



UNIVERSITÉ DE MONTRÉAL

DÉPARTEMENT DE L'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE
(D.I.R.O.)

IFT6521 - PROGRAMMATION DYNAMIQUE

Résolution d'un problème de Décision Séquentielle

Gestion de Portefeuille - Variante Avancée

Fait par :
André AGBANOU

Numéro Matricule :
20222919

Professeur :
Emma FREJINGER

18 avril 2023

Table des matières

1	Introduction	2
1.1	Description du problème	2
1.2	Importance du problème	2
1.3	Problème posé	2
2	Modélisation	3
2.1	Formulation numérique	3
2.2	Contraintes	3
2.3	Détails techniques pour la résolution du problème	3
3	Algorithmes	5
3.1	Méthode exacte : Chaînage arrière	5
3.2	Méthode approximative : Fitted value iteration	6
4	Résultats et Analyse	7
4.1	Petite instance	7
4.1.1	Chainage arrière	7
4.1.2	Méthode approximative	8
4.2	Moyenne instance	11
4.2.1	Chainage arrière	11
4.2.2	Méthode approximative	12
4.3	Grande Instance	13
4.3.1	Chainage arrière	14
4.3.2	Méthode approximative	14
5	Conclusion	16
5.1	Faits saillants	16
5.2	Discussion	16
	Bibliographie	17

1 Introduction

1.1 Description du problème

Le problème de décision séquentielle stochastique que nous avons choisi concerne la **gestion d'un portefeuille** d'investissement dans un marché financier qui comporte **un actif sans risque et des actifs risqués**. L'objectif ici consiste à maximiser le rendement du portefeuille tout en minimisant le risque associé à ces investissements. Cela implique de **prendre des décisions séquentielles dans le temps en ce qui concerne la répartition des investissements entre l'actif sans risque et les actifs risqués**, ainsi que la **réévaluation périodique de la répartition des investissements en fonction de l'évolution du marché**.

De prime abord, le marché financier est modélisé à l'aide de **processus stochastiques** pour représenter les fluctuations aléatoires des actifs risqués, ainsi que pour représenter les événements imprévus qui peuvent affecter les performances du marché. Les performances passées et futures des différents actifs sont simulées à l'aide de ces processus stochastiques.

Par ailleurs, le **risque associé** à chaque actif est mesuré en termes de **volatilité**, qui **représente la variabilité de la performance de l'actif**. Les choix de décision doivent donc **prendre en compte le compromis entre le risque et le rendement potentiel associé à chaque actif**. Plus précisément, les décisions de répartition des investissements doivent équilibrer le rendement potentiel de chaque actif risqué avec son risque de volatilité, tandis que la décision d'investir dans **l'actif sans risque ne comporte pas de risque de volatilité mais offre un rendement moins élevé**.

Aussi, le processus de prise de décision est influencé par l'incertitude entourant l'évolution future du marché. **Les choix de décision doivent donc être adaptatifs et flexibles pour s'adapter aux fluctuations du marché et aux changements de circonstances imprévus**. Les décisions doivent également **prendre en compte les contraintes telles que les limites de perte maximale ou les limites de concentration sur un actif particulier qui peuvent être imposées**.

Ainsi, d'après les notions étudiées en classe, on peut s'assurer que la résolution de ce problème de décision séquentielle stochastique de gestion de portefeuille d'investissement peut être effectuée à l'aide de la programmation dynamique stochastique. Cette technique nous permettra de **déterminer la meilleure stratégie d'investissement pour maximiser le rendement tout en minimisant le risque, en prenant en compte les différents éléments du problème tels que l'incertitude, la volatilité et les contraintes imposées**.

1.2 Importance du problème

Évidemment, **la raison principale qui nous motive à investir en bourse n'est ni plus ni moins que pour gagner de l'argent**. En investissant correctement, on se donne l'opportunité de faire fructifier notre capital, c'est-à-dire tout simplement de **faire travailler notre argent pour nous**.

De ce fait, le problème de décision séquentielle stochastique de gestion de portefeuille d'investissement est d'une grande importance pour les investisseurs, les gestionnaires de portefeuille et les analystes financiers.

La résolution de ce problème est importante pour les investisseurs car elle permet de maximiser le rendement potentiel de leurs investissements tout en minimisant les risques associés. Cela peut avoir un impact significatif sur leur capacité à atteindre leurs objectifs financiers à long terme, tels que la planification de la retraite ou la création d'un patrimoine.

Pour les gestionnaires de portefeuille, la résolution de ce problème est cruciale pour assurer la performance et la stabilité du portefeuille qu'ils gèrent. Les décisions de répartition des investissements doivent être prises en fonction de la stratégie d'investissement globale et des objectifs de l'investisseur, tout en prenant en compte les risques et les incertitudes associés.

Enfin, pour les analystes financiers, la résolution de ce problème est importante pour évaluer la performance des fonds d'investissement et des gestionnaires de portefeuille. En utilisant des modèles mathématiques sophistiqués pour simuler le comportement des investissements, les analystes peuvent mieux comprendre les forces et les faiblesses des différents fonds et gestionnaires de portefeuille et fournir des recommandations pour améliorer leur performance.

En somme, la résolution de ce problème de décision séquentielle stochastique de gestion de portefeuille d'investissement est d'une importance capitale pour les investisseurs individuels, les gestionnaires de portefeuille et les analystes financiers, car elle permet de prendre des décisions éclairées pour maximiser les rendements et minimiser les risques associés aux investissements.

1.3 Problème posé

Dans ce projet de gestion de portefeuille, nous disposons d'un capital dont nous souhaitons fructifier en allouant de manière réfléchie à un actif sans risque et d actifs risqués, en utilisant une approche de pro-

grammation dynamique stochastique. Nous supposons que le vecteur des actifs risqués suit un Mouvement Brownien Multivarié (MBM) avec une corrélation entre les différents actifs.

L'horizon de placement est fini et l'objectif est de maximiser la richesse finale du portefeuille W_T en prenant des décisions séquentielles à chaque étape t en fonction de l'état actuel du marché.

2 Modélisation

2.1 Formulation numérique

Soit un portefeuille constitué d'un actif sans risque et de $d = 3$ actifs risqués. L'horizon de placement est de cinq (05) ans soit 60 mois. Soit donc $t = 1, 2, \dots, T = 60$. À chaque mois t , nous avons un état du marché caractérisé par un vecteur de prix $S_t = (S_t^{(0)}, S_t^{(1)}, S_t^{(2)}, S_t^{(3)})$ où $S_t^{(0)}$ est le prix de l'actif et $S_t^{(i)}, i = 1, 2, 3$ est le prix de l'actif i . Le vecteur de prix des actifs risqués suit un mouvement brownien géométrique multivarié (MBGM) avec une corrélation entre les différents actifs risqués. Les variances de la matrice de covariance de ce vecteur illustrent la volatilité (les possibles fluctuations) du prix de chaque actif.

La simulation du MBGM nous permettra de simuler les prix des actifs risqués à chaque étape t afin de maximiser la richesse finale du portefeuille W_T en prenant des décisions séquentielles en fonction de l'état actuel du marché et des risques potentielles d'investissement.

2.2 Contraintes

En voulant résoudre ce problème, nous allons nous imposer des contraintes réalistes sur le problème de maximisation :

- Soit le vecteur des allocations $\alpha_t = (\alpha_t^0, \alpha_t^1, \alpha_t^2, \alpha_t^3)$ à l'étape t . La somme des allocations de fonds aux actifs à chaque étape t doit être égale à 1 :

$$\sum_{i=0}^3 \alpha_t^i = 1. \quad (1)$$

- À chaque étape t , l'objectif est de s'assurer que le risque d'investissement R_t soit inférieur à une valeur seuil R_0 . Cette dernière sera appelée **la tolérance au risque** :

$$R_t \leq R_0 \quad (2)$$

- L'objectif est d'atteindre une richesse finale minimale W_T à la fin de la période de 5 ans. Ceci consistera donc à maximiser son utilité en prenant soin de minimiser les risques d'investissement associés à chaque étape.

