

Sélection topologique dans l'algorithme génétique pour le contrôle optimal

C. Mbangue Lobe Yiphta^{1,a}, D. Razafindralandy^{1,b}, C. Allery^{1,c}, A. Hamdouni^{1,d}

¹ Laboratoire des Science de l'Ingénieur pour L'Environnement, La Rochelle Université

^aconstantini.mbangue_lobe_yiphta@univ-lr.fr, ^bdina.razafindralandy@univ-lr.fr, ^ccyrille.allery@univ-lr.fr, ^daziz.hamdouni@univ-lr.fr

Résumé — On s'intéresse aux algorithmes génétiques pour le contrôle optimal d'écoulements turbulents. On propose une méthode pour améliorer leur capacité à capter les oscillations rapides. Pour cela, on utilise un critère topologique dans la phase de sélection au lieu d'un critère fondé sur la fonction coût à minimiser. Ce critère topologique est basé sur la distance entre le diagramme de persistance de la cible et celui de l'approximation. On montre la pertinence de l'approche sur des solutions ératiques de l'équation de Burgers ayant un comportement similaire à celui d'un écoulement turbulent.

Mots clés — Contrôle optimal, Algorithme génétique, Analyse topologique des données, Homologie persistante.

1 Introduction

Les problèmes d'optimisation sous contrainte interviennent dans beaucoup de branches de la mécanique. Pour les résoudre numériquement, quelques méthodes comme l'algorithme génétique, la descente de gradient ou encore l'algorithme du gradient stochastique sont populairement utilisées [1]. Souvent, la fonction coût à minimiser dans de tels problèmes s'écrit sous forme de somme de normes L^2 (ou L^1). De ce fait, pour capter les petites oscillations, il faut imposer une tolérance très faible, augmentant ainsi le coût de calcul. Or il y a des problèmes où les petites oscillations peuvent avoir une grande importance. C'est le cas par exemple en aéro-acoustique où ce sont les petites fluctuations qui engendrent les bruits.

Dans cette communication, on propose une méthode permettant de capturer plus rapidement toutes les oscillations présentes dans la fonction cible. Pour cela, on va se focaliser sur l'algorithme génétique. Ce dernier se décompose en plusieurs phases dont une phase de sélection. Classiquement, lors de cette phase, on sélectionne les individus qui minimisent le plus la fonction coût. Ici, on se propose de sélectionner plutôt les individus qui ont les propriétés topologiques les plus proches de celles de la cible. Pour cela, on utilisera l'analyse topologique des données, et en particulier les diagrammes de persistance topologique [2]. En effet, lorsqu'on les applique à des fonctions, les diagrammes de persistance encodent les fluctuations de la fonction et les classent selon leur "taille" ou persistance. Cela rend notre méthode plus sensible aux fluctuations. À part la phase de sélection, on garde l'architecture classique de l'algorithme génétique.

Les diagrammes de persistance peuvent être vus comme une signature topologique. Leurs propriétés de robustesse en ont fait un outil d'analyse efficace [3, 4]. Notons que l'analyse topologique des données a déjà montré sa pertinence dans la résolution de problèmes de la mécanique. Par exemple, dans l'industrie automobile, les premières études ont permis, grâce aux diagrammes de persistance, de détecter l'état de stress d'un conducteur au volant [5]. D'autres études ont montré qu'on peut détecter l'endommagement d'un ouvrage d'art grâce à l'analyse topologique des données issus de capteurs, et cela indépendamment de facteurs extérieurs tels que la température [6]. La persistance topologique a aussi été utilisée pour analyser et caractériser des écoulements de fluide [7].

Nous souhaitons appliquer notre approche au contrôle d'écoulements turbulents, où la fonction cible est une solution des équations de Navier-Stokes, représentant un écoulement turbulent. Mais dans cette étape exploratoire, on se limitera à des solutions de l'équation de Burgers qui présentent un comportement ératique [8].

Dans la section 2, nous rappellerons rapidement les grands principes de l'algorithme génétique. Puis, dans la section 3, nous ferons une brève introduction à l'analyse topologique des données et au dia-

gramme de persistance. La présentation qui sera faite sera très simpliste. Nous rappellerons le théorème de robustesse. Dans la section 4, on présentera quelques résultats numériques.

