

Projet MADMC 2019–2020

Elicitation incrémentale et recherche locale pour le problème de sélection multi-objectifs

1. Introduction

On s'intéresse dans ce projet à la version suivante du problème de sélection multi-objectifs (objectifs à **maximiser**) :

Données :

- n objets,
- un entier k ,
- chaque objet i est valué par un vecteur c^i à p dimensions (c_1^i, \dots, c_p^i)

Solutions réalisables : tout sous-ensemble de k objets, caractérisé par un vecteur $x = (x_1, \dots, x_n)$ ($x_i = 1$ si l'objet i est sélectionné, 0 sinon).

But : déterminer une solution réalisable x maximisant une fonction d'agrégation paramétrée représentant les préférences du décideur sur les solutions. La fonction d'agrégation est de la forme $\Psi_\omega(y(x)) = \Psi_\omega(y_1(x), \dots, y_p(x))$, avec :

- ω : paramètre initialement inconnu parmi l'ensemble Ω des paramètres initialement admissibles ($\omega \in \Omega$)
- $y(x)$: évaluation d'une solution x donnée par $(\sum_{i=1}^n c_1^i x_i, \dots, \sum_{i=1}^n c_p^i x_i)$.

La fonction d'agrégation sera une fonction linéaire en ses paramètres, par exemple la somme pondérée, OWA ou l'intégrale de Choquet.

Le projet consiste à développer deux procédures de résolution du problème, basées sur l'élicitation incrémentale, et à comparer leur efficacité.

2. Une première procédure de résolution

Dans cette partie, nous allons mettre en oeuvre une procédure en deux temps pour déterminer la solution préférée du décideur : (1) déterminer une approximation des points non-dominés (au sens de Pareto), puis (2) déterminer la solution préférée du décideur parmi ceux-ci à l'aide d'une procédure d'élicitation incrémentale.

Pour la détermination de l'approximation des points non-dominés, vous utiliserez la recherche locale de Pareto (PLS) vue en cours : en partant d'une solution admissible générée aléatoirement, vous appliquerez une fonction de voisinage permettant de générer de nouvelles solutions admissibles. Vous lancerez de nouveau la fonction de voisinage à partir de toute nouvelle solution non-dominée trouvée, jusqu'à ce qu'il ne soit plus possible de générer de nouvelles solutions non-dominées avec la fonction de voisinage considérée (i.e. un optimum local est obtenu).

La fonction de voisinage sera un simple échange 1-1 : un objet est retiré de la sélection, un nouvel objet est ajouté. En partant d'une solution, tous les échanges 1-1 possibles seront considérés.

Ensuite, à partir de l'ensemble de solutions non-dominées obtenu, vous appliquerez une méthode d'aide à la décision par élicitation incrémentale des poids des critères, pour une fonction d'agrégation $\Psi_\omega(y(x))$ linéaire en ses paramètres (voir l'article numéro 1 joint (sections 3.1 et 3.2)).

On choisira un vecteur poids qui reste inconnu de la procédure d'élicitation pour simuler les réponses du décideur à des questions portant sur ses préférences. Pour cette partie, vous étudierez les deux points suivants :

- a) Evaluer à l'aide de plusieurs essais le nombre de questions nécessaires pour identifier la solution optimale pour le décideur (parmi les solutions générées en phase 1).
- b) Etudier comment varie le regret minimax en fonction du nombre de questions posées (pour calculer les regrets maximum, on utilisera le solveur Gurobi (<https://www.gurobi.com/>) qui est installé dans les salles en libre-service mais peut aussi être installé sur des machines personnelles en téléchargeant depuis une adresse de l'université).

3. Une deuxième procédure de résolution

La deuxième procédure de résolution est une procédure où élicitation incrémentale et recherche locale sont combinées afin de réduire le nombre de solutions générées et de pouvoir résoudre des problèmes de plus grande taille.

Le principe est le suivant : comme pour PLS, en partant d'une solution admissible générée aléatoirement, vous appliquerez une fonction de voisinage permettant de générer de nouvelles solutions admissibles. En revanche, contrairement à PLS, vous déterminerez la meilleure solution parmi l'ensemble des voisins et la solution courante (solution à partir de laquelle le voisinage a été appliqué) à l'aide d'une procédure d'élicitation incrémentale. Si la meilleure solution n'est pas la solution courante mais une solution dans son voisinage, vous relancerez la procédure à partir de cette solution voisine. Sinon la procédure s'arrête car un optimum local a été obtenu (voir l'article numéro 2 joint, section 3).

4. Données et comparaison des deux méthodes

Vous trouverez dans la section Projet du site Moodle un fichier de données nommé "Data". Ce fichier contient les évaluations selon 6 critères de 200 objets. Pour tester vos méthodes, vous pourrez utiliser des sous-ensembles de ces données, par exemple prendre uniquement les 20 premiers objets et les 3 premiers critères de façon à avoir un problème à $n=20$ objets et $p=3$ critères. La taille k de l'ensemble d'objets à sélectionner sera toujours prise égale à $\lfloor \frac{n}{2} \rfloor$.

Vous trouverez également sur le site les fronts de Pareto obtenus avec une méthode exacte pour quelques problèmes. Par exemple le fichier "PF_40_4.txt" contient l'ensemble des points Pareto non-dominés pour l'instance composée des 40 premiers objets, 4 premiers critères et $k=20$ (il y a 13015 points non-dominés). A l'aide de ces fichiers, vous mesurerez la qualité des approximations obtenues par PLS, en terme de pourcentage de points non-dominés trouvés.

Pour l'élicitation incrémentale, vous testerez en priorité comme fonction d'agrégation la somme pondérée, ensuite éventuellement OWA avec poids décroissants et l'intégrale de Choquet avec capacité super-modulaire (les conditions imposées visent à favoriser les vecteurs de performances équilibrés).

Vous comparerez expérimentalement la première et la seconde procédure de résolution en termes de :

- temps de calcul,

- erreur par rapport à la solution optimale du décideur,
- nombre de questions posées.

Vous ferez la moyenne des résultats obtenus pour au moins 20 jeux de paramètres différents de la fonction d'agrégation.

Notez aussi que la méthode PLS nécessite de nombreuses mises à jour d'ensemble de solutions non-dominées. Il est donc opportun d'utiliser des structures de données adéquates pour la mise à jour, telles que ND-Tree ou Quad-Tree (qui seront vues durant le cours du 3 décembre, dont les transparents sont disponibles dans la section Projet du site Moodle).

5. Organisation et dates

Le travail est à effectuer par binôme. Le choix du langage de programmation est libre. Toutes les possibilités de visualisation sont les bienvenues (bien que facultatives).

Les projets doivent être soumis le **dimanche 19 janvier 2020** au plus tard sur le lien de soumission du site Moodle. Votre livraison sera constituée d'une archive **zip** qui comportera les sources du programme, un fichier README détaillant comment compiler et exécuter le programme, et un rapport (un fichier au format **pdf**) comportant une description synthétique des procédures implantées et des résultats des différents tests numériques menés. Le plan du rapport suivra le plan du sujet.

Une **soutenance** avec transparents est prévue lors de la séance du 21 janvier.