2.3 Détails techniques pour la résolution du problème

De ce problème, les éléments nous servant à sa résolution sont :

- **Les étapes** : Les étapes de notre problème sont représentées par les mois $t = 1, 2, \dots, t + N = T = 60$ avec $N = 60$. Chaque étape représente un mois différent.
- **Les états** : Les états du marché à chaque mois t sont caractérisés par un vecteur de prix $S_t = (S_t^{(0)}, S_t^{(1)}, S_t^{(2)}, S_t^{(3)})$ où $S_t^{(0)}$ est le prix de l'actif sans risque et $S_t^{(i)}, i = 1, 2, 3$ sont les prix des actifs risqués.
- **Les actions** : Les actions possibles à chaque étape t consistent à allouer des fonds aux actifs risqués. On peut investir un pourcentage $0 \leq \alpha_t^i \leq 1$ de notre portefeuille dans chaque actif risqué i , avec $\sum_{i=0}^3 \alpha_t^i = 1$.
- **Les fonctions de transition** : Les fonctions de transition déterminent comment l'état du marché évolue d'une étape à l'autre. Les prix des actifs risqués suivent un mouvement brownien géométrique multivarié avec une corrélation entre les différents actifs.

En effet, comme l'actif $S^{(0)}$ est sans risque, il est donc déterministe ; on peut écrire :

$$S_{t+1}^{(0)} = S_t^{(0)}(1 + r) \quad (3)$$

avec r le taux d'intérêt mensuellement composé supposé constant.

Pour ce qui est des autres actifs $S^{(i)}$, $i = 1, 2, 3$, ils satisfont chacun l'équation différentielle stochastique :

$$dS_t^{(i)} = \mu_i S_t^{(i)} dt + \sigma_i dB_t^{(i)}. \quad (4)$$

Soit Σ la matrice de covariance associée au vecteur des actifs risqués. On peut donc décomposer Σ en écrivant $\Sigma = AA^T$ et alors :

$$dS_t^{(i)} = \mu_i S_t^{(i)} dt + \sum_{j=1}^3 A_{ij} S_t^{(i)} dW_t^{(j)}. \quad (5)$$

Ceci permet d'aboutir à l'algorithme de simulation suivant :

$$S_{t_{n+1}}^{(i)} = S_{t_n}^{(i)} \exp \left(\left(\mu_i - \frac{\sigma_i^2}{2} \right) (t_{n+1} - t_n) + \sqrt{t_{n+1} - t_n} \sum_{j=1}^3 A_{ij} Z_{k+1}^j \right) \quad (6)$$

avec $i = 1, 2, 3$ et $n = 0, \dots, N - 1$ où Z est un vecteur normal standard (Glasserman, 2013 [1]). Ainsi, les équations 3 et 6 consistent la fonction de transition du vecteur S_t :

$$f(S_t, \alpha_t, Z) = \begin{cases} S_t^{(0)}(1+r) \\ S_{t_n}^{(i)} \exp \left(\left(\mu_i - \frac{\sigma_i^2}{2} \right) (t_{n+1} - t_n) + \sqrt{t_{n+1} - t_n} \sum_{j=1}^3 A_{ij} Z_{k+1}^j \right) \end{cases} \quad (7)$$

- **Les coûts** : L'objectif de ce problème de ce problème consiste à maximiser l'espérance de l'utilité de la richesse finale en minimisant le risque possible lié aux allocations dans les actifs risqués. Au pire, on s'impose donc un seuil de tolérance au risque.

★ **Choix de la fonction d'utilité** : En ce qui concerne la fonction d'utilité, nous sommes dans une politique où l'actif sans risque ne donne pas un rendement conséquent pour atteindre le rendement escompté. Alors, l'investisseur est prêt à **accepter un certain niveau de risque pour obtenir un rendement potentiel plus élevé mais à un certain point**. On va donc utiliser la fonction d'utilité **CRRA** (Constant Relative Risk Aversion) employée par les investisseurs qui ont une aversion relative au risque. Son expression mathématique est donnée par :

$$U(W) = \begin{cases} \frac{W^{1-\gamma} - 1}{1-\gamma} & \text{si } \gamma \neq 1 \\ \log(W) & \text{si } \gamma = 1 \end{cases} \quad (8)$$

où W représente la richesse du portefeuille, γ le coefficient d'aversion au risque, (Michael Wickens, (2011) [2])

En pratique, $\gamma = \gamma_t$ et est généralement compris entre 0 et 10 mais, il peut aussi prendre des valeurs élevées ou plus faibles. En effet, si γ est grand, l'investisseur est très aversif au risque (préfère les investissements moins risqués et plus certains) et γ est petit, il est plus tolérant au risque. On choisira arbitrairement $\gamma = 1.2$ dans notre travail quelque soit l'instance.

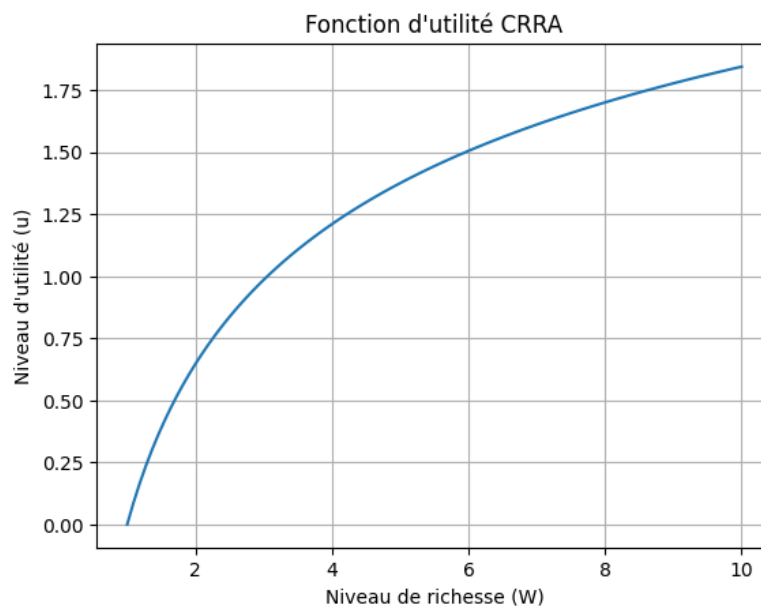


FIGURE 1 – Fonction d'utilité CRRA avec comme paramètre de risque d'aversion $\gamma = 1.2$

- ★ **Calcul du risque associé** : On suppose qu'on est à l'étape t . On compte allouer les fonds par le vecteur $\alpha_t = (\alpha_t^0, \alpha_t^1, \alpha_t^2, \alpha_t^3)$ avec α_t^i l'allocation à l'actif i . L'actif $S^{(0)}$ est sans risque et non corrélé aux actifs $S^{(1)}, \dots, S^{(d)}$ qui sont corrélés entre eux. On prend donc Σ la matrice de covariance associée utilisée dans la section des fonctions de transition. Le risque d'investissement à l'étape t est donc donné par :

$$R_t = \sqrt{\alpha_t^\top \Sigma \alpha_t} \quad (9)$$

R_t est contraint d'être inférieur à R_0 . On veut donc maximiser l'espérance de l'utilité de la quantité $(W_t - R_t W_t)$ avec

$$W_t = \sum_{i=0}^d \alpha_t^i S_t^{(i)} \quad (10)$$

À l'instant final $t = T$, on va considérer le coût comme étant **l'utilité du prix maximum entre les composantes du vecteur de prix** :

$$W_T = \max_{i=0, \dots, d} S_T^{(i)}$$

La fonction de coût associée g serait donc définie par :

$$g(S_t, \alpha_t) = \begin{cases} U(W_T) & \text{si } t = T \\ U(W_t - R_t W_t) & \text{si } t \neq T \text{ et } R_t \leq R_0 \\ 0 & \text{sinon} \end{cases} \quad (11)$$

- **Les équations de récurrence** : Soit \mathcal{U}_t l'ensemble des actions que peut prendre l'investisseur défini par :

$$\mathcal{U}_t = \left\{ \alpha_t = (\alpha_t^0, \alpha_t^1, \dots, \alpha_t^d) \in [0, 1]^d : \sum_{i=0}^d \alpha_t^i = 1, \quad R_t \leq R_0 \right\}. \quad (12)$$

On a donc les équations suivantes :

$$\begin{aligned} J_T(S_T) &= g(S_T) \\ J_t(S_t) &= \max_{\alpha_t \in \mathcal{U}_t} \mathbb{E}[U(S_T) \mid S_t] \\ &= \max_{\alpha_t \in \mathcal{U}_t} \mathbb{E}[g(S_t, \alpha_t) + J_{t+1}(S_{t+1})] \end{aligned} \quad (13)$$

3 Algorithmes

3.1 Méthode exacte : Chaînage arrière

L'algorithme de chaînage arrière est particulièrement approprié pour résoudre des problèmes d'optimisation dynamique où les décisions prises à chaque étape dépendent des décisions futures et où les états futurs sont incertains. Dans notre problème, on cherche à maximiser l'espérance de $J_{t+1}(S_{t+1})$ en choisissant la meilleure décision α_t à chaque étape t , en utilisant la fonction d'utilité CRRA pour évaluer les valeurs de $J_t(S_t)$ et donc de $J_{t+1}(S_{t+1})$. Comme cet algorithme permet également de stocker les décisions optimales α_t à chaque étape, il nous permettra, par conséquent, de déterminer la meilleure politique de décision pour le problème considéré.