2 Problème de contrôle optimal et algorithme génétique

On considère la classe des problèmes variationnels de la forme

$$\operatorname{argmin}_{\theta \in \mathcal{K}} J(u(\theta), \theta) \quad (1)$$

soumis à une contrainte

$$\mathcal{N}(u(\theta), \theta) = 0 \quad (2)$$

où $J : \mathcal{V} \times \mathcal{Z} \rightarrow \mathbb{R}$ désigne la fonctionnelle coût, et \mathcal{N} décrit une équation de contraintes appelée aussi équation d'état. De plus, nous supposons que $\mathcal{K} \subset \mathcal{Z}$ et \mathcal{Z} et \mathcal{V} sont des espaces de Hilbert. Les problèmes qui s'inscrivent dans le cadre des équations (1) et (2) sont assez généraux et se retrouvent dans de nombreux domaines d'application. Dans le contexte du contrôle optimal, la variable u est appelée variable d'état, θ le paramètre de contrôle. La fonctionnelle coût J est choisie de manière appropriée pour atteindre des performances spécifiques. Ici, on s'intéresse à un problème de contrôle optimal (1)-(2) où la fonction coût est de la forme

$$J(u(\theta), \theta) = \|u(\theta) - \hat{u}\|_{L^2(\Omega, T)}^2,$$

\hat{u} est un champ qu'on cherche à atteindre (champ cible) et la contrainte (2) est une équation aux dérivées partielles définies sur $\Omega \times]0, T]$. Les espaces fonctionnelles sont des espaces de Hilbert tels que

$$\mathcal{V} = L^2(0, T; H^1(\Omega)), \quad \mathcal{W} = L^2(0, T; L^2(\Omega)), \quad \mathcal{Z} = \mathbb{R}^d.$$

Quelques méthodes numériques existent pour la résolution du problème (1)-(2). Dans cette communication, on va se concentrer sur les algorithmes génétiques [9]. La première étape d'un algorithme génétique consiste à choisir aléatoirement des individus pour constituer une population initiale. Dans notre cas, cette population initiale est un ensemble $\{\theta_i, i = 1, \dots, I\}$ d'individus θ_i dont le champ $u(\theta_i)$ associé vérifie l'équation (2). On évalue ensuite la qualité de tous les individus en calculant pour chacun d'eux la fonction $J(u(\theta_i), \theta_i)$. Tant que le critère d'optimalité n'est pas vérifié à une tolérance ε donnée, une nouvelle population est générée de la manière suivante.

— Sélection : on sélectionne les meilleurs individus θ_i pour lesquels

$$J(u(\theta_i), \theta_i) \quad (3)$$

sont les plus faibles. Cela constitue une sous-population A . Pour qu'il y ait variété génétique, on ajoute à cette sous-population quelques individus parmi ceux qui n'ont pas été retenus. On obtient une sous-population B .

— Mutation : Dans l'algorithme génétique de base, les composantes θ_i^j de certains individus $\theta_i \in B$ choisis au hasard sont changées aléatoirement. On obtient une population C .

— Croisement : On choisit des paires d'individus distincts dans C et on interchange quelques composantes pour produire de nouveaux individus. Ces derniers seront rajoutés à la population C pour former la nouvelle génération.

Ces différentes phases de l'algorithme génétique sont représentées graphiquement sur la Figure 1. On se fixe généralement un nombre maximal d'itérations.

Dans notre approche, on propose de modifier l'étape de sélection. En effet, on propose dans cette étape de garder non pas les individus qui minimisent le plus la fonctionnelle J mais ceux qui satisfont le mieux un critère basé sur les propriétés topologiques. Ces propriétés sont encodées par les invariants topologiques (nombres de Betti) et leurs persistances qu'on introduira brièvement dans la section suivante.

