

**ARMAND MORIN
MALO PERRIEN
AMAURY REVEL**



CentraleSupélec

RAPPORT PROJET ST7

**Minimisation des risques sur les marchés
Optimisation de portefeuille**

**Gad Bouaziz
Volga Technologies**



Sommaire

1. Présentation du problème
2. Contraintes
3. Optimisation
4. Prédiction
5. Analyse des résultats

Présentation

$$F(P_t) = P_t \Sigma P_t - \alpha < \mu, P_t > + \gamma TC(P_t)$$



Optimisation de portefeuille (constitué d'ETFs)

Portefeuille = vecteur des positions prises pour chaque actif à l'instant t, + somme de cash investie dans le portefeuille

Cette représentation est simple et permet d'accéder rapidement à toute l'information, les vecteurs prix étant connus

Optimisation de risque sous contraintes

On cherche principalement à faire une optimisation par le risque, en choisissant une fonction de coût qui représente ce risque, à laquelle on rajoute des contraintes pour orienter l'optimisation

On travaille avec un petit nombre d'actifs possibles :

Une vingtaine d'actifs différents (ETFs) : equities, real estate, financial products, commodities (fournis) + Twitter, Intel, TSM (semi-conducteurs), US mutual fund

Contraintes proposées

01

Limitation de la variation de position par actif

En deux temps : d'abord on a imposé à chaque position de ne pas varier de +/- 10 (un nombre arbitraire), puis on a travaillé en pourcentages

02

Limitation de la variation de notionnel total investi

Pour tenter de limiter les trop grosses variations et obtenir des smooth paths, on impose que le notionnel total ne puisse pas varier de plus que +/- X%

03

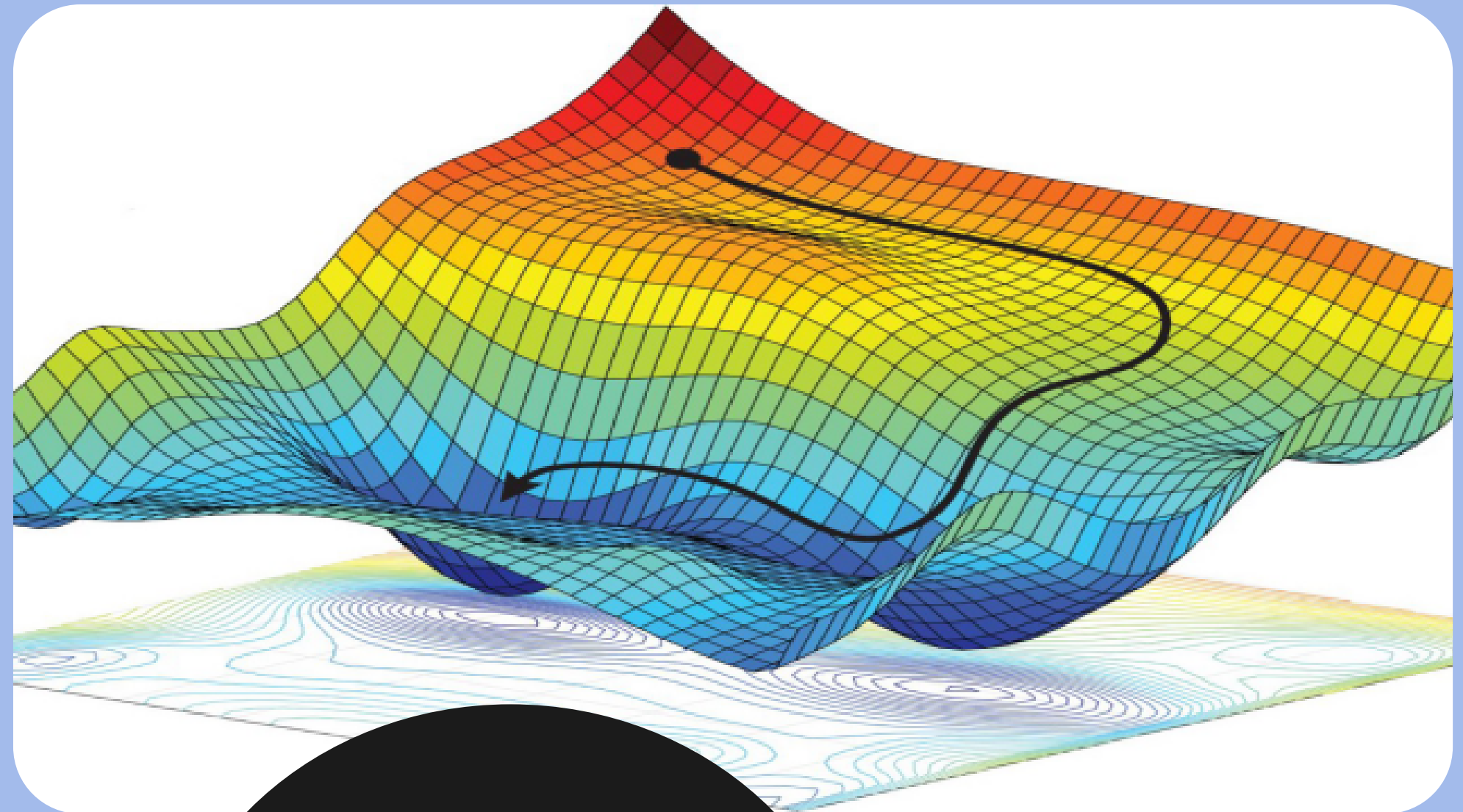
Long-only

Les positions de chaque actif doivent être positives ou nulles

Algorithme d'optimisation

On a utilisé la librairie `scipy.optimize` pour effectuer cette optimisation, en particulier la fonction `minimize`

- Méthodes d'optimisation : SLSQP, COBYLA (retenue), trust-bounds.
- Mémoïsation
- Initial guess : la valeur précédente du vecteur des positions (qui vérifie toutes les contraintes)
- Rebalancement à chaque étape : calcul du cash à partir des vecteurs prix, vecteur position inchangé, calcul de la corrélation, des expected returns
- Stockage des variables pour les graphes

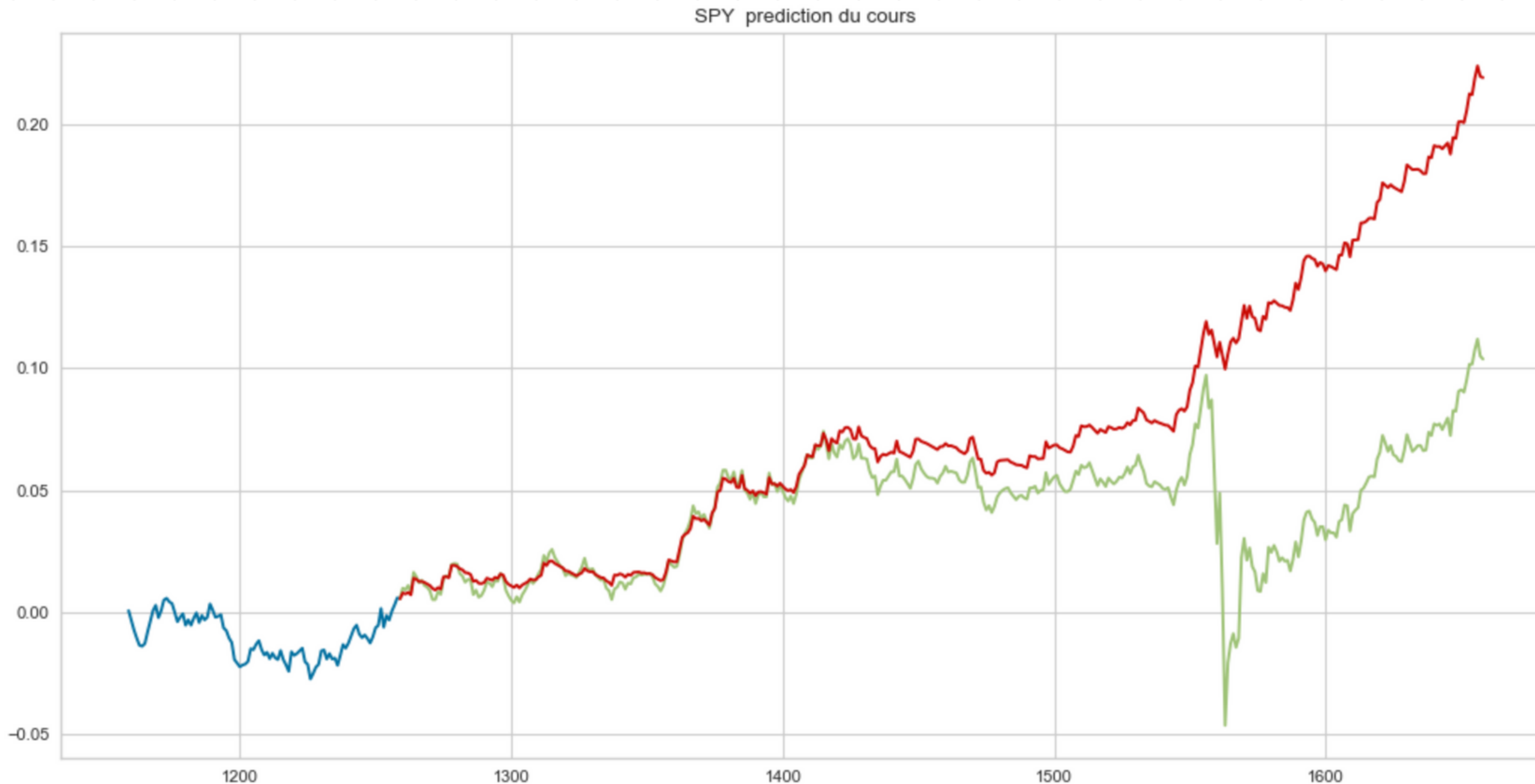


On a choisi de ne pas essayer `nlopt` parce que `scipy` était facile d'utilisation et pour se concentrer sur la gestion des contraintes et des hyperparamètres

Prédiction

Signaux utilisés:

RSI, MACD, SMA, EMA, BB, Volume, Low, High, Mom, StochasticRSI, ForceIndex,



Prédiction refaite tout les mois donc ce n'est pas un problème si la fin n'est pas correcte.

Prédiction que pour SPY.

Trop de temps de le faire pour toutes les stocks car Pycaret demande de valider les features numerical.

On n'a pas eu le temps de tester en direct l'efficacité des prédictions dans les performances du portefeuille par manque de temps. On n'a utilisé simplement les moyennes mobiles.

