthode avec remise. En effet, soit X le nombre d'itérations

avant d'obtenir le minimum global. En utilisant une loi hypergéométrique négative et en supposant encore une fois que le minimum global ne s'obtient qu'avec un seul partage spécifique, l'espérance de X est donnée par

$$\begin{aligned} \mathrm{E}[X] &= \frac{2^{2^n-1} - 1 + 1}{2} \\ &= 2^{2^n-2}. \end{aligned}$$

Il faut donc en moyenne parcourir deux fois moins de candidats que pour la méthode avec remise, ce qui est un avantage majeur considérant le nombre élevé de candidats lorsque le nombre de périodes n augmente.

Entrées : Initialisation des paramètres ;

1 Initialiser le nombre d'itérations ;
2 Générer une séquence d'entiers d'une distribution uniforme discrète sans remise sur le nombre de partitions ;
3 Fixer la fonction de coût minimum à ∞ ;
4 **pour** *ii = 1 au nombre d'itérations* **faire**
5 | Sélectionner le partage numéroté par le ii^{e} entier de la séquence ;
6 | Calculer la fonction de coût associée à ce partage ;
7 | **si** *la fonction de coût minimum est améliorée* **alors**
8 | | Mettre à jour la fonction de coût minimum ;
9 | **fin**
10 **fin**

Sorties : La fonction de coût minimum et son partage associé ;

Figure 3.6: Algorithme de recherche uniforme sans remise

Par contre, cette méthode soulève aussi quelques désavantages. Effectivement, puisque nous souhaitons conserver une trace des candidats déjà visités, il est

nécessaire de conserver en mémoire pour chaque itération le partage sélectionné. Que ce soit en les sélectionnant dès le début sans remise ou en vérifiant à chaque itération que l'on visite bel et bien un nouveau candidat, cet algorithme demande d'allouer beaucoup d'espace pour les partages déjà visités.

Le fait de garder en mémoire les candidats permet d'augmenter l'efficacité de la recherche stochastique, mais nécessite en contrepartie beaucoup d'espace mémoire. La prochaine approche présentée tente de conserver une partie des candidats visités tout en prenant avantage du modèle binomial pour augmenter considérablement son efficacité.

3.2.3 ... *Markovian change-when-improve*

Pour cette approche, nous désirons instaurer une certaine structure dans la visite des candidats pour procéder de façon plus efficace. L'inspiration pour cette méthode nous est provenue de la lecture de différentes métaheuristiques, entre autres l'algorithme de colonies de fourmis, en anglais *Ant Colony Optimization* (Dorigo *et al.*, 1996). Notre contribution décrite dans cette section est l'algorithme nommé *Markovian change-when-improve.*

Supposons qu'initialement, nous débutons avec un certain partage que l'on notera par les groupes `P1` et son complémentaire `P2`. Ainsi, si nous combinons toutes les étiquettes présentes dans ces deux groupes, nous nous retrouvons avec les 2^n valeurs de la richesse terminale.

À chaque itération, nous mettons à jour les objets `P1` et `P2` d'une façon bien particulière. En effet, nous sélectionnons aléatoirement une des étiquettes que l'on change de partage, en prenant bien soin qu'aucun groupe ne soit vide.

Maintenant, en ce qui a trait à la procédure à chaque itération, nous débutons à un candidat choisi au hasard. Ensuite, nous réalisons la transition et vérifions si le

nouveau partage donne une mesure de risque moins élevée. Si tel est le cas, nous mettons à jour le partage et nous réitérons les étapes. Dans le cas contraire, nous écartons le partage proposé. En résumé, nous changeons l'état de l'algorithme, c'est-à-dire le partage, uniquement lorsque une amélioration est détectée, d'où le terme *change-when-improve.* Le fait de modifier l'état seulement lorsque la mesure de risque s'améliore peut potentiellement permettre de se diriger vers le minimum global plus rapidement en termes d'itérations. Cet algorithme tire profit du modèle binomial, contrairement aux méthodes uniformes qui pige aléatoirement les candidats.

Nous souhaitons tout de même conserver une partie des candidats visités en mémoire. Pour ce faire, nous créons une variable mémoire qui contient les étiquettes de 1 à 2^n. La variable mémoire est réinitialisée lorsque nous changeons l'état, ce qui signifie que l'algorithme garde une trace d'au plus les 2^n derniers candidats examinés. Ceci est moins problématique que la recherche uniforme discrète sans remise expliquée précédemment qui conserve tous les états visités, soit $2^{2^n-1} - 1$.

Advenant la variable de mémoire vide, ce qui signifie que tous les candidats adjacents ont été testés et qu'aucune amélioration n'a été détectée, l'algorithme réinitialise l'état de départ et recommence la procédure. Cette méthode peut ainsi être exécutée pour un nombre très élevé d'itérations. L'algorithme décrit est présenté sous forme de pseudo-code dans la figure 3.7. Le code Matlab est aussi disponible en Annexe A.3, bien que sa lecture soit plus aride.

L'algorithme contient dans son nom *Markovian*, puisque nous pouvons reformuler celui-ci comme un processus stochastique, plus particulièrement une chaîne de Markov. Comme le nombre d'états pour décrire un tel processus est très élevé, on omet volontairement l'expression analytique de cette chaîne de Markov pour se concentrer uniquement sur la performance de l'algorithme.

Entrées : Initialisation des paramètres ;

1 Initialiser le nombre d'itérations ;

2 Sélectionner au hasard un partage (i.e. P1 et P2) ;

3 Fixer la fonction de coût minimum à la fonction de coût associée au partage initial ;

4 Créer une variable mémoire, par exemple un vecteur avec les étiquettes 1 à 2^n ;

5 **pour** *ii = 2 au nombre d'itérations* **faire**

6 Sélectionner une des étiquettes dans la mémoire et la retirer de celle-ci ;

7 Changer l'étiquette de groupe par rapport à P1 et P2;

8 Calculer la fonction de coût associée ;

9 **si** *la fonction de coût minimum est améliorée* **alors**

10 Mettre à jour la fonction de coût minimum ;

11 **fin**

12 **si** *la fonction de coût est améliorée par rapport à la fonction de coût associée à P1 et P2* **alors**

13 Mettre à jour le partage P1 et P2 avec l'étiquette modifiée;

14 Réinitialiser la variable mémoire ;

15 **sinon si** *la mémoire est vide* **alors**

16 Réinitialiser le partage (i.e. P1 et P2) et la variable mémoire ;

17 Calculer la fonction de coût associée ;

18 **fin**

19 **fin**

Sorties : La fonction de coût minimum et son partage associé ;

Figure 3.7: Algorithme *Markovian change-when-improve*

CHAPITRE IV

COMPARAISON DES ALGORITHMES STOCHASTIQUES

Au chapitre précédent, nous avons proposé trois méthodes de recherche stochastique, soit la méthode uniforme discrète avec remise, la méthode uniforme discrète sans remise et l'algorithme *Markovian change-when-improve*, abrégé par *MCWI* dans les figures et tableaux. Dans ce chapitre, nous allons uniquement comparer l'algorithme de recherche uniforme sans remise et notre algorithme proposé. Il est clair que la recherche uniforme avec remise est moins performante que la méthode sans remise, ce pourquoi nous l'ignorons dans ces études de simulation. Cela nous permettra de voir à quel rythme ces méthodes peuvent trouver le minimum global et quels algorithmes sont les plus efficaces.

4.1 Paramètres des études de simulation...

Nous utilisons la notation définie dans le chapitre III. Pour tous les graphiques et tableaux présentés, le nombre de périodes n varie pour observer le comportement de la méthode. Par contre, nous fixons les autres paramètres aux valeurs suivantes :

- Richesse initiale : $X_0 = 1$
- Richesse terminale espérée : $X^* = 6/5$
- Probabilité d'un rendement élevé du titre risqué : $p = 0.75$
- Rendement élevé du titre risqué : $U = 1$

— Rendement faible du titre risqué : $L = -2$

— Seuil pour la mesure de risque : $\alpha = 0.3$

Ce choix de paramètres est arbitraire, les comportements des méthodes étant similaires avec d'autres combinaisons de paramètres. Encore une fois, rappelons que les facteurs sont actualisés, c'est-à-dire que le rendement du titre sans risque $r \equiv 0$. Les simulations ont été réalisées avec le logiciel Matlab et la totalité des scripts nécessaires pour ces études de simulation est fournie au lecteur (voir Annexes A.2 et A.3).

