

Heuristique de reconnaissance des préférences 2-Euclidiennes

Bruno Escoffier^{1,2}, Olivier Spanjaard¹, Magdalena Tydrichová¹

¹ Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

{bruno.escoffier, olivier.spanjaard, magdalena.tydrichova}@lip6.fr

² Institut Universitaire de France

Mots-clés : *choix social computationnel, préférences Euclidiennes, heuristique de reconnaissance*

1 Introduction et motivation

Nous proposons une heuristique de reconnaissances de *préférences Euclidiennes dans le plan* (ou *2-Euclidiennes*). Etant donné un profil de préférences de n votants sur un ensemble de m candidats, on cherche à placer les candidats et votants dans le plan de manière à ce que plus le point représentant le candidat c est proche (au sens de la norme Euclidienne classique ℓ_2) du point correspondant au votant v , plus ce dernier préfère c . Si un tel placement (appelé *représentation* dans ce qui suit) est possible, on dit que le profil est 2-Euclidien.

Il s'agit d'une restriction de domaine très utilisée dans diverses disciplines comme le choix social computationnel, l'économie, la psychologie ou les sciences politiques. Malgré cette popularité, les préférences 2-Euclidiennes restent relativement mal comprises contrairement à d'autres restrictions de domaine classiques (comme par exemple les préférences unimodales sur un axe, unimodales sur un arbre, *single-crossing*, ou 1-Euclidiennes). Peters a prouvé que la reconnaissance des préférences 2-Euclidiennes est un problème NP-difficile nécessitant un nombre exponentiel de bits pour exprimer la solution [5]. Il a également prouvé la conjecture de Chen et Grottko [2] selon laquelle les préférences 2-Euclidiennes ne peuvent pas être caractérisées par un ensemble de sous-structures interdites dont la taille serait polynomiale en m .

Au fil des décennies, de nombreux algorithmes ont été développés pour trouver, étant donné un profil (qui peut être ou ne pas être Euclidien), une représentation approchée qui minimise la distance (au sens de la norme ℓ_2) à une solution exacte (notons que même si le profil à l'entrée n'est pas Euclidien, nous pouvons toujours trouver une représentation qui respecte le plus possible la définition d'une représentation euclidienne). En pratique, ces algorithmes rencontrent tous le même problème, à savoir la tendance à renvoyer des représentations dites dégénérées, avec les candidats placés approximativement en cercle, et les votants formant un nuage autour de son centre. Bien que la distance à une représentation exacte soit ainsi numériquement petite, cette solution n'apporte aucune information sur la structure des préférences des votants.

2 Heuristique de reconnaissance

2.1 L'idée de l'heuristique

Contrairement aux heuristiques susmentionnées, nous ne cherchons pas une représentation approchée, mais, si elle existe, une représentation exacte du profil, et ceci en deux étapes :

1. Nous utilisons deux résultats connus pour en faire des conditions nécessaires que chaque profil 2-Euclidien doit vérifier - il s'agit de la taille maximale (en nombre de votes distincts) d'un profil 2-Euclidien en fonction du nombre de candidats [1], et de la caractérisation des profils 2-Euclidiens sur un ensemble de quatre candidats [3]. Si le profil d'entrée ne vérifie pas une de ces conditions, il n'est pas 2-Euclidien, et l'algorithme s'arrête.
2. Sinon, l'heuristique répète les trois points suivants afin de deviner une représentation 2-Euclidienne du profil :
 - elle place les candidats dans le plan d'une manière aléatoire,
 - elle détermine l'ensemble des préférences compatibles avec ce placement de candidats,
 - si le profil en entrée est inclus dans cet ensemble, alors il est 2-Euclidien.

Si aucune représentation n'est trouvée dans le temps imparti, l'heuristique s'arrête sans pouvoir décider si le profil est 2-Euclidien ou pas.

2.2 Contribution

Voici un bref résumé de nos contributions :

- Afin de rendre plus opérationnelle cette heuristique, nous proposons diverses améliorations basées sur des propriétés géométriques du problème, ainsi que de nombreuses optimisations algorithmiques. Ces optimisations nous ont permis d’obtenir une heuristique opérationnelle pour des profils comportant jusqu’à 9 candidats. Le code sera bientôt rendu public et pourra, on l’espère, servir à de futurs travaux dans le domaine.
- Nous avons pu décider, pour 95% des profils du monde réel disponibles sur Preflib [4], s’ils étaient 2-Euclidiens ou non. Nous avons pu identifier des classes de profils sur lesquelles notre heuristique était performante/peu performante.
- Nous avons mené des expériences sur des données synthétiques pour tester les limites de l’heuristique. La plupart de profils avec au plus 6 candidats sont reconnus, et une proportion importante des profils avec au plus 8 candidats. Bien que ces tailles de profils puissent paraître modestes, il est à noter que cette performance est meilleure que celle des autres approches existantes : à titre de comparaison, la résolution avec Gurobi d’un programme quadratique que nous avons conçu ne permet de reconnaître que très peu de profils de 6 préférences sur 5 candidats, et ne reconnaît pratiquement aucun profil plus grand.
- Nous observons une transition de phases dans le problème de reconnaissance : la plupart des profils avec $n \leq 6$ sont 2-Euclidiens, tandis que la plupart des profils avec $n \geq 10$ ne le sont pas. Le domaine de valeurs de n où il y a une forte incertitude est donc assez réduit, et ne change pas beaucoup en fonction du nombre de candidats.
- Enfin, l’heuristique nous a permis d’avoir quelques éléments de réponses à des questions plus théoriques. Nous avons notamment pu avoir quelques indications sur le nombre de profils 2-Euclidiens maximaux en fonction du nombre de candidats.

3 Conclusion

L’heuristique proposée, qui se base sur une idée assez simple, fonctionne bien en pratique grâce aux différentes améliorations effectuées. Néanmoins, il reste beaucoup à faire : il serait très intéressant de trouver des conditions nécessaires plus fortes - en effet, il semblerait que cela soit principalement la détection insuffisante des profils non Euclidiens qui serait responsable de nombreux profils non décidés. Par ailleurs, notre algorithme est optimisé pour des profils comportant au plus 10 candidats - une optimisation pour des profils plus grands serait bienvenue. Enfin, il serait intéressant d’approfondir la compréhension théorique du problème, notamment l’étude du nombre de profils 2-Euclidiens, ou encore la compréhension de la transition de phase.

Références

- [1] Joseph F. Bennett and William L. Hays. Multidimensional unfolding : Determining the dimensionality of ranked preference data. *Psychometrika*, 25(1) :27–43, Mar 1960.
- [2] Jiehua Chen and Sven Grottko. Small one-dimensional Euclidean preference profiles. *Social Choice and Welfare*, pages 1–28, 2021.
- [3] Hidehiko Kamiya, Akimichi Takemura, and Hiroaki Terao. Ranking patterns of unfolding models of codimension one. *Advances in Applied Mathematics*, 47(2) :379–400, 2011.
- [4] Nicholas Mattei and Toby Walsh. Preflib : A library of preference data [HTTP://PREFLIB.ORG](http://preflib.org). In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer, 2013.
- [5] Dominik Peters. Recognising multidimensional Euclidean preferences. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.