Résultats

I) Influence des paramètres
(sans contraintes)

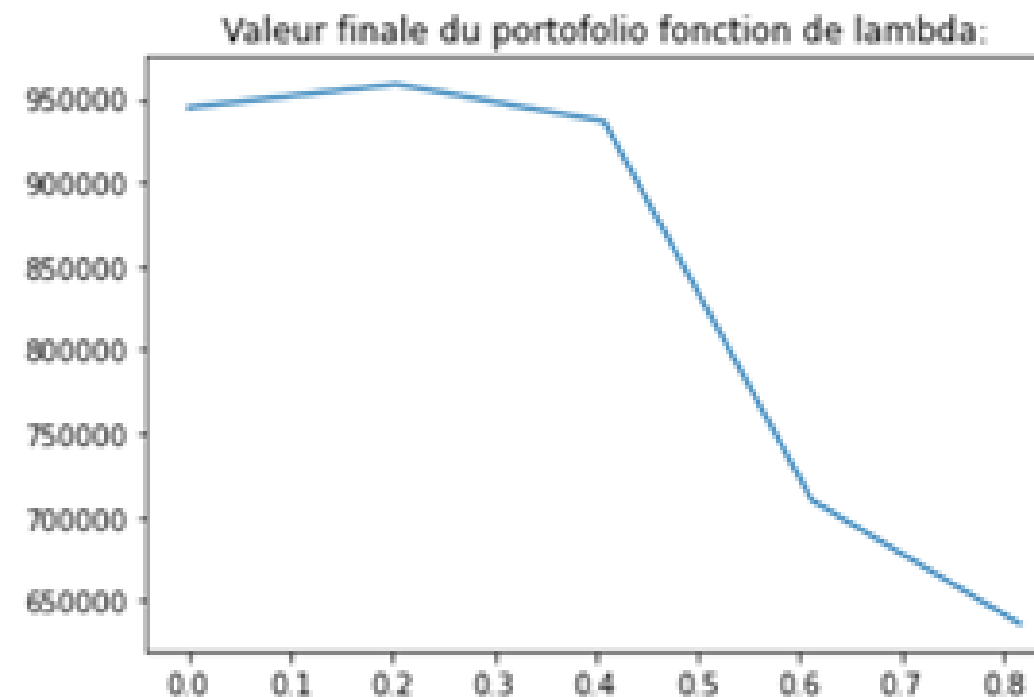
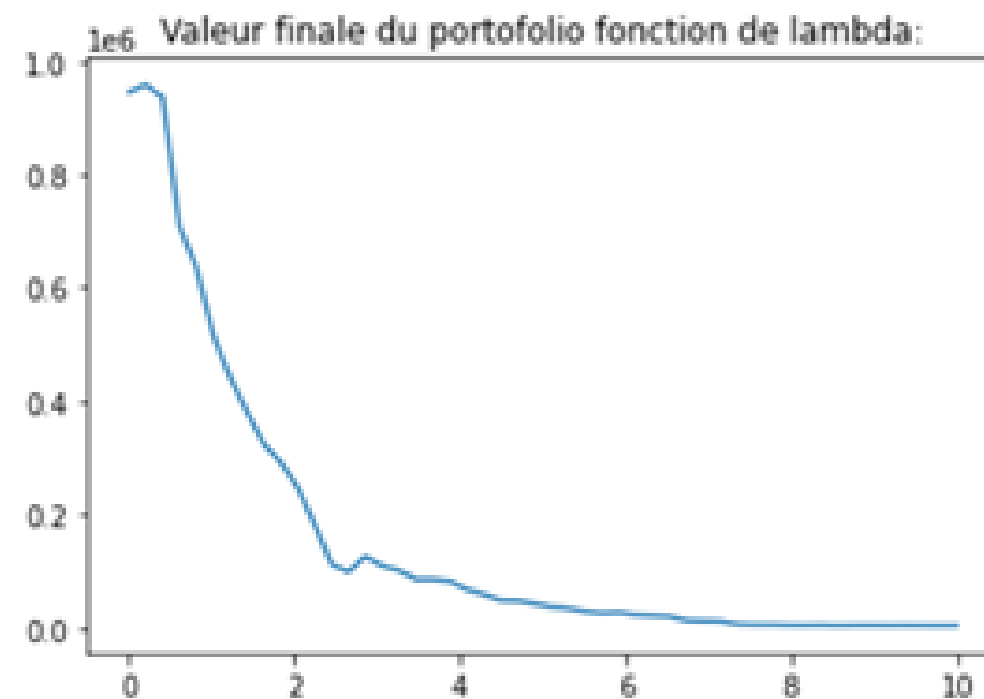
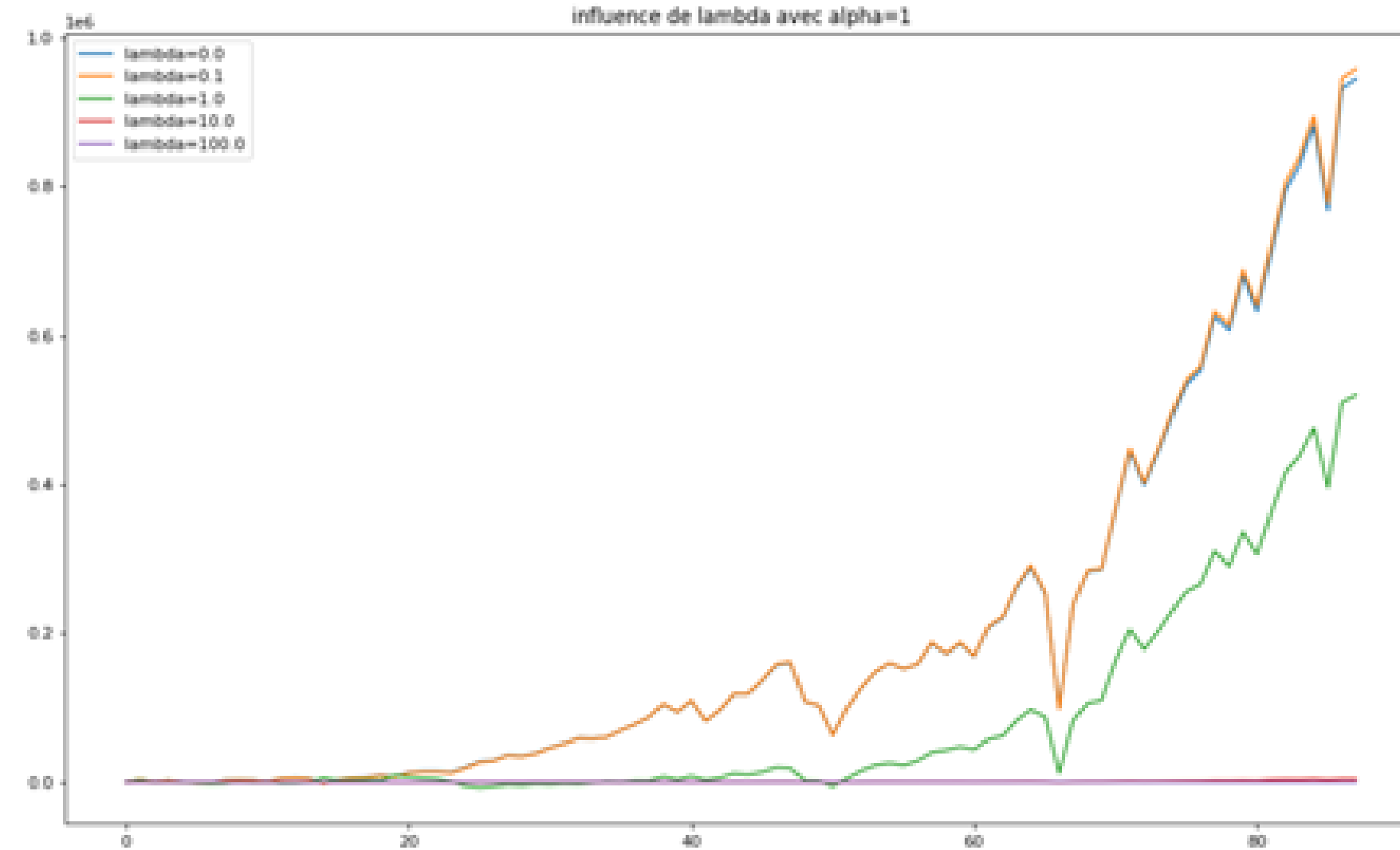
Influence de alpha

