

Implémentation C++ générique et comparaison expérimentale d'algorithmes de sélection de points dans un front de Pareto

Habib KHTEIRA

LERIA – Université d'Angers

Soutenance de stage M1 Informatique

Plan

1 Introduction

2 Critères de sélection

- p-dispersion : Formules
- Hypervolume : Illustration et formule

3 État de l'art

- énumération exhaustive
- Programmation dynamique
- Algorithme PMm
- Algorithme PMSN

4 Analyse Expérimentale

- Nombre de solutions optimales
- Compatibilité entre critères
- Robustesse de MSN

5 Conclusion

- Optimisation multiobjectif : plusieurs critères à minimiser.
- Front de Pareto : ensemble des solutions non dominées.
- Problème : trop de points dans le front.
- Objectif : sélectionner un sous-ensemble représentatif.

Introduction aux critères p-dispersion

- Les critères **p-dispersion** mesurent la dispersion géométrique d'un sous-ensemble S de p points sélectionnés parmi n sur un front de Pareto.
- Les variantes classiques comme **PMm**, **PMS**, **PMSm**, **PMmS** sont indépendantes de la dimension : elles s'appliquent à tout front de Pareto en $\mathbb{R}^d$.
- Ces critères ont été initialement étudiés pour la localisation d'infrastructures, notamment dans *E. Erkut S. Neuman, Comparison of Four Models for Dispersing Facilities* [4].
- Le critère **PMSN** est une variante spécifique aux fronts **bidimensionnels ordonnés**. Il exploite la structure croissante du front et a été introduit dans le papier *N. Dupin, Polynomial algorithms for p-dispersion problems in a planar Pareto Front* [1].

p-dispersion : définitions mathématiques (1/2)

- **Max-Min (PMm)** : maximise la distance minimale entre chaque paire de points sélectionnés.

$$\max_{S \subset E, |S|=p} \min_{i < j} d(x_i, x_j)$$

- **Max-Sum (PMS)** : maximise la somme totale des distances entre les points sélectionnés.

$$\max_{S \subset E, |S|=p} \sum_{i < j} d(x_i, x_j)^\alpha$$

- **Max-Sum-Min (PMSm)** : maximise la somme des distances entre chaque point sélectionné et le point sélectionné le plus proche.

$$\max_{S \subset E, |S|=p} \sum_{i=1}^p \min_{j \neq i} d(x_i, x_j)^\alpha$$

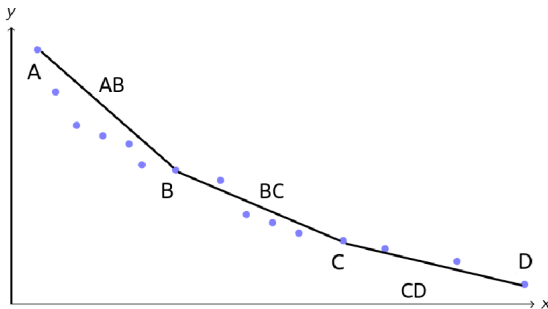
- **Max-Min-Sum (PMmS)** : qui maximise la somme minimale des distances entre chaque point sélectionné et les autres points sélectionnés.

$$\max_{S \subseteq E, |S|=p} \min_{i=1}^p \sum_{j \neq i} d(x_i, x_j)^\alpha$$

- **Max-Neighbor-Sum (PMSN)** : maximise la somme des distances entre points consécutifs dans un ordre donné.

$$\max_{1 \leq i_1 < \dots < i_p \leq n} \sum_{j=1}^{p-1} d(x_{i_j}, x_{i_{j+1}})^\alpha$$

