



# 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  $\text{budget\_max}$  :
    - i. Si le coût de l'action est inférieur ou égal au budget :
      - $dp[i][\text{budget}] = \max(\text{bénéfice action précédente pour budget, meilleure profit action} + \text{profit})$
    - ii. Sinon :
      - $dp[i][\text{budget}] = \text{bénéfice action précédente pour budget}$
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 beaucoup de mémoire en fonction du budget et du nombre d'actions, et peut être inefficace pour des montants de budgets très élevés ou un 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

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

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

# Comparaison des résultats - Dataset 1

|                       | Résultat algorithme   | Résultat Sienna |
|-----------------------|---|-----------------|
| Actions sélectionnées | Share-NFKM, Share-PNBE,<br>Share-LQXO, Share-IXFI,<br>Share-PUDT, Share-QIID,<br>Share-URDO, Share-OFWH,<br>Share-LFGX, Share-TECI,<br>Share-BDMW, Share-IZCN,<br>Share-LGSU, Share-ODFR,<br>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 €   |

# Rapport d'exploration de l'ensemble des données

| Ensemble de Données 1  | Ensemble de Données 2  |
|--|--|
| <p>Nombre d'entrées : 1001<br/>Colonnes : name, price, profit<br/>Types de données : name (object), price (float64), profit (float64)</p> <p>Statistiques descriptives :</p> <p>Prix :<br/>Moyenne : 24.41<br/>Écart-type : 19.65<br/>Minimum : -2.73<br/>25% : 15.17<br/>Médiane : 24.87<br/>75% : 33.55<br/>Maximum : 498.76</p> <p>Profit :<br/>Moyenne : 20.33<br/>Écart-type : 11.44<br/>Minimum : 0.00<br/>25% : 10.87<br/>Médiane : 20.07<br/>75% : 30.68<br/>Maximum : 39.98</p> | <p>Nombre d'entrées : 1000<br/>Colonnes : name, price, profit<br/>Types de données : name (object), price (float64), profit (float64)</p> <p>Statistiques descriptives :</p> <p>Prix :<br/>Moyenne : 12.61<br/>Écart-type : 16.24<br/>Minimum : -9.95<br/>25% : 0.00<br/>Médiane : 9.37<br/>75% : 27.16<br/>Maximum : 51.46</p> <p>Profit :<br/>Moyenne : 19.66<br/>Écart-type : 11.92<br/>Minimum : 0.15<br/>25% : 8.98<br/>Médiane : 19.81<br/>75% : 30.57<br/>Maximum : 39.98</p> |



# Rapport d'exploration de l'ensemble des données

- Ensemble de Données 2
- Aperçu des données :