- 1/ sur le portefeuille
- 2/ sur le drawdown



Influence de lambda

- 1/ sur le portefeuille
- 2/ sur la valeur finale du portefeuille

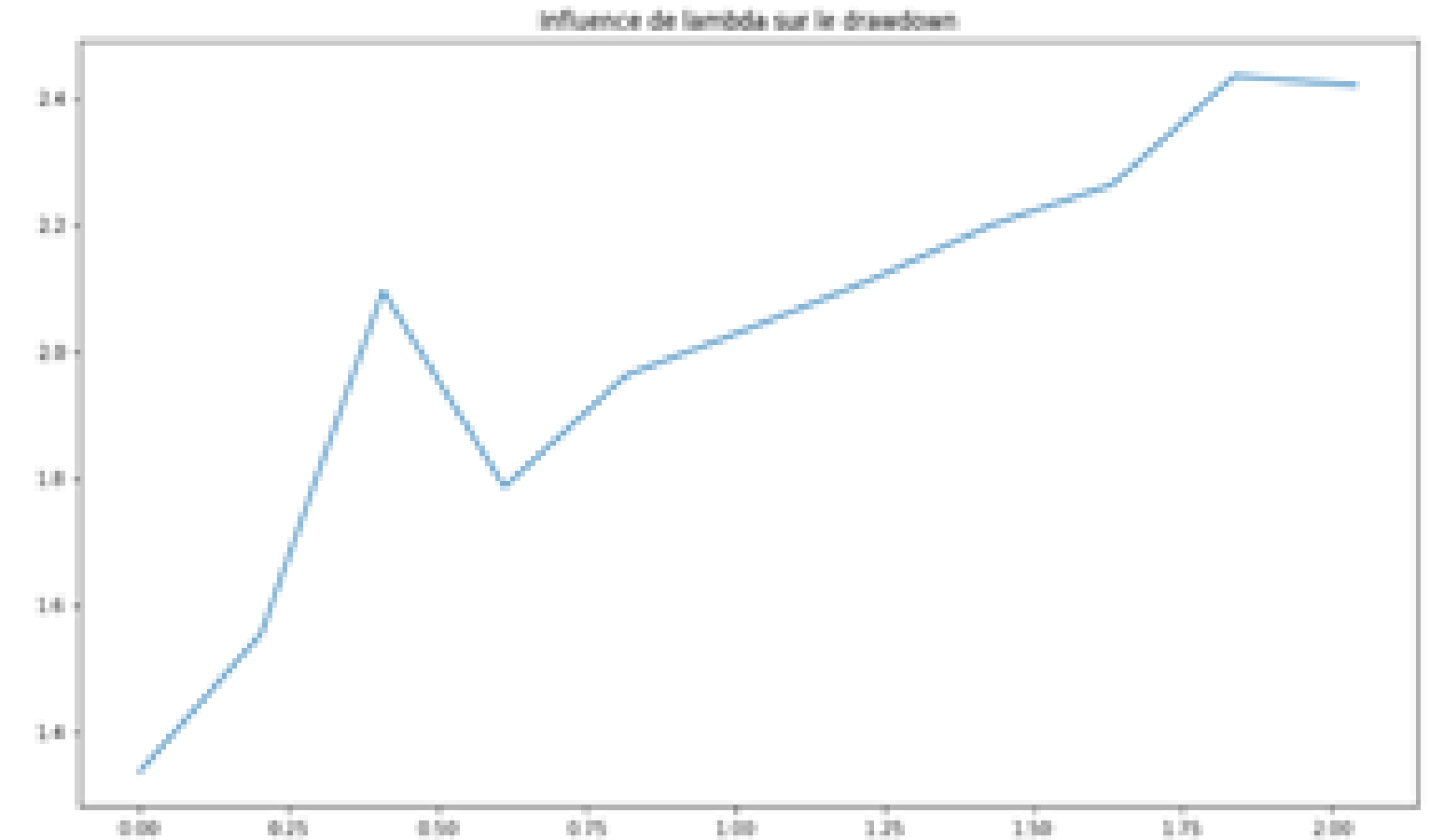
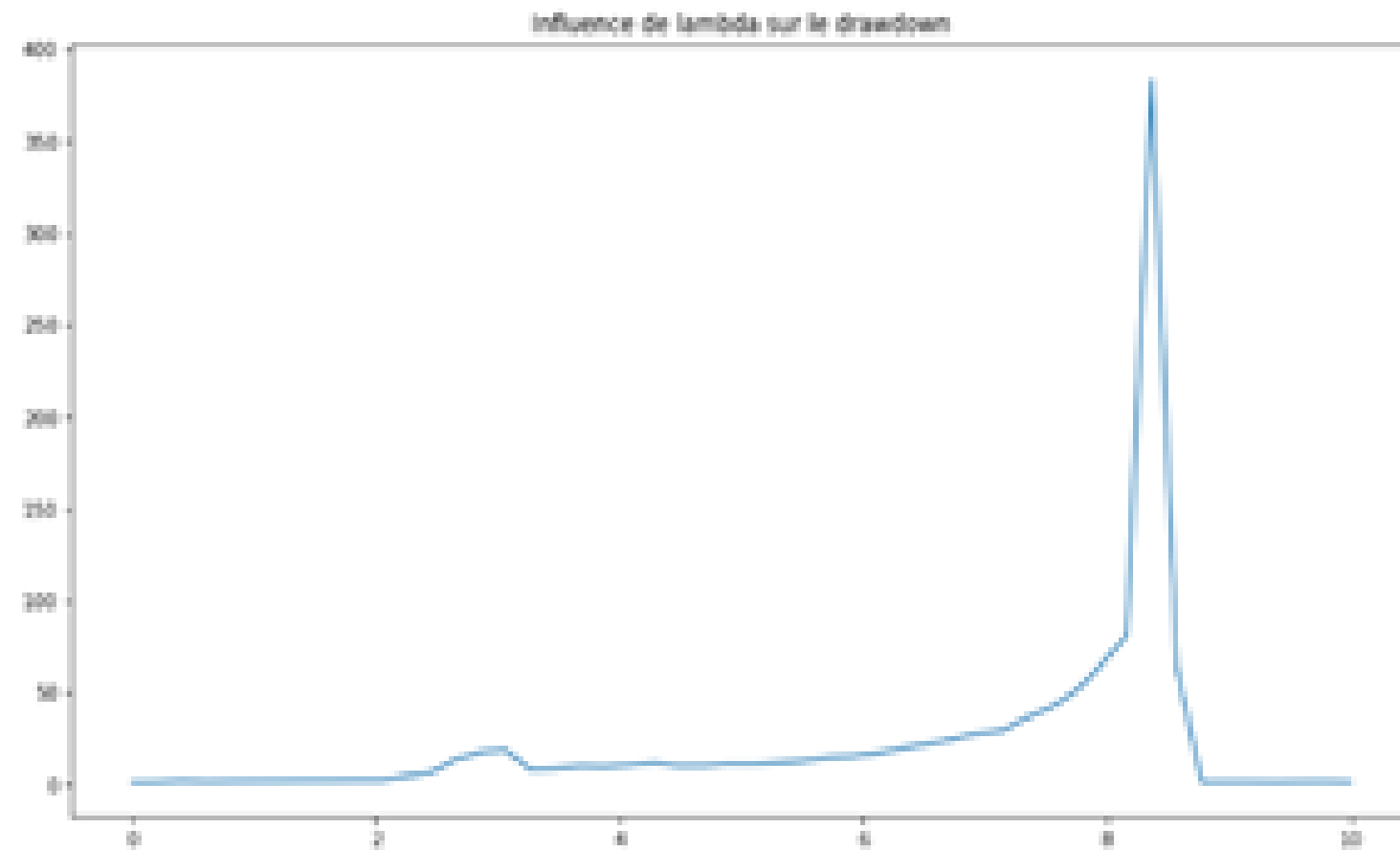