4.1.1 ... pour la distribution du nombre d'itérations avant l'obtention du minimum global

Attardons-nous au nombre d'itérations avant l'obtention du minimum global. Ainsi, en laissant les algorithmes fonctionner très longtemps, c'est-à-dire avec un nombre d'itérations très élevé, nous pouvons espérer obtenir le minimum global comme solution. Nous nous intéressons alors à la distribution du nombre d'itérations avant d'obtenir le minimum global. Cela peut nous donner une idée empirique du nombre moyen ou maximal d'itérations avant de produire une solution optimale.

Pour la méthode uniforme discrète sans remise, nous connaissons la distribution théorique. En effet, nous savons que le nombre moyen d'itérations est de 2^{2^n-2} et que le nombre maximal d'itérations est de $2^{2^n-1} - 1$ en utilisant une loi hypergéométrique négative. Nous pouvons aussi aisément trouver ces valeurs empiriquement. Par contre, pour l'algorithme *Markovian change-when-improve*, nous allons les estimer empiriquement. Pour ce faire, nous laissons l'algorithme fonctionner jusqu'à ce qu'il obtienne le minimum global, calculé en amont. Nous réalisons plusieurs réplications pour ensuite estimer la distribution du nombre d'itérations avant l'obtention du minimum global. La figure 4.1 illustre ces distributions pour

les problèmes à respectivement deux, trois, quatre et cinq périodes.

Nous pouvons maintenant résumer le tout dans un tableau récapitulatif pour comparer les deux méthodes. La table 4.1 compare ces distributions que l'on visualise dans la figure 4.1. Notons toutefois que les valeurs de la méthode uniforme sont théoriques et les valeurs de l'algorithme *Markovian change-when-improve* sont estimés. Si nous excluons le problème à deux périodes, qui ne comporte que sept candidats pouvant être le minimum global, la recherche *Markovian change-when-improve* semble beaucoup plus efficace que la méthode uniforme.

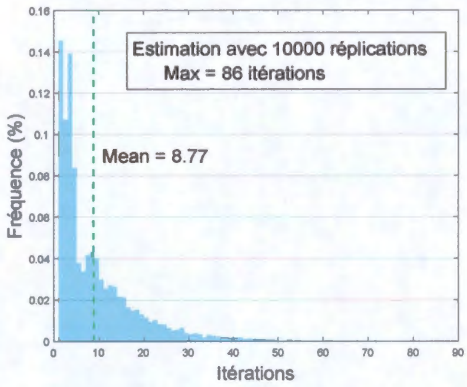
Tableau 4.1: Comparaison des distributions du nombre d'itérations pour trouver le minimum global pour les deux méthodes

	Nombre	Uniforme sans remise		*Markovian CWI*	
	de combinaisons	Moy.	Max.	Moy.	Max.
$n = 2$	7	**4**	**7**	8.77	86
$n = 3$	127	64	**127**	**25.73**	208
$n = 4$	32 767	16 384	32 767	**98.27**	**640**
$n = 5$	2.1×10^9	1.0×10^9	2.1×10^9	**295.95**	**1 584**

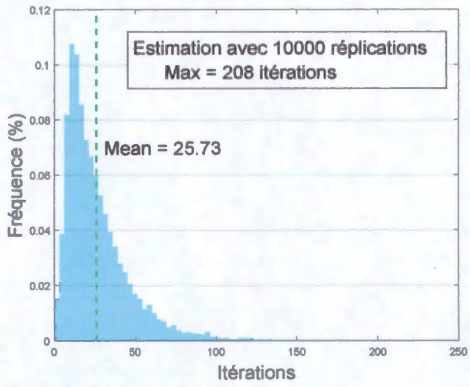
Valeurs moyennes (Moy.) et maximales (Max.) du nombre d'itérations pour trouver le minimum global où les valeurs sont exactes pour la méthode uniforme et estimées pour la méthode *MCWI*.

4.1.2 ... pour l'efficacité en un nombre fixé d'itérations

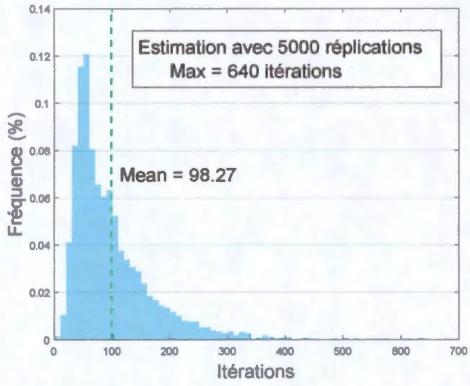
Continuons cet examen de performance en comparant cette fois-ci l'efficacité des algorithmes en un nombre fixé d'itérations. Dans ce cas, il est fort probable que la recherche ne trouve pas le minimum global. Malgré cela, nous pouvons tout de même calculer la plus petite valeur trouvée par l'algorithme à la dernière itération et les comparer entre elles. Par souci d'équité pour les deux méthodes, autant la



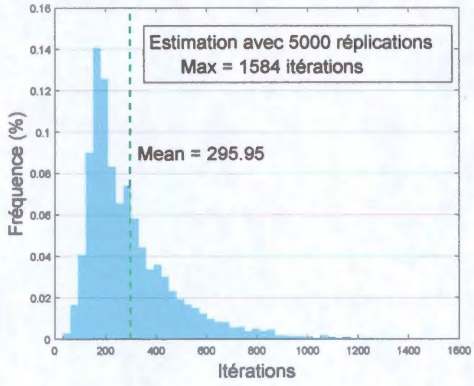
(a) 2 périodes



(b) 3 périodes



(c) 4 périodes



(d) 5 périodes

Figure 4.1: Distribution du nombre d'itérations pour trouver le minimum global avec la méthode *MCWI*

recherche uniforme discrète sans remise que le *Markovian change-when-improve*, nous débutons la recherche au même candidat. Nous pourrons ainsi évaluer de façon similaire les deux méthodes de recherche.

Commençons par le cas à $n = 2$ périodes en limitant le nombre d'itérations à 4. Comme il n'y a que sept candidats possibles dans ce problème, nous sommes portés à croire que l'algorithme de recherche uniforme sans remise est supérieur au *Markovian change-when-improve.* En effet, lorsque nous observons la figure 4.2, nous constatons que l'algorithme de recherche uniforme obtient une valeur strictement inférieure à l'autre algorithme dans environ 28% des réplications comparativement à 24% pour l'autre. L'algorithme parvient aussi à trouver le minimum global plus fréquemment que l'algorithme *Markovian change-when-improve.* Effectivement, la méthode uniforme trouve le minimum global 57% des fois comparativement à 48% pour l'autre méthode. Ceci nous montre que lorsque le nombre de candidats est si peu élevé, inclure l'aléatoire dans la recherche du minimum global nuit à l'efficacité de la recherche. Les statistiques sont calculées ici avec 10000 réplications.

Maintenant, attardons-nous au problème à $n = 3$ périodes en fixant le nombre d'itérations à 65, toujours en réalisant 10000 réplications. Cette fois-ci, l'algorithme *Markovian change-when-improve* sort davantage vainqueur du duel, comme l'illustre la figure 4.3. Tout d'abord, dans environ 47% des réplications, le *Markovian change-when-improve* a obtenu une valeur strictement inférieure à la méthode recherche uniforme et cette proportion augmente à 96% si nous considérons plutôt inférieur ou égal. De plus, l'algorithme de recherche uniforme parvient à obtenir le minimum global au terme des itérations dans 51% des cas comparativement à 95% pour l'autre méthode.

