Projet : Élicitation incrémentale et recherche locale pour le problème de sélection multi-objectifs

MU5IN254 - Modèles et algorithmes pour la décision multicritère et collective

Réalisé par : Madina Traoré

SORBONNE UNIVERSITÉ

Sommaire

Présentation du problème de sélection multi-objectifs

Présentation du problème de sélection multi-objectifs

Données: n objets, un entier k d'objets à sélectionner, chaque objet i est valué par un vecteur c^i à p dimensions $(c^i_1, ..., c^i_p)$

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

Objectif : Déterminer une solution réalisable x maximisant une fonction d'agrégation paramétrée représentant les préférences du Décideur sur les solutions.

Fonction d'agrégation $\psi_\omega(y(x)) = \psi_\omega(y_1(x), ..., y_p(x))$, 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 $(\sum_{i=1}^{n} c^1_i x_i, ..., \sum_{i=1}^{n} c^p_i x_i)$.

Première procédure de résolution

Première procédure de résolution

- Déterminer une approximation des points Pareto non-dominés en appliquant une recherche locale de Pareto

- Déterminer la solution préférée du Décideur parmi ceux-ci à l'aide d'une procédure d'élicitation incrémentale

Première procédure de résolution

Recherche locale de Pareto

- On génère aléatoirement une solution admissible x
- On génère l'ensemble de ses voisins
- On détermine les solutions non-dominées parmi x et ses voisins
- Tant que l'on trouve de nouvelles solutions non-dominées, on détermine, pour chacune d'elle, ses voisins et on regarde s'il existe de nouvelles solutions non-dominées parmi elle et ses voisins

Première procédure de résolution

Élicitation incrémentale

- On dispose d'un ensemble $\mathcal{X}$ de solutions
- On cherche à déterminer les coefficients d'un vecteur de poids ω qui devra représenter au mieux les préférences du Décideur
- Pour cela on lui pose des questions du type "Quelle solution préférez-vous entre x_1 et x_2 ?" ($x_1 \succcurlyeq x_2$?)
- S'il répond qu'il préfère x_1 à x_2 ($x_1 \succcurlyeq x_2$), on restreint l'ensemble Ω de poids possibles aux poids ω tels que $\psi_\omega(y(x_1)) \geq \psi_\omega(y(x_2))$

ψ : fonction d'aggrégation linéaire en ses paramètres (somme pondérée ou OWA par exemple)

Deuxième procédure de résolution

Deuxième procédure de résolution

Recherche locale interactive (ILS)

1. On génère aléatoirement une solution admissible x
2. On génère l'ensemble N des voisins de x
3. On détermine la meilleure solution $x^* \in N \cup \{x\}$ en effectuant une élicitation incrémentale. Si $x^* \in N$, on recommence à partir de l'étape 2 en générant cette fois-ci les voisins de x^*. Sinon, la procédure s'arrête car un optimum local (x^*) a été trouvé

Utilisation d’un ND-Tree

Utilisation d’un ND-Tree

- La recherche locale de Pareto, utilisées dans les deux procédures, nécessite de nombreuses mises à jour d’ensembles de solutions non-dominées

- La structure ND-Tree permet d’effectuer ces mises à jour de manière efficace en s’appuyant notamment sur des approximations du point nadir et du point idéal de chaque noeud de l’arbre

Démonstration du code

Étude et comparaison des procédures de résolution

Étude et comparaison des procédures de résolution

Pourcentage de points non-dominés trouvés par PLS et efficacité du ND-Tree implémenté

		PLS		PLS avec ND-Tree	
n	p	Temps	Pareto trouvés	Temps	Pareto trouvés
10	2	0.0372s	100%	0.0117s	100%
10	3	0.1470s	100%	0.0951s	100%
10	4	0.3807s	100%	0.1277s	100%
10	5	1.7604s	100%	0.3794s	100%
10	6	2.8176s	100%	0.9507s	100%
20	2	1.4673s	100%	0.4090s	100%

Tableau 1: Résultats obtenus pour 20 exécutions de chaque méthode (**PLS()** et **PLS_nd_tree()**) pour chaque jeu de paramètres et ψ la somme pondérée

Étude et comparaison des procédures de résolution

Première procédure de résolution : évolution du regret minimax en fonction du nombre de questions posées