$$F(P_t) = P_t \Sigma P_t - \alpha < \mu, P_t > + \gamma TC(P_t)$$

$$\gamma TC(P_t) = \gamma \sum_i \lambda |P - t(i) - P_{t-1}(i)|$$

Influence de lambda

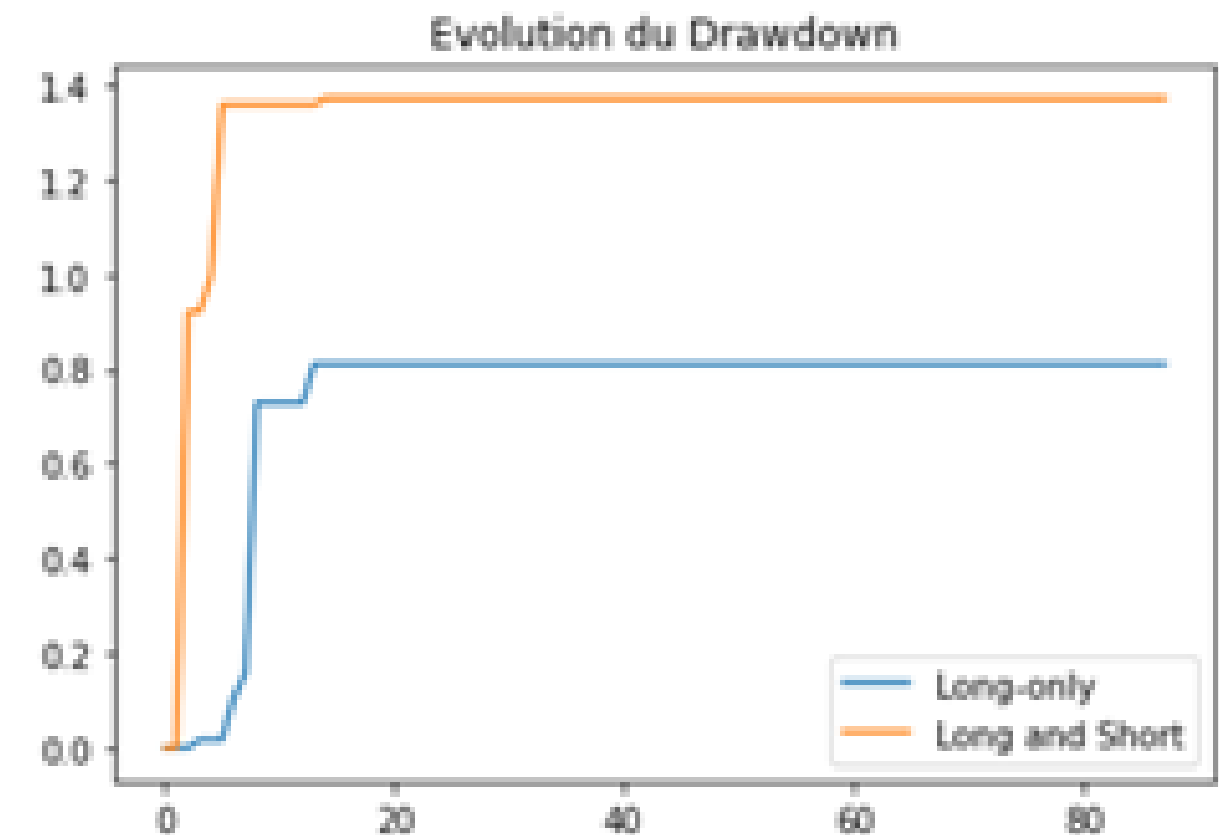
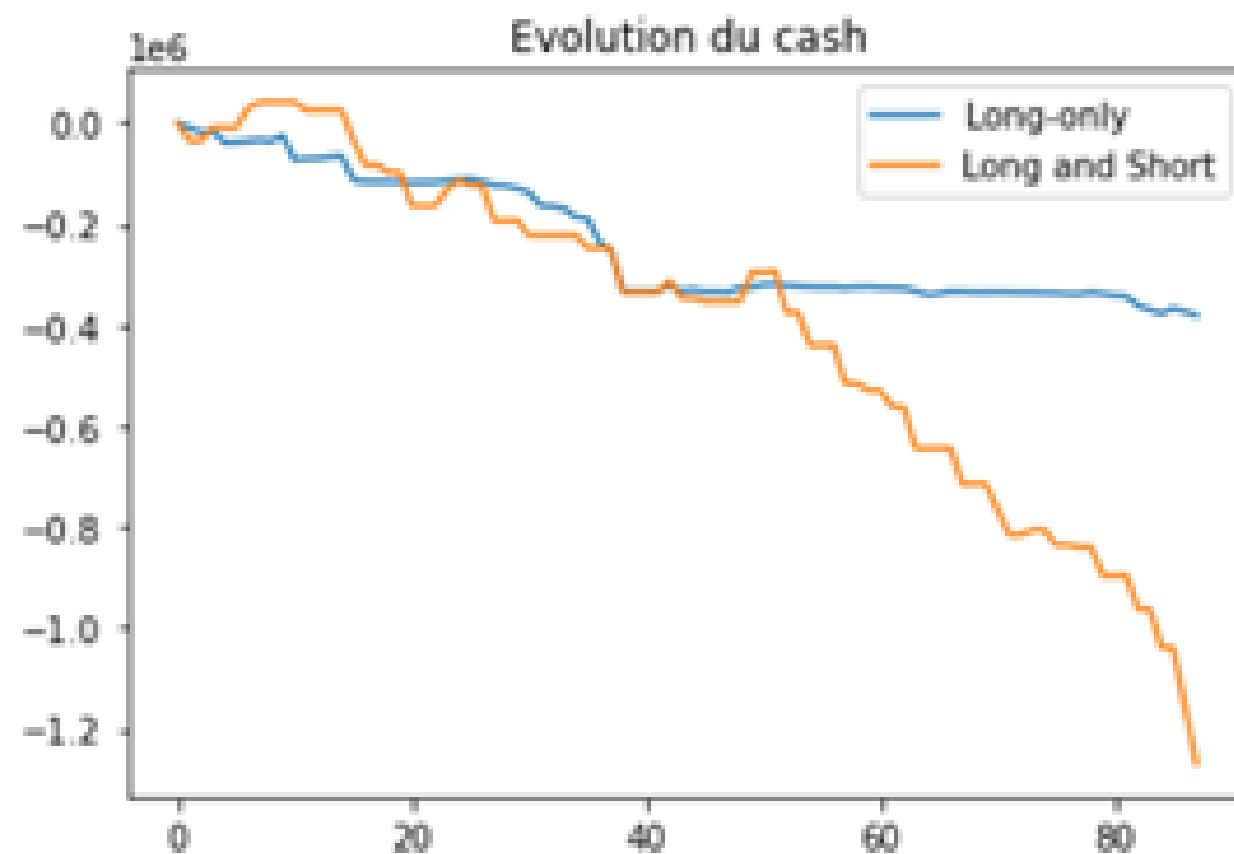
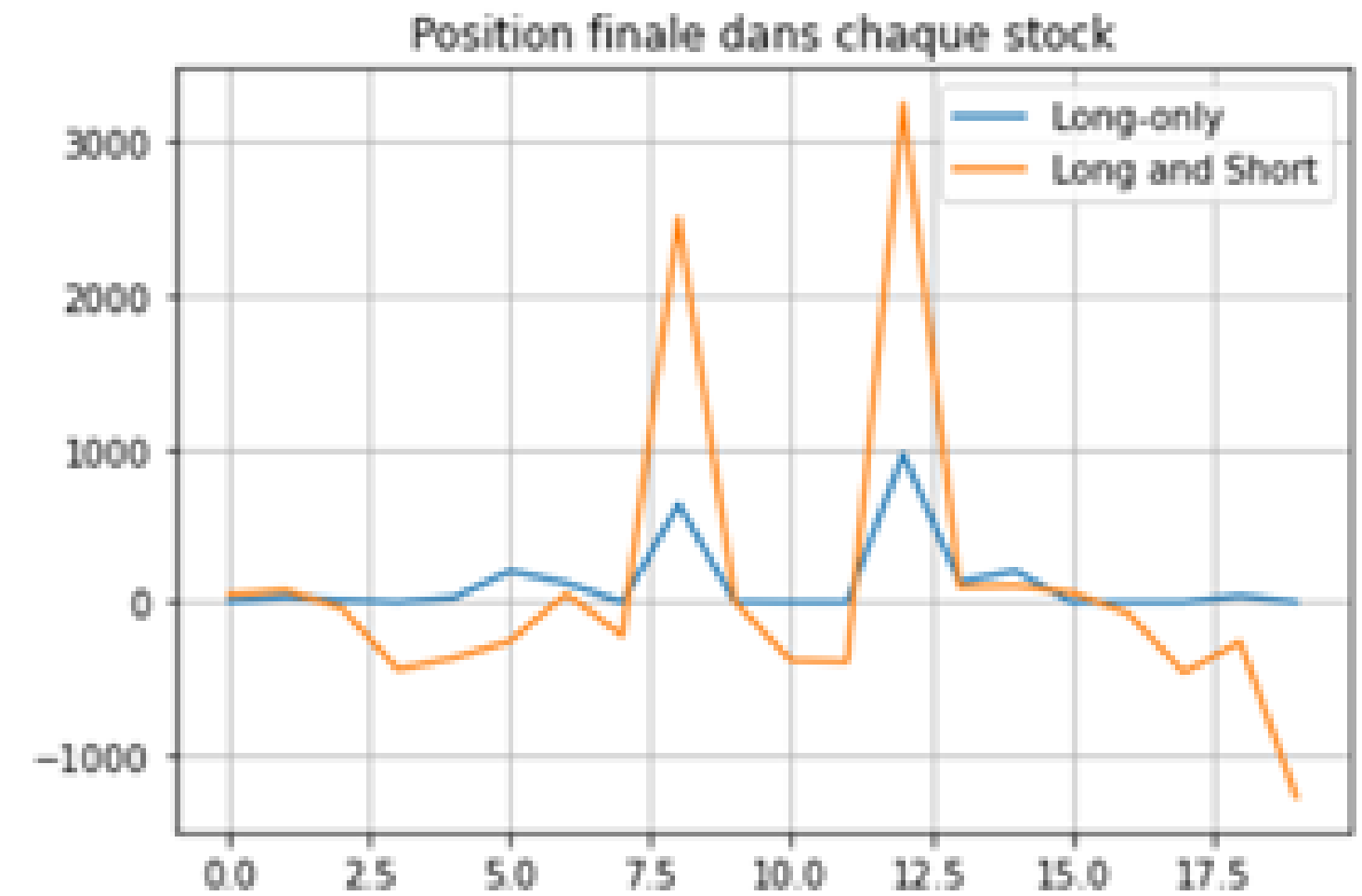
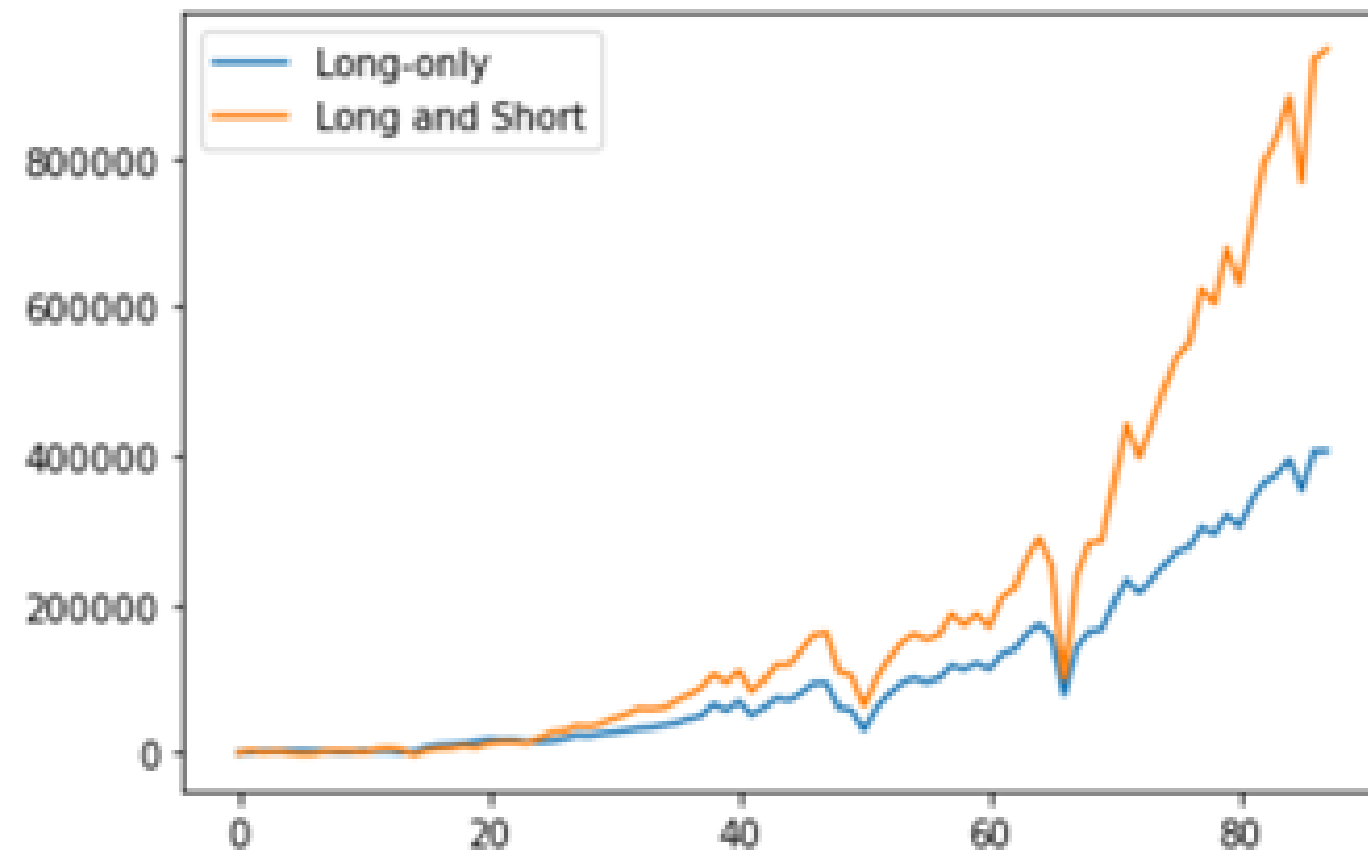
sur le drawdown