Lorsque nous nous attaquons au problème à $n = 4$ périodes, le *Markovian change-when-improve* fait l'unanimité en ce qui a trait à l'efficacité. Encore une fois,

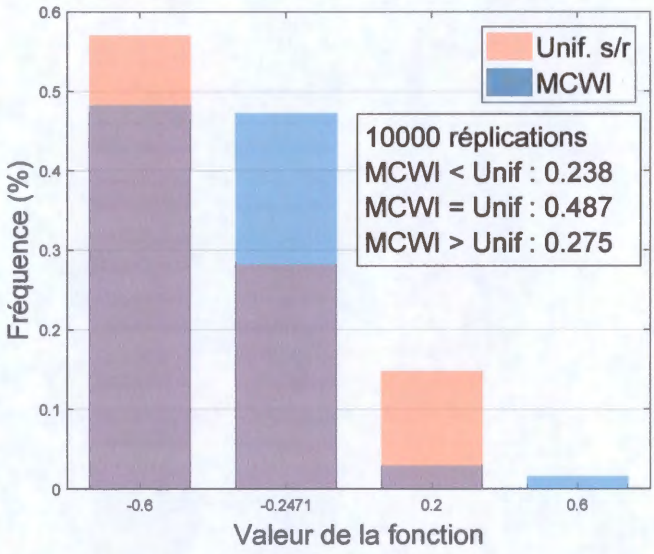


Figure 4.2: Distribution de la valeur minimale obtenu après 4 itérations pour le problème à deux périodes

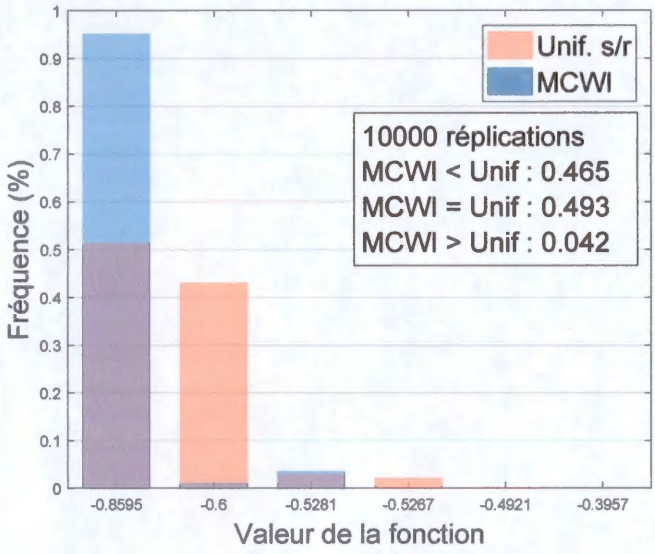


Figure 4.3: Distribution de la valeur minimale obtenu après 65 itérations pour le problème à trois périodes

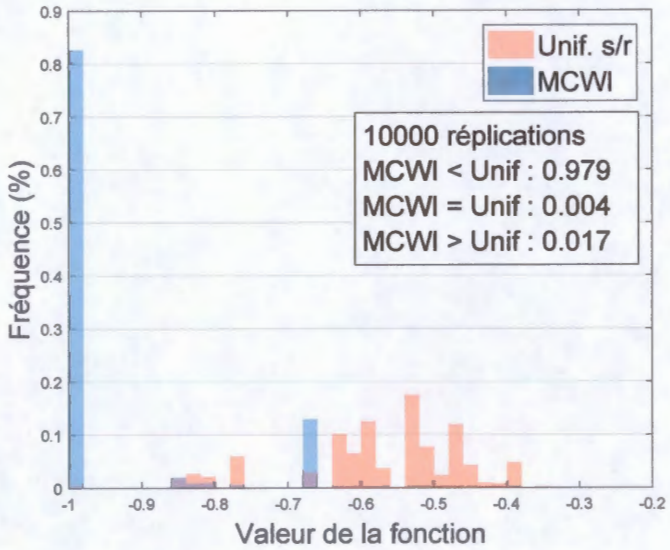


Figure 4.4: Distribution de la valeur minimale obtenu après 150 itérations pour le problème à quatre périodes

réalisons 10000 réplications, mais fixons le nombre d'itérations à 150. La figure 4.4 illustre que non seulement la valeur minimale après 150 itérations du *Markovian change-when-improve* est strictement inférieure à celle de recherche uniforme dans 98% des cas, mais le minimum global est atteint dans 82% des réplications. À titre comparatif, la méthode de recherche uniforme sans remise parvient à atteindre le minimum global dans 0.37% des réplications. L'algorithme de recherche *Markovian change-when-improve* procure un gain significatif dans l'efficacité de la recherche du minimum global.

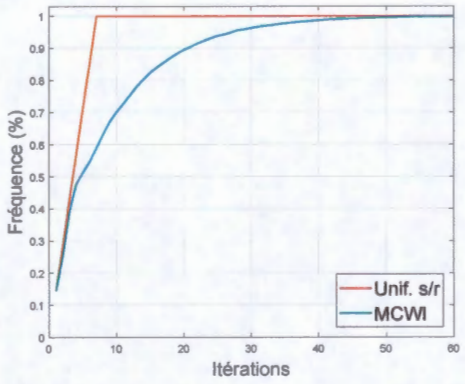
Nous pouvons aussi regarder la proportion des réplications qui trouvent le minimum global, ce qui nous donne une idée de la probabilité de trouver le minimum global en un nombre fixé d'itérations. Pour chaque réplication, nous allons alors vérifier à toutes les itérations si le minimum global est atteint ou non. Nous pouvons par la suite tracer une courbe de la progression qui nous donne une idée de la

fonction de répartition. La figure 4.5 montre cette courbe pour les deux méthodes avec 10000 réplications.

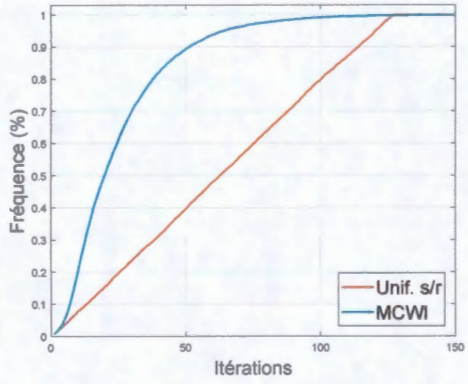
Pour le problème à deux périodes, le méthode de recherche uniforme sans remise trouve assurément le minimum global en moins de 7 itérations, ce qui lui permet d'être plus efficace. Par contre, pour les problèmes avec un nombre de candidats à visiter plus élevé, la méthode *Markovian change-when-improve* a toujours une probabilité empirique plus grande ou égale de trouver le minimum global comparativement à la recherche uniforme, comme on le remarque sur la figure 4.5.

Lors de ces études de simulation, nous avons limité le nombre de périodes au problème d'optimisation à $n = 5$ pour montrer la supériorité de notre algorithme proposé. Or, par la suite, rien ne nous empêche de trouver des solutions pour le problème à plus de cinq périodes. Par contre, l'énumération exhaustive de tous les candidats pour trouver le minimum global est très coûteuse, voire impossible à faire. Nous pouvons ainsi trouver un minimum avec la procédure, mais nous n'avons rien à quoi le comparer.

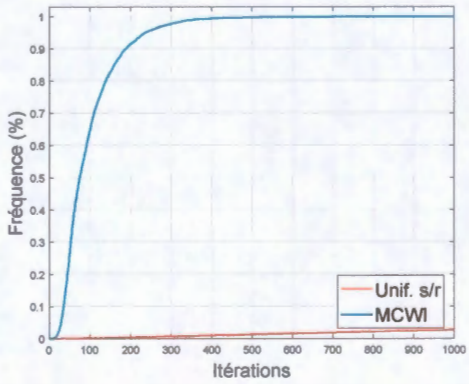
De la façon que nous avons codé les scripts Matlab, l'algorithme de recherche uniforme sans remise fonctionne pour un nombre de périodes inférieur à six, sinon l'ordinateur avec lequel nous travaillons pose une erreur d'allocation de mémoire. Au contraire, notre algorithme *Markovian change-when-improve* sur le même appareil fonctionne au minimum jusqu'à 12 périodes, tests à l'appui.



(a) 2 périodes



(b) 3 périodes



(c) 4 périodes

Figure 4.5: Proportion des 10000 réplications ayant trouvé le minimum global

CONCLUSION

Ce mémoire a permis une étude approfondie du problème moyenne-shortfall à une période et une exploration du même problème dans un contexte multipériodique.

Dans les deux premiers chapitres, nous avons fait un bref tour d'horizon des mesures de risque existantes dans la littérature avant d'arrêter notre choix sur une généralisation de l'*expected-shortfall*, une décision justifiée entre autres par sa cohérence. Nous avons ensuite analysé le problème moyenne-shortfall à une période et ces propriétés, autant théoriquement qu'empiriquement. Cette étude a permis d'établir les bases sur lesquelles construire pour généraliser ce type de problème de gestion de portefeuilles.

