

Aide au choix d'une solution optimale sur un front Pareto à l'aide de méthodes de segmentation.

Ilyas RAHHALI

Directeurs : Charles Audet, Jonathan Jalbert

Polytechnique Montreal
Gerad

Montréal, 8 avril 2020

- 1 Introduction
- 2 Problèmes biobjectifs et Front Pareto
- 3 Algorithmes de segmentation
- 4 Méthodologie proposée
- 5 Exemple pratique

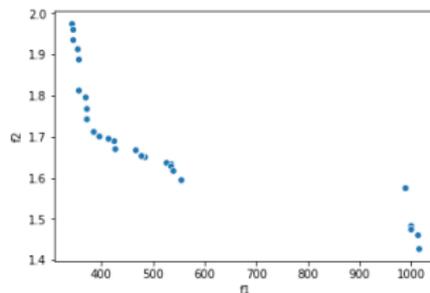


Figure 1 – Solutions optimales d'un problème biobjectif

- Choisir une solution peut être difficile
- On doit vérifier les caractéristiques des solutions une par une

Comment choisir une solution qui vérifie les critères de l'ingénieur ?

Problème biobjectif [1]

Un problème biobjectif se formule de la manière suivante :

$$\min_{x \in \Omega} F(x) = (f_1(x), f_2(x)) \quad (1)$$

Points dominés [1]

Soient $x, y \in \Omega$

On dit que x domine y si et seulement si $f_i(x) \leq f_i(y)$ pour $i \in \{1, 2\}$ avec au moins l'une des inégalités doit être stricte

On écrit alors $x \prec y$

Points non dominés ou Pareto optimaux [1]

Soit $x \in \Omega$

x est dit non dominé ou Pareto optimal si et seulement si $\nexists y \in \Omega$ tel que $y \prec x$

Définition de segmentation

Apprentissage non supervisé ou segmentation

- Catégorie de l'apprentissage machine qui tente de découvrir une structure latente à la distribution d'un jeu de données
- Permet d'identifier des groupements de points
- Un groupement est un ensemble de points regroupés à partir de certaines similarités

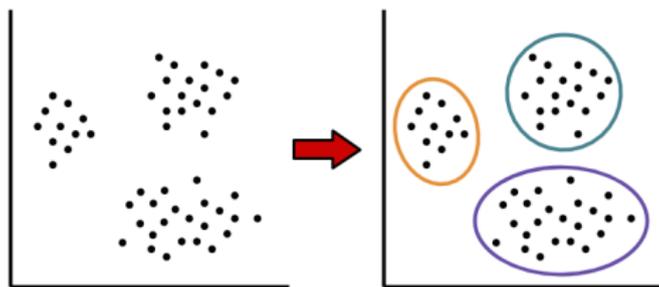


Figure 2 – Exemple de segmentation [2]

kmeans [3]

- Présuppose un nombre de groupements prédéfini K
- Algorithme assignant chaque point au centroïde le plus proche itérativement
- Le but est de minimiser la variance au sein de chaque groupement notée s

Algorithme 1 Kmeans

- 1: *Trouver un bon nombre de groupements K*
 - 2: *Initialiser K centroïdes $\{c_1, \dots, c_K\}$*
 - 3: **faire**
 - 4: $s = \sum_{k=1}^K \sum_{i=1}^N (x_i - c_k)^2$
 - 5: *Assigner chaque point x_i au centroïde c_k le plus proche*
 - 6: *Mettre à jour les centroïdes*
 - 7: **tant que** *Changement de centroïdes*
-

Meanshift [4]

- Aussi appelé algorithme de recherche de mode
- Considère l'espace de paramètres comme une fonction de densité de probabilité dont les points qui lui sont fournis sont considérés comme échantillons.
- Considère que s'il existe des groupements dans les données, ils correspondent aux modes ou maxima locaux de cette fonction de densité.
- Pas besoin de spécifier un nombre de groupements à l'avance
- Retourne un groupement pour chaque mode trouvé

Algorithme 2 Meanshift

- 1: *Sélectionner une fenêtre F*
 - 2: **pour** chaque point p du jeu de données **faire**
 - 3: **tant que** pas encore convergé **faire**
 - 4: *Calculer la moyenne des points dans une fenêtre F autour du point p*
 - 5: *Déplacer la fenêtre F pour la centrer sur la nouvelle moyenne*
 - 6: **fin tant que**
 - 7: **fin pour**
-

HDBscan [5]

- Extension de l'algorithme par densité DBscan
- Fait peu d'hypothèses sur la structure des données
- Plutôt que de chercher des groupements d'une certaine forme, il cherche les régions denses de l'espace
- Excellent pour les données bruitées
- Très sensible aux hyperparamètres