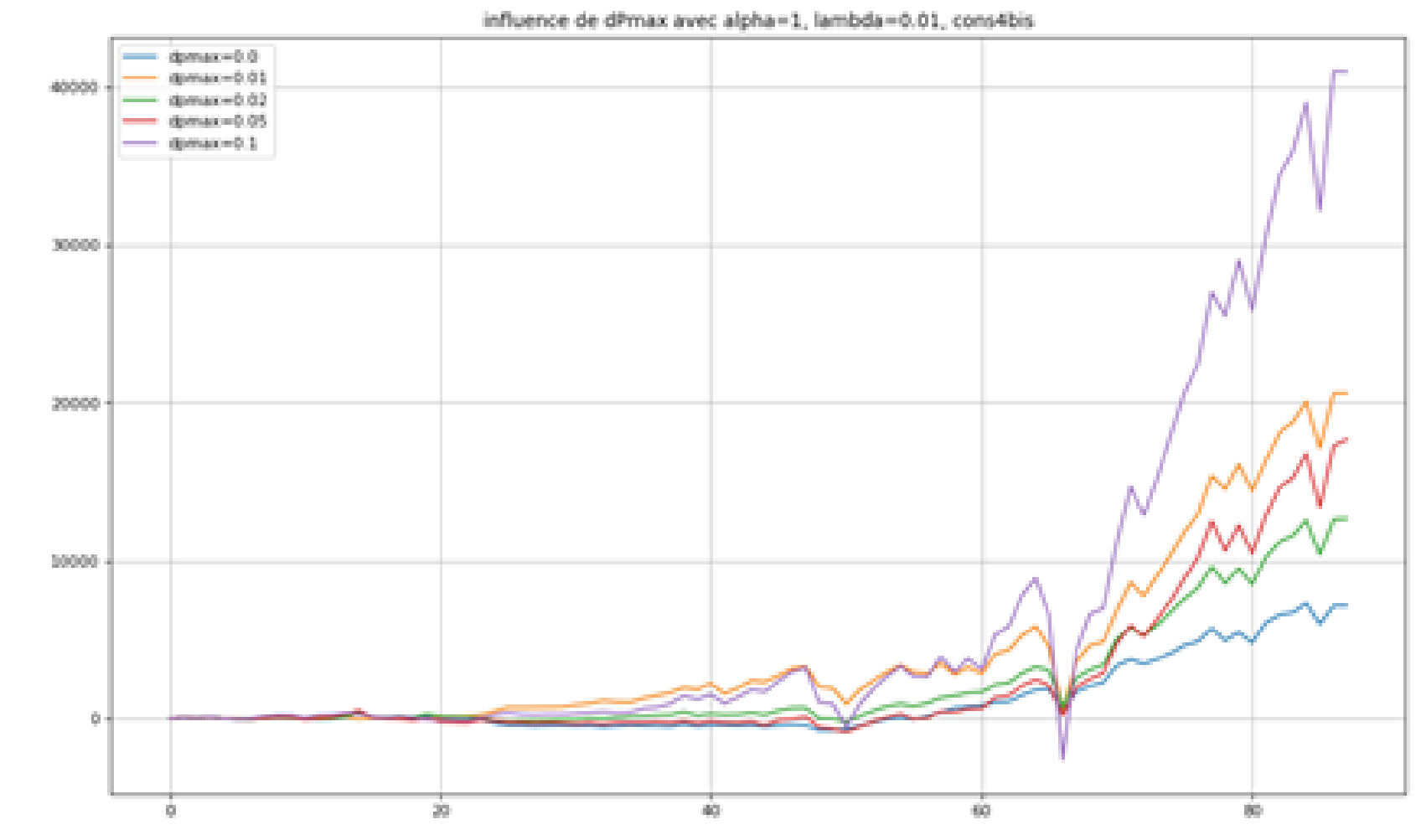
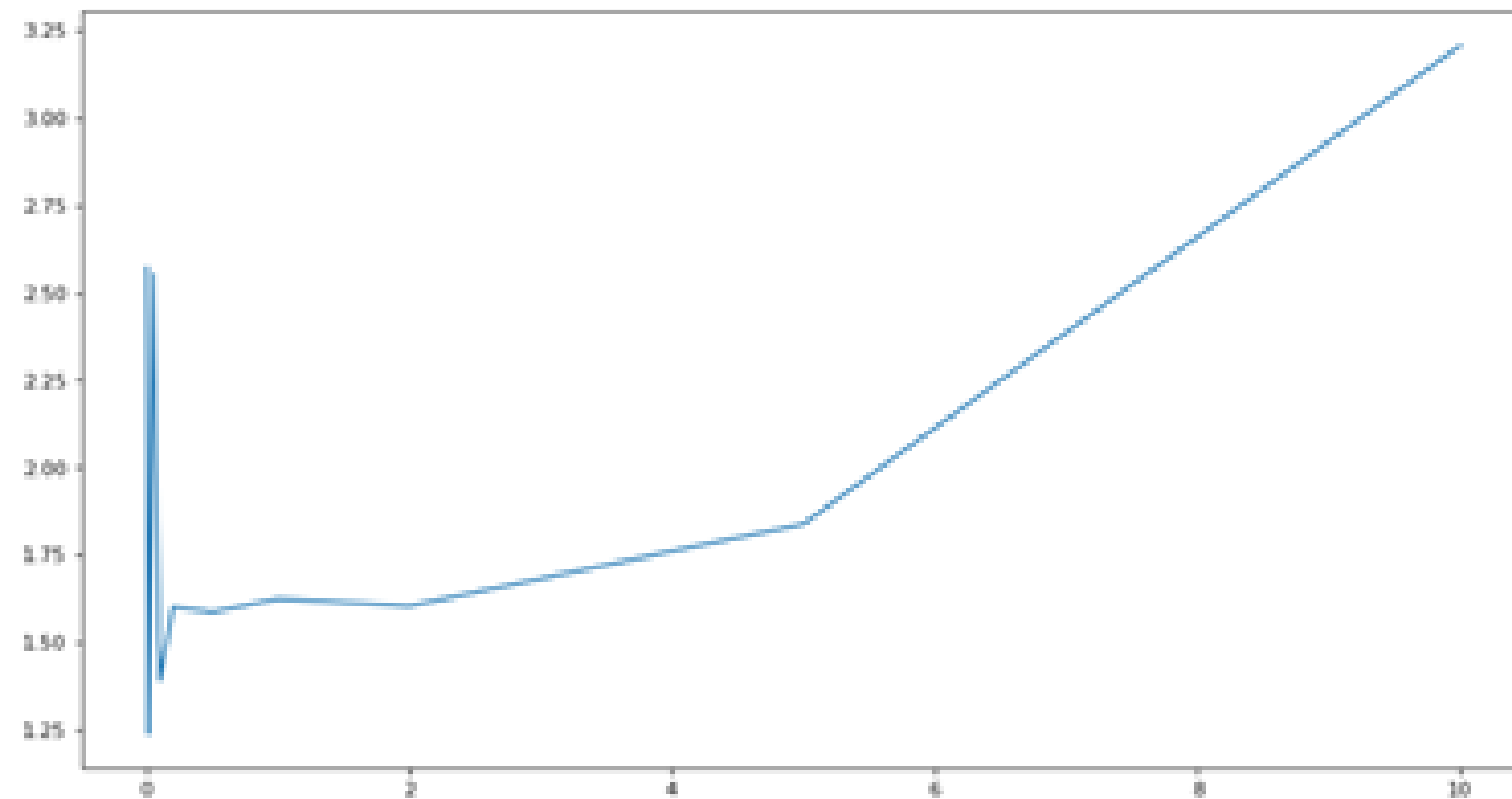
Résultats

II) Influence des contraintes

Contrainte long-only

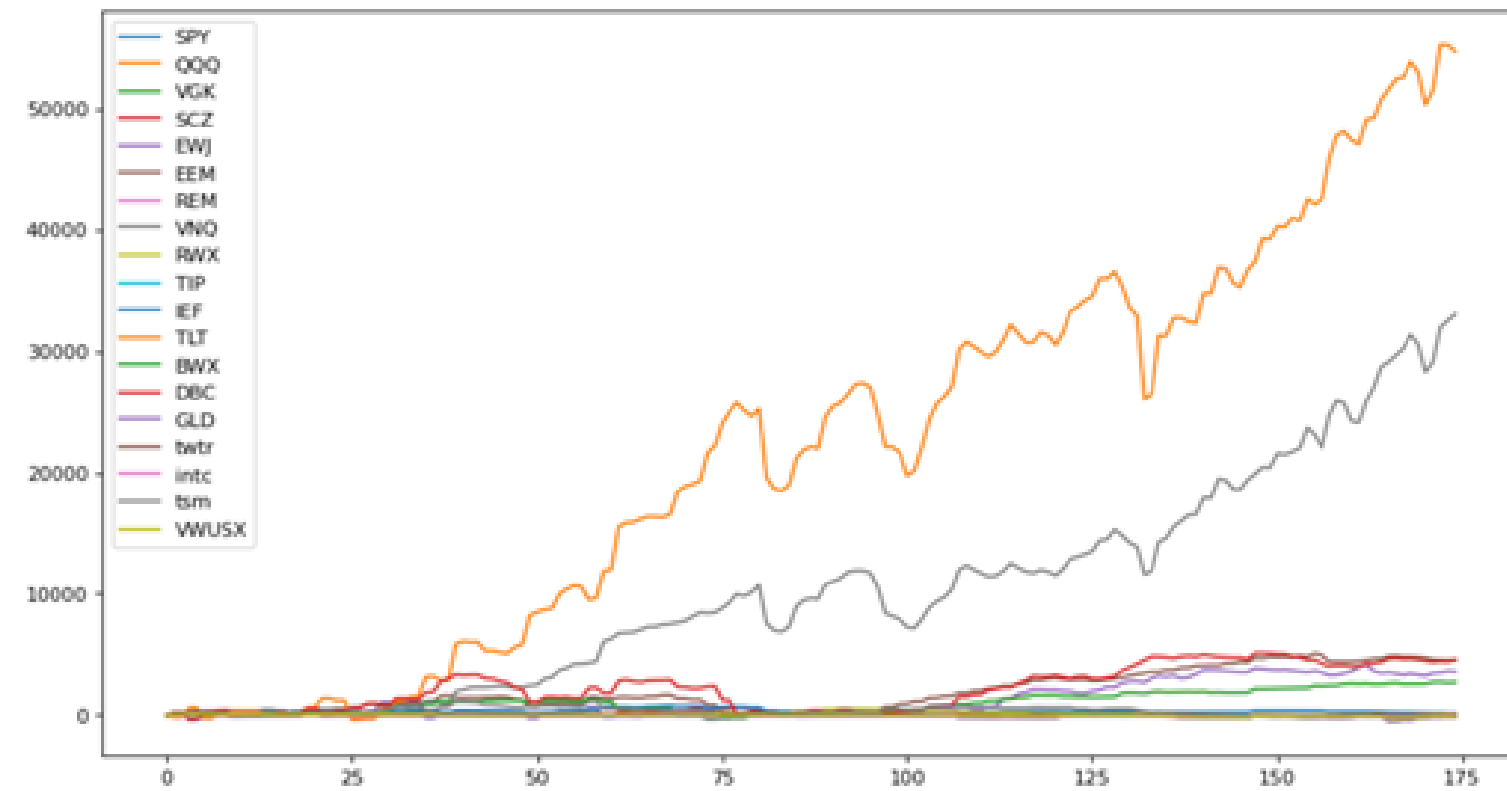
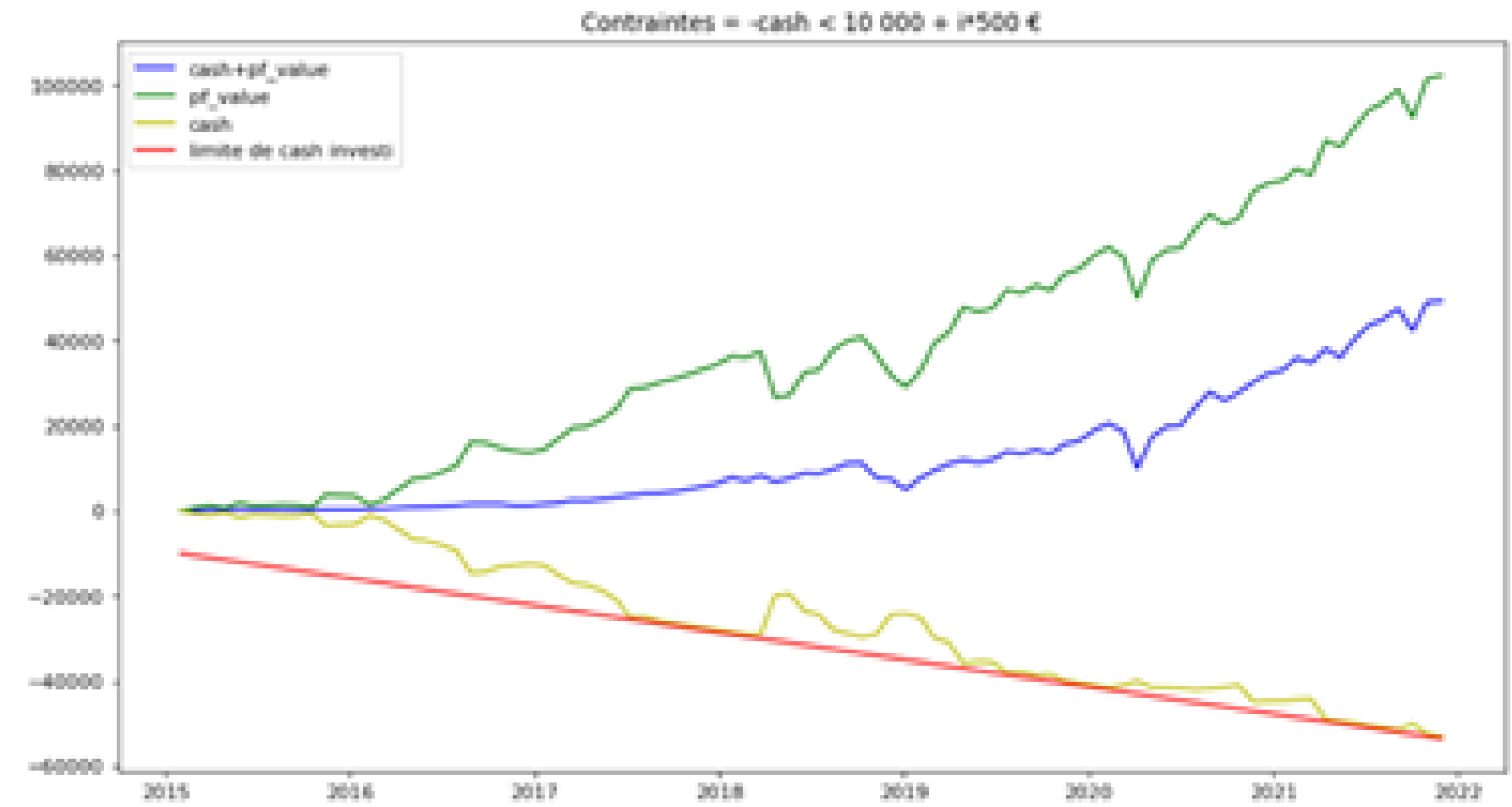


