



Analyse de l'algorithme de force brute

- L'algorithme de force brute génère toutes les combinaisons possibles d'actions et calcule le coût et le bénéfice pour chaque combinaison.
- Il sélectionne ensuite la combinaison qui respecte le budget et maximise le bénéfice.
- Complexité temporelle : $O(n * 2^n)$
- Complexité spatiale : $O(2^n)$
- Limites : Très inefficace pour un grand nombre d'actions en raison de l'explosion combinatoire.



Pseudocode de la solution optimisée

1. Initialiser une table dp de taille $(n+1) \times (\text{budget_max}+1)$ avec des zéros.
2. Pour chaque action i de 1 à n :
 - a. Pour chaque budget de 0 à budget_max :
 - i. Si le coût de l'action est inférieur ou égal au budget :
 - $dp[i][\text{budget}] = \max(dp[i-1][\text{budget}], dp[i-1][\text{budget} - \text{cost}] + \text{profit})$
 - ii. Sinon :
 - $dp[i][\text{budget}] = dp[i-1][\text{budget}]$
3. Retrouver les actions sélectionnées en retraçant la table dp .

Description de l'algorithme optimisé et ses limites

- L'algorithme optimisé utilise la programmation dynamique pour résoudre le problème du sac à dos (Knapsack). Il maximise le bénéfice tout en respectant le budget en utilisant une table pour stocker les résultats intermédiaires.
- Complexité temporelle : $O(n * \text{budget_max})$
- Complexité spatiale : $O(n * \text{budget_max})$
- Limites : Nécessite une quantité de mémoire proportionnelle au budget et au nombre d'actions. Peut ne pas être efficace pour des budgets très élevés ou un très grand nombre d'actions.



Comparaison des performances entre brute force et optimisé

Algorithme	Force brute	Optimisé
Temps d'exécution	3 secondes	Moins d'une seconde
Complexité temporelle	$O(n * 2^n)$	$O(n * \text{budget_max})$
Complexité spatiale	$O(2^n)$	$O(n * \text{budget_max})$

- L'algorithme optimisé est généralement préférable en raison de sa meilleure efficacité temporelle et spatiale.

Comparaison des Algorithmes d'Investissement

Les deux fichiers ont été nettoyés (suppression des prix négatifs ou nuls, filtrage des bénéfices excessifs (supérieurs à 40%)).

Analyse des résultats de l'algorithme
optimisé par rapport aux décisions
de Sienna

Comparaison des résultats - Dataset 1

	Résultat algorithme	Résultat Sienna
Actions sélectionnées	Share-NFKM, Share-PNBE, Share-LQXO, Share-IXFI, Share-PUDT, Share-QIID, Share-URDO, Share-OFWH, Share-LFGX, Share-TECI, Share-BDMW, Share-IZCN, Share-LGSU, Share-ODFR, Share-SJAK	Share-GRUT
Coût total	499,57 €	498,76 €
Bénéfice total	196,58 €	196,61 €

Comparaison des résultats - Dataset 2

	Résultat algorithme	Résultat Sienna
Actions sélectionnées	Share-ICHU, Share-RGUQ, Share-UUEZ, Share-MXQF, Share-UWVI, Share-ACFX, Share-URZP, Share-NGBD, Share-EGHA, Share-FXQO, Share-TWII, Share-RHIE, Share-GJHJ, Share-MZYQ, Share-PWAT	Share-ECAQ, Share-IXCI, Share-FWBE, Share-ZOFA, Share-PLLK, Share-YFVZ, Share-ANFX, Share-PATS, Share-NDKR, Share-ALIY, Share-JWGF, Share-JGTW, Share-FAPS, Share-VCAX, Share-LFXB, Share-DWSK, Share-XQII, Share-ROOM
Coût total	498,11 €	489,24 €
Bénéfice total	189,81 €	193,78 €