

Vers une conception optimale de nos véhicules:

Adaptation des stratégies d'évolution pour résoudre des problèmes d'optimisation multidisciplinaire.

Informations et candidatures : Youssef.Diouane@isae-superaero.fr.

Contexte

La conception de produits est plus que jamais collaborative. Les équipes de concepteurs, bien souvent experts dans une discipline (fluide, structure, propulsion, ...) doivent donc disposer à la fois de logiciels qui puissent collaborer entre eux mais aussi d'une plateforme qui centralise dans un même espace l'ensemble de calculs nécessaires à la conception du produit. On parle alors de conception multidisciplinaire. Une fois cette chaîne de processus de calcul automatisée, une surcouche d'intelligence artificielle est utilisée pour optimiser la conception ou la production en fonction des attentes. Par exemple, dans la conception d'une aile d'avion, la discipline "structure" a pour souhait d'augmenter l'épaisseur pour avoir une meilleure résistance, tandis que le but de la discipline "CFD" (Computational Fluid Dynamics ou dynamique des fluides numérique) est au contraire de réduire l'épaisseur de l'aile afin de diminuer la traînée. L'optimisation multidisciplinaire est donc un moyen de trouver un compromis entre les différentes disciplines.

Dans ce sens, l'optimisation sans dérivée a connu un regain d'intérêt ces dernières années, principalement motivée par le besoin de résoudre les problèmes d'optimisation définis par des fonctions dont les valeurs sont calculées par simulation (par exemple, la conception technique, la restauration d'images médicales ou de nappes phréatiques). Ces dernières années, un certain nombre de méthodes d'optimisation sans dérivée ont été développées/évoluées et en particulier les méthodes dites stratégies d'évolution (voir Fig. 1).

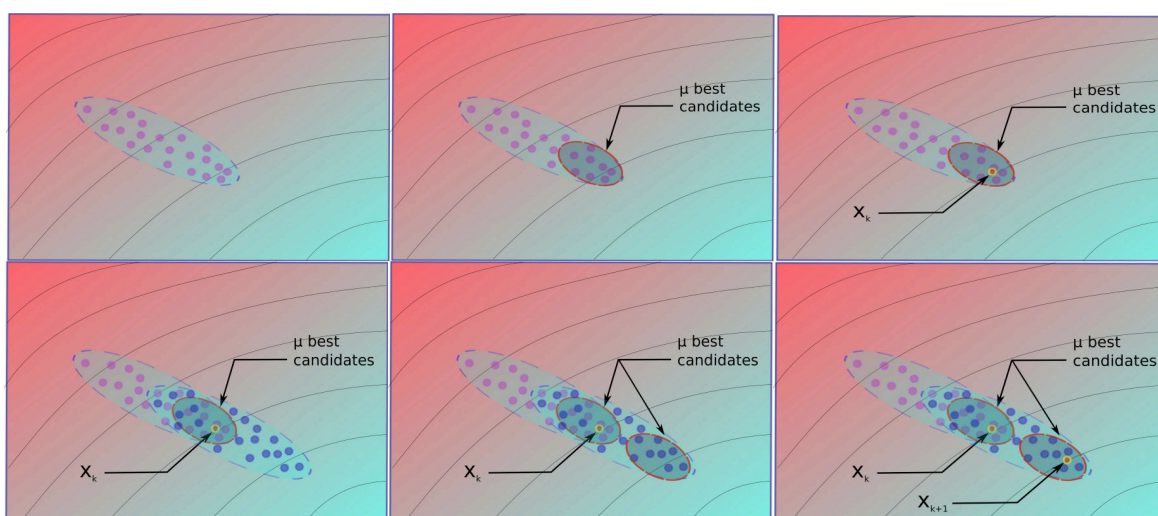


Fig. 1 : schéma explicatif d'une classe des stratégies d'évolutions où on utilise un ensemble de μ « parents » pour produire λ « enfants ». L'algorithme est noté $(\mu/\mu_w, \lambda)$ -ES.

