AlgoInvest&Trade

Algorithmes d’optimisation d’investissement

OpenClassroom - Formation DA Python - P7

Objectif

- Maximiser le profit réalisé après deux ans par les clients d’AlgoInvest&Trade en sélectionnant la liste d’ actions la plus rentable parmi une sélection.

Contraintes

- L’investissement maximal est de 500€.
- Chaque action ne peut être achetée qu’une seule fois.
- Les actions sont indivisibles.

Algorithme de Force brute

- On génère toutes les combinaisons possibles pour les tailles de 1 jusqu’au nombre total d’actions du dataset.
- Pour chaque combinaison, on vérifie si le coût total est inférieur ou égal au budget.
- Si oui, le bénéfice de la combinaison est calculé.
- Si le bénéfice de la combinaison actuelle est supérieur à la précédente meilleure combinaison, on stocke la combinaison actuelle à la place.
- Puisque toutes les combinaisons sont testées, la solution trouvée est optimale.

Algorithme dynamique

Action (prix, ROI) \ Budget	0	1	2	3	4	5
Aucune action	0	0	0	0	0	0
Action 1 (2, 3)	0	0	3	3	3	3
Action 2 (3, 13)	0	0	3	13	13	16
Action 3 (3, 10)	0	0	3	13	13	16
Action 4 (2, 7)	0	0	7	13	13	20

Remarque : cette méthode nécessite des nombre entiers.
Si les valeurs contiennent des chiffres après la virgule, il faut les convertir en cents.

Le bénéfice optimal est la dernière entrée.

- Pour chaque entrée de la matrice, si le prix de l'action actuelle est inférieur au budget actuel, on calcule le meilleur bénéfice possible au budget actuel en incluant l'action actuelle : Bénéfice de l'action actuelle + bénéfice maximum précédent au coût : (budget actuel - prix de l'action actuelle)
- Si le bénéfice est meilleur avec l'action actuelle que sans, on enregistre ce résultat. Sinon on enregistre le résultat sans l'action actuelle (résultat de la ligne supérieure).
- La solution finale est optimale, et est ainsi construite progressivement, à partir des solutions aux sous-problèmes précédents (n=1, n=2 etc.) évitant ainsi de nombreux re-calculs.

Détermination de la meilleure combinaison

Action (prix, ROI) \ Budget	0	1	2	3	4	5
Aucune action	0	0	0	0	0	0
Action 1 (2, 3)	0	0	3	3	3	3
Action 2 (3, 13)	0	0	3	13	13	16
Action 3 (3, 10)	0	0	3	13	13	16
Action 4 (2, 7)	0	0	7	13	13	20