Voici les étapes (*Cours*, [3]) pour l'algorithme de chaînage arrière pour notre sujet à cet effet :

Algorithme 1 : Procédure Chaînage Arrière

```

 $J_T(S_T) \leftarrow U(W_T)$ 
pour  $t = T-1, \dots, 1$  faire
  pour  $S_t \in X_t$  faire
     $J_t(S_t) \leftarrow \max_{\alpha_t \in \mathcal{U}_t} \mathbb{E}[g(S_t, \alpha_t) + J_{t+1}(S_{t+1})]$ 
     $\alpha_t^*(S_t) \leftarrow \arg \max_{\alpha_t \in \mathcal{U}_t} \mathbb{E}[g(S_t, \alpha_t) + J_{t+1}(S_{t+1})]$ 
  fin
fin

```

3.2 Méthode approximative : Fitted value iteration

L'utilisation de la méthode d'approximation dans l'algorithme de "Fitted Value Iteration" est cruciale pour résoudre des problèmes de décision séquentielle avec des espaces d'états et d'actions complexes. En effet, elle est souvent nécessaire lorsque l'espace d'états est continu. Dans les problèmes de décision séquentielle avec des espaces d'états continus, il est souvent impossible ou inefficace de rechercher exhaustivement toutes les valeurs possibles de la fonction de valeur ou de la fonction d'action, en raison du grand nombre de points dans l'espace d'états. L'approximation permet de représenter ces fonctions de valeur ou d'action de manière compacte (optimale $\tilde{J}_t(\mathbf{S}_t, r_t)$ ou la fonction d'action optimale $\tilde{Q}_t(\mathbf{S}_t, \mathbf{A}_t, r_t)$ à chaque étape de temps t . Cela permet de généraliser les valeurs ou les actions optimales à partir d'un ensemble limité d'exemples d'états et de récompenses, obtenus par échantillonnage Monte Carlo.), ce qui rend l'algorithme de résolution plus rapide et plus efficace.

Voici les étapes (Cours, [3]) pour cet algorithme pour notre sujet à cet effet :

Algorithme 2 : Fitted value iteration : Simulation Monte Carlo Simple

Données : Architecture de $\tilde{J}_t(\mathbf{S}_t, r_t)$, $t = 1, \dots, T - 1$ et paramètres du problème

Résultats : Vecteur \hat{r}_t , $t = 1, \dots, T - 1$ et donc les fonctions $\tilde{J}_t(\mathbf{S}_t, \hat{r}_t)$, $t = 1, \dots, T - 1$

$q = Q$ // Nombre d'échantillons par étapes t

$\tilde{J}_T(\mathbf{S}_T) = U(W_T)$

$\hat{r}_T = 0$

$t = T - 1$

tant que $t \geq 0$ **faire**

$s = 1$

tant que $s \leq q$ **faire**

 Échantillonner un état \mathbf{S}_t^s // Une méthode Monte Carlo

$\beta_k^s = \max_{\alpha_t \in \mathcal{U}_t(\mathbf{S}_t^s)} \mathbb{E} [\tilde{J}_{t+1}(f_t(\mathbf{S}_t^s, \alpha_t), Z_t), \hat{r}_{t+1}]$

 Sauvegarder l'exemple $(\mathbf{S}_t^s, \beta_k^s)$

$s = s + 1$

fin

$\hat{r}_t \in \arg \min_r \sum_{s=1}^q (\tilde{J}_t(\mathbf{S}_t^s, r) - \beta_t^s)^2$ // Apprentissage sur les données

$t = t - 1$

fin

Cependant, il est important de noter que l'approximation introduit également des erreurs d'approximation dans les estimations des fonctions de valeur ou d'action, ce qui peut affecter la qualité des politiques de décision obtenues. Le choix de la méthode d'approximation et des paramètres d'apprentissage peut avoir un impact sur la performance de l'algorithme, et nécessite une validation et une expérimentation appropriées pour s'assurer de la qualité des résultats obtenus.

4 Résultats et Analyse

4.1 Petite instance

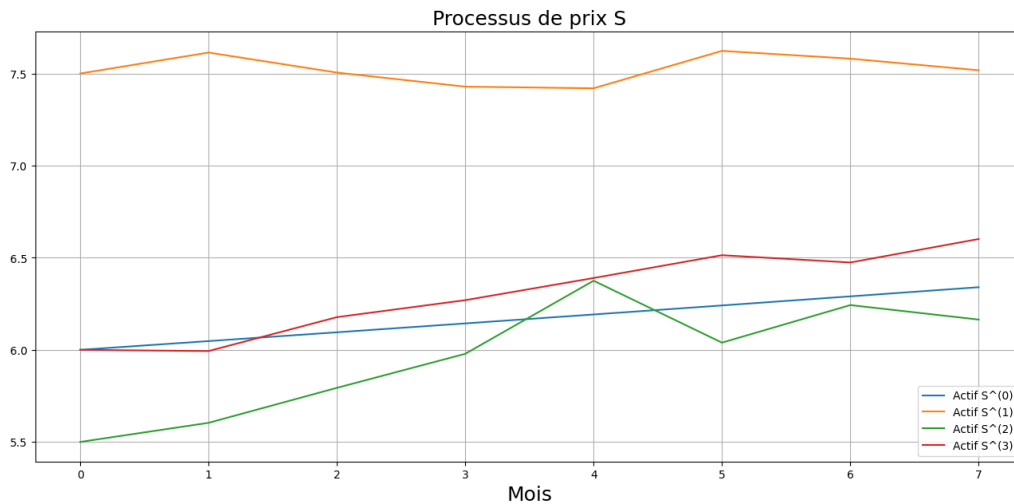


FIGURE 2 – Processus de prix $S_t^{(i)}$ avec $t = 1, \dots, 8$

Le processus de prix sur lequel nous avons travaillé est celui donné par la figure(2) On suppose qu'à chaque étape, le processus de prix ne peut transiter que dans **3 états**.

4.1.1 Chainage arrière

La politique à adopter à l'instant initiale est donné par la sortie de code suivant :

```
Étape : 1
Prix S[1] = [6.04743416 7.60861865 5.5385559 5.96498117]
J[etat 1, allocation optimale = [0.52 0.46 0.01 0.01]] = 13.303578812616632

Prix S[1] = [6.04743416 7.66292058 5.56080985 5.9594844 ]
J[etat 2, allocation optimale = [0.49 0.49 0.01 0.01]] = 13.248713737942186

Prix S[1] = [6.04743416 7.58528052 5.62303254 5.94217088]
J[etat 3, allocation optimale = [0.47 0.43 0.01 0.09]] = 13.261227307532899
```

Elle suggère à l'investisseur d'investir au départ, dans les actifs $S^{(0)}$ et $S^{(1)}$. Ceci peut se comprendre car l'investisseur est assez prudent dès son entrée en bourse et est donc favorable à miser sur l'actif risqué pour assurer un rendement positif et dans un actif dont le prix est **élevé en espérant que le prix augmentera avec le temps**.

