

# Sujet de stage Master 2

## Sélection et combinaison d’algorithmes sous contraintes de ressources

**Thématique** : sélection d’algorithmes, réglage des paramètres d’un solveur, optimisation combinatoire, algorithmes évolutionnaires, complexité, apprentissage automatique

**Équipe d’accueil** : Statistiques et Algorithmique pour la Biologie (SaAB)

**Laboratoire d’accueil** : Informatique Appliquées de Toulouse, INRAE

**Lieu** : Castanet-Tolosan (près de Toulouse), France

**Encadrants** : Samir Loudni (samir.loudni@imt-atlantique.fr), Simon de Givry (simon.de-givry@inrae.fr), Charles PRUD’HOMME (charles.prudhomme@imt-atlantique.fr)

**Gratification** :  $\approx$  550 euros / mois

### Contexte

La Programmation Par Contraintes (PPC) est une discipline de l’Intelligence Artificielle qui vise à proposer des langages de modélisation mathématique permettant d’exprimer des problèmes combinatoires et offrant des algorithmes génériques pour les résoudre. Deux solveurs open-source, CHOCO [6] en Java et TOULBAR2 [2] en C++, intègrent une collection d’algorithmes collectés depuis plus de vingt ans. La ligne de commande de TOULBAR2 contient une cinquantaine de paramètres et il existe bien d’autres réglages internes contrôlant le prétraitement des instances, les algorithmes de recherche et leurs heuristiques. Il en va de même pour le solveur CHOCO. L’amélioration des performances de ces solveurs par un meilleur réglage est une clé importante pour la réduction de leur consommation énergétique et leur potentiel *embarquabilité* (e.g., Android apps).

L’équipe TASC poursuit des recherches théoriques et appliquées en PPC qu’elle intègre au sein du solveur CHOCO (<https://choco-solver.org/>). L’équipe SaAB mène des travaux en optimisation combinatoire dans les sciences du vivant et développe le solveur TOULBAR2 (<https://miat.inrae.fr/toulbar2>). Les deux solveurs ont remporté plusieurs compétitions en PPC (<https://xcsp.org/competitions>, <https://www.minizinc.org/challenge>) et sur les modèles graphiques probabilistes (<https://uaicompetition.github.io/uci-2022>). Dans chaque compétition, les ressources de calcul sont limitées en temps et en espace.

### Sujet

L’objectif du stage est d’étudier de nouvelles stratégies de sélection et de combinaison d’algorithmes capables de traiter des jeux d’instances très hétérogènes en respectant les limites imposées sur le temps de calcul et la mémoire. Les techniques actuelles de sélection d’algorithmes [3] se heurtent au coût temporel d’une évaluation d’une stratégie donnée sur un jeu d’entraînement. Une piste à explorer consistera à extraire un ou des sous-problèmes caractéristiques du jeu d’instances pour en accélérer l’évaluation. Différentes techniques d’extraction seront considérées dont la décomposition arborescente. Un autre domaine prometteur est celui des algorithmes évolutionnaires qui visent à fournir un cadre général pour modifier le code d’un programme existant par un processus inspiré des algorithmes génétiques [5, 4, 1]. Le stage commencera par identifier les algorithmes disponibles, leurs complexités théoriques et pratiques (via du profiling) pour définir les briques élémentaires sur lesquelles tester un processus évolutionnaire visant à sélectionner et contrôler une combinaison d’algorithmes adaptée aux instances cibles. L’originalité de l’approche sera de prendre en compte dans ce contrôleur les limites imposées en temps et en espace.

Possibilité de poursuite en thèse au sein du projet ANR GMLaS (2025-2029) qui porte sur l’amélioration des techniques d’optimisation combinatoire à l’aide de la fouille de données et de l’apprentissage automatique.

### Modalités de candidature

Envoyer votre dossier de candidature (CV, notes de licence et master, lettre de motivation) par email aux trois encadrants avant le 1 décembre 2024.

### Références

- [1] Robert Andrew Bennetto. *Evolutionary search strategies in constraint programming*. PhD thesis, Stellenbosch University, 2020.
- [2] B Hurley, B O’Sullivan, D Allouche, G Katsirelos, T Schiex, M Zytnicki, and S de Givry. Multi-Language Evaluation of Exact Solvers in Graphical Model Discrete Optimization. *Constraints*, 21(3) :413–434, 2016.
- [3] Pascal Kerschke, Holger H Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection : Survey and perspectives. *Evolutionary computation*, 27(1) :3–45, 2019.
- [4] Su Nguyen, Dhananjay Thiruvady, Mengjie Zhang, and Kay Chen Tan. A genetic programming approach for evolving variable selectors in constraint programming. *IEEE Transactions on Evolutionary Computation*, 25(3) :492–507, 2021.
- [5] Justyna Petke, William B. Langdon, and Mark Harman. Applying genetic improvement to minisat. In Günther Ruhe and Yuanyuan Zhang, editors, *Search Based Software Engineering*, pages 257–262, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [6] Charles Prud’homme and Jean-Guillaume Fages. Choco-solver : A java library for constraint programming. *Journal of Open Source Software*, 7(78) :4708, 2022.