AlgoInvest&Trade

Algorithmes et Analyse d’un portefeuille d’actions

1. Algorithme brute force
2. Algorithme optimisé
3. Analyses et comparaisons
4. Rapport d’exploration

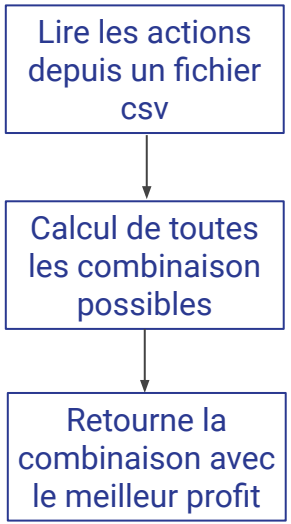
Algorithme Brute-Force

L’algorithme génères toutes les combinaisons possible présentes dans une liste d’actions.

Le programme nous renvoie la combinaisons rapportant le plus grand profit en fonction du budget maximal du client.

Avantages et cas limites

Organigramme



Avantages

Permet le calcul de toutes les possibilités

- Permet un profit maximum

Cas limites

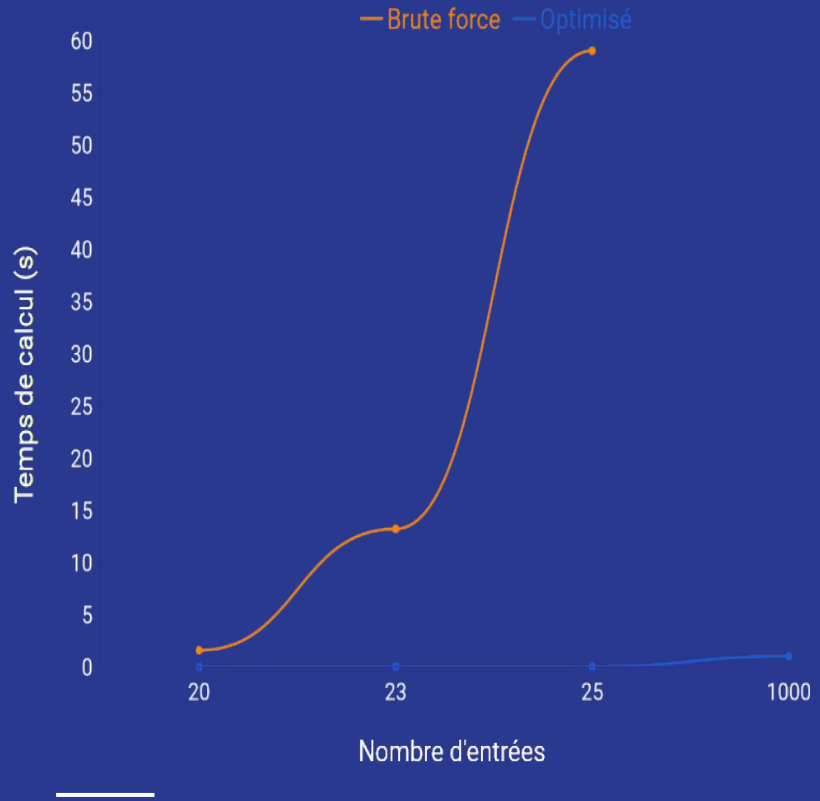
Temp de calcul exponentiel

- Ne permet pas le traitement de grand volume d’entrée

Notation BigO

Analyse du temps de traitement

- Brute-force = O(2n)
- Optimisé = O(n)



Analyse de la solution optimisé

Organigramme

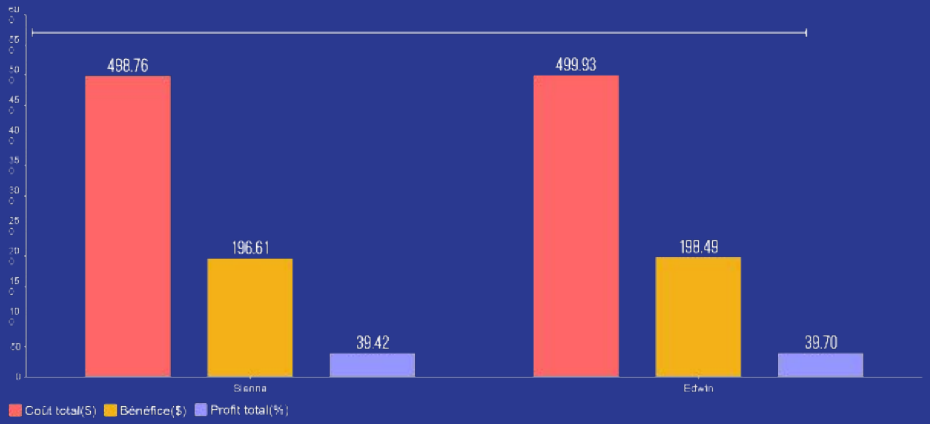
- ➔ **L'algorithme optimisé formate et trie les données**
  - ◆ **Retire les actions faussées ou trop peu rentable**
- ➔ **Calcul le ratio bénéfice/coût en fonction du coût et budget maximal**
- ➔ **Normalise les scores de 0 à 100**
  - ◆ **Permet le choix d'un score élevé pour optimiser le % de profit général**
- ➔ **Calcul en moins d'une seconde**