Algorithme 3 HDBscan

- 1: *Transformer l'espace suivant la densité*
 - 2: *Construire un arbre couvrant minimal à partir du graphe pondéré*
 - 3: *Construire une hiérarchie de segmentation à partir des composants connectés*
 - 4: *Condenser cette hiérarchie en se basant sur la taille de groupement minimale*
 - 5: *Extraire les groupements stables de l'arbre condensé*
-

Algorithme spectral [6]

- Algorithme n'utilisant pas de mesure de distance mais plutôt la matrice de similarité lui octroyant une robustesse en haute dimension
- Se base sur les valeurs propres de la matrice de similarité pour réduire la dimensionnalité des données avant de performer la segmentation

Algorithme 4 Segmentation spectrale [7]

- 1: *Construire le graphe de similarité. Soit W sa matrice d'adjascence pondérée*
 - 2: *Calculer le laplacien normalisé L*
 - 3: *Calculer les k premier vecteurs propres de L u_1, \dots, u_k de L (k étant le nombre de groupements)*
 - 4: *Construire la matrice T à partir des u_i comme colonnes puis normaliser ses lignes suivant la norme 1*
 - 5: *Effectuer une segmentation par k means sur les lignes de la matrice T*
-

Démarche suivie

- Génération et préparation des données
- Choix d'un algorithme de segmentation
- Présentation des groupements et leurs caractéristiques
- Choix de représentant par groupement

Génération et préparation des données

- Récupérer la cache de l'algorithme d'optimisation biobjectif
- Normaliser les données
- Extraire les points non dominés noté P
- Étendre P en y ajoutant les points à un certain rayon r de chacun des points dans l'espace (f_1, f_2)

Génération et préparation des données

Soit C la cache du problème biobjectif définie comme suit :

$$C = \{(x, f(x), c(x)) \in \mathbf{R}^n \times \mathbf{R}^2 \times \mathbf{R}^m : x \in \{x^1, x^2, \dots, x^p\}\}$$

où x^1, \dots, x^p sont les points visités par l'algorithme.

Algorithme 5 Extraction des points non dominés

Entrée: *cache* : C

```
1: fonction EXTRACTPARETO
2:    $P = \emptyset \leftarrow$  Front Pareto
3:   pour  $x \in C$  faire
4:     si  $x$  est non dominé dans  $C$  et  $x \in \Omega$  alors
5:        $P \leftarrow P \cup \{x\}$ 
6:     fin si
7:   fin pour
8:   retourne  $P$ 
9: fin fonction
```

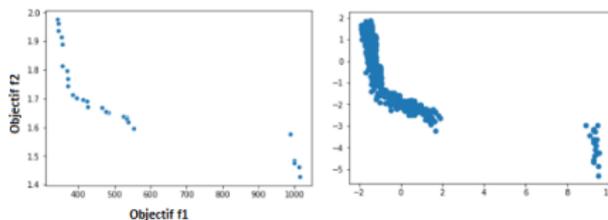


Figure 3 – Relaxation du Front Pareto

Algorithme 6 Relaxation du front Pareto

Entrée: cache C , pareto P , rayon r

```
1: fonction RELAXPARETO
2:    $R = \emptyset \leftarrow$  front relâché
3:   pour  $x$  dans  $P$  faire
4:     pour  $y$  dans  $C \setminus \{x\}$  faire
5:       si  $y \in B_r(F(x))$  alors
6:          $R \leftarrow R \cup \{y\}$ 
7:       fin si
8:     fin pour
9:   fin pour
10:  retourne  $R$ 
11: fin fonction
```

Choix d'un algorithme de segmentation

- Comparer la performance des 4 algorithmes de segmentation présentés précédemment
- En choisir un qui performe bien dans un cas général
- Vérifier sur un problème réel

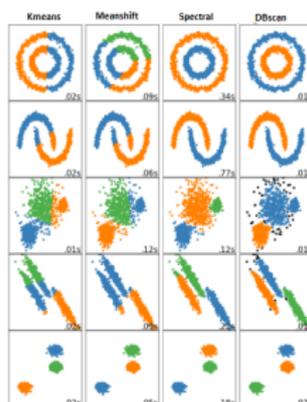


Figure 4 – Comparaison des 4 algos [8]

Choix d'un représentant par groupement

Deux méthodes sont proposées, l'utilisateur pourra choisir selon ses besoins

- Choix du point pareto le plus proche du centroïde de chaque groupement

Algorithme 7 choix de représentant

Entrée: Ensemble de groupements : $G = \{G_k\}, k \in \{1..p\}$

1: **fonction** HEURISTIQUE 1

2: **pour** $G_k \in G$ **faire**

3: $c_k = \left(\frac{1}{|G_k|} \sum_{x \in G_k} f_1(x), \frac{1}{|G_k|} \sum_{x \in G_k} f_2(x) \right)$ dans l'espace (f_1, f_2)