Ce stage a pour objectif de montrer comment adapter une classe d'algorithmes évolutionnaires pour traiter les problèmes d'optimisation multidisciplinaire issus de la

conception des systèmes complexes tel que les avions. On s'intéressera notamment à un cas de test industriel, nommé MOPTA08, issu de l'industrie automobile (General Motors). Il s'agit d'une problématique de réduction de masse d'un véhicule sous contraintes de crash, de réduction de vibration, et de résistance mécanique.

Mission

A l'heure actuelle, la méthode CMA-ES (Covariance Matrix Adaptation – Evolution Strategy) se présente comme une des meilleures stratégies d'évolution. On souhaite adapter cette méthode pour résoudre des problèmes d'optimisation sous contraintes. En effet, les problèmes de conception des systèmes présentent souvent des contraintes à respecter lors de la procédure d'optimisation (ex. MOPTA,...). Un deuxième volet de ce stage, consiste à enrichir la méthode CMA-ES avec des modèles de substitution afin d'obtenir une meilleure performance pour le cas test MOPTA.

On attend que le stagiaire utilise un code existant de la méthode CMA-ES pour ensuite l'adapter/améliorer pour résoudre des problèmes MDO. Le code sera ensuite intégré dans la plate-forme dans PyOpt Sparse pour servir le processus de conception avion via la plateforme OpenMDAO.

Le stage se décompose en 5 grandes étapes :

1. Comprendre, à partir de la bibliographie sur le sujet, les différentes notions: les stratégies d'évolution, CMA-ES, modèles réduits/de substitution, ...
2. Se familiariser avec le code CMA-ES pour traiter les problèmes d'optimisation sans contrainte.
3. Adapter la méthode CMA-ES pour résoudre des problèmes d'optimisation sous contraintes.
 - a. Assimiler les références existantes.
 - b. Modifier CMA-ES pour traiter les contraintes.
 - c. Valider l'implémentation sur des cas test académiques.
 - d. Évaluer le code sur des cas test plus réalistes (ex. MOPTA,...).
4. Hybrider la méthode CMA-ES par différents modèles réduits, évaluer l'impact de ce couplage sur différents cas de test.
 - a. Assimiler les références existantes.
 - b. Réaliser le couplage CMA-ES et modèles réduits.
 - c. Valider l'implémentation sur des cas test académiques.
 - d. Évaluer le code sur des cas test plus réalistes (ex. MOPTA,...).
5. Intégrer tous les travaux (issus des étapes 3 et 4) pour obtenir un code plus performant.

Toutes les implémentations sont prévues en Python, mais la connaissance préalable du langage Python n'est pas indispensable.

Etudiant :

Ecole d'ingénieur ou master de recherche en Recherche Opérationnelle, Optimisation Numérique ou autre discipline proche.

Encadrement

Y. Diouane et J. Morlier, ISAE-SUPAERO, Toulouse.

N. Bartoli, DCPS/SAE, équipe des systèmes aéronautiques, ONERA, centre de Toulouse.

Rémunération : 554.40€ /mois (net), facilités d'hébergement sur le campus.

Durée : 3 à 6 mois.

Lieu de stage :

Au sein du Département d'Ingénierie des Systèmes Complexes (DISC), ISAE-SUPAERO, Toulouse.

Bibliographie

- N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. Eshelman, editor, Proceedings of the Sixth International Conference on Genetic Algorithms, Pittsburgh, pages 57–64, 1995.
- Y. Diouane, S. Gratton and L. N. Vicente, Globally convergent evolution strategy for constrained optimization. Journal of Computational Optimization and Application, 2015.
- R. Chocat, L. Brevault, M. Balesdent and S. Defoort, Modified Covariance Matrix adaptation – Evolution Strategy algorithm for constrained optimization under uncertainty, application to rocket design, International Journal for Simulation and Multidisciplinary Optimization, Vol. 6 :A1, EDP Sciences, 2015.
- OpenMDAO, <http://openmdao.org/docs/tutorials/index.html>
- Pyopt, <http://www.pyopt.org/>
- PyOpt Sparse, <https://bitbucket.org/mdolab/pyoptsparse>