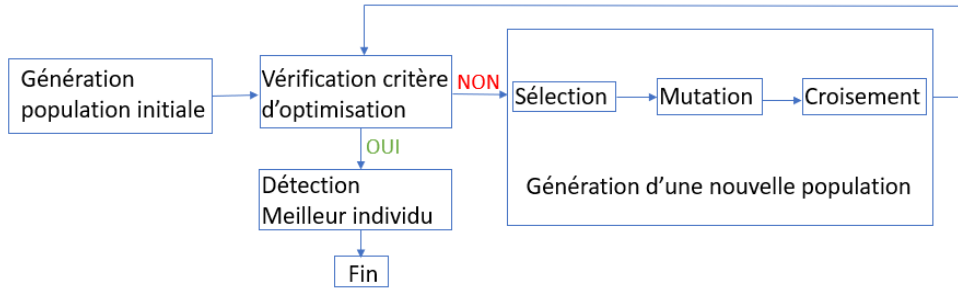


FIGURE 1 – Étapes de l’algorithme génétique

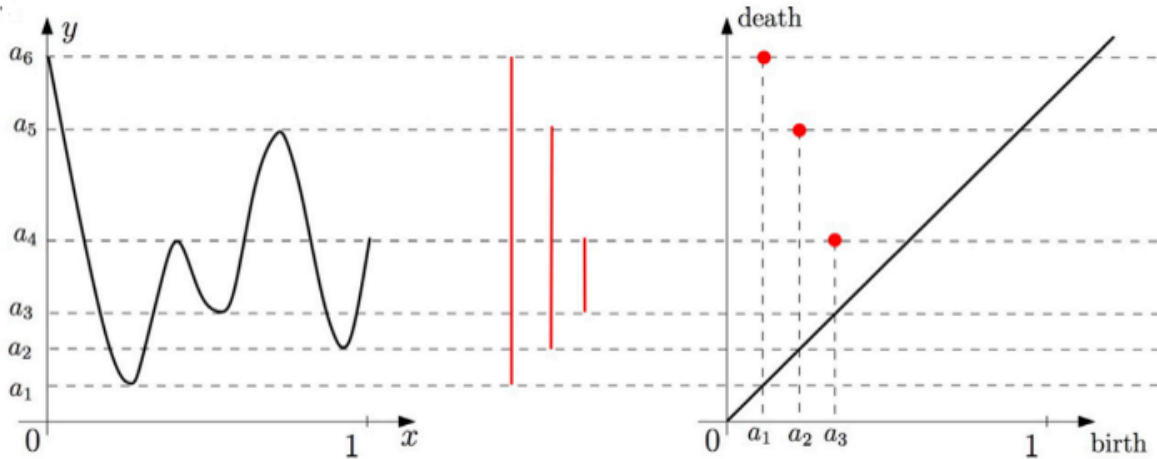


FIGURE 2 – Exemple d’une fonction $f : [0, 1] \rightarrow \mathbb{R}$ (à gauche), de son code-barre (au milieu) et son diagramme de persistance (à droite). Tiré de [10]

3 Homologie persistante et modification de l’algorithme génétique

Avant de donner des définitions formelles, on illustre l’homologie persistante sur un exemple.

Soient $f : [0, 1] \rightarrow \mathbb{R}$ la fonction représentée sur la Figure 2 et $(F_r)_{r \in \mathbb{R}}$ la famille des ensembles de sous-niveau

$$F_r = f^{-1}([-\infty, r]), \quad r \in \mathbb{R}$$

de f . Pour un r fixé, un ensemble de sous-niveau F_r est soit vide, soit une union d’intervalles, de sorte que la seule information topologique non triviale est le nombre de composantes connexes, qui est la dimension du groupe d’homologie d’ordre 0.

