

Optimisation des Algorithmes d'Investissement



AlgoInvest&Trade

Présentation technique
Par : Julie Estevan

Analyse de l'algorithme de force brute

- Approche : explorer toutes les combinaisons possibles d'actions
- Complexité : $O(2^n)$
- Avantage : garantit la solution optimale
- Inconvénient : devient très lent dès que $n > 20$
- Exemple : avec 20 actions \rightarrow 1 048 576 combinaisons

Solution optimisée (Programmation dynamique)

Pseudocode (Knapsack 0/1)

- Créer une matrice $DP[n+1][Budget+1]$
- Pour chaque action i :
Pour chaque budget w :
 - Si $coût[i] \leq w \rightarrow$ choisir $\max(bénéfice + DP[i-1][w-coût], DP[i-1][w])$
 - Sinon $DP[i][w] = DP[i-1][w]$
- Reconstruire la solution optimale en retraçant les choix

Algorithme choisi et limites

Programmation dynamique (Knapsack 0/1)

Avantages :

- Temps d'exécution rapide ($O(n \times \text{Budget})$)
- Trouve la solution optimale

Limites :

- Consomme de la mémoire (matrice DP)

Cas limites :

- Actions dont le coût $>$ budget
- Données invalides (bénéfices négatifs, coûts nuls)

Comparaison des performances

Force brute :

- Complexité : $O(2^n)$
- Impraticable pour $n > 25$

Programmation dynamique :

- Complexité : $O(n \times \text{Budget})$
- Exemple avec $n=100$, $\text{Budget}=500 \rightarrow 50\,000$ opérations
- Résultats en moins d'une seconde

Comparaison des résultats – Dataset 1

Critère	Sienna	Algo Optimisé
Actions choisies	Share-GRUT	22 actions (KMTG, GHIZ, ...)
Coût total (€)	498.76	499.96
Profit total (€)	196.61	198.55
Respect du budget	Oui	Oui

Comparaison des résultats – Dataset 2

Critère	Sienna	Algo Optimisé
Nombre d'actions	17	20
Coût total (€)	489.24	499.92
Profit total (€)	193.78	197.96
Respect du budget	Oui	Oui

Conclusion de comparaison

- ✓ L'algorithme exploite mieux le budget en diversifiant les choix.
- ✓ Dans les deux datasets, il obtient un profit supérieur au portefeuille de Sienna.
- ✓ Limites : arrondis financiers, actions achetables uniquement en unités entières.