Budget restant = 0 : Fin

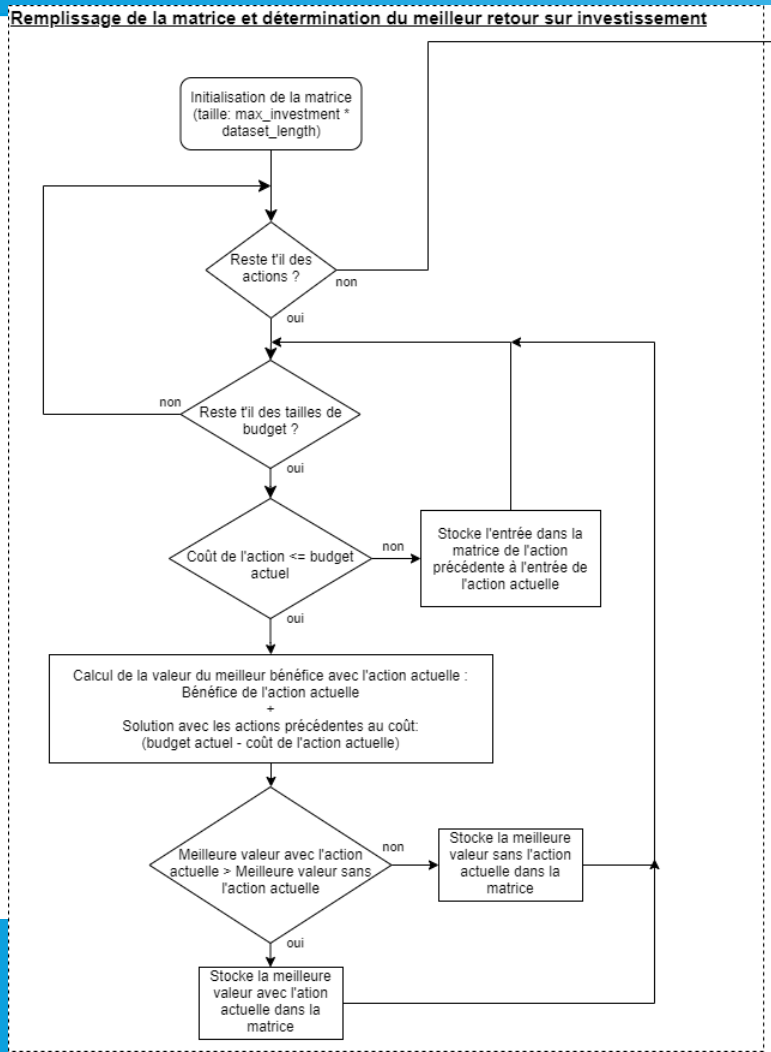
13 = 13 + 0 : Action incluse

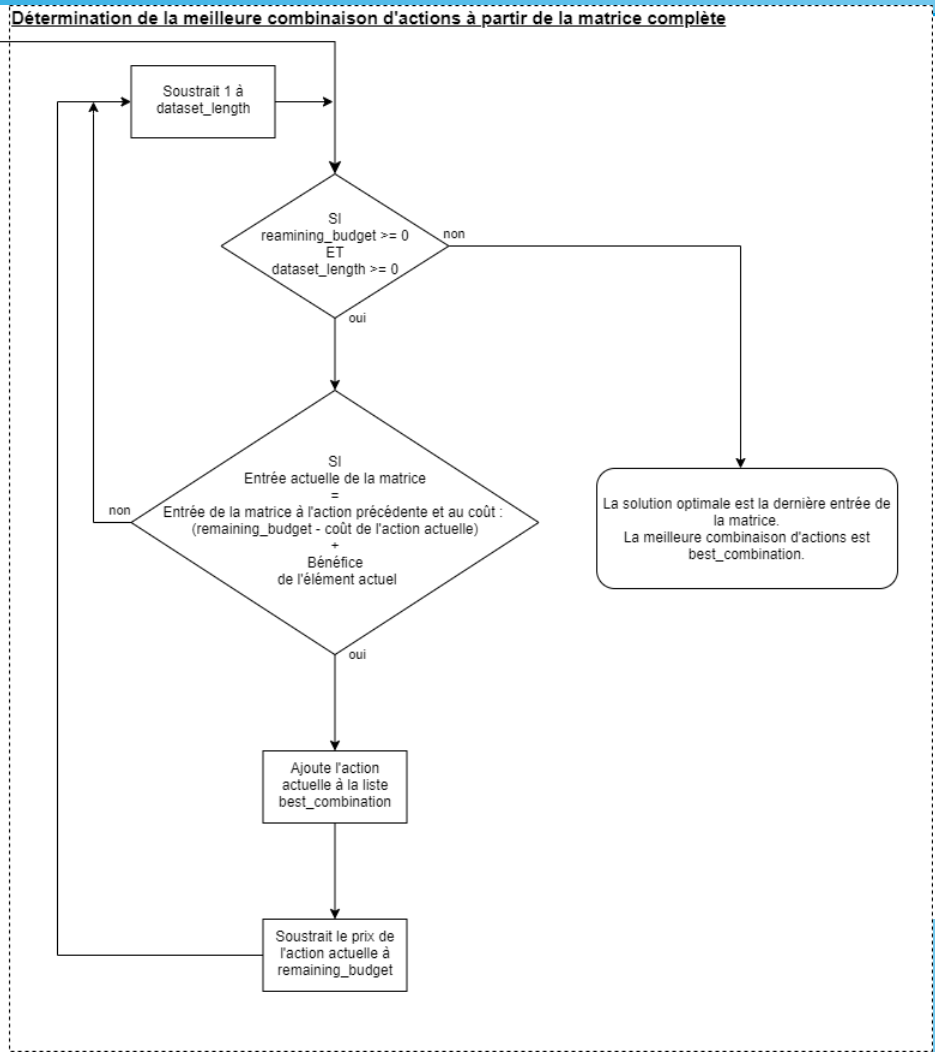
10 ≠ 13 + 0 : Action exclue

20 = 7 + 13 : Action incluse

- En partant de la fin (dernière action et budget max), pour chaque action, on vérifie si :
  entrée actuelle = bénéfice de l'action actuelle + entrée pour les actions précédente au budget : (budget actuel - prix de l'action actuelle)
- Si oui, cela signifie que cette action fait partie de la solution. On ajoute alors l'action actuelle à la combinaison finale et on soustrait son coût au budget restant. Sinon on ne fait rien.
- On passe à l'action précédente, jusqu'à épuisement des actions ou du budget restant.
- Solution finale de l'exemple : Actions 2 et 4 pour un coût de 5€ et un bénéfice de 20€.

Diagramme Algorithme optimisé





Analyse de performance des algorithmes

n = nombre total d’actions
i = budget en euros

Force brute

- Complexité temporelle : O(2^n)
- Complexité spatiale : O(n)

Dynamique

- Complexité temporelle : O(100i*n)
- Complexité spatiale : O(100i*n)

Limites de l’algorithme dynamique

- Bien que l’algorithme trouve la solution optimale, il peut être assez long pour un très grand nombre d’actions et/ou un budget important.
- On pourrait lui préférer un algorithme glouton qui ne donnera pas forcément la solution optimale, mais toujours au moins une bonne approximation, tout en étant plus économe en calcul.
- L’algorithme en l’état ne peut pas trouver la solution si l’on décide de modifier ou d’ajouter des contraintes (fractions d’actions sélectionnables, multiples critères de préférence…)

