

ED-MDAO, un composant Python pour le développement d'applications de MDAO (analyse et optimisation de conception multidisciplinaire)

Aziz JEGHAM¹, Armelle LE GALL¹, Gérald PETITJEAN¹, Charles POMBET¹,

¹ EURODECISION, 9A RUE DE LA PORTE DE BUC, 78000 VERSAILLES, FRANCE


{aziz.jegham,armelle.legall, gerald.petitjean, charles.pombet}@eurodecision.com


Mots-clés : optimisation, modèles statistiques, plans d'expériences, MDAO, Python, Scikit-learn, pymoo, XGBoost, SALib, OpenTURNS, pyDOE, seaborn.


Résumé


Certains systèmes sont trop complexes à modéliser et ne peuvent pas s'exprimer sous une forme analytique. Des simulateurs ou essais réels sont utilisés pour estimer leur performance. La MDAO (optimisation multi-disciplinaire) est le domaine scientifique qui répond aux souhaits des experts métier d'optimiser les paramètres du système pour que ce dernier satisfasse un cahier des charges. Une méthodologie classique pour la MDAO consiste à construire des modèles statistiques, un par observation/critère caractérisant le comportement du système. Des optimisations basées sur ces modèles statistiques permettent de rechercher la conception optimale. Plusieurs itérations d'optimisation sont nécessaires. Les modèles statistiques sont systématiquement réestimés pour tirer parti des nouvelles simulations effectuées.

La richesse des bibliothèques existantes disponibles depuis un environnement python nous a permis de concevoir la suite logicielle ED-MDAO couvrant les fonctionnalités suivantes :

 Génération de plans d'expériences : méthodes Monte-Carlo et quasi-Monte-Carlo, plans contraints, plans fractionnaires ...

 Modèles statistiques de substitution (classification et régression) : automatisation de la sélection de modèles, cross validation, définition de plusieurs métriques d'évaluation, gestion de plusieurs sorties en simultané, normalisation ou non des données, transformation des variables et des réponses ...

 Optimisation multi-objectifs : optimisation blackbox et algorithmes évolutionnaires, prise en compte de variables entières et qualitatives ...

 Analyse et visualisation des résultats (essais et modèles statistiques) : Morris, Sobol, corrélations ...

Nous présenterons l'architecture générale du composant ED-MDAO et les différentes bibliothèques Python open source utilisées (Scikit-learn, pymoo, XGBoost, SALib, OpenTURNS, pyDOE, seaborn).

De plus, réaliserons une démonstration du composant ED-MDAO, mettant en valeur le workflow des différentes étapes, les principales fonctionnalités et les sorties graphiques.