Dans l’exemple de la fonction de la figure 2, F_r est vide lorsque $r < a_1$. Lorsque $r = a_1$, une première composante connexe apparaît dans F_{a_1} . L’homologie persistante enregistre ainsi a_1 comme heure de naissance d’une composante connexe et commence à en garder la trace en créant un intervalle commençant à a_1 . Ensuite, F_r reste connexe jusqu’à ce que r atteigne la valeur a_2 , où une deuxième composante connexe apparaît. L’homologie persistante commence à suivre ce nouveau composant connexe en création d’un deuxième intervalle à partir de a_2 . De même, lorsque sont atteints a_3 , une nouvelle composante connexe apparaît et l’homologie persistante crée un nouvel intervalle commençant à a_3 . Lorsque sont atteints a_4 , les deux composantes connexes créés à a_1 et a_3 fusionnent pour donner une seule composante plus grande. À cette étape, l’homologie persistante suit la règle selon laquelle il s’agit de la composante la plus récemment apparue dans la filtration qui meurt ; l’intervalle commencé à a_3 se termine donc à a_4 . Ainsi, on a une composante connexe qui est née à $r = a_3$ et meurt à $r = a_4$. Cela est représentée par un intervalle de persistance (a_3, a_4) et graphiquement par la barre la plus à droite du code-barre de la figure 2. Lorsque r atteint a_5 , la composante née à a_2 meurt et l’intervalle de persistance (a_2, a_5) est créé et la barre du milieu du code-barre est entièrement constituée. L’intervalle créé à a_1 reste jusqu’à la fin de la filtration, donnant lieu à l’intervalle persistant (a_1, a_6) , si la filtration est arrêtée à a_6 , ou $(a_1, +\infty)$, si

r va à $+\infty$ (noter que dans ce dernier cas, la filtration reste constante pour $r > a_6$). On peut constater que le code-barre encode les fluctuations de la fonction ; les barres les plus courtes étant celles les moins persistantes, représentant les plus petites fluctuations.

On peut représenter la vie des intervalles de persistance de plusieurs manières dont par le diagramme de persistance. En effet, chaque intervalle (a, a') peut être représenté par le point de coordonnées (a, a') dans le plan \mathbb{R}^2 . L'ensemble de points qui en résulte est appelé diagramme de persistance de f , noté ici $dgm(f)$. Les points les plus proches de la diagonale $\Delta = \{(b, d) \in \mathbb{R}^2 \mid b = d\}$ du plan encodent les plus petites fluctuations. Notez qu'une fonction peut avoir plusieurs copies du même intervalle dans son code-barre de persistance. En conséquence, dans le diagramme de persistance de f , chaque point de $dgm(f)$ a une multiplicité à valeurs entières. Enfin, pour des raisons techniques qui deviendront claires plus tard, on ajoute à $dgm(f)$ tous les points de la diagonale Δ du plan avec une multiplicité infinie.

Plus généralement, comme la date de naissance est inférieure à la date de la mort, un diagramme de persistance est un ensemble $dgm \subset \{(b, d) \in \mathbb{R}^2 \mid b \leq d\}$ contenant Δ , dont les points possèdent une multiplicité et les points de Δ ont une multiplicité infinie. L'intérêt de représenter les fluctuations de la fonction par un diagramme de persistance est que \mathbb{R}^2 possède des métriques naturelles pour les comparaisons. Grâce à ces métriques, on peut munir l'ensemble des diagrammes de persistance d'une métrique. Par exemple, si on choisit la norme $\|\cdot\|_{L^\infty}$ de \mathbb{R}^2 , on peut définir la métrique suivante de l'ensemble des diagrammes de persistance.

Definition 1 La distance de bottleneck entre deux diagrammes de persistance dgm_1 et dgm_2 est définie par

$$d_b(dgm_1, dgm_2) = \inf_{m \in M} \max_{(p, q) \in m} \|p - q\|_\infty \quad (4)$$

où M est l'ensemble de tous les appariements possibles entre les éléments de dgm_1 et dgm_2 .

Notons que le nombre d'éléments hors diagonaux de dgm_1 et dgm_2 n'est pas forcément le même. Si on n'avait pas inclut la diagonale Δ dans les diagrammes de persistance d'une fonction, on ne pourrait pas toujours réaliser les appariements.

La métrique de bottleneck est une métrique de type $\|\cdot\|_{L^\infty}$. Elle souffre donc des mêmes inconvénients que les normes habituelles L_∞ , c'est-à-dire qu'elle est complètement déterminée par la plus grande distance parmi les paires et ne tient pas compte de la proximité du reste paires de points. Une variante pour surmonter ce problème est la distance de Wasserstein entre les diagrammes.