Les chapitres 3 et 4 illustrent l'exploration du problème moyenne-shortfall dans un contexte multipériodique. Nous avons tout d'abord explicité le problème en imposant un marché financier d'un seul titre risqué où son prix est modélisé par un modèle d'arbre binomial, la motivation principale provenant du fait que ces modèles peuvent approximer des distributions continues fréquemment utilisées dans la littérature. Nous avons finalement comparé différentes approches stochastiques pour trouver une solution optimale et il s'avère que la méthode que nous proposons, le *Markovian change-when-improve*, performe plutôt bien pour ce type particulier de problème.

Il serait intéressant dans le futur d'utiliser cette approche stochastique pour déceler des propriétés de la solution analytique du problème moyenne-shortfall multipériodique. Ceci pourrait éventuellement permettre d'analyser le problème

moyenne-shortfall à temps continu avec des modèles tels que celui de Black et Scholes.

APPENDICE A

ANNEXES

A.1 Script Matlab pour la comparaison moyenne-variance et moyenne-shortfall à une période

Les paramètres de la simulations pouvant être manuellement modifiés sont respectivement le nombre de simulations de rendements des titres risqués, le taux d'intérêt du titre sans risque, la richesse initiale, la richesse terminale espérée et le seuil de la mesure de risque. L'utilisateur peut aussi changer la valeur de la variable `elliptique` pour générer d'une distribution elliptique ou non.

```
%*****************************************************************
% Nom du programme : MoyShortfall_VS_MoyVariance
%
% Comparaison des problemes Moy-ES et Moy-Var; une periode et
% deux actions (titres risques); Choix entre une distribution
% elliptique (normale) ou non (gamma bivariee).
%
% Moyenne-Shortfall:
%       min ES_{alpha}[ X(T) ] s.t. E[ X(T) ] = X_star
% Moyenne-Variance:
%       min Var[ X(T) ] s.t. E[ X(T) ] = X_star
%
% Anthony Coache - Mars 2019
%*****************************************************************
```

```
% Ajout des sous-dossiers
addpath(genpath(pwd))
% Generateur aleatoire
rng(9662);
% Distribution elliptique? (oui = 1, non = 0)
elliptique = 1;
% Parametres de la simulation
Nbr_sim_ratios = 5000;
r = 0.1;
X_0 = 100;
X_star = 120; %lambda*X_0*(1+r) + (1-lambda)*120
alpha = 0.45;

%-------------------------------------------------------------------------%

% Temps d'execution (depart)
tic;
% Rendements des titres risques
if elliptique == 1
    % Distribution normale
    mu_S = [1  3];
    sigma_S = [1  0.75; 0.75  1];
    action = mvnrnd(mu_S, sigma_S, Nbr_sim_ratios) - 1;
else
    % Distribution gamma bivariee
    [a, b] = BivGamRND(Nbr_sim_ratios, 1, 5, 1, 3, 0.9);
    action = [a b] - 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Richesse(s) terminale(s)
```

```
X_terminale = @(w) reshape(X_0*(1 + r) + w(1) .* ...
    (action(:,1) - r) + w(2) .* (action(:,2) - r), [], 1);
% Construction de la contrainte du probleme d'optimisation
contr_prob = @(w) mean(X_terminale(w));
nonlcon_Rtilde = @(w) contrainte_Rtilde(w, contr_prob, X_star);
% Fonction a minimiser dans les deux problemes
min_shortfall = @(w) EShortfall(X_terminale(w), alpha);
min_variance = @(w) var(X_terminale(w));
% Optimisation
pi_shortfall = fmincon(min_shortfall, [0.3 0.3], ...
    [], [], [], [], [], [], nonlcon_Rtilde);
pi_variance = fmincon(min_variance, [0.3 0.3], ...
    [], [], [], [], [], [], nonlcon_Rtilde);
% Temps d'execution (fin)
tps_exec = toc;

%-------------------------------------------------------------------------%

% Affichage - Comparaison moyenne-shortfall / moyenne-variance
fprintf(1, '.....\n\n');
fprintf(1, 'Simulation et resultats\n');
fprintf(1,'Temps_d''execution (secondes) = %6.6f\n\n', ...
    tps_exec);
fprintf(1, 'Minimum dans le Moy-Shortfall = %11.5f\n', ...
    min_shortfall(pi_shortfall));
fprintf(1, 'Minimum dans le Moy-Variance = %11.5f\n\n', ...
    min_variance(pi_variance));
fprintf(1, 'pi dans le Moy-Shortfall = [ %7.5f , %7.5f]\n ', ...
    pi_shortfall(1), pi_shortfall(2));
fprintf(1, 'pi dans le Moy-Variance = [ %7.5f , %7.5f]\n\n', ...
    pi_variance(1), pi_variance(2));
fprintf(1, '.....\n');
```

```
%-------------------------------------------------------------------------%

% Affichage - Condition de la solution du moyenne-shortfall
Spi = pi_shortfall(1).*action(:,1) + pi_shortfall(2).*action(:,2);
Quant_Spi = (Spi < quantile(Spi, alpha));
% Termes des deux cotes de chaque condition
Cond1_actif1 = (X_star - X_0*(1+r)) * ...
    (r - mean(action(:,1) .* Quant_Spi)/alpha) / ...
    (r*sum(pi_shortfall) - mean(Spi .* Quant_Spi)/alpha);
Cond1_actif2 = (X_star - X_0*(1+r)) * ...
    (r - mean(action(:,2) .* Quant_Spi)/alpha) / ...
    (r*sum(pi_shortfall) - mean(Spi .* Quant_Spi)/alpha);
Cond2_actif1 = mean(action(:,1)) - r;
Cond2_actif2 = mean(action(:,2)) - r;
% Nettoyage
clear condition Spi Quant_Spi
% Affichage
fprintf(1, '.....\n\n');
fprintf(1, 'Conditions de la solution optimale\n');
fprintf(1, '2 termes de la condition - Actif 1 = %7.5f, %7.5f\n',...
    Cond1_actif1, Cond2_actif1);
fprintf(1, '2 termes de la condition - Actif 2 = %7.5f, %7.5f\n',...
    Cond1_actif2, Cond2_actif2);
fprintf(1, '.....\n\n');

%-------------------------------------------------------------------------%

% Fonction pour creer la contrainte non lineaire
function [c,ceq] = contrainte_Rtilde(w, contr_prob, X_star)
    ceq = contr_prob(w) - X_star;
    c = [];
end
% Fonction pour calculer l'expected-shortfall
```

```
function [ estimator ] = EShortfall( random , alpha )
%*****************************************************************
% Estimation de l'expected shortfall au seuil alpha sur
% le vecteur passe en parametre
%
%       Parametres d'entree:
%   random : Le vecteur des simulations
%   alpha : Le quantile recherche
%
%   Parametres de sortie:
%   estimator : La valeur de l'expected shortfall
%*****************************************************************
if sum(alpha < 0) || sum(alpha > 1)
    error('Le parametre alpha doit etre compris entre 0 et 1.')
elseif ~isvector(random)
    error('Le_parametre_random_doit etre un vecteur.')
else
    % Longueur du vecteur
    if size(random,2) == 1
        N = size(random,1);
        random = random';
    else
        N = size(random,2);
    end
    % Expected Shortfall
    estimator = (-1/alpha) * ...
        ( mean( random .* (random < quantile(random, alpha)) ) + ...
        quantile(random, alpha)*...
        (alpha - sum(random < quantile(random, alpha))/N));

end
end
```