Voici une sortie du code lorsqu'on se retrouve à l'étape 7 donc à $t = 6$:

```
Étape : 7
Prix S[7] = [6.34001874 7.51324243 6.18656946 6.63151214]
J[etat 1, allocation optimale = [0.09 0. 0. 0.91]] = 3.35211179724291

Prix S[7] = [6.34001874 7.46572065 6.22818575 6.6249277 ]
J[etat 2, allocation optimale = [0.08 0.05 0.05 0.82]] = 3.3354784784080493

Prix S[7] = [6.34001874 7.49533823 6.08393392 6.63111718]
J[etat 3, allocation optimale = [0.25 0.24 0. 0.51]] = 3.332219973562793
```

Pour cette sortie, on voit que l'investisseur **doit beaucoup investir dans l'actif $S^{(3)}$ plus que les autres** pour obtenir un meilleur rendement au mois suivant. En effet, lorsqu'on se calque sur la figure (2), on remarque bien que le prix de l'actif $S^{(3)}$ a bien augmenté au mois suivant ce qui donne de **la crédibilité** sur la politique d'allocation.

Aussi, par rapport à l'état 3 du processus, on remarque que l'investisseur **doit investir près d'un quart de ses fonds dans l'actif $S^{(1)}$** alors que sur la figure (2), son prix décroît au mois suivant. Cela n'est pas

anodin car en fait, **les actifs $S^{(1)}$ et $S^{(3)}$ sont corrélés négativement**. La matrice de corrélation associée est en effet :

```
array([[ 1.          ,  0.21559353, -0.23061348],
       [ 0.21559353,  1.          , -0.1984001 ],
       [-0.23061348, -0.1984001 ,  1.          ]])
```

Cependant, **il ne doit pas trop y investir car cet actif est plus volatile que l'actif $S^{(3)}$** . Ceci justifie pourquoi, **dans ce cas, il doit faire une "bonne allocation dans l'actif sans risque $S^{(0)}$ pour couvrir d'éventuelles pertes"**.

Analyse : La politique obtenue garantit que l'investisseur fera des allocations de manière à "rester à flot". Cela signifie que l'investisseur allouera suffisamment de ressources pour minimiser le risque de perdre tout son capital, en veillant à ce que les allocations soient au moins couvertes par l'actif sans risque.

La politique garantit également que le capital de l'investisseur reste positif tout au long de l'horizon de temps considéré. Cela est important pour assurer la viabilité financière de l'investisseur et lui permettre de continuer à investir dans les états futurs.

En se basant sur cette politique, on obtient une maximisation du coût J dont le tracé est donné par la figure (3). Les données sont très sensiblement proches à chaque étapes. C'est la raison pour laquelle on a représenté l'exponentielle des coûts.

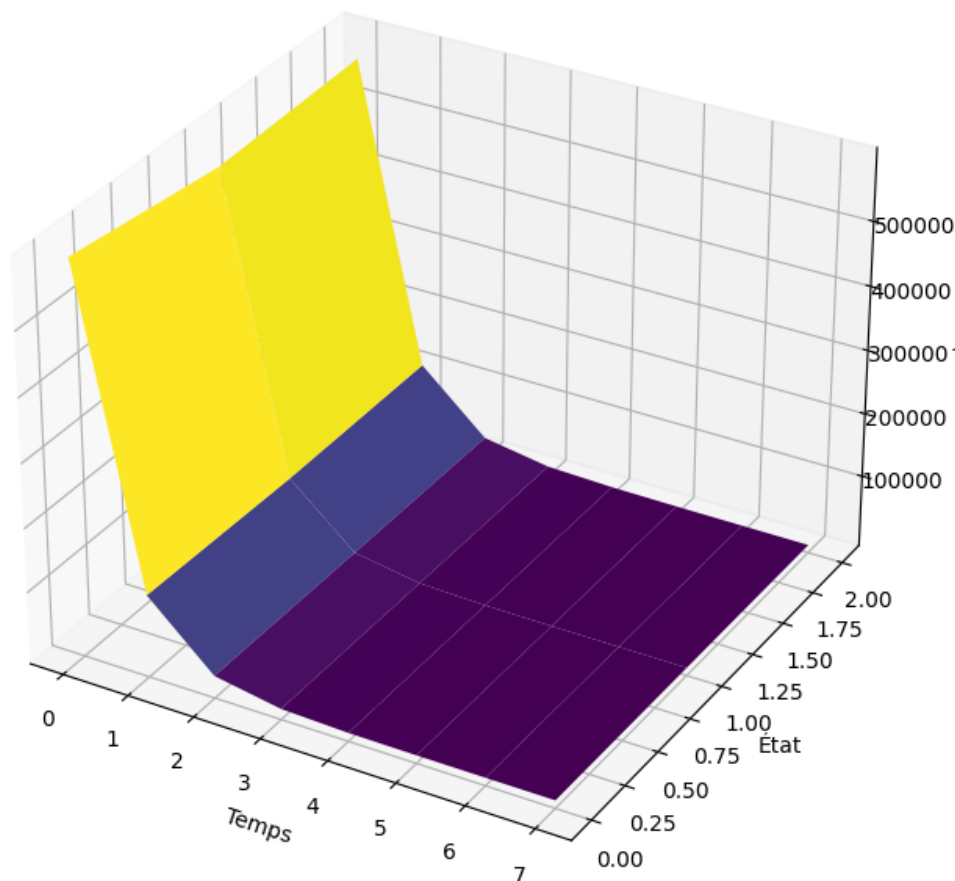


FIGURE 3 – Exponentielle du coût J en fonction de la politique optimale

4.1.2 Méthode approximative

- **Fitted Value Iteration :**

La politique optimale obtenue à partir de la méthode approximative offre de mauvaises allocations. En effet, on observe avec cette régression qu'il faut continuer à allouer beaucoup de fonds à l'actif $S^{(1)}$ en allouant le reste dans l'actif sans risque pour couvrir les possibles pertes et dans l'actif $S^{(3)}$ qui a aussi un bon rendement ; comme le montre ce vecteur des allocations :

```
array([0.13042803, 0.69285562, 0.0378025 , 0.13891385])
```

Cependant, avec cette figure, l'algorithme a maximisé le coût ce qui est donné par la figure (4).

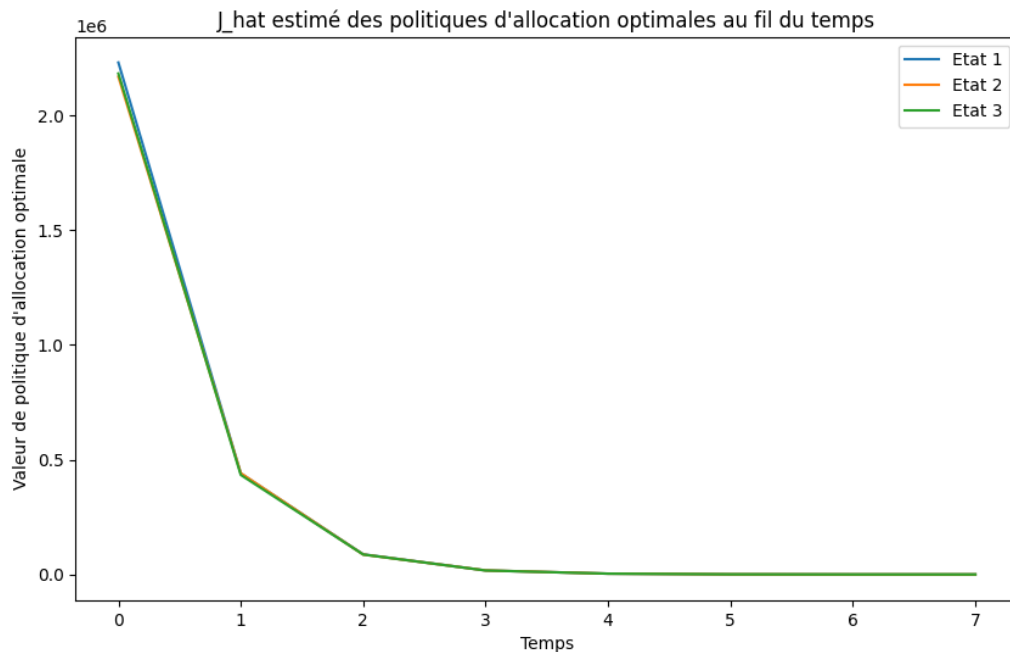


FIGURE 4 – Exponentielle du coût \hat{J} en fonction de la politique optimale

On a essayé d'évaluer l'erreur absolue et celle relative entre les politiques de l'algorithme et celles de la méthode exacte. Les figures (5) et (6) montre que plus on se rapproche de l'étape initiale, l'erreur tend vers zéro ce qui est bon signe.