Definition 2 Etant donné $p \geq 1$, La distance de Wasserstein entre deux diagrammes de persistance dgm_1 et dgm_2 est définie par

$$W_p(dgm_1, dgm_2) = \left(\inf_{m \in M} \sum_{(p, q) \in m} \|p - q\|_\infty^p \right)^{1/p}. \quad (5)$$

Une propriété fondamentale de l'homologie persistante est que les diagrammes de persistance sont stables par rapport à certaines perturbations des données [3, 4]. Par exemple, avec la distance bottleneck, on a le théorème suivant.

Theorem 1 Soient f et $g : M \rightarrow \mathbb{R}$ deux fonctions à valeurs réelles définient assez régulière¹ sur un espace topologique alors

$$d_b(dgm(f), dgm(g)) \leq \|f - g\|_\infty = \sup_{x \in M} |f(x) - g(x)| \quad (6)$$

Ici, on s'est restreint aux propriétés topologiques d'ordre 0 (les composantes connexes) car cela suffit pour nos applications. Mais les diagrammes de persistance peuvent encoder des propriétés topologiques d'ordre supérieur (qui représentent les cycles, les tunnels, ...).

Revenons maintenant à l'algorithme génétique. On propose de choisir dans la phase de sélection de l'algorithme génétique les individus qui minimisent la distance

$$J_T(u(\theta_i), \hat{u}) = W_2(dgm(u(\theta_i)), dgm(\hat{u})) \quad (7)$$

entre les diagrammes de persistance topologique au lieu du critère (3). Ce faisant, on accorde plus d'importance aux fluctuations. Dans la section suivante, on va mener quelques expériences numériques.

1. Voir par exemple [2]

4 Expériences numériques

On considère un problème de minimisation sous la contrainte de l'équation de Burgers

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = f. \quad (8)$$

Dans [8], des solutions ératiques de cette équation ont été proposées. Elles sont sous la forme de somme de fonctions de type

$$\phi_{\theta_i}(t, x) = A_{\theta_i}^{-1} \sin^{21}(f_{\theta_i}) \quad \text{avec} \quad A_j = e^{j^{0.31}}, \quad f_j = A_j \pi(x + 0.8 + t). \quad (9)$$

Plus j est élevé, plus ϕ_{θ_i} est fluctuante. Pour simplifier le problème, on va fixer la valeur du temps à $t = 0.7$ (voir [8]). On mènera quatre expériences.

Dans un premier test, qu'on désignera par Test1, on choisit comme solution la fonction

$$u(x) = A_{\theta_1}^{-1} \sin^{21}(f_{\theta_1}) + A_{\theta_2}^{-1} \sin^{21}(f_{\theta_2}). \quad (10)$$

La cible correspond à $(\theta^1, \theta^2) = (3, 13)$, composée d'une partie basse fréquence et d'une partie haute fréquence. Cette fonction est tracée dans la figure 3. On effectue deux séries de 100 simulations. Dans la première série, on utilise le critère classique (3). Dans la deuxième série, on utilise le critère topologique (7). La valeur de la tolérance relative (basée sur J) est fixée à $\varepsilon = 10^{-2}$. On ne garde que les résultats où la convergence a été atteinte (le nombre d'itérations est inférieur à 101). Les simulations où la convergence n'a pas été atteinte n'ont jamais dépassé 3%.

Dans toutes les simulations, et avec les deux critères, la valeur du paramètre de basse fréquence $\theta_1 = 3$ a été trouvée. En revanche, ce n'est pas le cas pour θ_2 . La figure 4 montre les valeurs obtenues avec les deux méthodes. On constate qu'avec le critère L^2 classique, la bonne valeur de θ_2 a été trouvée dans 82% des simulations, comme cela a été reporté dans le tableau 1, Cas 1. Dans les 18% restantes, des valeurs fausses ont aussi été trouvées. Avec le critère de sélection topologique, on a trouvé la valeur exacte $\theta_2 = 13$ dans toutes les simulations.

