

Rapport TP1

ALGORITHME GENETIQUE APPLIQUEE AU
KNAPSACK PROBLEM

OUBRAIM Walid | AI52

14 octobre 2024

1. Faites varier le nombre d'objets, leurs poids, la capacité maximale du sac et le nombre de générations. Que constatez-vous ? Expliquez les causes.

- Variation du nombre d'objets : L'augmentation du nombre d'objets rend l'espace de recherche plus complexe ne permettant pas à l'algorithme de trouver des solutions faisables donc arrivé à un certain nombre plus rien ne marche. En faisant diminuer ce nombre, on a moins de chance de maximiser les gains par rapport à la capacité donnée donc une fitness moins bonne, car il y a moins de possibilités à explorer.
- Modification des poids des objets : Des objets plus légers augmentent la diversité des combinaisons réalisables, ce qui permet à l'algorithme d'explorer davantage de solutions et d'atteindre une meilleure fitness, car plus d'objets peuvent être ajoutés au sac sans dépasser la capacité. Si les poids sont élevés, le sac atteint rapidement sa limite, ce qui réduit la qualité des solutions. Lorsque les objets ont des poids identiques, l'algorithme se concentre sur les objets ayant les valeurs les plus élevées, permettant ainsi de valider l'efficacité de la sélection.
- Capacité maximale du sac : En augmentant la capacité, l'algorithme a plus de marge pour inclure des objets de grande valeur, ce qui améliore les solutions obtenues. Toutefois, une capacité trop importante réduit la contrainte du problème, ce qui peut générer des solutions moins optimisées. À l'inverse, une capacité trop faible rend la sélection des objets plus difficile, menant à des solutions moins satisfaisantes.
- Nombre de générations : Un nombre élevé de générations permet à l'algorithme de converger vers des solutions de meilleure qualité en augmentant l'exploration de l'espace de recherche. Cependant, au-delà d'un certain point, les améliorations deviennent négligeables. Un nombre de générations réduit limite l'exploration, entraînant des solutions de moindre qualité.

2. Apporter les modifications nécessaires pour aboutir seulement à des solutions faisables, si elles existent. Voici les approches qu'il faut coder et comparer :

a. Mettre une fitness négative proportionnelle au dépassement de la capacité du sac

En cas de dépassement de la capacité maximale, la valeur de la fitness est ajustée pour refléter l'excès, calculée comme :

$$\text{Fitness} = \text{Capacité} - \text{Somme des poids}$$

b. Vérifier lors de la génération des solutions, du croisement et de la mutation que ces solutions respectent la capacité du sac.

Cette approche implique d'ajuster les solutions qui dépassent la capacité autorisée. Pour chaque individu de la population, si la somme des poids excède la capacité, les bits correspondant aux objets sélectionnés sont inversés aléatoirement jusqu'à ce que le poids total soit inférieur ou égal à la capacité.

Comparaison entre les deux approches :

Pour comparer les performances des deux méthodes, quatre instances du problème de scénarios différents ont été générées. La taille de la population est fixée à 8 et le nombre de générations à 150. Voici les quatre instances :

1. Instance de base : Cette instance servira de référence pour évaluer les performances de base de l'algorithme génétique.

Nombre d'objets : 10

Poids des objets : Générez des poids variés entre 1 kg et 15 kg.

Valeurs des objets : Générez des valeurs variées entre 50 € et 350 €.

Capacité maximale du sac : 30 kg

2. Instance avec beaucoup d'objets et capacité limitée : Cette instance teste la capacité de l'algorithme à gérer un grand nombre d'objets avec une capacité limitée.

Nombre d'objets : 40

Poids des objets : Générez des poids variés entre 1 kg et 15 kg.

Valeurs des objets : Générez des valeurs variées entre 50 € et 200 €.

Capacité maximale du sac : 20 kg

3. Instance avec objets légers et valeurs élevées : Cette instance teste la capacité de l'algorithme à gérer des objets légers avec des valeurs élevées.

Nombre d'objets : 25

Poids des objets : Générez des poids variés entre 1 kg et 5 kg.

Valeurs des objets : Générez des valeurs variées entre 100 € et 400 €.

Capacité maximale du sac : 30 kg

4. Instance avec objets de poids égaux et valeurs variées : Cette instance teste le comportement de l'algorithme dans une situation particulière pour évaluer l'impact de ces caractéristiques.

Nombre d'objets : 10

Poids des objets : Poids égal à 5 kg pour tous objets.

Valeurs des objets : Générez des valeurs variées entre 50 € et 350 €.