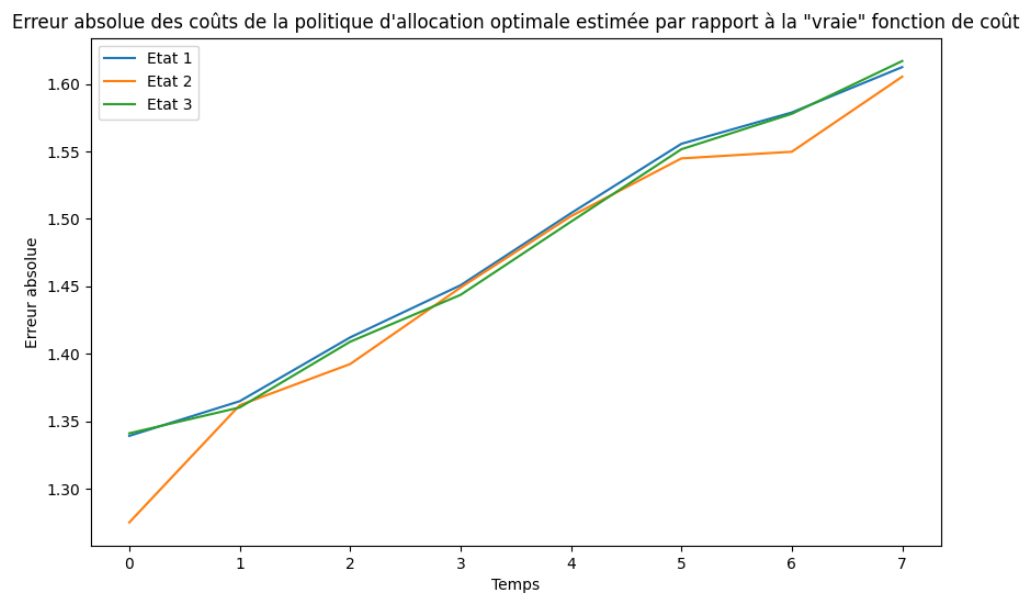


FIGURE 5 – Erreur absolue

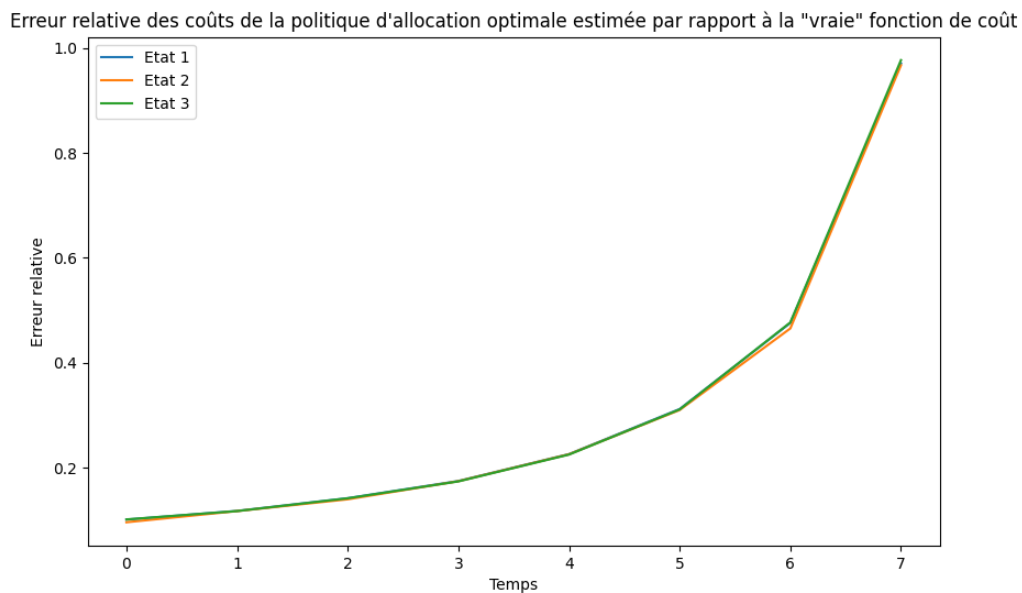


FIGURE 6 – Erreur relative

Toutefois, l'erreur relative croît rapidement à partir de $t = 4$. On peut supposer que cela est dû au choix répitif de continuer à allouer beaucoup de fonds à l'actif $S^{(1)}$ ce qui n'est pas le cas avec la méthode exacte.

On peut émettre l'hypothèse que l'algorithme de fitted value ne semble pas bonne pour une petite instance dans notre contexte même si elle semble donner de bonnes politiques pour les premières étapes.

- **Réseau de neurones** : On obtient les erreurs absolues et relatives montrées dans les figures suivantes :