Cible	Sélection classique	Sélection topologique
Test1	82 %	100 %
Test2	91%	100 %
Test3	90 %	100 %

TABLE 1 – Pourcentages de réussite

Dans un deuxième test, désigné Test2, la solution est une somme

$$u(x) = \sum_{j=1}^{\theta} A_j^{-1} \sin^{21}(f_j). \quad (11)$$

Elle est représentée dans la figure 5. Le champ cible correspond au paramètre $\theta = 13$.

On mène des séries de 1000 simulations pour chacun des deux critères. On trouve bien alors la valeur cible dans 91% des simulations avec le critère classique de sélection, comme reporté dans le tableau 1, Test2. Dans les 9% de cas restants, des multitudes de valeurs ont été trouvées, ce qui rend difficile la représentation sous forme d'histogramme comme dans la figure 3. Avec le critère de sélection topologique, on trouve la valeur cible dans 100% des simulations, comme montré dans le tableau.

Notons que pour représenter les solutions, la solution a été discrétisée avec 1500 points, ce qui est suffisant pour des valeurs relativement petites de θ . Dans un dernier test, Test3, on va examiner le cas où la solution cible est sous-échantillonnée. Pour cela, on reprend la solution (10) mais avec comme valeurs cibles $(\theta^1, \theta^2) = (3, 76)$. La valeur cible de θ^1 a toujours été trouvée par les deux algorithmes. Les pourcentages de réussite pour trouver θ^2 sont reportés dans le tableau 1. Comme on peut constater, on trouve la bonne valeur avec la sélection topologique dans tous les cas, alors que dans 10% des cas, le critère classique ne permet pas de trouver θ_2 .

Dans un dernier test, désigné Test4, on reprend la solution (11) mais avec comme paramètre cible $\theta = 76$. Le champs cible est sous-échantillonné. Cette fois, aucune des deux approches n'a permis de

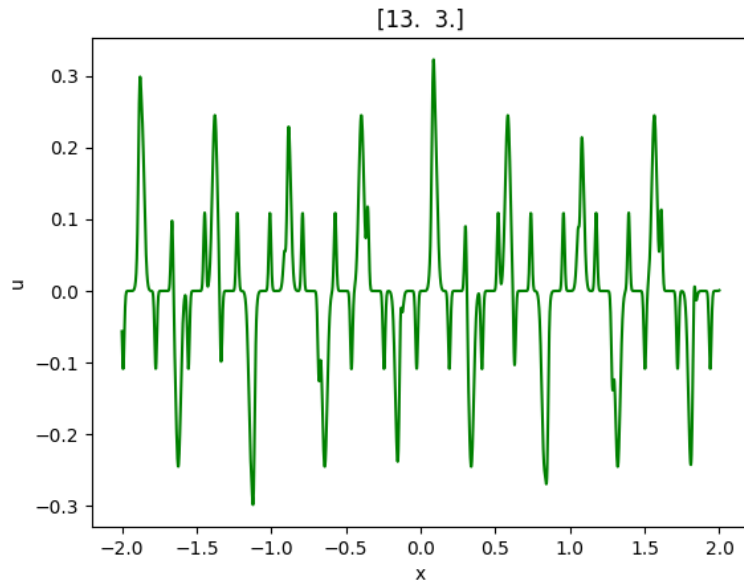


FIGURE 3 – Solution (10)

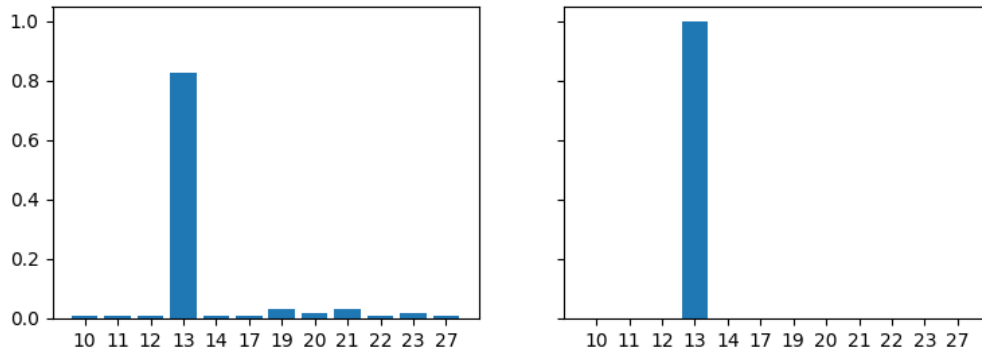


FIGURE 4 – A gauche les résultats obtenus par l'approche L2 et à droite, ceux obtenus par l'approche topologique pour Test1.