Illustration p-dispersion



Variant	Dispersion of A,B,C,D
Mm	: $\min(AB^\alpha, BC^\alpha, CD^\alpha) = \min(AB, BC, CD)^\alpha$
MS	: $AB^\alpha + AC^\alpha + AD^\alpha + BC^\alpha + BD^\alpha + CD^\alpha$
MSN	: $AB^\alpha + BC^\alpha + CD^\alpha$
MSm	: $AB^\alpha + \min(AB^\alpha, BC^\alpha) + \min(BC^\alpha, CD^\alpha) + CD^\alpha$
MmS	: $\min(AB^\alpha + AC^\alpha + AD^\alpha, AB^\alpha + BC^\alpha + BD^\alpha, AC^\alpha + BC^\alpha + CD^\alpha, AD^\alpha + BD^\alpha + CD^\alpha)$

Figure – Illustration des variantes p-dispersion [2]

Illustration de l'hypervolume

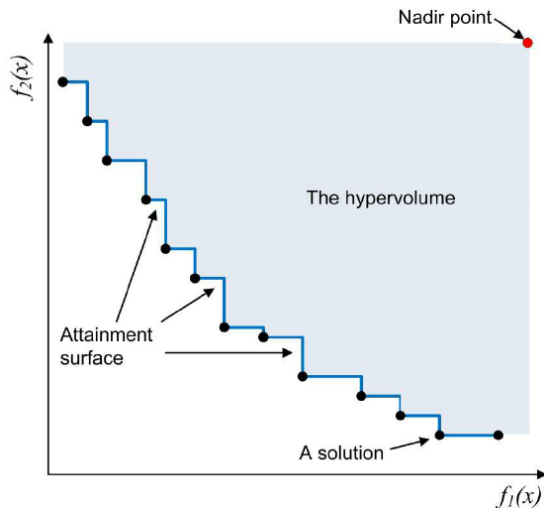


Figure – Exemple d'hypervolume dominé [3]

Hypervolume 2D : Formule exacte

- L'hypervolume mesure la surface dominée par les points sélectionnés par rapport à un point de référence.
- Soit $S = \{x_{i_1}, \dots, x_{i_p}\}$ trié selon x croissant.
- Point de référence : $r = (x_{\text{ref}}, y_{\text{ref}})$ dominé par tous les points de S .
- **Formule de l'hypervolume 2D :**

$$HV(S) = \sum_{j=1}^{p-1} (y_{\text{ref}} - y_{i_j})(x_{i_{j+1}} - x_{i_j})$$

- De nombreux travaux ont abordé le problème de la sélection de sous-ensembles sur un front de Pareto.
- Les approches incluent des algorithmes exacts, heuristiques, de recherche locale, ou encore de type glouton.
- Dans le cadre 2D, certains algorithmes exacts s'appuient sur la structure ordonnée du front et permettent des solutions en temps polynomial.
- Nous nous concentrerons ici sur deux algorithmes efficaces de **programmation dynamique**, appliqués aux critères PMm et PMSN, tels que proposés dans *N. Dupin, Polynomial algorithms for p-dispersion problems in a planar Pareto Front* [2].

Méthodes brutes : énumération exhaustive

- Pour des petites valeurs de p (typiquement $p \leq 5$), on peut résoudre les variantes de p -dispersion par énumération complète des combinaisons possibles.
- On sélectionne systématiquement les **points extrêmes** du front 2D.
- Cela réduit l'espace de recherche à $\binom{n-2}{p-2}$ combinaisons pour les $p - 2$ points restants.
- Utilisables comme **références expérimentales** ou pour vérifier la correction des algorithmes dynamiques.
- Pour $p \gg 5$, cette approche devient prohibitive $\Rightarrow$ recours nécessaire à la **programmation dynamique**.

Programmation dynamique : principe général

- Les fronts 2D peuvent être triés.
- Cela induit une **structure ordonnée**, compatible avec une approche en programmation dynamique.
- Chaque sous-problème correspond à une préfixe du front avec un nombre de points fixé.
- Une table $C[k][i]$ peut mémoriser la meilleure solution de k points se terminant au point i .

