

## Pourquoi cette UE ?

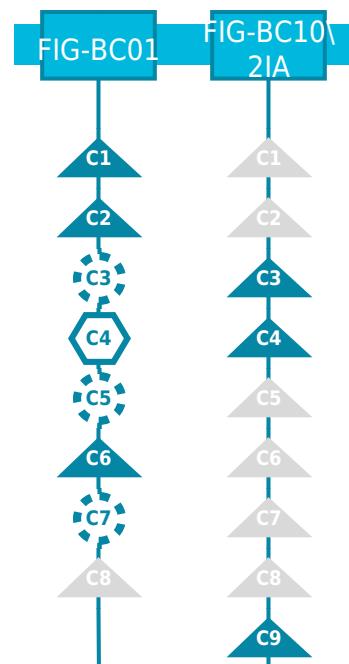
Au-delà de l'importance indéniable des techniques d'optimisation pour la résolution de nombreux problèmes, un grand nombre de problématiques liées à l'Intelligence Artificielle et en particulier à l'apprentissage machine sont aussi formulées comme des problèmes d'optimisation. Ce module propose de traiter différentes notions mathématiques importantes pour l'optimisation et l'Apprentissage machine. Il introduit plusieurs méthodes et notions importantes en Apprentissage machine. Il présente différentes approches heuristiques utiles pour la résolution de problèmes d'optimisation combinatoires NP-difficiles qui ne peuvent être résolus de manière exacte.

## Eléments constitutifs de l'UE

	coefficients	
2IAiasd_9_4-1 Approches heuristiques pour l'optimisation combinatoire	1	
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
70	50	4

Alignement curriculaire

## Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?



- BC1 L'UE contribue à ce bloc de compétences
- BC1 L'UE ne contribue pas à ce bloc de compétences
- C1 Compétence non adressée dans cette UE
- C1 Compétence mise en œuvre dans cette UE
- C1 Compétence enseignée dans cette UE
- C1 Compétence évaluée dans cette UE
- C1 Compétence enseignée et évaluée dans cette UE

## 2IAisd\_9\_4-1 Approches heuristiques pour l'optimisation combinatoire

**Contexte et enjeux de l'enseignement**

De nombreux problèmes d'optimisation combinatoires mènent à une explosion combinatoire qui rend impossible une résolution « exacte », c'est notamment le cas des problèmes NP-complets. Ainsi, pour résoudre ces problèmes pour des instances de grande taille, il est nécessaire d'employer des méthodes « métaheuristiques » (ou « approchées » en opposition à « exactes »), permettant de prendre des raccourcis dans l'exploration (on parle de configurations, de voisinage et d'exploration d'un espace de recherche).

**Prise en compte des dimensions socio-environnementales****Modalités d'enseignement et d'évaluation**

	Nb d'heures
Cours	8
Cours intégré (cours + TD)	
TD	
TP	12
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	
Travail personnel	10

**Prérequis**

Algorithmique et complexité (2IA-8.2)

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Développer la compétence des élèves ingénieurs :  
 - à implémenter une méthode réaliste de résolution de problèmes réels à variables discrètes,  
 - à évaluer la qualité des solutions obtenues et l'efficacité, en termes de ressources CPU, de la méthode.

**Activités**

(CM, TD, TP, projet, sortie terrain, etc.)

Les séances de cours présenteront les concepts principaux sur lesquels reposent ces heuristiques. Les TP consisteront à l'implémentation et l'expérimentation d'une approche de ce type pour traiter un ensemble d'instances d'un problème combinatoire donné (benchmark).  
 Le découpage est prévu comme suit :  
 - 8h de cours découpés en 2 chapitres principaux ;  
 - 12h de mise en application.

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Comptes-rendus de TP notés

Retour sur l'évaluation fait à l'élève : 3 semaines après la dernière séance.

## Plan de cours

1. Recherche locale : concepts principaux et application.
2. Algorithmes évolutionnaires et bio-inspirés: concepts principaux et application.

## Ressources et références

Supports de cours et références bibliographiques

**Contexte et enjeux de l'enseignement**

At a time of unprecedented growth in artificial intelligence applications, it's easy to get lost in the mass of "ready-to-use" tools. Yet many of these tools are based on the enormous work of applied mathematics. In particular, mathematical optimization and the panoply of associated software tools are the basis of a very large number of machine learning tools. The aim of this course is to explain the contribution of these mathematical tools in order to better understand their behavior and results.

**Prise en compte des dimensions socio-environnementales****Prérequis**

Calculus: Mathematical Analysis, Linear Algebra, Python, Operational Research. Module 8.2 : Algorithmique Optimisation

**Modalités d'enseignement et d'évaluation**

	Nb d'heures
Cours	18
Cours intégré (cours + TD)	
TD	14
TP	
Projets	8
Travail en autonomie encadré	8
Contrôles et soutenances	2
Travail personnel	40

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Know how to formulate a learning problem as a mathematical optimization problem; recognize the main characteristics of the problems in order to choose the right solver; apply these solvers to significant cases.

**Activités**

(CM, TD, TP, projet, sortie terrain, etc. )

The TD will be carried out on students' personal computers.  
The breakdown is as follows:  
- 20h lectures;  
- 18h application of various theoretical concepts.  
- 10h to finalize the mini-projects that accompany the course

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

1. Contrôle écrit sur papier
2. Contrôle continu pendant les travaux dirigés
3. Réalisation et rendu d'exercices significatifs

## Plan de cours

- Convex Optimization;
- Duality, Support Vector Machines and important samples;
- Regularization, Big dimension problems and important features; Sparse Learning.
- Generative methods, parametric probability estimation

## Ressources et références

Polycopie of lectures and indications for Python software needs