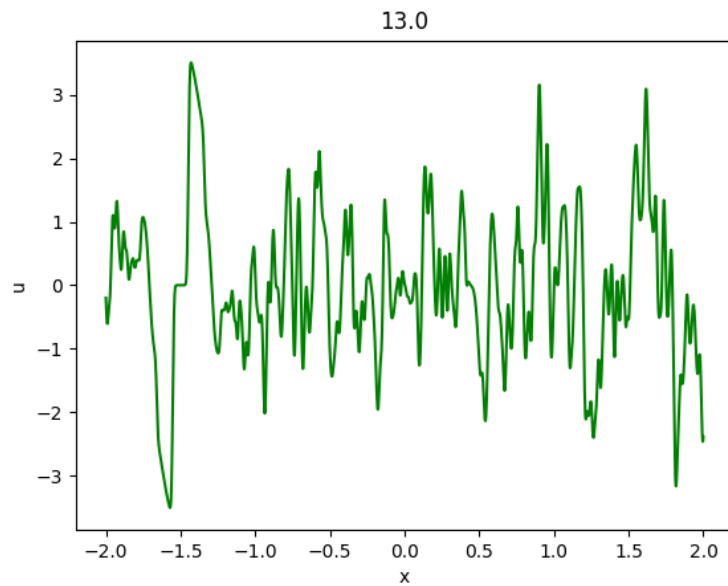


FIGURE 5 – Solution (11)

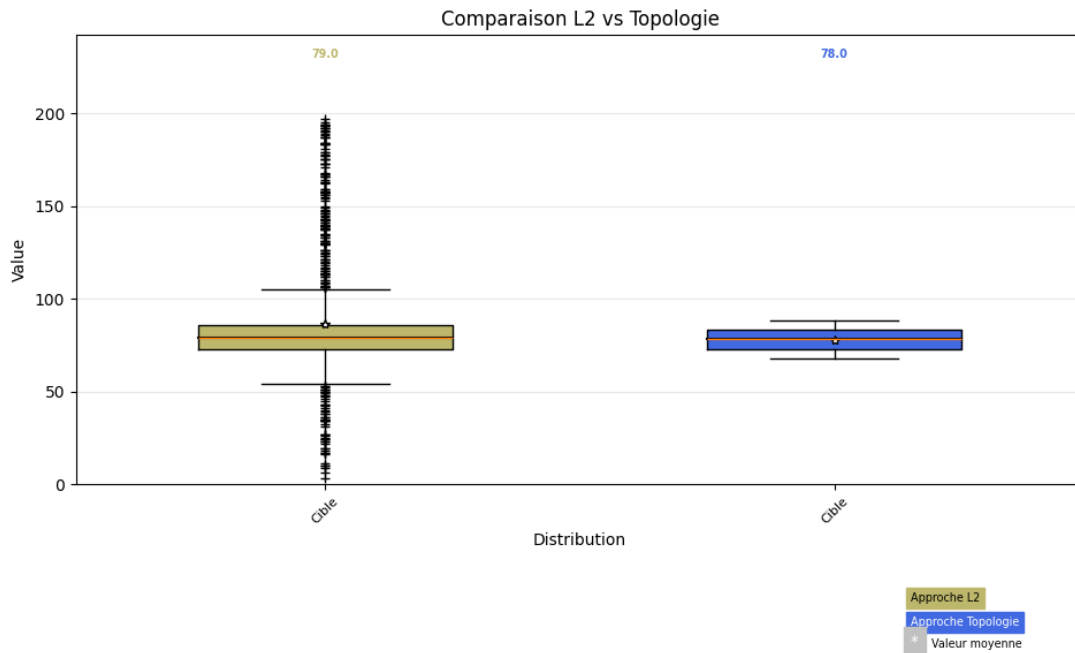


FIGURE 6 – Boxplot de Test4. A gauche les résultats obtenus par l’approche L2 et à droite, ceux obtenus par l’approche topologique avec pour cible la fonction u_4 .