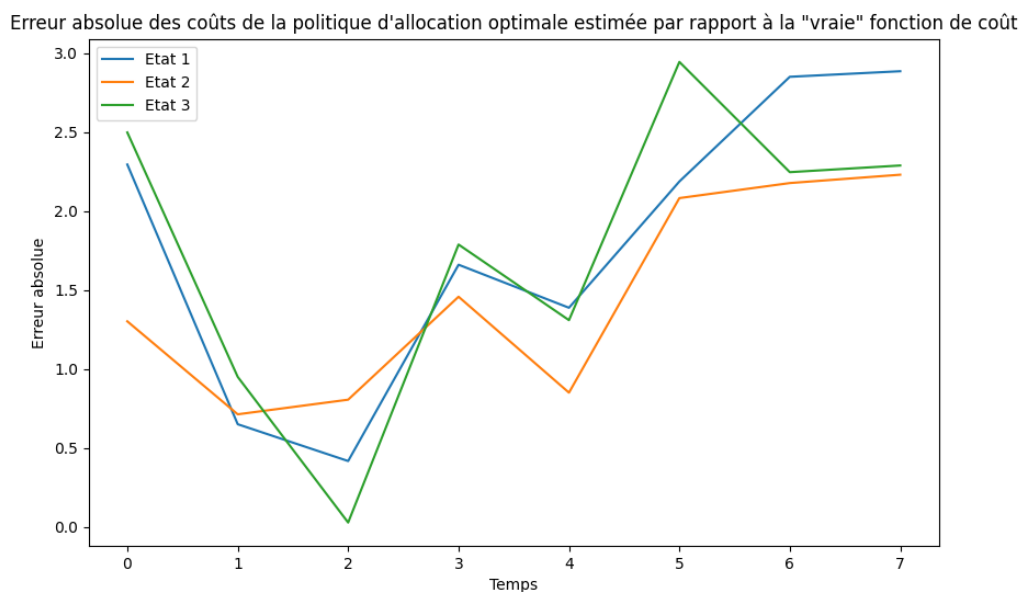


FIGURE 7 – Erreur absolue

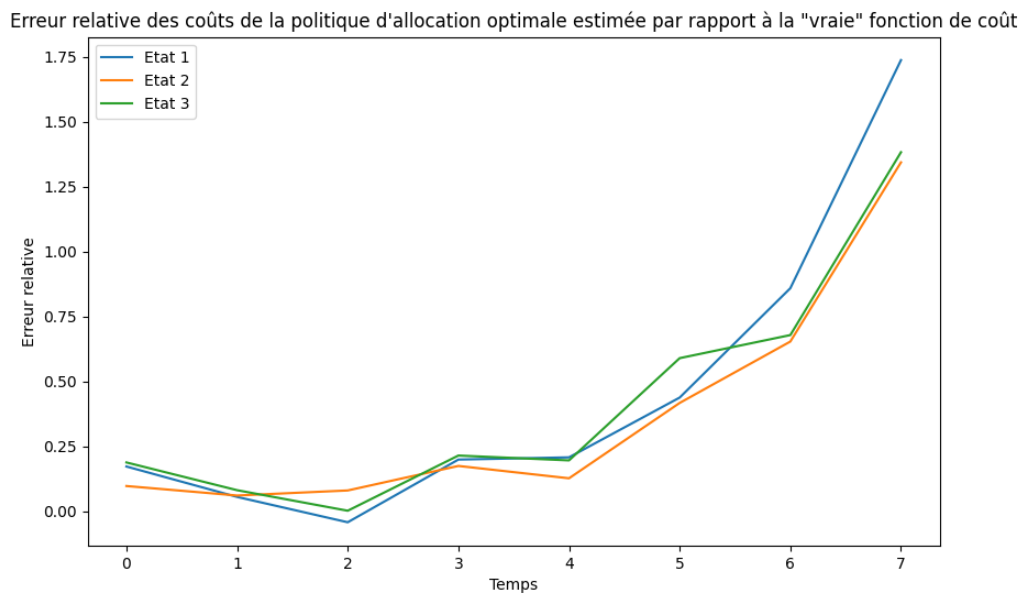
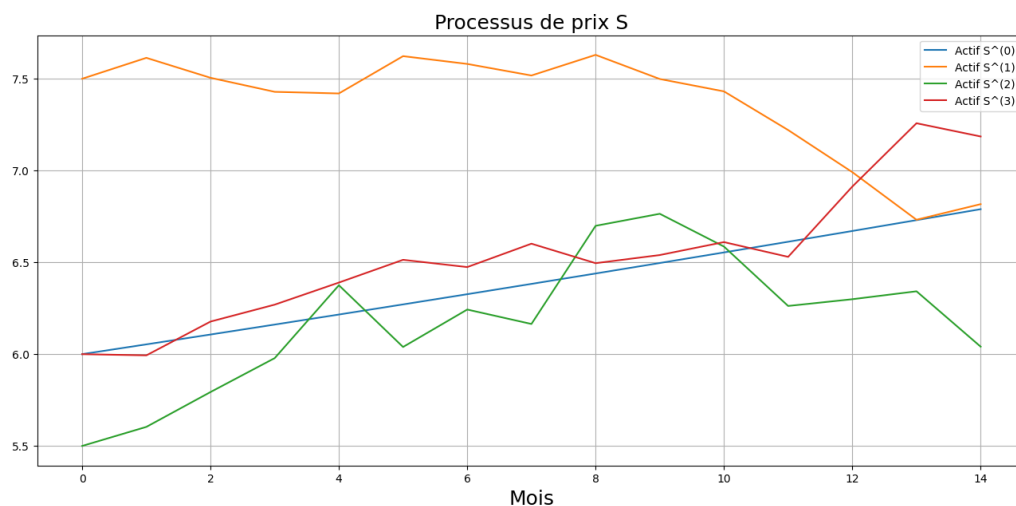


FIGURE 8 – Erreur relative

L'erreur absolue observée donne une structure **affligeante** ! Ces deux figures impliquent les mêmes analyses que celles de l'algorithme "fitted value iteration". On peut conclure :

Pour une petite instance (avec T assez petit et peu d'états à supposer), l'algorithme de chaînage arrière est préférable même si la méthode approximative donne de bons résultats lorsqu'on se rapproche des premières étapes.

4.2 Moyenne instance

FIGURE 9 – Processus de Prix pour $t = 1, \dots, 15$

Le processus de prix sur lequel nous avons travaillé est celui donné par la figure(9) On suppose qu'à chaque étape, le processus de prix ne peut transiter que dans **10 états**.

4.2.1 Chainage arrière

Cette instance nous donne les mêmes politiques et analyses que celles de la petite instance. **Cependant, on commence à noter une certaine lenteur de l'algorithme !**

4.2.2 Méthode approximative

- **Fitted Value Iteration** : On commence à avoir de très bons résultats. En effet, cela se voit par les figures (10) et (11) qui illustrent les erreurs absolues et relatives des \hat{J} ajustés par rapport aux J de la méthode exacte. Même si **l'erreur absolue semble continuellement croître**, on remarque en effet, que **les politiques sont bonnes sur un bon nombre d'étapes**. L'erreur relative commence à sérieusement croître qu'à partir de $t = 11$, ce qui est très bon signe. En effet, on a une sortie de la politique d'allocations optimale qui est au mois 14 :

[0.20975744, 0.20206452, 0.00879607, 0.57938197]

Même si les allocations sont approximatives, elles sont bien en accord avec l'exemple de la petite instance, c'est-à-dire qu'on doit beaucoup investir dans l'actif $S^{(3)}$ et investir un peu dans l'actif $S^{(1)}$ (à cause de la corrélation négative) en s'assurant de couvrir sa forte volatilité par l'actif sans risque.

On commence donc à avoir une politique similaire à celle de la petite instance, celle de "rester à flôt".

Erreur absolue des coûts de la politique d'allocation optimale estimée par rapport à la "vraie" fonction de coût

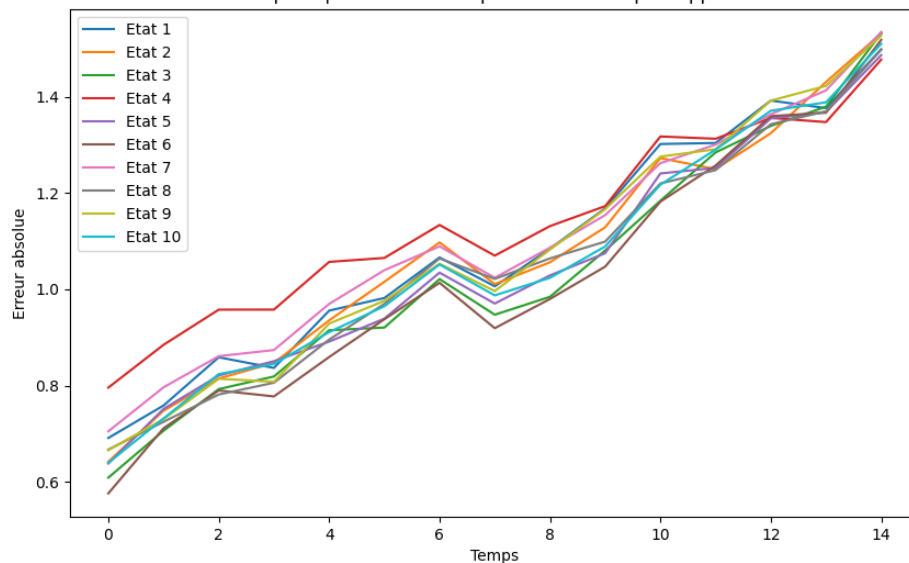


FIGURE 10 – Erreur absolue

Erreur relative des coûts de la politique d'allocation optimale estimée par rapport à la "vraie" fonction de coût

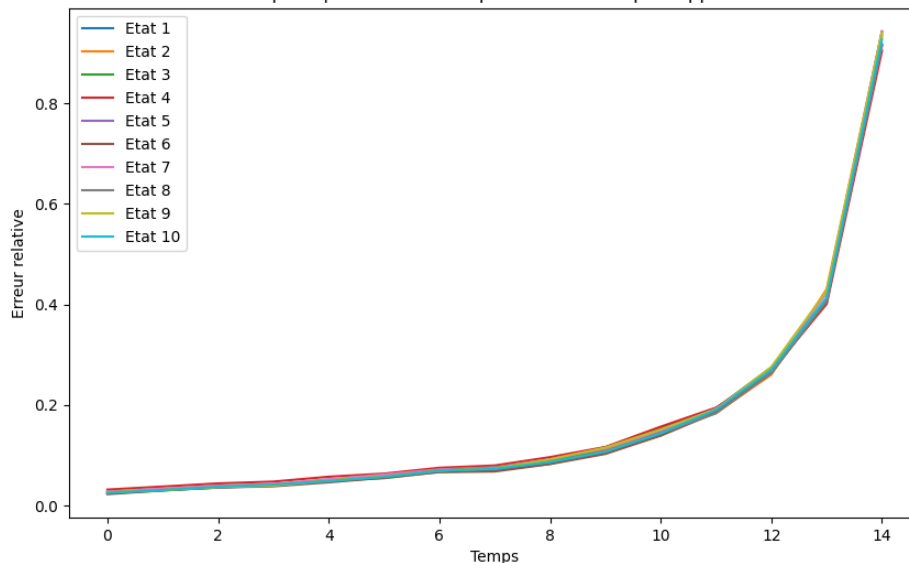


FIGURE 11 – Erreur relative

- Réseau de neurones :