Contrainte sur les variations de positions

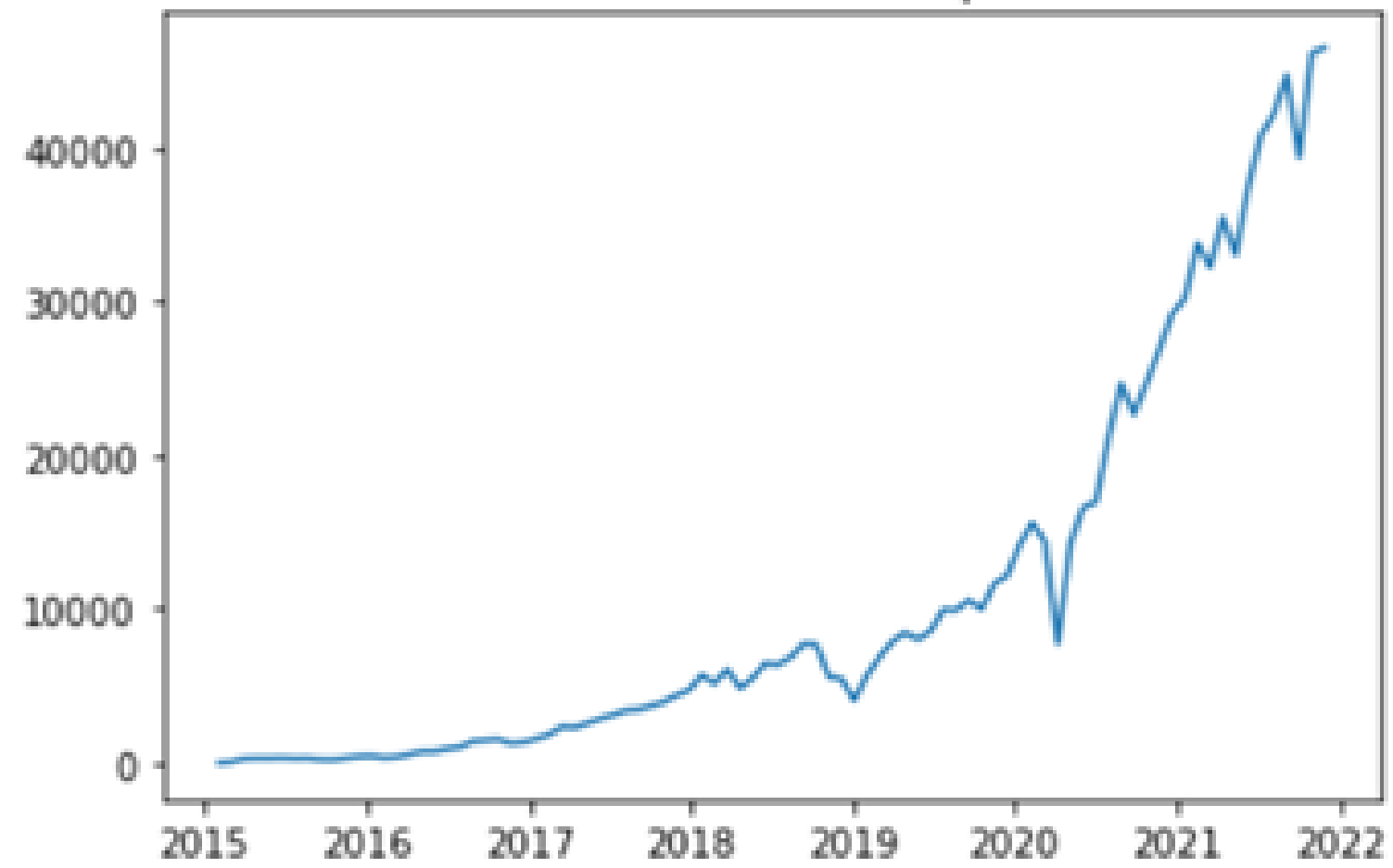


```
def cons4bis(x,pos,dPmax):  
    # pour chaque actif, on ne peut en changer (en notionnel) plus de dPmax% de la valeur déjà investie dans l'actif  
    return min([dPmax*abs(pos[i]) - abs(x[i]-pos[i]) for i in range(nb_actifs)])
```

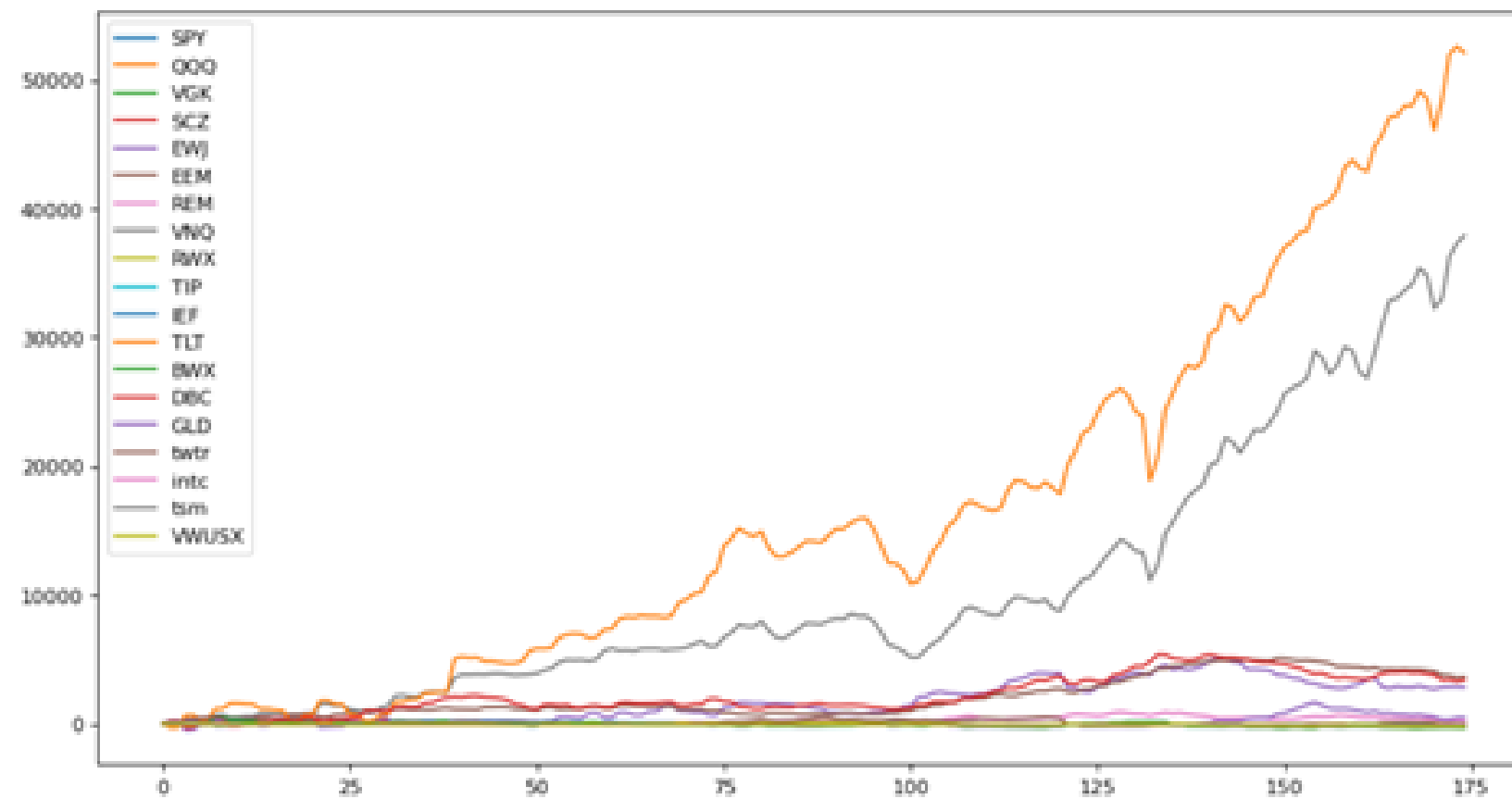
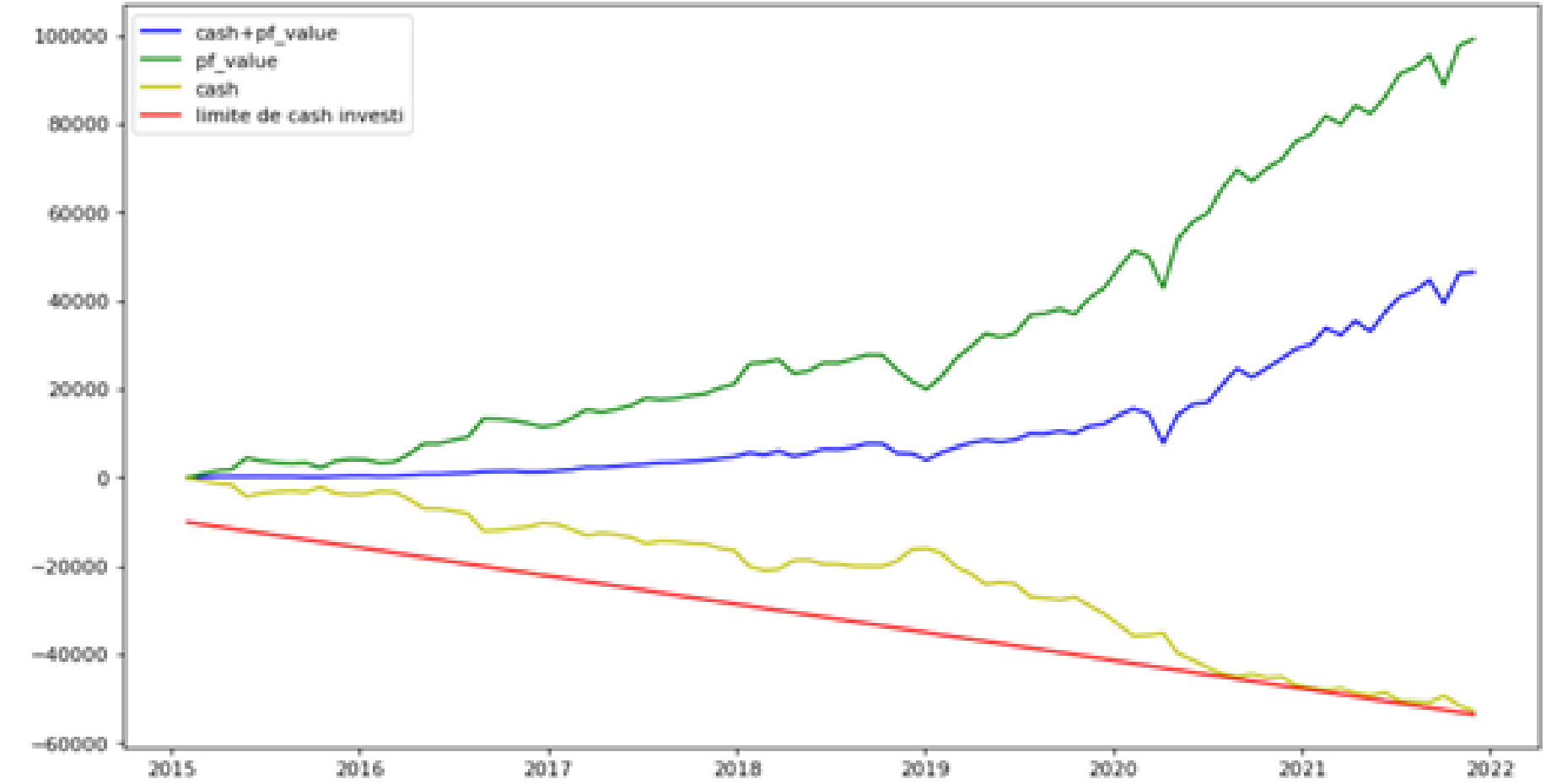
Evolution de la valeur du portfolio