Algorithme PMm : Max-Min

- Objectif : maximiser la distance minimale entre chaque paire de points sélectionnés.
- Démarche : approche de type **dichotomie** sur la distance minimale possible.
- Cas $p = 3$: complexité en $\mathcal{O}(n)$ en temps et $\mathcal{O}(1)$ en espace.
- Cas général ($p > 3$) : complexité en $\mathcal{O}(pn \log n)$ à l'aide d'une recherche dichotomique et d'un test efficace.
- Espace utilisé : $\mathcal{O}(n)$.

Algorithme PMSN : Max-Neighbor-Sum

- Objectif : maximiser la somme des distances entre points consécutifs d'un sous-ensemble ordonné.
- Idée : calculer dynamiquement le meilleur score obtenu en sélectionnant k points terminant en x_i .
- Table $C[k][i]$ mise à jour par récurrence :

$$C[k][i] = \max_{j < i} (C[k-1][j] + d(x_j, x_i))$$

- Initialisation : $C[2][i] = d(x_1, x_i)$.
- Reconstruction par backtracking des indices optimaux.
- Complexité : $\mathcal{O}(pn^2)$ en temps, $\mathcal{O}(pn)$ en espace.

Test 1 : Nombre de solutions optimales

- Objectif : mesurer combien de sous-ensembles distincts optimaux existent pour chaque critère.
- Contexte : front de Pareto 2D généré aléatoirement, de taille n , avec $\alpha \in 0,5, 1, 2$.

Conclusion :

- PMm génère de nombreuses solutions optimales (jusqu'à 331 pour $p = 5$), ce qui complique l'analyse comparative.
- Les autres critères (MS, MSm, PMSN, HSS) produisent une seule solution optimale, ce qui les rend plus discriminants.
- Cela suggère une grande variabilité dans les résultats PMm, et une meilleure stabilité pour les autres critères.

Résultats Test 1 : Nombre de solutions optimales

Table – Nombre moyen de solutions optimales ($n = 50$, $\alpha = 1$)

p	HSS	Mm	MSN	MSm	MmS	MS
3	1.00	6.60	1.00	1.00	1.00	1.00
4	1.00	24.70	1.00	1.00	1.00	1.00
5	1.00	331.20	5.50	3.60	1.00	1.00

Test 2 : Compatibilité inter-critères

- Objectif : évaluer la similarité entre les sous-ensembles sélectionnés selon différents critères.
- Méthode : on compare, pour chaque critère, les performances croisées des solutions optimales obtenues selon d'autres critères, afin de mesurer leur compatibilité.
- Cela permet de détecter les affinités structurelles entre critères.

Conclusion :

- Forte compatibilité entre MS, MSm, MmS et PMSN : ces critères mènent souvent aux mêmes sous-ensembles.
- Le critère HSS produit des solutions plus éloignées des autres, car il favorise la couverture du front plutôt que sa dispersion.
- Mm (Max-Min) est peu compatible à cause de sa forte variabilité.

Résultats Test 2 : Compatibilité inter-critères

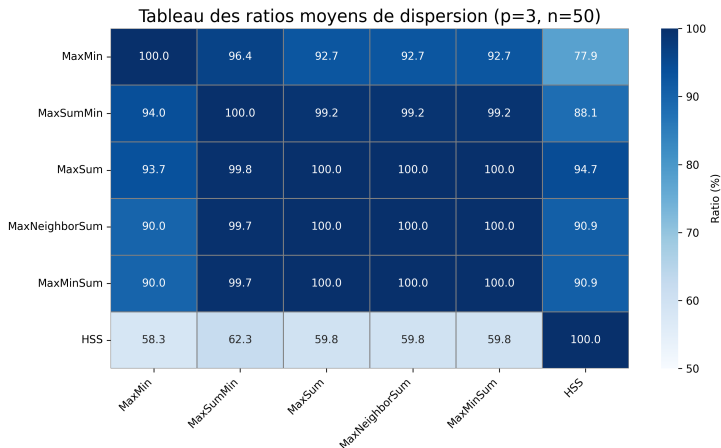


Figure – Ratios de dispersion ($p=3$, $n=50$)