Capacité maximale du sac : 30 kg

1^{ère} approche

2^{ème} approche

Instance n°	Solution optimale	Temps de calcul (en s)	Solution optimale	Temps de calcul (en s)
1	Fitness: 1114 Nbr objets: 5 Poids: 24kg	0.127	Fitness: 1296 Nbr objets: 5 Poids: 28kg	0.165
2	Fitness: 1046 Nbr objets: 7 Poids: 30kg	0.113	Fitness: 1378 Nbr objets: 9 Poids: 29kg	0.197
3	Fitness: 4049 Nbr objets: 13 Poids: 30kg	0.127	Fitness: 4702 Nbr objets: 16 Poids: 30kg	0.148
4	Fitness: 1453 Nbr objets: 6 Poids: 30kg	0.113	Fitness: 1816 Nbr objets: 6 Poids: 30kg	0.1515

Les résultats montrent que la deuxième approche est plus efficace en termes de résultats par rapport à la première approche. La vérification des solutions générées, croisées et mutées pour garantir le respect de la capacité du sac implique certes un coût plus élevé en temps de calcul mais permet néanmoins d'avoir de meilleures solutions. Aussi, on peut voir la robustesse des deux approches dans des scénarios proposant un nombre d'objets élevés ou de valeurs élevées (2 et 3). De plus, le scénario 4 permet de voir réellement les meilleures combinaisons d'objets proposées par chaque approche et là aussi la 2^{ème} est la meilleure.

3. Adapter le programme à la gestion du portefeuille.

Pour adapter l'algorithme génétique à la gestion de portefeuille, les modifications suivantes ont été apportées :

- Formulation du problème: Chaque objet représente un titre boursier, avec la possibilité d'acheter plusieurs actions, sous réserve d'une limite de 20 % du

budget pour un même titre. La fitness mesure le rendement attendu. La fitness mesurera le rendement attendu plutôt que la valeur brute.

- Représentation des solutions : Les individus de la population représentent des portefeuilles, chaque gène correspondant au nombre d'actions d'un titre.
- Fonction de fitness adaptée : La fitness doit maintenant refléter le rendement attendu du portefeuille tout en respectant la contrainte de minimisation du risque. Si la solution dépasse le budget, une fitness nulle est affectée. Sinon, mettre la somme des actions avec une pénalité proportionnelle au dépassement de la contrainte risque (20% du budget) si elle n'est pas validée.
- Adaptation des opérateurs génétiques : Lors de la génération de solutions, une vérification a été ajoutée pour s'assurer que la somme des actions achetées respecte le budget total. En plus de ça, une vérification de la contrainte de minimisation du risque et corriger le faite de trouver plus d'un titre avec plusieurs actions ont été ajoutés après le croisement et la mutation. Le croisement et la mutation ont été ajustés pour garantir que les solutions restent faisables du point de vue du budget.

4. Choisir 4 instances des données du problème et stocker les dans des fichiers csv. Améliorer les performances de l'algorithme génétique sur ces quatre instances. Décrivez dans le rapport les améliorations réalisées et les motivations.

En s'inspirant de la réponse de la question 2, on peut définir 4 scénarios différents. Les paramètres fixés sont : La taille de la population à 8, le nombre de générations à 150 et le nombre max d'actions achetées à 20. Voici les quatre instances :

1. Instance de base

Nombre de titres : 10

Valeurs des actions : Générez des valeurs variées entre 50 € et 350 €.

Budget total : 1000 €

2. Instance avec beaucoup de titre et budget limitée

Nombre de titres : 40

Valeurs des objets : Générez des valeurs variées entre 50 € et 200 €.

Budget total : 2000 €

3. Instance 3 quelconque

Nombre de titres : 25

Valeurs des actions : Générez des valeurs variées entre 10 € et 100 €.

Budget totale : 2000 €

4. Instance 4 quelconque

Nombre de titres : 20

Valeurs des actions : Générez des valeurs variées entre 50 € et 350 €.

Budget total : 1000 €

Axes d'améliorations de l'algorithme génétique :

- Utiliser d'autres méthode de sélection, telles que la sélection par tournoi ou la sélection par roulette, pour influencer la diversité de la population et la rapidité de la convergence. Ici la méthode de sélection par tournoi a été implémentée.
- Taux de croisement et de mutation qui changent au fil des générations pour équilibrer l'exploration et l'exploitation.
- Interchanger les valeurs de deux objets lors de la mutation au lieu d'inverser qu'un seul.
- Augmenter le nombre de bit à muter : une mutation plus fréquente permet une exploration plus large de l'espace des solutions.
- Essayer d'autres méthodes de croisement : ici la méthode de croisement à 2 points a été implémentée.

Discussion des résultats :

Après amélioration

<i>Instance n°</i>	Fitness de la solution optimale	Temps de calcul (en s)
1	895	0.107
2	2000	0.233
3	1481	0.230
4	936	0.129

- Rajouter une autre méthode de croisement s'avère être plus efficace, malgré un léger coût de temps d'exécution.
- Nous pouvons tout de même remarquer que les améliorations nouvellement ajoutées ne sont pas forcément meilleures que les méthodes de base. En effet, les résultats ne montent pas une grande différence entre elles.