Evolution de la valeur du portfolio

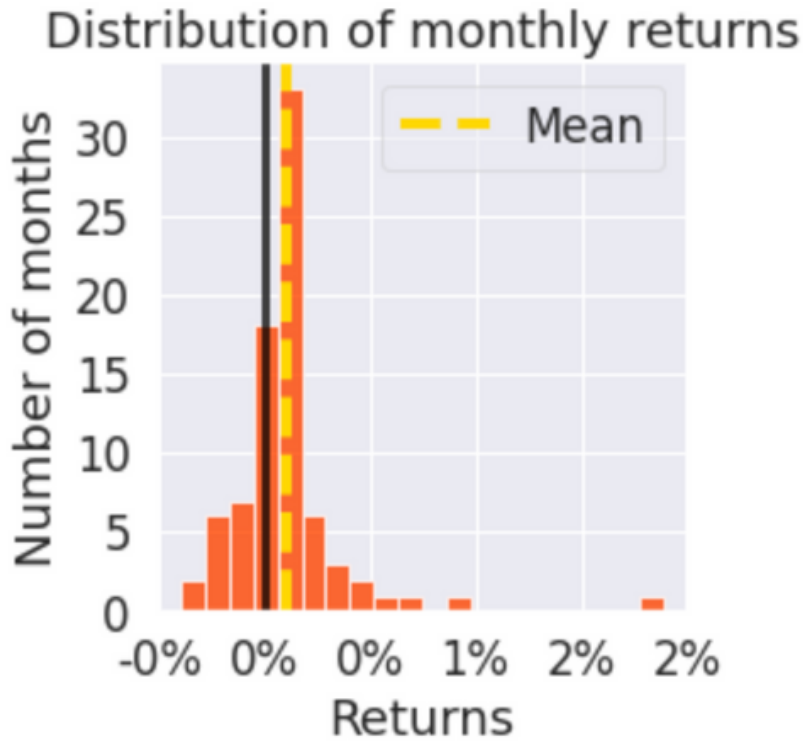
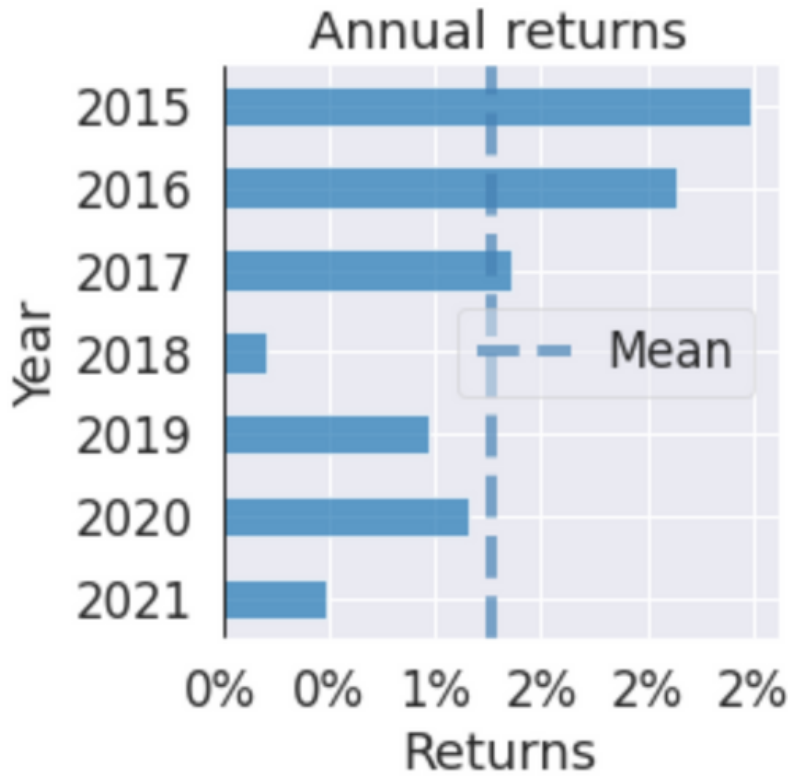
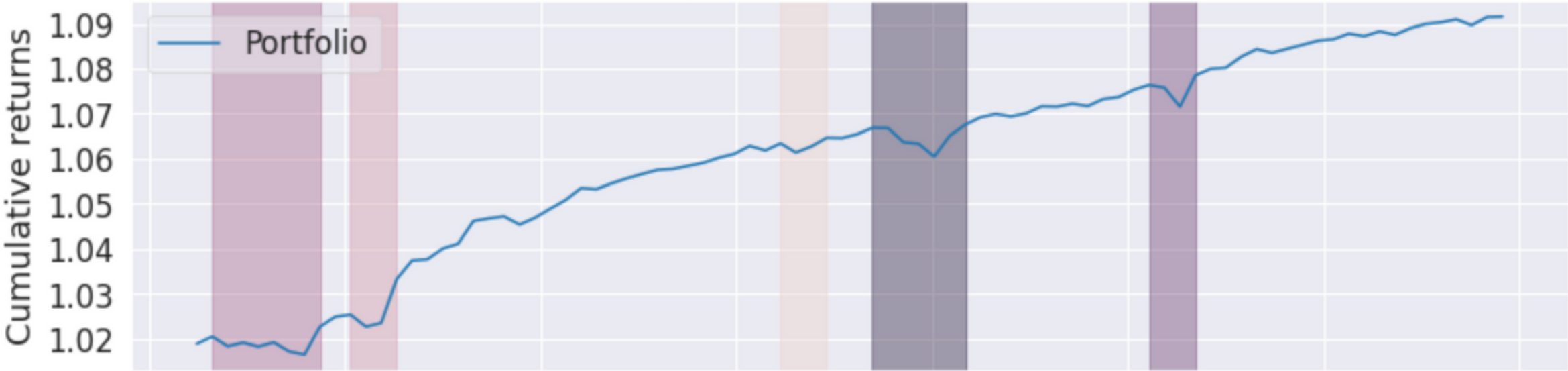


Contraintes = $-\text{cash} < 10\,000 + i \cdot 500 \text{ €}$



Pyfolio (alpha = 50 lambda = 0.05)

