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 → 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] <= w → 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 × 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 × Budget)
- Exemple avec n=100, Budget=500 → 50 000 opérations
- Résultats en moins d’une seconde