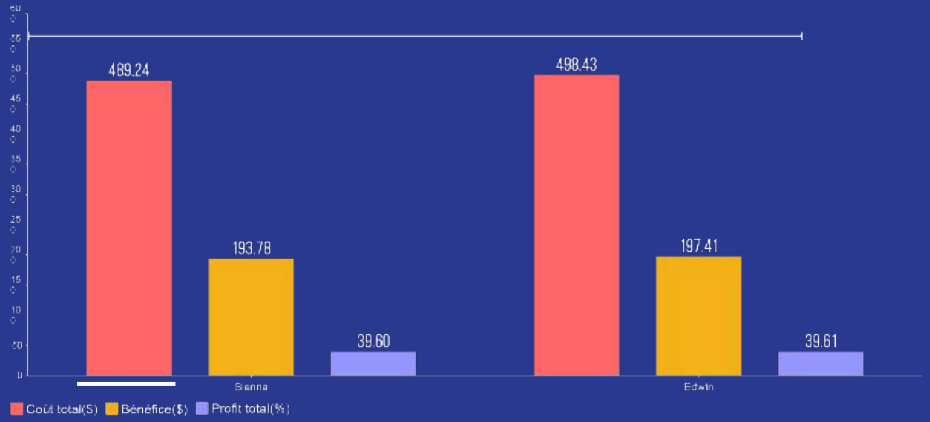
Comparaison Brute Force -> Optimisé

	Algorithme Brute Force	Algorithme optimisé
Avantages	Sélectionne la meilleure combinaison d'action possible	Permet le calcul de gros volume efficacement
Limites	Ne permet pas le calcul volume d'entrées modéré ou grand	Optimisation du meilleur choix laissé au profit de la vitesse
Complexité Temporelle	Exponentielle : 0(2**n)	Linéaire : Trie des données : O(nlogn) Traitement : O(n)
Complexité Spatiale	O(n) : Mais le temps de calcul peut poser des problèmes de mémoire.	O(n) : Stock les actions triées
Analyse	Simple à implémenter, ne permet cependant pas le traitement de gros volume. Inefficace et non viable pour de vrai calcul de données.	Offre des performance bien plus rapide, efficacité modulable, calcul de gros volume. Le choix de la version optimisé est évidente

Rapport d’exploration des données

Comparaison des résultats





1. Set de données n°1

- ➔ Sur un ensemble de 1000 entrées
- ➔ Ne peut contenir deux fois la même action
- ➔ Budget maximal de 500$

	Sienna	Algorithme optimisé
Coût total	498.76 $	499.93 $
Profit	196.61 $	198.49 $
Bénéfice total	39.42 %	39.70 %
Nbr d'actions	1	24
Analyse	Choix de l'action ayant le meilleur bénéfice sans prendre en compte une meilleure combinaison	Permet le choix de plusieurs petites actions ayant un meilleur rendement final

1. Set de données n°2

- ➔ Sur un ensemble de 1000 entrées
- ➔ Ne peut contenir deux fois la même actions
- ➔ Budget maximal de 500$

	Sienna	Algorithme optimisé
Coût total	489.24 $	498.43 $
Profit	193.78 $	197.41 $
Bénéfice total	39.6 %	39.61 %
Nbr d'actions	18	20
Analyse	Rendement égale à l'algorithme	Cherche à combler le budget au maximum

Optimisation grâce au score

- Il possible d'optimiser le pourcentage de bénéfice total en affinant la recherche avec un score élevé.
  Avec un score > de 99 les résultats s'améliore au détriment du budget fixé.

Data_set 2	Algorithme: score > 99
Coût total	255.48 $
Profit	101.84 $
Bénéfice total	39.86 %

➔ Malgré un budget revue en baisse le bénéfice total est plus élevé

➔ L'algorithme sépare aussi les erreurs du fichier d'origine pour traitement futur

Conclusion

- Permet le traitement immédiat de grands ensembles de données.
- Affinage pour des résultats optimisés.
- Évolutions sur le long terme