Top 5 drawdown periods



Start date 2015-03-31

End date 2021-11-30

Total months 4

Backtest

Annual return 29.266%

Cumulative returns 9.156%

Annual volatility 4.356%

Sharpe ratio 5.92

Calmar ratio 48.79

Stability 0.94

Max drawdown -0.6%

Omega ratio 4.15

Sortino ratio 19.60

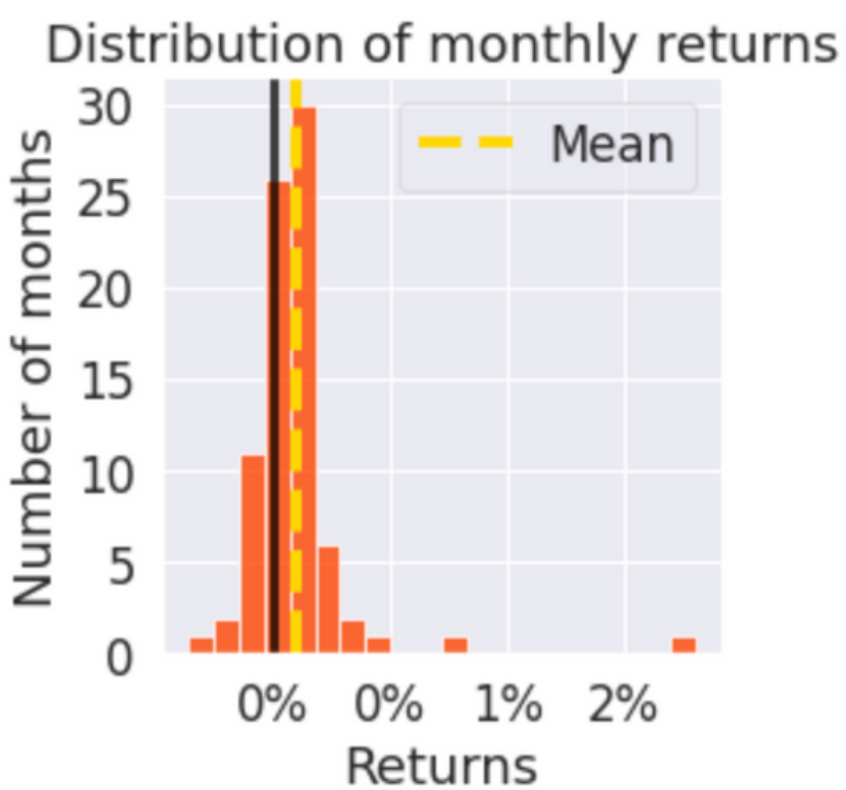
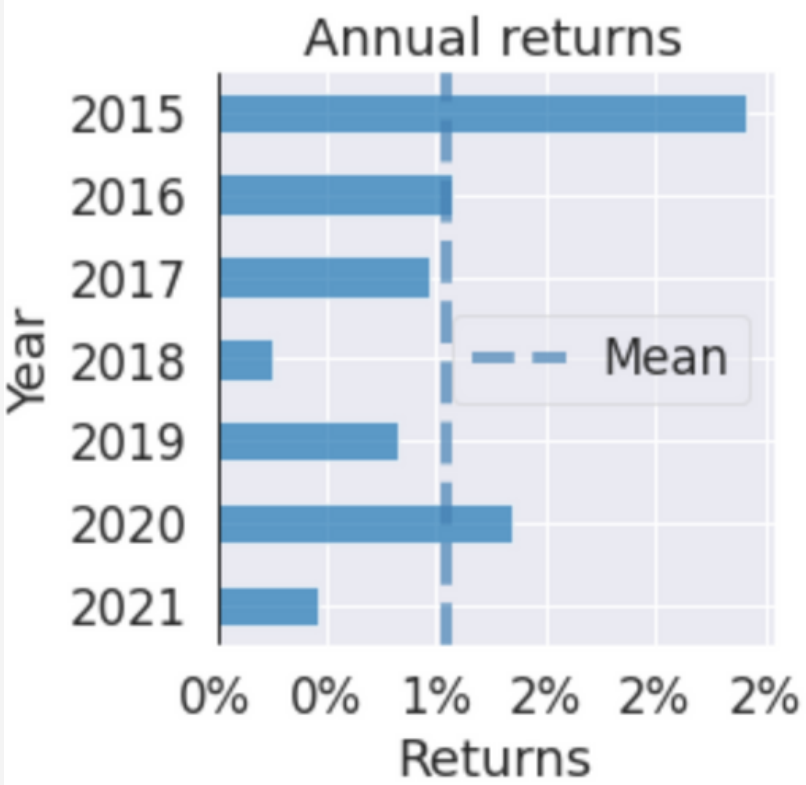
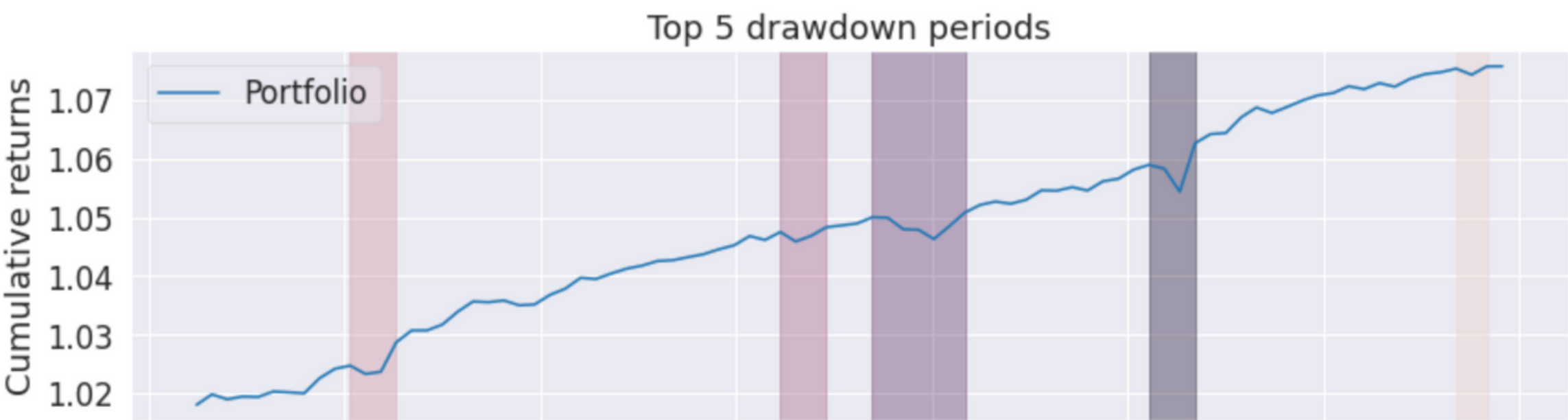
Skew 3.59

Kurtosis 20.16

Tail ratio 2.31

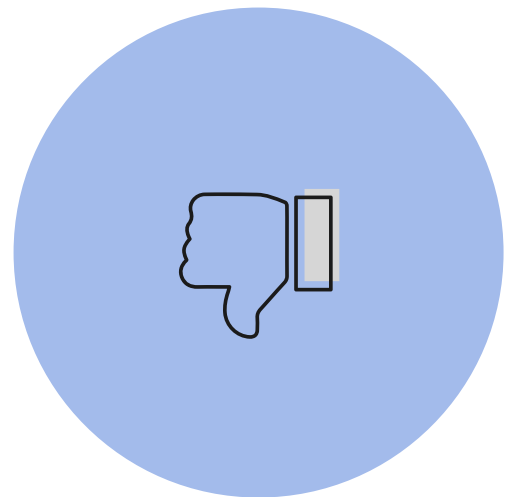
Daily value at risk -0.447%

Pyfolio (alpha = 10 lambda = 0.05)



Start date	2015-03-31
End date	2021-11-30
Total months	4
Backtest	
Annual return	23.857%
Cumulative returns	7.575%
Annual volatility	3.696%
Sharpe ratio	5.81
Calmar ratio	55.26
Stability	0.98
Max drawdown	-0.432%
Omega ratio	5.16
Sortino ratio	23.45
Skew	4.98
Kurtosis	33.68
Tail ratio	1.90
Daily value at risk	-0.38%

Ce qui n'a pas marché



Le respect des contraintes

Des erreurs dans l'algorithme de minimisation sont parfois apparues, liées au

- non respect de certaines contraintes
- absence de minimiseur qui respecte
- on obtient des résultats cohérents en simplifiant

SLSQP et trust-bounds

2 algorithmes d'optimisation qui traitent des contraintes:

- pas satisfaction, car
- trop lents
- les contraintes non respectées dans la solution proposée (SLSQP)

Pyfolio/zipline

Impossible de tirer le meilleur parti de pyfolio, notamment à cause de

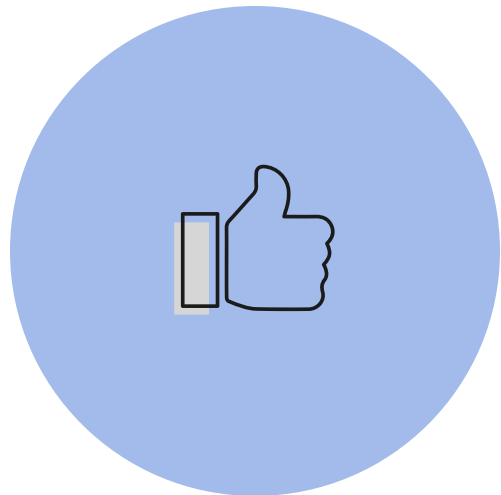
- la librairie zipline qui n'est plus mise à jour depuis la disparition de Quantopian,
- scraping difficile pour écrire la stratégie sous zipline.
- analyses avec pyfolio en lui donnant les returns calculés

Pycaret

Difficile à mettre en place de manière optimale et maîtrisée

- intervalle de temps limité,
- les prédictions obtenues semblaient correctes
- pas encore implémentées dans le minimiseur

Ce qui a bien marché



Scipy.optimize.minimize

- Simple d'utilisation,
- documentation complète
- variété intéressante de méthodes d'optimisations à essayer

Modulabilité

- algorithme facilement adaptable à tout les hyperparamètres
- lancer rapidement un grand nombre de simulation pour tester les stratégies

Rebalancement

- rapide à compréhension du rebalancement pour pouvoir calculer la valeur du portfolio à chaque instant ainsi que la valeur du cash

yfinance

- Import facile de toutes les données grâce à la bibliothèque

COBYLA

- Rapide
- efficace

Le résultat final

- portfolios avec rendements satisfaisants
- drawdowns corrects

Portfolio Management CentraleSupelec

Dash de contrôle

Market to compare performances to

☐ ^GSPC ☐ ^DJI ☐ ^IXIC ☐ ^FCHI

Stocks equity

☒ SPY ☒ QQQ ☒ VGK ☒ SCZ ☒ EWJ ☒ EEM

Stocks finance

☒ TIP ☒ IEF ☒ TLT ☒ BWX

Stocks comodity

☒ DBC ☒ GLD

Stocks real estate

☒ REM ☒ VNQ ☒ RWX

Cryptos

☐ BTC ☐ ETH ☐ AVAX ☐ USDT ☐ BNB

Others

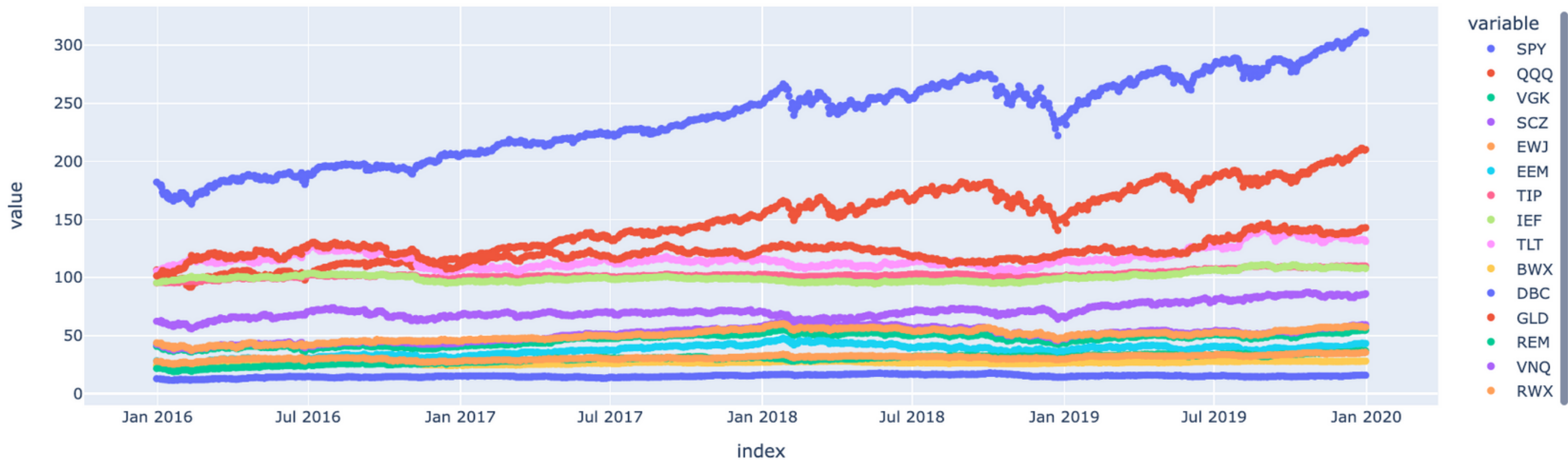
☐ TWTR ☐ INTC ☐ TSM ☐ VWUSX ☐ AMC

Range Simulation: Start Date / End Date



Duration of Rebalancing

☒ 1 month ☐ 3 months ☐ 5 months



Merci !

N'hésitez pas à nous contacter si vous avez des questions.



Lien du Github

https://github.com/Armand-Morin/Portfolio_Optimization_Volga-Technologies