4: Choisir $x_{repr_k} = \min_{x \in P} d(x - c_k)$

5: **fin pour**

6: **fin fonction**

- Construire une fonction de coût à partir des critères de l'utilisateur qui sera utilisée pour trier ces points, ou effectuer une optimisation mono-objectif dessus

Définition du problème

On souhaite synthétiser un fluide caloporteur de température de fusion et d'activité de solution solides minimaux à partir de huit sels. Ce problème peut donc être reformulé sous forme de problème d'optimisation comme suit :

$$\min_{x \in \Omega} F(x) = (f_1(x), f_2(x))$$

$$\text{avec } F : \mathbf{R}^8 \longrightarrow \mathbf{R}^2 \text{ et } x \in \Omega \subseteq \mathbf{R}^8$$

Exemple pratique

Lecture des données :

	LiCl	NaCl	KCl	CaCl_2	MgCl_2	MnCl_2	NiCl_2	CoCl_2	f1	f2
0	33.0	54.0	66.0	49.0	6.0	28.0	9.0	66.0	413.2337	1.111307
1	90.0	85.0	81.0	54.0	93.0	66.0	100.0	16.0	564.5982	0.880445
2	15.0	39.0	51.0	21.0	54.0	61.0	91.0	18.0	658.5661	0.808576
3	66.0	24.0	83.0	35.0	70.0	30.0	19.0	91.0	472.5585	1.086090
4	23.0	20.0	54.0	77.0	65.0	77.0	47.0	57.0	615.7641	0.845349

Figure 5 – Aperçu des variables du problème

Normalisation des données :

	x1	x2	x3	x4	x5	x6	x7	x8	f1	f2
0	0.106109	0.173633	0.212219	0.157556	0.019293	0.090032	0.028939	0.212219	0.396616	0.806089
1	0.153846	0.145299	0.138462	0.092308	0.158974	0.112821	0.170940	0.027350	0.541894	0.638632
2	0.042857	0.111429	0.145714	0.060000	0.154286	0.174286	0.260000	0.051429	0.632083	0.586502
3	0.157895	0.057416	0.198565	0.083732	0.167464	0.071770	0.045455	0.217703	0.453555	0.787798
4	0.054762	0.047619	0.128571	0.183333	0.154762	0.183333	0.111905	0.135714	0.591002	0.613176

Figure 6 – Normalisation des données

Distribution des solutions et extraction des points Pareto

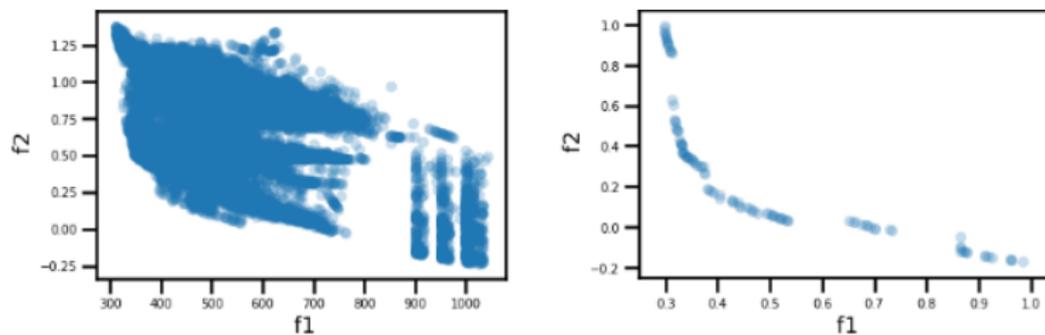


Figure 7 – Solutions dans l'espace (f_1, f_2) (gauche), Points Pareto (droite)

Relaxation du front Pareto :

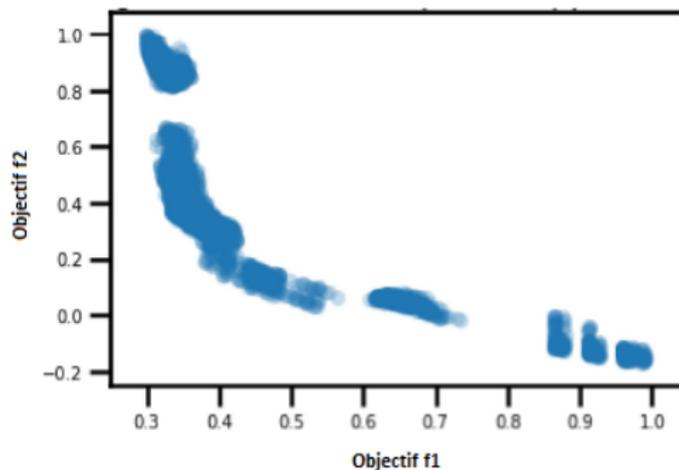


Figure 8 – Points approximés autour des points non dominés

Exemple pratique

Application des algorithmes de segmentation :