Test 3 : Robustesse MSN

- Objectif : évaluer la robustesse de MSN lorsque le paramètre α varie.
- On évalue les performances de la variante MSN par rapport aux autres critères pour différentes valeurs de α .
- Cela permet de tester la stabilité du comportement de MSN face à une variation de la sensibilité à la dispersion.

Conclusion :

- Pour $\alpha \leq 1$, MSN reste proche des solutions optimales selon les autres critères.
- Pour $\alpha > 1$, les performances relatives de MSN se dégradent, notamment face à MSm et MS.
- MSN peut être considéré comme un compromis efficace tant que la sensibilité à la dispersion reste modérée.

Résultats Test 3 : Robustesse MSN

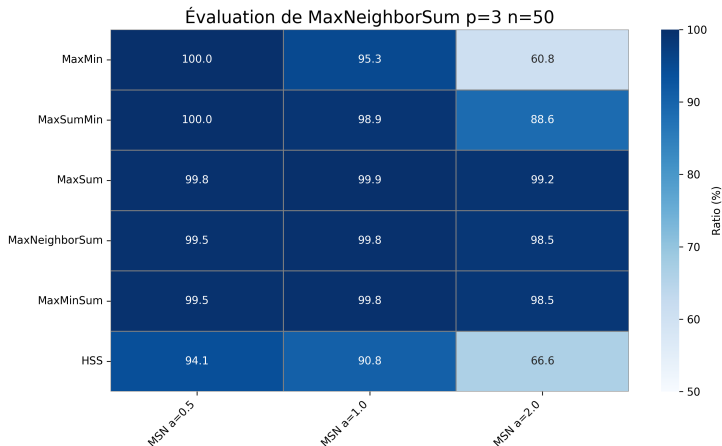


Figure – Évaluation des solutions MSN selon les autres critères

Conclusion

- Ce stage m'a permis de travailler à l'interface entre algorithmique, mathématiques et ingénierie logicielle.
- J'ai développé une architecture orientée objet en C++, compatible avec solverCO, permettant de comparer facilement plusieurs solveurs sur différents critères de sélection.
- L'analyse expérimentale a mis en évidence plusieurs tendances :
 - Certains critères sont très discriminants, comme **MS** ou **MmS**.
 - D'autres, comme **PMSN**, sont robustes et polyvalents.
 - Le critère **PMm**, bien qu'intéressant théoriquement, manque de fiabilité pratique à cause de la multiplicité des solutions optimales.
- Perspectives :
 - Comparaison avec des méthodes issues du **clustering**.
 - Extension aux fronts de Pareto en dimension $d > 2$.

References



N. Dupin.

Polynomial algorithms for p -dispersion problems in a planar Pareto Front.

RAIRO-Operations Research, 57(2) :857–880, 2023.



N. Dupin, F. Nielsen, E. Talbi. *Unified Polynomial Dynamic Programming Algorithms for p -Center Variants in a 2D Pareto Front*. arXiv :2002.11830, 2020. <https://arxiv.org/pdf/2002.11830>



Alexander Brownlee, Alain L. Dias, Richard Wood, Alan Chappell, John R. H. Rossiter, Jonathan M. Cooper. *Multi-objective optimization of cellular fenestration by an evolutionary algorithm*. Engineering Applications of Artificial Intelligence, Volume 43, 2015, Pages 23–32. doi : 10.1016/j.engappai.2015.03.003.



Erhan Erkut and Susan Neuman. *Comparison of Four Models for Dispersing Facilities*. Location Science, Volume 1, Issue 3, Pages 199–214, 1991.

Questions ?