  Il faudra lui préférer une approche linéaire issue de la théorie des graphes.

Rapport d’exploration dataset1

Échantillon de 956 actions valides, investissement maximum de 500€

Algorithme	Sienna	Dynamique
Investissement	498,76€	499,50€
Nombre d’action(s)	1	22
Bénéfice	196,61€	198,54€
Liste d’action(s)	Share-GRUT	Share-XJMO, Share-KMTG, Share-MTLR Share-GTQK, Share-LRBZ, Share-WPLI, Share-GIAJ, Share-GHIZ, Share-ZSDE, Share-IFCP, Share-FKJW, Share-NHWA, Share-LPDM, Share-QQTU, Share-USSR, Share-EMOV, Share-LGWG, Share-QLMK, Share-SKKC, Share-UEZB, Share-KZBL, Share-MLGM

La solution de Sienna est sous-optimale. Elle n’inclut qu’une action coûtant presque l’investissement maximal avec un rendement très correct.
La solution dynamique, optimale dans sa conception, inclut plus d’actions moins chères mais avec un rendement légèrement supérieur. La différence de bénéfice reste inférieure à 1%.

Rapport d’exploration dataset2

Échantillon de 541 actions valides, investissement maximum de 500€

Algorithme	Sienna	Dynamique
Investissement	489,24€	499,90€
Nombre d’action(s)	18	20
Bénéfice	193,78€	197,96€
Liste d’action(s)	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	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, Share-SCWM, Share-LXZU

Là encore la solution de Sienna est sous-optimale.
On remarque que l’algorithme dynamique choisit les mêmes actions plus deux supplémentaires (pour un total de 10,66€) sans dépasser l’investissement maximum.
Le bénéfice est donc automatiquement supérieur, la différence dépassant les 2%.