Projet MADMC

Ariana Carnielli

déc. 19, 2020

Created on Sun Dec 13 23 :44 :18 2020

Projet MADMC Sélection bi-objectifs avec coefficients intervalles

@author : Ariana Carnielli

projet_madmc.**f_i**(*y*, *I*)

Calcule la valeur de la fonction f_I sur le vecteur y avec l'intervalle I.

Paramètres

- **y** (*tuple(float, float)*) – Le vecteur qu'on veut appliquer la fonction f_I.
- **I** (*tuple(float, float)*) – L'intervalle [alpha_min, alpha_max] utilisé dans le calcul.

Renvoie _ – La valeur de la fonction f_i(y) dans l'intervalle I.

Type renvoyé float

projet_madmc.**gen_vecteur**(*n*, *m*)

Crée une liste de n vecteurs. Chaque vecteur est représenté comme un tuple de Python à 2 éléments où chaque élément a des valeurs tirées aléatoirement avec la loi normale d'espérance m et d'écart-type m/4.

Paramètres

- **n** (*int*) – La quantité de vecteurs créés.
- **m** (*int*) – L'espérance de la loi normale utilisée.
- **_** (*list((float, float))*) – Liste de taille n avec des tuples de floats de taille 2 représentant les vecteurs.

projet_madmc.**i_solver**(*list_vec*, *k*, *I*)

Implémente la procédure en deux temps pour déterminer l'image d'une solution minimax dans l'espace des objectifs. Fait comme décrit dans la partie 4 du projet en utilisant la I-dominance.

Paramètres

- **list_vec** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des coûts des objets.
- **k** (*int*) – Quantité d'objets à prend dans la solution.
- **I** (*tuple(float, float)*) – L'intervalle [alpha_min, alpha_max] utilisé dans le calcul.

Renvoie _ – Vecteur avec l'image d'une solution minimax dans l'espace des objectifs.

Type renvoyé tuple(float, float)

projet_madmc.**non_domine_i**(*list_vec*, *I*)

Détermine les vecteurs non I-dominés dans une liste de vecteurs. Réalise d'abord une transformation des vecteurs par la fonction pi, puis calcule les vecteurs Pareto non-dominés parmi ceux-ci et enfin transforme ces vecteurs à l'aide de la fonction pi-1 pour identifier les vecteurs non I-dominés.

Paramètres

- **list_vec** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des vecteurs.
- **I** (*tuple(float, float)*) – L'intervalle [alpha_min, alpha_max] utilisé dans le calcul.

Renvoie _ – liste des vecteurs non I-dominés.

Type renvoyé list((float, float))

projet_madmc.**non_domine_p**(*list_vec*)

Détermine les vecteurs Pareto non-dominés dans une liste de vecteurs. Réalise d'abord un tri lexicographique des vecteurs, puis un seul parcours de la liste obtenue pour identifier les vecteurs non-dominés.

Paramètres **list_vec** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des vecteurs.

Renvoie **res** – liste des vecteurs non dominés.

Type renvoyé list((float, float))

projet_madmc.**non_domine_p_naif**(*list_vec*)

Détermine les vecteurs Pareto non-dominés dans une liste de vecteurs. Procède avec des comparaisons par paires systématiques de vecteurs et retourne une liste avec les vecteurs non dominés.

Paramètres **list_vec** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des vecteurs.

Renvoie **res** – liste avec les vecteurs non dominés.

Type renvoyé list((float, float))

projet_madmc.**pareto_dyn**(*tab_couts*, *k*)

Utilise la programmation dynamique pour calculer l'image des sous-ensembles Pareto-optimaux de taille k d'un ensemble de taille n d'objets bi-valués.

Paramètres

— **tab_couts** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des coûts des objets.

— **k** (*int*) – Quantité d'objets à prend dans la solution.

Renvoie **_** – Liste des images des solutions Pareto non-dominés.

Type renvoyé list((float, float))

projet_madmc.**pareto_solver**(*list_vec*, *k*, *I*)

Implémente la procédure en deux temps pour déterminer l'image d'une solution minimax dans l'espace des objectifs. Fait comme décrit dans la partie 3 du projet en utilisant la dominance de Pareto.

Paramètres

— **list_vec** (*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des coûts des objets.

— **k** (*int*) – Quantité d'objets à prend dans la solution.

— **I** (*tuple(float, float)*) – L'intervalle [alpha_min, alpha_max] utilisé dans le calcul.

Renvoie **_** – Vecteur avec l'image d'une solution minimax dans l'espace des objectifs.

Type renvoyé tuple(float, float)

projet_madmc.**tester_temps_alpha**(*fonction*)

Implémente le test de temps de calcul démandé à la Question 12. Teste une fonction de calcul d'un image d'une solution minimax dans l'espace des objectifs passée en paramètre en créant 500 ensembles de vecteurs tirés aléatoirements pour chaque valeur de épsilon entre 0.025 et 0.5, avec un pas de 0.025, et en calculant le temps moyen de calcul pour chaque valeur d'épsilon.

Paramètres **fonction** (*Function*) – La fonction qu'on veut tester le temps d'exécution.

Renvoie **res** – Dictionnaire dont les clés sont les épsilons utilisés pour créer les intervalles I et les valeurs le temps moyen de calcul pour chaque épsilon.

Type renvoyé dict(int : float)

projet_madmc.**tester_temps_n**(*fonction*)

Implémente le test de temps de calcul démandé à la Question 5. Teste une fonction de calcul des points Pareto non-dominés passée en paramètre en créant 50 ensembles de vecteurs tirés aléatoirements pour chaque valeur de n entre 200 et 1000, avec un pas de 200, et en calculant le temps moyen de calcul pour chaque valeur de n.

Paramètres **fonction** (*Function*) – La fonction qu'on veut tester le temps d'exécution.

Renvoie **res** – Dictionnaire dont les clés sont les tailles utilisées dans la création des vecteurs et les valeurs le temps moyen de calcul pour chaque taille.

Type renvoyé dict(int : float)

projet_madmc.**vec_minimax**(*list_vec*, *I*)

Calcule un vecteur minimax dans une liste de vecteurs. Utilise la fonction f_i pour calcular la valeur F_i de chaque vecteur.

Paramètres

— **list_vec**(*list((float, float))*) – Liste avec des tuples de floats de taille 2 représentant des vecteurs.
— **I**(*tuple(float, float)*) – L'intervalle [alpha_min, alpha_max] utilisé dans le calcul.

Renvoie **vec_res** – Un vecteur minimax de la liste des vecteurs.

Type renvoyé tuple(float, float)