Erreur absolue des coûts de la politique d'allocation optimale estimée par rapport à la "vraie" fonction de coût

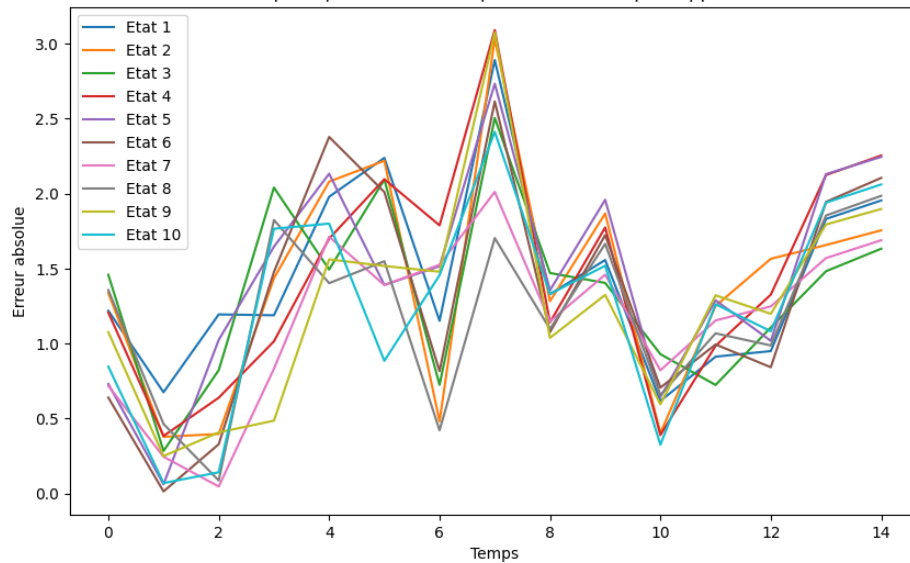


FIGURE 12 – Erreur absolue

Erreur relative des coûts de la politique d'allocation optimale estimée par rapport à la "vraie" fonction de coût

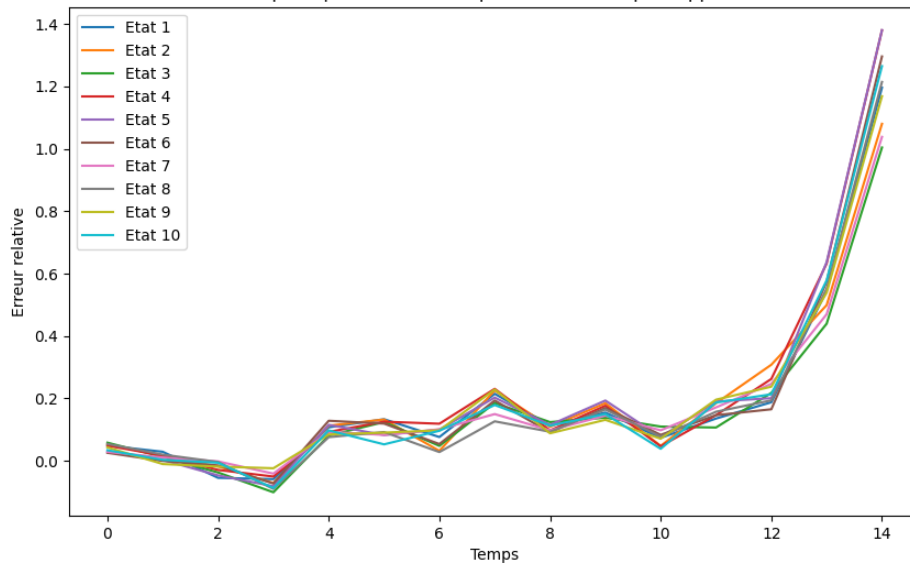


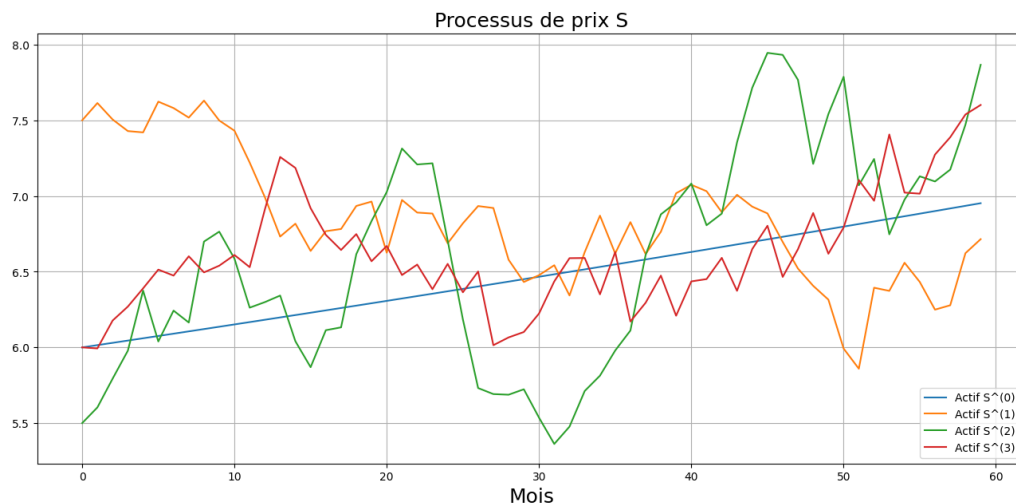
FIGURE 13 – Erreur relative

En se basant plus sur la deuxième figure, on voit qu'on obtient plus vite de bonnes approximations d'allocations. En effet, l'erreur croît sérieusement qu'à partir de $t = 12$, même si ce n'est qu'un mois de plus, le J avec la régression neuronale donne de bons résultats.

On peut commencer vraiment par envisager utiliser une méthode approximative car **la convergence des résultats est très rapide** et les allocations sont assez proches de celles de la "vraie politique optimale".

4.3 Grande Instance

Le processus de prix sur lequel nous avons travaillé est celui donné par la figure(14) On suppose qu'à chaque étape, le processus de prix ne peut transiter que dans **20 états**.

FIGURE 14 – Processus de prix pour $t = 1, \dots, 60$

4.3.1 Chainage arrière

Ceci est mon avis personnel : **Il ne faut pas envisager l'utiliser dans ce contexte lorsque les données deviennent de plus en plus grandes!** En effet, on pourrait avoir d'allocations optimales "exactes" mais **le temps de calcul est désespérément long!** En effet, ma méthode exacte a pris un temps de compilation de :

```
time: 30min 14s
```

Je voudrais attirer l'attention sur le fait que **n'avons considérer que $T = 60$ mois et juste 20 états de transition pour le prix des actifs.**

Cependant, on tombe toujours sur de bonnes allocations. Voici, en effet, un exemple :

```
Prix S[37] = [6.58071918 6.61082523 6.6719957 6.35388561]
J[etat 9, allocation optimale = [0.27 0.28 0.29 0.16]] = 40.63051607788534
```

Cette politique suggère que pour cet état du processus de prix au mois 36, l'investisseur doit faire une allocation qui semble équitable entre les actifs. C'est évident qu'il doit investir dans l'actif $S^{(2)}$ mais doit rester modéré car cet investissement conditionne un grand risque. Heureusement, il majore son risque par R_0 . Pour s'assurer donc une croissance au mois 37, il investit dans les actifs S^1 et $S^{(3)}$ car négativement corrélés entre eux donc un sûrement va monter. Pour couvrir tout le risque d'investissement R_t , il doit donc aussi investir une somme conséquente dans l'actif sans risque $S^{(0)}$. Ainsi, **il est à flôt!**

4.3.2 Méthode approximative

- **Fitted Value Iteration :**

D'abord, bonne nouvelle : le temps de calcul n'est que de :

```
time: 3min 8s
```

C'est environ **10 fois moins que celle de la méthode exacte**. Cependant, il aboutit à des allocations approximatives. Là où la méthode exacte suggère de ne pas prendre trop de risque alors que le prix de l'actif S^2 ne fait qu'augmenter, lui donne une politique d'allocation pas forcément la meilleure mais réaliste :

```
array([0.0556412 , 0.11416579, 0.71887441, 0.1113186 ])
```

On investit plus donc dans cet actif et on fait une combinaison d'allocations entre les actifs $S^{(1)}$ et $S^{(3)}$ pour gérer le risque d'allocations, ce qu'il est aussi faisable car ces deux actifs sont négativement corrélés. La figure (15) montre **la convergence de l'erreur relative vers 0** lorsqu'on s'approche de l'instant finale. Cette est "trop rapide" car elle donne de très "bonnes erreurs" à partir de $t = 57$. Que dire ! C'est excellent car on assure d'avoir de bons résultats sur un grand nombre d'étapes, à moindre coût en terme de temps de calcul.