A.2 Fonction Matlab pour calculer la mesure de risque d'un candidat

Les paramètres d'entrée de cette fonction sont respectivement les indices du partage d'un candidat, les différentes valeurs de la richesse terminale et leur probabilité associée, le nombre de périodes au problème d'optimisation et le seuil de la mesure de risque.

```
function cout = CostFunction(indice, poss, prob, n_per, alpha)
%*****************************************************************
% Nom de la fonction : CostFunction
%
% Fonction a minimiser, soit la mesure de risque expected-shortfall
%
%       Parametres d'entree:
%       indice : Vecteur des indices
%       poss : Matrice des richesses terminales en fonction des \pi
%       prob : Vecteur de probabilites associees aux
%                       richesses terminales
%       n_per : Nombre de periodes
%       alpha : Seuil de l'expected-shortfall
%
%       Parametres de sortie:
%       cout : Valeur de l'expected shortfall
%
% Anthony Coache - Mai 2019
%*****************************************************************
% Creation du systeme d'equations lineaires
% Les deux groupes sont notes AA (temp_AA) et BB (temp_BB)
num = 1:2^n_per;
temp_AA = num(ismember(1:2^(n_per), indice));
temp_BB = num(~ismember(1:2^(n_per), indice));
AA = [];
BB = [];
```

```
for ii = 1:length(temp_AA)-1
        AA = [AA ; poss(temp_AA(1), :) - poss(temp_AA(ii+1), :)];
end
for ii = 1:length(temp_BB)-1
        BB = [BB ; poss(temp_BB(1), :) - poss(temp_BB(ii+1), :)];
end
% Resolution du systeme d'equations lineaires
if isempty(BB)
        barr = linsolve(AA(:,2:(2^n_per-1)), ...
                -AA(:,1))';
else
        barr = linsolve([AA(:,2:(2^n_per-1)) ; ...
                BB(:,2:(2^n_per-1))], -[AA(:,1) ; BB(:,1)])';
end
% Obtention des valeurs de la richesse terminale
% (donne uniquement deux valeurs distinctes)
X_final = poss * [1, barr]';
% Ordonner les valeurs et probabilites de la richesse terminale
[X_finalsort, index] = sort([X_final(temp_AA(end)) ...
        X_final(temp_BB(end))]);
probsort = [sum(prob(temp_AA)) sum(prob(temp_BB))];
probsort = probsort(index);
cumprobsort = cumsum(probsort);
% Calcul de l'indice du quantile
quant = sum(cumprobsort < alpha) + 1;
E_totale = X_finalsort .* probsort;
% Calcul de la mesure de risque
cout = (-1/alpha)*( sum(E_totale(1:quant)) + ...
        X_finalsort(quant)*(alpha - cumprobsort(quant)) );
end
```

A.3 Scripts Matlab pour comparer la méthode uniforme sans remise et le *Markovian change-when-improve*

Ce script effectue une étude de simulation entre la méthode de recherche uniforme sans remise et le *Markovian change-when-improve*. Notons bien que le script doit nécessairement avoir un nombre de périodes inférieur à six (6) pour être exécuté tel quel, sinon la méthode de recherche uniforme sans remise pose une erreur d'allocation de mémoire.