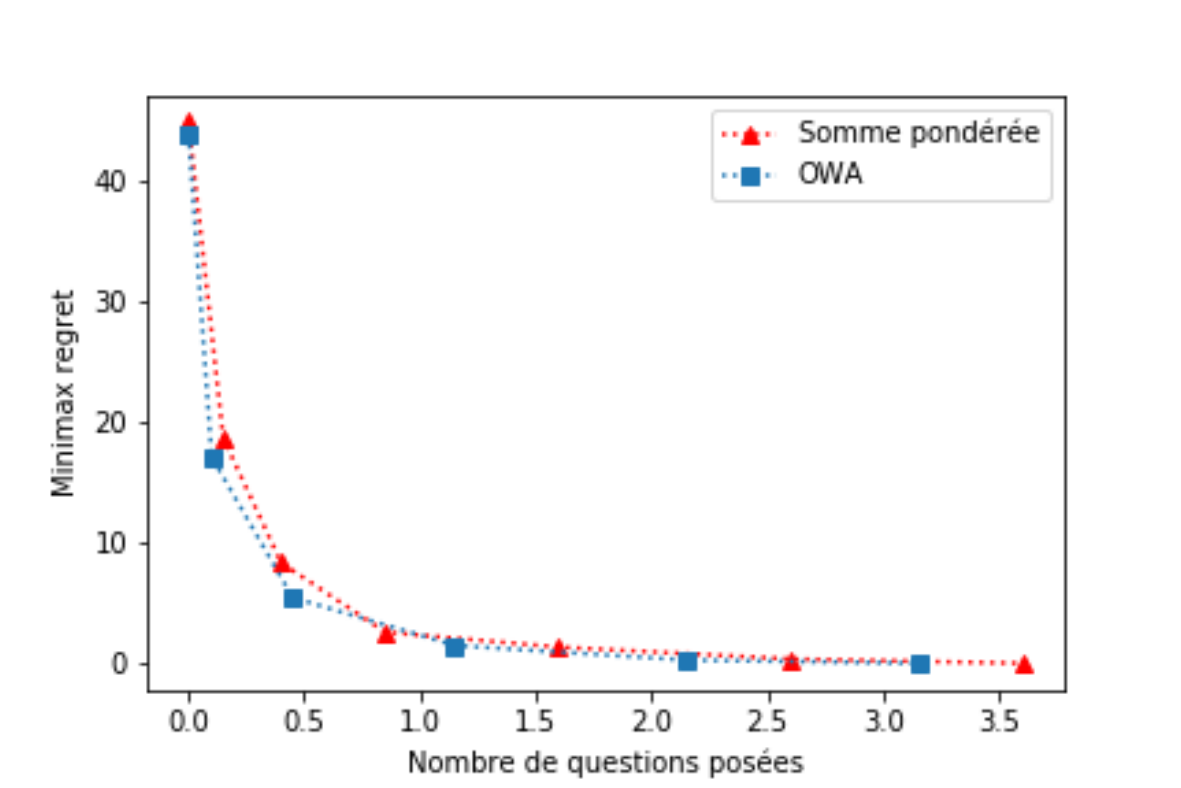

Figure 1: Évolution du regret minimax en fonction du nombre de questions posées ($n = 20$, $p = 2$, moyenne sur 20 jeux de poids différents)

Étude et comparaison des procédures de résolution

Comparaison des deux procédures de résolution en terme de temps de calcul, d'erreur par rapport à la solution optimale du Décideur et de nombre de questions posées

		Procédure 1			Procédure 2		
n	p	Temps	Erreur	Queries	Temps	Erreur	Queries
10	2	0.0178s	0	1	0.0212s	0	2.8
10	3	0.1328s	0	3.5	0.0462s	0	6.45
10	4	1.5034s	0	8.6	0.3035s	0	12.7
10	5	10.0967s	0	10.1	1.7184s	0	19.4
20	2	1.4029s	0	3.55	0.0653s	0	6.75

Tableau 2: Résultats obtenus en appliquant les deux procédures de résolution sur une partie des données (moyenne sur 20 jeux de poids différents pour chaque jeu de paramètre)

Étude et comparaison des procédures de résolution

Comparaison des deux procédures de résolution en terme de temps de calcul, d'erreur par rapport à la solution optimale du Décideur et de nombre de questions posées

		Procédure 2		
n	p	Temps	Erreur	Queries
30	2	0.2038s	0	6.35
40	2	0.5446s	0	8.1
50	2	2.1057s	0	8.7
60	2	3.3249s	0	9.95

Tableau 3: Résultats obtenus en appliquant la $2^{ème}$ procédure de résolution sur une partie des données (moyenne sur 20 jeux de poids différents pour chaque jeu de paramètre)

Étude et comparaison des procédures de résolution

Conclusion

- La structure ND-Tree permet d'effectuer une recherche locale de Pareto au moins deux fois plus rapide
- La deuxième procédure de résolution est globalement plus rapide que la première et permet de traiter des instances beaucoup plus grandes mais nécessite de poser plus de questions au Décideur
- En situation réelle, on pourrait préférer la première procédure de résolution si l'on souhaite déranger le moins possible le Décideur et la deuxième si l'on souhaite recommander un sous-ensemble d'articles parmi un très grand ensemble d'articles

Références

- Nawal Benabbou, Christophe Gonzales, Patrice Perny, Paolo Viappiani. *Minimax Regret Approachesfor Preference Elicitation with Rank-Dependent Aggregators*. EURO Journal on Decision Processes, 2015, 3 (1-2), pp.29-64.

- Nawal Benabbou, Cassandre Leroy, Thibaut Lust, Patrice Perny. *Combining Local Search and Elicitation for Multi-Objective Combinatorial Optimization*. ADT 2019 - 6th International Conference onAlgorithmic Decision Theory, Oct 2019, Durham, NC, United States. hal-02170910

- Andrzej Jaszkiewicz, Thibaut Lust. *ND-Tree-based update : a Fast Algorithm for the Dynamic Non-Dominance Problem*. 2018. hal-01900840

Merci pour votre attention