

Les métaheuristiques utilisées pour résoudre les problèmes d’optimisation difficiles ont généralement plusieurs paramètres qui doivent être définis de manière appropriée de sorte que certains aspects des performances soient optimisés. Dans ce chapitre, nous passons en revue F-Race, un algorithme de course pour la configuration automatique des algorithmes. F-Race est basé sur une approche statistique pour sélectionner la meilleure configuration parmi un ensemble de configurations candidates sous évaluations stochastiques. Nous présentons également une extension de l’algorithme F-Race initial, qui conduit à une famille d’algorithmes appelée F-Race itérée (I-Race) que nous allons utiliser plus tard dans le calibrage automatiques de paramètres des métaheuristiques que nous avons implémentés.

## 0.1 F-Race

### 0.1.1 Approche par Course (Racing)

F-Race, qui s’inspire de l’algorithme Hoeffding race [?, ?] en machine learning pour la sélection de modèles, a été proposé dans [?] et étudié de manière approfondie dans [?]. L’idée essentielle de la méthode de course est:

- d’évaluer l’ensemble des configurations de manière incrémentale sur un flux d’instances.
- dès que des preuves (statistiques) suffisantes sont rassemblées contre certains candidats, ces configurations sont rejetées et la course continue sur les candidats survivants.
- Dans F-Race, après chaque cycle d’évaluation des configuration, le test de Friedman non paramétrique est utilisé pour vérifier s’il existe des preuves qu’au moins une des configurations candidates est significativement différente des autres en termes de mesures de performance.
  - Si l’hypothèse nulle d’absence de différences est rejetée:
    - \* des comparaisons par paires entre la configuration la mieux classée et les autres configurations candidates sont effectuées.
    - \* tous les candidats qui entraînent des performances nettement inférieures à la meilleure configuration sont éliminés et n’apparaîtront pas lors du prochain cycle d’évaluation [?].

- Le processus est répété jusqu'à ce qu'il ne reste que deux candidats, et le meilleur des deux est pris comme résultat pour un problème de calibrage automatique des paramètres.

F-Race utilise la puissance de calcul plus efficacement qu'une évaluation répétée dans une approche par force brute. Il peut également arrêter le processus de recherche par lui-même, c'est-à-dire s'arrêter lorsqu'il ne reste qu'une configuration. Cependant, si l'algorithme cible a un grand nombre de paramètres et / ou chaque paramètre a une large gamme de valeurs possibles, un très grand nombre de configurations candidates doivent être évaluées pour obtenir des résultats de haute qualité. Dans de tels cas, l'adoption de F-Race pourrait devenir peu pratique ou prohibitive sur le plan des calculs.

## 0.2 I/F-Race (F-Race Itéré)

Pour atténuer ce problème, Balaprakash et al. [?] a proposé l'application itérative de la F-Race, qui est abrégée en F-Race itéré ou I / F-Race. F-Race itérée, comme son nom l'indique, utilise une procédure itérative pour trouver les configurations de paramètres optimales:

- A chaque itération:
  - tout d'abord, un ensemble de configurations est généré à partir de l'espace des paramètres selon un modèle probabiliste,
  - puis une course F standard est effectuée sur l'ensemble candidat
  - les candidats survivants sont utilisés pour mettre à jour le modèle probabiliste qui sera utilisé dans la prochaine itération.
- Ce cycle est répété jusqu'à ce que la condition d'arrêt, telle que le budget de calcul maximal, soit satisfaite.

Il est à espérer de concentrer les configurations autour de la région prometteuse dans l'espace des paramètres en utilisant des configurations élites choisies par F-Race à chaque itération pour biaiser l'échantillonnage de nouvelles configurations [?]. L'efficacité de la procédure de recherche est ainsi améliorée de cette manière.

### 0.3 Le package IRace

Le package irace implémente une procédure de course itérée, qui est une extension de la course F itérée (I / F-Race). L'utilisation principale d'irace est la configuration automatique des algorithmes d'optimisation et de décision, c'est-à-dire la recherche des paramètres les plus appropriés d'un algorithme étant donné un ensemble d'instances d'un problème. Il s'appuie sur le package de F-Race par Birattari et il est implémenté dans R. Le package irace est disponible dans [?]. La version que nous avons utilisée est irace 3.4.1.