Une fois le script exécuté, deux blocs dont la condition sont à `false` permettent d'obtenir des graphiques. Le premier donne des histogrammes de la distribution du nombre d'itérations pour trouver le minimum global. Le deuxième donne des histogrammes de la distribution de la valeur minimale obtenue après un nombre fixé d'itérations. Pour obtenir des graphiques identiques à ceux présents dans le mémoire, nous devons modifier certains commandes. Par exemple, la figure 4.1 nécessite de retirer l'histogramme pour la méthode uniforme.

```
%*****************************************************************
% Nom du script : Comparaison_UnifVsImproveReset
%
% Comparaison de l'algorithme de recherche uniforme sans remise et
% de l'algorithme Markovian change-when-improve.
%
% Graphiques de sortie
%       - Histogrammes de la distribution du nombre d'iterations
%               pour trouver le minimum global
%
%       - Histogrammes de la distribution de la valeur minimale
%               obtenue apres un nombre fixe d'iterations
%
% Anthony Coache - Mai 2019
%*****************************************************************
```

```
% Parametres de la simulation
rng(9662);
% Parametres du modele
x_0 = 1;
x_star = 6/5;
p = 0.75;
U = 1;
L = -2;
alpha = 0.3;
n_per = 3;
% Parametres de la simulation
Nbr_iterations = 150;
Nbr_replications = 10000;
% Esperance des rendements
E_R = U*p + L*(1-p);

%-------------------------------------------------------------------------%

% Obtention des differentes valeurs de la richesse terminale
% (le premier cas est la valeur U, le deuxieme est la valeur L...)
poss = [x_0 + (x_star - x_0)/E_R*U ; ...
    x_0 + (x_star - x_0)/E_R*L];
prob = 1;
for jj = 2:n_per
    temp = kron(poss,ones(2,1));
    prob = [prob .* p , prob .* (1-p)];
    temp1 = repmat(prob, 2^(jj), 1);
    temp2 = [U .* ones(2^(jj-1)) ; L .* ones(2^(jj-1))];
    temp3 = kron(eye(2^(jj-1)), [U; L]);
    poss = [temp , temp3 - temp1.*temp2];
end
prob = [prob .* p , prob .* (1-p)];
```

```
% Initialisation des vecteurs reponses
IterMoyen_Unif = zeros(1, Nbr_replications);
IterMoyen_Trans = zeros(1, Nbr_replications);
ValeurMin_Unif = zeros(1, Nbr_replications);
ValeurMin_Trans = zeros(1, Nbr_replications);
AtteintMin_Unif = zeros(1, Nbr_replications);
AtteintMin_Trans = zeros(1, Nbr_replications);
% Minimum global theorique dans ce probleme precis
MinGlobal
for kk = 1:Nbr_replications
% Indicateur de progression
if mod(kk,100) == 0
        fprintf('At iteration %d...\n',kk);
end

%-------------------------------------------------------------------%
% ALGORITHME UNIFORME SANS REMISE
% Initialisation des vecteurs reponses
CostMinEvolution = zeros(1, ...
        min(Nbr_iterations, 2^(2^(n_per)-1)-1));
CostEvolution = zeros(1, ...
        min(Nbr_iterations, 2^(2^(n_per)-1)-1));
% Initialisation de la suite d'uniforme sans remise
unif_partage = datasample(1:(2^(2^(n_per)-1)-1), ...
        min(Nbr_iterations, 2^(2^(n_per)-1)-1), ...
        'Replace', false);
% Partage initial
% (les deux groupes sont notes partage1 et partage2)
num = 1:2^n_per;
cond = de2bi(unif_partage(1), 2^n_per);
Partage_init = num(logical(cond));
partage1 = num(ismember(1:2^(n_per), Partage_init));
partage2 = num(~ismember(1:2^(n_per), Partage_init));
```

```
% Cout du partage initial
if(length(partage1) >= length(partage2))
        Cout_min = CostFunction(sort(partage1), poss, ...
                        prob, n_per, alpha);
else
        Cout_min = CostFunction(sort(partage2), poss, ...
                        prob, n_per, alpha);
end
% Conserver l'evolution de l'algorithme
CostEvolution(1) = Cout_min;
CostMinEvolution(1) = Cout_min;
% Nombre d'iterations
for ii = 2:min(Nbr_iterations, 2^(2^(n_per)-1)-1)
        % Determiner le nouveau partage
        cond = de2bi(unif_partage(ii), 2^n_per);
        Partage_temp = num(logical(cond));
        partage1 = num(ismember(1:2^(n_per), Partage_temp));
        partage2 = num(~ismember(1:2^(n_per), Partage_temp));
        % Calcul de la fonction a minimiser
        if(length(partage1) >= length(partage2))
                Cout_temp = CostFunction(sort(partage1), poss, ...
                                prob, n_per, alpha);
        else
                Cout_temp = CostFunction(sort(partage2), poss, ...
                                prob, n_per, alpha);
        end
        % Mettre a jour le minimum global
        if(Cout_temp < Cout_min)
                Cout_min = Cout_temp;
        end
        % Conserver l'evolution de l'algorithme
        CostEvolution(ii) = Cout_temp;
        CostMinEvolution(ii) = Cout_min;
```

```
        % Terminer la boucle si le minimum global est trouve
        if(Cout_min == MinGlobal)
                break;
        end
end
% Cout minimum atteint
ValeurMin_Unif(kk) = CostMinEvolution(ii);
% Nombre d'iterations moyenne avant d'obtenir la plus petite valeur
IterMoyen_Unif(kk) = find(...
        CostMinEvolution == CostMinEvolution(ii), 1, 'first');
% Atteinte du minimum
AtteintMin_Unif(kk) = CostMinEvolution(ii) == MinGlobal;

%-----------------------------------------------------------------------%
% ALGORITHME TRANSITION MARKOV

% Initialisation des vecteurs reponses
CostMinEvolution = zeros(1, Nbr_iterations);
CostEvolution = zeros(1, Nbr_iterations);
% Initialisation du point de depart
% Utilisation de Partage_init de l'algorithme precedent
% (les deux groupes sont notes partage1 et partage2)
partage1 = num(ismember(1:2^(n_per), Partage_init));
partage2 = num(~ismember(1:2^(n_per), Partage_init));
% Cout du partage initial
if(length(partage1) >= length(partage2))
        Cout_min = CostFunction(sort(partage1), poss, ...
                        prob, n_per, alpha);
else
        Cout_min = CostFunction(sort(partage2), poss, ...
                        prob, n_per, alpha);
end
% Conserver l'evolution de l'algorithme
```

```
CostEvolution(1) = Cout_min;
CostMinEvolution(1) = Cout_min;
% Initialisation de la memoire
memory = 1:2^(n_per);
% Initialisation du cout precedent
% (pour determiner quand changer le partage)
Cout_prec = Cout_min;
% Nombre d'iterations
for ii = 2:Nbr_iterations
        % Determiner le nouveau partage
        % Cas ou partage1 contient seulement un etiquette
        if(length(partage1) == 1)
                memory = memory(memory ~= partage1);
                ind_mod = datasample(memory, 1, 'Replace', false);
                partage1_temp = [partage1 ind_mod];
                partage2_temp = partage2(partage2 ~= ind_mod);
        % Cas ou partage2 contient seulement un etiquette
        elseif(length(partage2) == 1)
                memory = memory(memory ~= partage2);
                ind_mod = datasample(memory, 1, 'Replace', false);
                partage2_temp = [partage2 ind_mod];
                partage1_temp = partage1(partage1 ~= ind_mod);
        else
                ind_mod = datasample(memory, 1, 'Replace', false);
                if(ismember(ind_mod, partage1))
                partage2_temp = [partage2 ind_mod];
                partage1_temp = partage1(partage1 ~= ind_mod);
                else
                partage1_temp = [partage1 ind_mod];
                partage2_temp = partage2(partage2 ~= ind_mod);
                end
        end
        % Enlever l'indice de la liste en memoire
```

```
memory = memory(memory ~= ind_mod);
% Calcul de la fonction a minimiser
if(length(partage1_temp) >= length(partage2_temp))
        Cout_temp = CostFunction(sort(partage1_temp), ...
                poss, prob, n_per, alpha);
else
        Cout_temp = CostFunction(sort(partage2_temp), ...
                poss, prob, n_per, alpha);
end
% Mettre a jour le minimum global
if(Cout_temp <= Cout_min)
        Cout_min = Cout_temp;
end
% S'il y a amelioration compare a la precedente iteration
if(Cout_temp <= Cout_prec)
        % Reinitialiser la memoire
        memory = 1:2^n_per;
        Cout_prec = Cout_temp;
        % Modifier les groupes partage1 et partage2
        partage1 = partage1_temp;
        partage2 = partage2_temp;
% Si la memoire est vide
elseif(isempty(memory))
        % Reinitialiser le partage
        cond = zeros(1,2^n_per);
        while(sum(cond) == 0 || sum(cond) == 2^n_per)
                cond = binornd(1,0.5,1,2^n_per);
        end
        Partage_init = num(logical(cond));
        % Nouveau partage (groupes partage1 et partage2)
        partage1 = num(ismember(1:2^(n_per), Partage_init));
        partage2 = num(~ismember(1:2^(n_per), Partage_init));
        % Reinitialiser la memoire et le cout precedent
```

```
                memory = 1:2^n_per;
                if(length(partage1) >= length(partage2))
                        Cout_prec = CostFunction(sort(partage1), ...
                                poss, prob, n_per, alpha);
                else
                        Cout_prec = CostFunction(sort(partage2), ...
                        poss, prob, n_per, alpha);
                end
        end
        % Conserver l'evolution de l'algorithme
        CostEvolution(ii) = Cout_temp;
        CostMinEvolution(ii) = Cout_min;
        % Terminer la boucle si le minimum global est trouve
        if(Cout_min == MinGlobal)
                break;
        end
end
% Cout minimum atteint
ValeurMin_Trans(kk) = CostMinEvolution(ii);
% Nombre d'iterations moyenne avant d'obtenir la plus petite valeur
IterMoyen_Trans(kk) = find(...
        CostMinEvolution == CostMinEvolution(ii), 1, 'first');
% Atteinte du minimum
AtteintMin_Trans(kk) = CostMinEvolution(ii) == MinGlobal;
end

%-------------------------------------------------------------------------%

% Proportion de replications que le minimum global est atteint
prop_min_global_Unif = sum(AtteintMin_Unif) / Nbr_replications;
prop_min_global_Trans = sum(AtteintMin_Trans) / Nbr_replications;
% Proportion de replications que chacune des
% methodes est plus performante que l'autre
```

```
prop_min_trans_inf = ...
        sum(ValeurMin_Trans < ValeurMin_Unif) / Nbr_replications;
prop_min_trans_egal = ...
        sum(ValeurMin_Trans == ValeurMin_Unif) / Nbr_replications;
prop_min_trans_sup = ...
        sum(ValeurMin_Trans > ValeurMin_Unif) / Nbr_replications;
% Comparaison des distributions des iterations
if false
h = figure;
p1 = histogram(IterMoyen_Unif, ...
        'Normalization','probability', ...
        'FaceAlpha', 0.5, 'FaceColor', [0.9 0.3 0], ...
        'EdgeColor', 'none');
hold on;
p2 = histogram(IterMoyen_Trans, ...
        'Normalization','probability', ...
        'FaceAlpha', 0.5, 'FaceColor', [0 0.4 0.6], ...
        'EdgeColor', 'none');
p3 = line([mean(IterMoyen_Unif) mean(IterMoyen_Unif)], ...
        get(gca,'ylim'),'Color',[0.9 0.3 0], ...
        'LineStyle', '--', 'LineWidth', 1.5);
p4 = line([mean(IterMoyen_Trans) mean(IterMoyen_Trans)], ...
        get(gca,'ylim'),'Color',[0 0.4 0.6], ...
        'LineStyle', '--', 'LineWidth', 1.5);
xlabel('Iterations', 'FontSize', 18);
ylabel('Frequency (%)', 'FontSize', 18);
legend({'Unif. w/o','MCWI'}, 'FontSize', 18);
hold off;
set(gca,'XGrid','off');
set(gca,'YGrid','on');
set(gca, 'GridAlpha', 1);
set(gca, 'GridColor', [0.75 0.75 0.75]);
set(h,'Units','Inches');
```

```
pos = get(h,'Position');
set(h,'Units','Inches','position', ...
        [pos(1), pos(2)-1, pos(3)+1, pos(4)+1]);
set(h,'PaperPositionMode','Auto','PaperUnits','Inches',...
        'PaperSize',[pos(3)+1, pos(4)+1])
p5 = annotation('textbox', [0.49 0.0 0.8 0.75], ...
 'String', ...
 {[num2str(Nbr_replications,'%6.0f'), ' replications'], ...
 ['Max for Unif w/o : ',num2str(max(IterMoyen_Unif),'%6.0f')], ...
 ['Max for MCWI : ',num2str(max(IterMoyen_Trans),'%6.0f')]}, ...
 'FitBoxToText','on', 'FontSize', 18, ...
 'BackgroundColor', 'white');
print(h,'dist_IterMoyen.pdf','-dpdf','-r0')
end

% Comparaison des distributions des valeurs minimales
if false
h = figure;
%Utiliser histogram(categorical(round(ValeurMin, 4))
% et 'BarWidth',0.7 dans les options si peu de categories
p1 = histogram(ValeurMin_Unif, ...
        'FaceAlpha', 0.5, 'FaceColor', [0.9 0.3 0], ...
        'EdgeColor', 'none', ...
        'Normalization','probability');
hold on;
p2 = histogram(ValeurMin_Trans, ...
        'FaceAlpha', 0.5, 'FaceColor', [0 0.4 0.6], ...
        'EdgeColor', 'none', ...
        'Normalization','probability');
xlabel('Function_value', 'FontSize', 18);
ylabel('Frequency (%)', 'FontSize', 18);
legend({'Unif. w/o','MCWI'}, 'FontSize', 18);
hold off;
```

```
set(gca,'XGrid','off');
set(gca,'YGrid','on');
set(gca, 'GridAlpha', 1);
set(gca, 'GridColor', [0.75 0.75 0.75]);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h,'Units','Inches','position', ...
[pos(1), pos(2)-1, pos(3)+1, pos(4)+1]);
set(h,'PaperPositionMode','Auto','PaperUnits','Inches',...
        'PaperSize',[pos(3)+1, pos(4)+1]);
p3 = annotation('textbox', [0.51 0.0 0.8 0.75], ...
  'String', ...
  {[num2str(Nbr_replications,'%6.0f'), ' replications'], ...
  ['MCWI_<_Unif_:_',num2str(prop_min_trans_inf,'%4.3f')], ...
  ['MCWI_=_Unif_:_',num2str(prop_min_trans_egal,'%4.3f')],...
  ['MCWI_>_Unif_:_',num2str(prop_min_trans_sup,'%4.3f')]}, ...
  'FitBoxToText','on', 'FontSize', 18, ...
  'BackgroundColor', 'white');
print(h,'dist_ValeurMin.pdf','-dpdf','-r0')
end
```