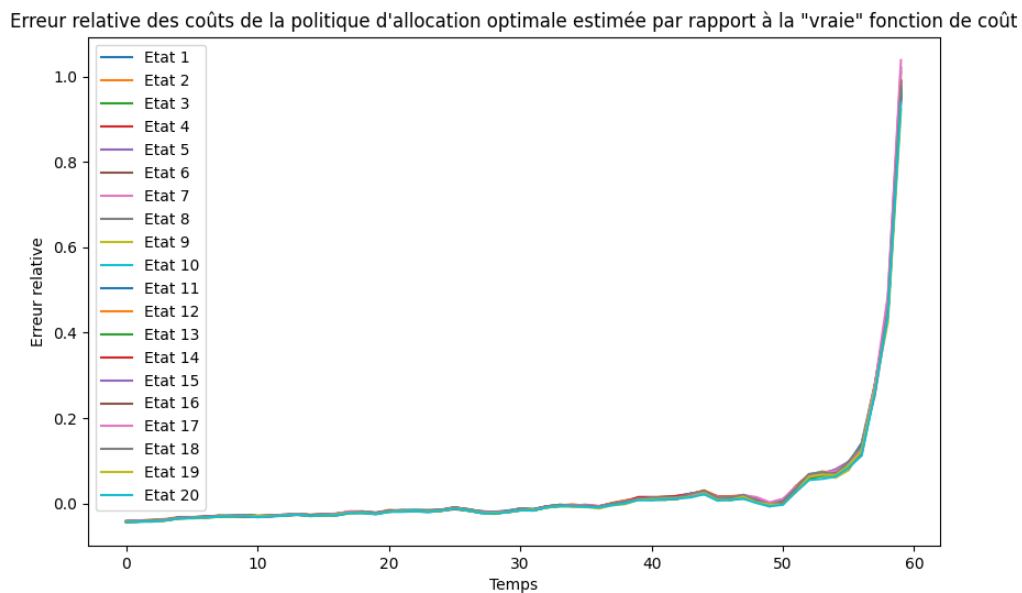


FIGURE 15 – Erreur relative

- **Réseau de neurones** : Approximativement le même temps de calcul :

time: 3min 4s

Avec, la figure (16), on a que le modèle apprend "très mal" dès le début de la récurrence mais converge très rapidement vers les bonnes valeurs de coûts J de méthode exacte déjà à partir de $t = 59$. C'est bon signe ! Même si les allocations obtenues sont approximatives, on préfère opter pour cette méthode.

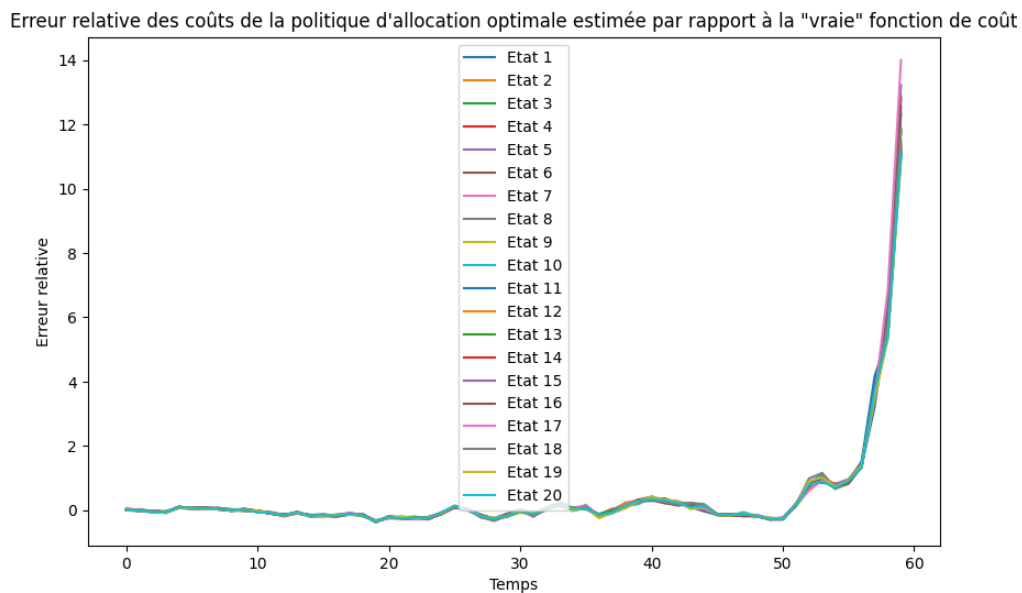


FIGURE 16 – Erreur relative

De tout ce qui précède, on voit qu'opter pour une méthode approximative lorsque l'instance devient de plus en plus grande est conseillé d'autant plus que la convergence vers les bons résultats est très rapide et cette méthode est moins couteuse en temps de calcul.

5 Conclusion

5.1 Faits saillants

Ce travail a apporté son lot de problème. De base, ce sujet est passionnant de par sa complexité et demande beaucoup d'investissement, tant les données que nous devrions utiliser devraient paraître "réalistes".

Ce travail nous permet d'obtenir une politique optimale basée sur la "maintenance à flôt" de l'investisseur. Celle-ci peut paraître prudente. Cependant, cette politique vient des hypothèses que nous nous sommes fixées. On arrive bien à atteindre la richesse finale sans faire des mises folles dans les actifs risqués. Cependant, cette politique peut bien changer si on change l'état de l'aversion de l'investisseur par rapport au risque R_t de ses allocations. Plus l'instance devient grande, plus on a donc tendance à opter pour une méthode approximative moins couteuse en temps de calcul et qui donne pourtant des politiques satisfaisantes.

5.2 Discussion

En fait, les méthodes d'approximation peuvent fonctionner sur ce problème. Cependant, elles peuvent donner des résultats moins corrects à la longue si on suppose par exemple que la fonction de coût est non linéaire. Aussi, dans notre travail, on suppose que le processus de prix suit un mouvement brownien géométrique ce qui n'est pas forcément le cas, comme on peut le deviner. En fait, **avec des processus de Levy, de Merton, on peut arriver à perdre des données lorsqu'on effectue un calcul d'espérance par la méthode de Monte Carlo.** Il faudrait donc, non pas complètement jeter certaines méthodes, mais **les ajuster** de manière à capturer le maximum d'information pour avoir des politiques jugées acceptables.

Références

- [1] P. Glasserman (2013), *Monte Carlo methods in financial engineering*. Springer Science & Business Media, Vol. 53
- [2] Michael Wickens (2011), *Macroeconomic Theory : A Dynamic General Equilibrium Approach*. Princeton University Press, Princeton & Oxford, 2nd Edition, ISBN 978-0-691-15286-8 : 616
- [3] Emma Frejinger, *Programmation Dynamique : Introduction & Approximation paramétrique dans l'espace de valeurs - horizon fini*

Personnes Consultées :

- Williams (Auxiliaire du cours)
- Emma Frejinger (Chargée du cours)
- Ange Maria Mousissa M'bella (Étudiante)
- Youssef Cherkani Hassani (Étudiant)
- Kevin Kiemtore (Étudiant)
- Blanche Kokoré