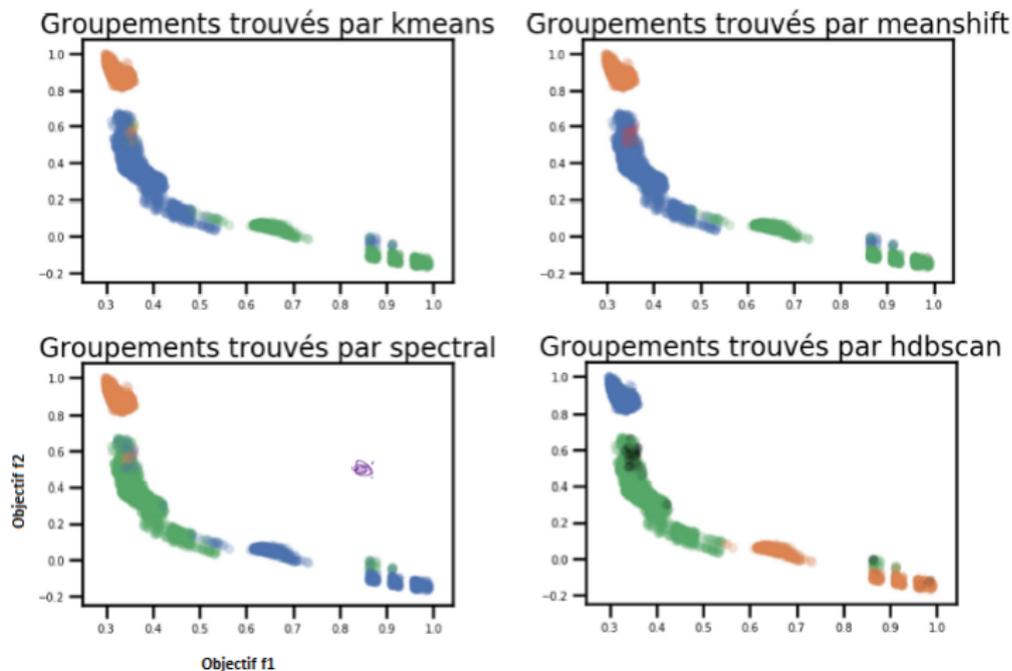


Figure 9 – Groupements trouvés par chaque algorithme

Sélection d'un représentant par groupement :

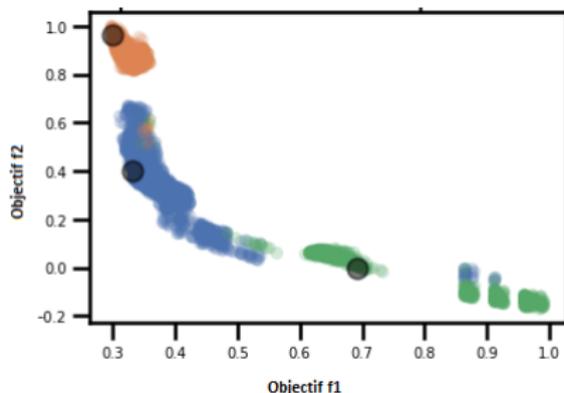


Figure 10 – Représentants par groupement

References I



C.Audet and W. Hare (2017)
Biobjective Optimization
Derivative-free and blackbox optimization
Springer, pp247–262.



Stanford university
<https://thedishonscience.stanford.edu/posts/clustering-breast-cancer-data/images/unsupervised-learning-example.png>



MacKay David (2003)
Chapter 20. An Example Inference Task : Clustering
Information theory, inference and learning algorithms
Cambridge university press, pp284–292.



K. Fukunaga, L. Hostetler (1975)
The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition
IEEE Transactions on Information Theory vol 21 no. 1 pp32–40



Campello R.J.G.B., Moulavi D., Sander J. (2013)
Density-Based Clustering Based on Hierarchical Density Estimates.
Advances in Knowledge Discovery and Data Mining.



Ng, Andrew Y and Jordan, Michael I and Weiss, Yair (2002).

On spectral clustering : analysis and an algorithm
Advances in Neural Information Processing Systems.



Ulrike von Luxburg (2007).

A Tutorial on Spectral Clustering
Springler, Statistics and Computing, 17 (4), 2007



Sklearn library

<https://scikit-learn.org/stable/modules/clustering.html>