Ce deuxième script, plus demandant en mémoire, permet d'apprécier visuellement la progression des deux algorithmes. En effet, ce script trace la proportion des réplications ayant trouvé le minimum global en fonction du nombre d'itérations. Encore une fois, certaines commandes doivent être modifiées pour obtenir des graphiques identiques à ceux présents dans le mémoire.

```
%*****************************************************************
% Nom du script : Comparaison_UnifVsImproveReset_prog
%
% Comparaison de l'algorithme de recherche uniforme sans remise et
% de l'algorithme Markovian change-when-improve.
%
```

```
% Graphiques de sortie
%       - Courbes de la proportion des replications ayant trouve
%               le minimum global en fonction du nombre d'iterations
%
%
% Anthony Coache - Avril 2019
%*****************************************************************

% Parametres de la simulation
rng(9662);
% Parametres du modele
x_0 = 1;
x_star = 6/5;
p = 0.75;
U = 1;
L = -2;
alpha = 0.3;
n_per = 2;
% Parametre de la simulation
Nbr_iterations = 1000;
Nbr_replications = 5000;
% Esperance des rendements
E_R = U*p + L*(1-p);

%-----------------------------------------------------------------%

% Obtention des differentes droites / richesses terminales
% Le premier cas est la valeur U, le deuxieme est la valeur L
poss = [x_0 + (x_star - x_0)/E_R*U ; ...
    x_0 + (x_star - x_0)/E_R*L];
prob = 1;
for jj = 2:n_per
    temp = kron(poss,ones(2,1));
```

```
    prob = [prob .* p , prob .* (1-p)];
    temp1 = repmat(prob, 2^(jj), 1);
    temp2 = [U .* ones(2^(jj-1)) ; L .* ones(2^(jj-1))];
    temp3 = kron(eye(2^(jj-1)), [U; L]);
    poss = [temp , temp3 - temp1.*temp2];
end
prob = [prob .* p , prob .* (1-p)];
% Initialisation des vecteurs reponses
ValeurMin_Unif = zeros(1, Nbr_replications);
ValeurMin_Trans = zeros(1, Nbr_replications);
AtteintMin_Unif = zeros(1, Nbr_replications);
AtteintMin_Trans = zeros(1, Nbr_replications);
% Minimum global theorique pour ce probleme precis
MinGlobal
CostMinEvolution_unif = zeros(Nbr_replications, ...
        min(Nbr_iterations, 2^(2^(n_per)-1)-1));
CostMinEvolution_trans = zeros(Nbr_replications, Nbr_iterations);
for kk = 1:Nbr_replications
% Indicateur de progression
if mod(kk,100) == 0
        fprintf('At iteration %d...\n',kk);
end

%-------------------------------------------------------------------------%
% ALGORITHME UNIFORME SANS REMISE
% Initialisation des vecteurs reponses
CostEvolution = zeros(1, min(Nbr_iterations, 2^(2^(n_per)-1)-1));
% Initialisation de la suite d'uniforme sans remise
unif_partage = datasample(1:(2^(2^(n_per)-1)-1), ...
        min(Nbr_iterations, 2^(2^(n_per)-1)-1), ...
        'Replace', false);
% Partage initial
% (les deux groupes sont notes partage1 et partage2)
```

```
num = 1:2^n_per;
cond = de2bi(unif_partage(1), 2^n_per);
Partage_init = num(logical(cond));
partage1 = num(ismember(1:2^(n_per), Partage_init));
partage2 = num(~ismember(1:2^(n_per), Partage_init));
% Cout du partage initial
if(length(partage1) >= length(partage2))
        Cout_min = CostFunction(sort(partage1), poss, ...
                prob, n_per, alpha);
else
        Cout_min = CostFunction(sort(partage2), poss, ...
                prob, n_per, alpha);
end
% Conserver l'evolution de l'algorithme
CostEvolution(1) = Cout_min;
CostMinEvolution_unif(kk,1) = Cout_min;
% Nombre d'iterations
for ii = 2:min(Nbr_iterations, 2^(2^(n_per)-1)-1)
        % Determiner le nouveau partage
        cond = de2bi(unif_partage(ii), 2^n_per);
        Partage_temp = num(logical(cond));
        partage1 = num(ismember(1:2^(n_per), Partage_temp));
        partage2 = num(~ismember(1:2^(n_per), Partage_temp));
        % Calcul de la fonction a minimiser
        if(length(partage1) >= length(partage2))
                Cout_temp = CostFunction(sort(partage1), poss, ...
                                prob, n_per, alpha);
        else
                Cout_temp = CostFunction(sort(partage2), poss, ...
                                prob, n_per, alpha);
        end
        % Mettre a jour le minimum global
        if(Cout_temp < Cout_min)
```

```
                Cout_min = Cout_temp;
        end
        % Conserver l'evolution de l'algorithme
        CostEvolution(ii) = Cout_temp;
        CostMinEvolution_unif(kk,ii) = Cout_min;
        % Terminer la boucle si le minimum global est trouve
        if(Cout_min == MinGlobal)
                break;
        end
end
% Cas special du break
if(ii ~= min(Nbr_iterations, 2^(2^(n_per)-1)-1))
        CostMinEvolution_unif(kk,ii:min(Nbr_iterations, ...
                2^(2^(n_per)-1)-1)) = CostMinEvolution_unif(kk,ii);
end
% Cout minimum atteint
ValeurMin_Unif(kk) = CostMinEvolution_unif(kk,ii);
% Atteinte du minimum
AtteintMin_Unif(kk) = CostMinEvolution_unif(kk,ii) == MinGlobal;

%-------------------------------------------------------------------------%
% ALGORITHME TRANSITION MARKOV
% Initialisation des vecteurs reponses
CostEvolution = zeros(1, Nbr_iterations);
% Initialisation du point de depart
% Utilisation de Partage_init de l'algorithme precedent
% (les deux groupes sont notes partage1 et partage2)
partage1 = num(ismember(1:2^(n_per), Partage_init));
partage2 = num(~ismember(1:2^(n_per), Partage_init));
% Cout du partage initial
if(length(partage1) >= length(partage2))
        Cout_min = CostFunction(sort(partage1), poss, ...
                        prob, n_per, alpha);
```

```
else
        Cout_min = CostFunction(sort(partage2), poss, ...
                        prob, n_per, alpha);
end
% Conserver l'evolution de l'algorithme
CostEvolution(1) = Cout_min;
CostMinEvolution_trans(kk,1) = Cout_min;
% Initialisation de la memoire
memory = 1:2^(n_per);
% Initialisation du cout precedent
% (pour determiner quand changer le partage)
Cout_prec = Cout_min;
% Nombre d'iterations
for ii = 2:Nbr_iterations
        % Determiner le nouveau partage
        % Cas ou partage1 contient seulement un etiquette
        if(length(partage1) == 1)
                memory = memory(memory ~= partage1);
                ind_mod = datasample(memory, 1, 'Replace', false);
                partage1_temp = [partage1 ind_mod];
                partage2_temp = partage2(partage2 ~= ind_mod);
        % Cas ou partage2 contient seulement un etiquette
        elseif(length(partage2) == 1)
                memory = memory(memory ~= partage2);
                ind_mod = datasample(memory, 1, 'Replace', false);
                partage2_temp = [partage2 ind_mod];
                partage1_temp = partage1(partage1 ~= ind_mod);
        else
                ind_mod = datasample(memory, 1, 'Replace', false);
                if(ismember(ind_mod, partage1))
                partage2_temp = [partage2 ind_mod];
                partage1_temp = partage1(partage1 ~= ind_mod);
                else
```

```
        partage1_temp = [partage1 ind_mod];
        partage2_temp = partage2(partage2 ~= ind_mod);
        end
end
% Enlever l'indice de la liste en memoire
memory = memory(memory ~= ind_mod);
% Calcul de la fonction a minimiser
if(length(partage1_temp) >= length(partage2_temp))
        Cout_temp = CostFunction(sort(partage1_temp), ...
                poss, prob, n_per, alpha);
else
        Cout_temp = CostFunction(sort(partage2_temp), ...
                poss, prob, n_per, alpha);
end
% Mettre a jour le minimum global
if(Cout_temp <= Cout_min)
        Cout_min = Cout_temp;
end
% S'il y a amelioration compare a la precedente iteration
if(Cout_temp <= Cout_prec)
        % Reinitialiser la memoire
        memory = 1:2^n_per;
        Cout_prec = Cout_temp;
        % Modifier les groupes partage1 et partage2
        partage1 = partage1_temp;
        partage2 = partage2_temp;
% Si la memoire est vide
elseif(isempty(memory))
        % Reinitialiser le partage
        cond = zeros(1,2^n_per);
        while(sum(cond) == 0 || sum(cond) == 2^n_per)
                cond = binornd(1,0.5,1,2^n_per);
        end
```

```
                Partage_init = num(logical(cond));
                % Nouveau partage (groupes partage1 et partage2)
                partage1 = num(ismember(1:2^(n_per), Partage_init));
                partage2 = num(~ismember(1:2^(n_per), Partage_init));
                % Reinitialiser la memoire et le cout precedent
                memory = 1:2^n_per;
                if(length(partage1) >= length(partage2))
                        Cout_prec = CostFunction(sort(partage1), ...
                                poss, prob, n_per, alpha);
                else
                        Cout_prec = CostFunction(sort(partage2), ...
                                poss, prob, n_per, alpha);
                end
        end
        % Conserver l'evolution de l'algorithme
        CostEvolution(ii) = Cout_temp;
        CostMinEvolution_trans(kk,ii) = Cout_min;
        % Terminer la boucle si le minimum global est trouve
        if(Cout_min == MinGlobal)
                break;
        end
end
% Cas special du break
if(ii ~= Nbr_iterations)
        CostMinEvolution_trans(kk,ii:Nbr_iterations) = ...
                CostMinEvolution_trans(kk,ii);
end
% Cout minimum atteint
ValeurMin_Trans(kk) = CostMinEvolution_trans(kk,ii);
% Atteinte du minimum
AtteintMin_Trans(kk) = CostMinEvolution_trans(kk,ii) == MinGlobal;
end
```

```
%-------------------------------------------------------------------------%

if false
limite = 1000;
prop_SansRemise = zeros(1,limite);
for ll = 1:min(limite, 2^(2^(n_per)-1)-1)
        prop_SansRemise(ll) = ...
        mean(CostMinEvolution_unif(:,ll) == MinGlobal);
end
%for ll = (2^(2^(n_per)-1)):limite
%    prop_SansRemise(ll) = ...
%                mean(CostMinEvolution_unif(:,end) == MinGlobal);
%end
prop_MCWI = zeros(1,limite);
for ll=1:limite
prop_MCWI(ll) = mean(CostMinEvolution_trans(:,ll)==MinGlobal);
end
h = figure;
p1 = plot(1:limite, prop_SansRemise, ...
        '-', 'LineWidth', 1.5, 'Color', [0.9 0.3 0]);
hold on;
p2 = plot(1:limite, prop_MCWI, ...
        '-','LineWidth', 1.5, 'Color', [0 0.4 0.6]);
xlim([0 limite])
ylim([0 1.03])
xlabel('Iterations', 'FontSize', 18);
ylabel('Frequency (%)', 'FontSize', 18);
legend({'Unif. w/o','MCWI'}, 'FontSize', 18, ...
        'Location', 'southeast');
hold off;
set(gca,'XGrid','on');
set(gca,'YGrid','on');
set(gca, 'GridAlpha', 0.80);
```

```
set(gca, 'GridColor', [0.75 0.75 0.75]);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h,'Units','Inches','position', ...
        [pos(1), pos(2)-1, pos(3)+1, pos(4)+1]);
set(h,'PaperPositionMode','Auto','PaperUnits','Inches',...
        'PaperSize',[pos(3)+1, pos(4)+1])
print(h,'GlobalMin_prog.pdf','-dpdf','-r0')
end
```