trouver la bonne valeur de θ à 100%. L’approche classique a donné en moyenne une valeur autour de 86 tandis que l’approche topologique a trouvée une valeur moyenne de 78, ce qui est beaucoup plus proche de la valeur exacte. Cela est reporté dans le tableau 2, où on peut trouver d’autres informations. Par ailleurs, avec l’approche classique, on obtient beaucoup de valeurs abérantes, comme montré dans la figure 6 qui analyse la distribution des valeurs trouvées sous forme de boxplot (boîte à moustache).

	Sélection classique	Sélection topologique
Moyenne	86	78
Médiane	79	78
50% des valeurs	Entre 73 et 86	Entre 73 et 83

TABLE 2 – Analyse des valeurs trouvées pour Test4

5 Conclusion

On a comparé l’algorithme génétique classique avec une variante hybride où la phase de sélection est basée sur des critères topologiques. On l’a testé sur des configurations où la fonction cible possède plusieurs fréquences. On a constaté que l’approche hybride permet de retrouver la plupart du temps les paramètres de hautes fréquences alors que ce n’est pas toujours le cas avec l’approche classique. Le seul cas où l’approche topologique n’a pas trouvé la valeur cible à 100%, c’est lorsque la fonction cible est composée de plusieurs fréquences et qu’elle est très sous-échantillonnée. Mais même dans ce cas, le résultat est meilleur qu’avec l’approche classique.

Pour se rapprocher des cas réalistes, on traitera le cas où le temps varie et on montrera les résultats lors de la présentation orale.

Références

- [1] David E Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms*, volume 1, pages 69–93. Elsevier, 1991.
- [2] H. Edelsbrunner and J. Harer. *Computational topology. An introduction*. AMS, 2000.

- [3] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Direct & Computational Geometry*, 37, 2007.
- [4] P. Skraba and K. Turner. Wasserstein stability for persistence diagrams. *arXiv :2006.16824*, 2021.
- [5] Tarek Frahi, Francisco Chinesta, Antonio Falco, Alberto Badias, Elias Cueto, Hyung Yun Choi, Manyong Han, and Jean-Louis Duval. Empowering advanced driver-assistance systems from topological data analysis. *Mathematics*, 9(6) :634, 2021.
- [6] Tristan Gowdridge, Nikolaos Dervilis, and Keith Worden. On topological data analysis for structural dynamics : an introduction to persistent homology. *ASME Open Journal of Engineering*, 1, 2022.
- [7] Miroslav Kramár, Rachel Levanger, Jeffrey Tithof, Balachandra Suri, Mu Xu, Mark Paul, Michael F Schatz, and Konstantin Mischaikow. Analysis of kolmogorov flow and rayleigh–bénard convection using persistent homology. *Physica D : Nonlinear Phenomena*, 334 :82–98, 2016.
- [8] Solution filtering technique for solving Burgers’ equation. In *Proceedings of the fourth international conference on dynamical systems and differential equations, May 24-27, 2002, Wilmington, NC, USA*, 2002.
- [9] John H Holland. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [10] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis : fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4 :667963, 2021.
- [11] Adrien Poulénard, Primoz Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, volume 37, pages 13–25. Wiley Online Library, 2018.
- [12] Mourad Oulghelou and Cyrille Allery. Non-intrusive reduced genetic algorithm for near-real time flow optimal control. *International Journal for Numerical Methods in Fluids*, 92(9) :1118–1134, 2020.
- [13] Alexandra Tallet, Cyrille Allery, and Cédric Leblond. Optimal flow control using a pod-based reduced-order model. *Numerical Heat Transfer, Part B : Fundamentals*, 70(1) :1–24, 2016.