RÉFÉRENCES

Acerbi, C. et Tasche, D. (2002a). Expected shortfall : a natural coherent alternative to value at risk. *Economic notes*, *31*(2), 379–388.

Acerbi, C. et Tasche, D. (2002b). On the coherence of expected shortfall. *Journal of Banking & Finance*, *26*(7), 1487–1503.

Artzner, P., Delbaen, F., Eber, J.-M. et Heath, D. (1999). Coherent measures of risk. *Mathematical finance*, *9*(3), 203–228.

Balbás, A., Garrido, J. et Mayoral, S. (2009). Properties of distortion risk measures. *Methodology and Computing in Applied Probability*, *11*(3), 385.

Bertsimas, D., Lauprete, G. J. et Samarov, A. (2004). Shortfall as a risk measure : properties, optimization and applications. *Journal of Economic Dynamics and control*, *28*(7), 1353–1381.

Casella, G. et Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.

Delbaen, F. (2002). Coherent risk measures on general probability spaces. In *Advances in finance and stochastics* 1–37. Springer.

Dorigo, M., Maniezzo, V., Colorni, A. *et al.* (1996). Ant system : optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B : Cybernetics*, *26*(1), 29–41.

Föllmer, H. et Schied, A. (2008). Convex and coherent risk measures. *Encyclopedia of Quantitative Finance*, 355–363.

Goh, C. (2002). *Duality in optimization and variational inequalities*, volume 2. CRC Press.

Jarrow, R. A. et Rudd, A. T. (1983). *Option pricing.* Richard d Irwin.

Kelker, D. (1970). Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā : The Indian Journal of Statistics, Series A*, 419–430.

Kim, Y., Stoyanov, S., Rachev, S. et Fabozzi, F. (2016). Multi-purpose binomial model : Fitting all moments to the underlying geometric brownian motion. *Economics Letters*, *145*, 225–229.

Landsman, Z. M. et Valdez, E. A. (2003). Tail conditional expectations for elliptical distributions. *North American Actuarial Journal*, *7*(4), 55–71.

Markowitz, H. (1952). Portfolio selection. *The journal of finance*, *7*(1), 77–91.

Markowitz, H. (1968). *Portfolio Selection : Efficient Diversification of Investments.* Cowles Foundation Monograph : No. 16. Yale University Press. Récupéré de `https://books.google.ca/books?id=GZDyAAAAQBAJ`

Rockafellar, R. T. et Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, *26*(7), 1443–1471.

Rosenthal, J. S. (2006). *A first look at rigorous probability theory.* World Scientific Publishing Co Inc.

Scaillet, O. (2004). Nonparametric estimation and sensitivity analysis of expected shortfall. *Mathematical Finance : An International Journal of Mathematics, Statistics and Financial Economics*, *14*(1), 115–129.

Schmeiser, B. W. et Lal, R. (1982). Bivariate gamma random vectors. *Operations Research*, *30*(2), 355–374.

Tasche, D. (2000). *Risk Contributions and Performance Measurement.* Rapport technique, research paper, Zentrum Mathematik (SCA).

Watier, F. (2002). *Optimisation stochastique dans le cadre de la gestion de portefeuilles.* (Thèse de doctorat). Université de Sherbrooke.