

## Description du module

# Algorithmique

**Généralités****Nombres de crédits ECTS**

3

**Sigle du module**

TSM\_Alq

**Version**

10.10.2015

**Responsable du module**

Eric Taillard, HES-SO

**Langue**

	Lausanne	Berne	Zurich
Enseignement	<input type="checkbox"/> E <input checked="" type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input checked="" type="checkbox"/> D <input type="checkbox"/> E
Documentation	<input checked="" type="checkbox"/> E <input checked="" type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input checked="" type="checkbox"/> E
Questions d'examen	<input type="checkbox"/> E <input checked="" type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input checked="" type="checkbox"/> E

**Catégorie du module**

- Bases théoriques élargies
- Approfondissement technique et scientifique
- Modules de savoirs contextuels

**Périodes**

- 2 périodes d'enseignement frontal et une période d'exercice par semaine
- 2 périodes d'enseignement frontal par semaine

**Breve description /Explication des objectifs et du contenu du module en quelques phrases**

Ce module présente différentes catégories d'algorithmes avancés ainsi que leurs domaines d'application typiques. La première partie du module approfondira les connaissances sur les structures de données qui permettent de gérer efficacement des ensembles de données très grands, complexes ou dynamiques, voire combinant ces trois caractéristiques. À l'issue du module, les étudiants seront capables de sélectionner le meilleur algorithme et de l'appliquer à des tâches telles que l'indexation, la recherche, l'extraction, l'insertion ou la mise à jour de données volumineuses, comme celles que l'on rencontre dans les systèmes d'informations géographiques, les hypertextes, l'intelligence artificielle ou en simulation de molécules. La seconde partie du module présentera des techniques de base pour la conception d'algorithmes appliqués à des problèmes d'optimisation difficiles. La combinaison de ces techniques de base —modélisation, construction de solution, amélioration de solution, décomposition de problème, apprentissage— conduit à des méta-heuristiques classiques comme les algorithmes génétiques, les colonies de fourmi artificielles ou la recherche avec tabous. À l'issue du module, les étudiants seront capables de concevoir et d'appliquer ces techniques à des problèmes d'optimisation difficiles.

**Objectifs, contenu et méthodes****Objectifs d'apprentissage et compétences visées**

Ce module donne une vue d'ensemble diverses classes d'algorithmes fréquemment utilisés en pratique. Sur la base de connaissances solides, l'étudiant est capable de concevoir et d'implanter les algorithmes les plus efficaces et les plus appropriés pour ses propres applications. L'étudiant :

- connaît différentes catégories d'algorithmes avancés et leurs domaines d'application ;
- a une bonne connaissance en structures de données avancées et leur utilisation pour gérer efficacement des données volumineuses, complexes et/ou dynamiques ;
- est en mesure d'évaluer quels algorithmes sont appropriés pour réaliser efficacement certaines tâches telles que l'indexation, la recherche, l'extraction, l'insertion ou la mise à jour de données volumineuses ;
- connaît les algorithmes dynamiques utilisés en robotique, en intelligence artificielle ou en sciences moléculaires.

**Contenu du module avec pondération des contenus d'enseignement**

- Géométrie computationnelle et structures de données multidimensionnelles Pondération: 50%
  - algorithmes géométriques
  - géométrie computationnelle
  - structures de données et algorithmes multidimensionnels
- Algorithmes basés sur les méta-heuristiques Pondération: 50%
  - Méthodes constructives
  - Recherches locales
  - Méthodes d'apprentissage
  - Méta-heuristiques classiques: GRASP, Colonies de fourmis artificielles, recherche avec tabous, recuit simulé, méthodes de bruitage, algorithmes génétiques, essais particuliers, etc.

semaine	matière
1	<b>Partie I : Géométrie computationnelle</b> Rappels : <ul style="list-style-type: none"> <li>• Notation asymptotique</li> <li>• Complexité d'algorithmes récursifs</li> <li>• Structures de données de base (tableau, liste, pile, queue, arbre binaire, tas, table de hachage)</li> </ul>
2	Rappels : <ul style="list-style-type: none"> <li>• Algorithmes de base (tri, indexation et recherche monodimensionnelles)</li> <li>• Graphes et réseaux</li> <li>• Algèbre linéaire</li> </ul>
3	<ul style="list-style-type: none"> <li>• Problème introductif : visibilité (carte de visibilité, polygones, opérations booléennes, intersection de segments, problèmes numériques)</li> <li>• Objets élémentaires et primitives (points, segments, polygones, produits scalaire et vectoriel, intersection, bibliothèques existantes)</li> </ul>
4	<ul style="list-style-type: none"> <li>• Technique de balayage (paire de points les plus proches, détection des intersections de segments)</li> <li>• Enveloppe convexe (un algorithme complet pour les points du plan, survol d'autres paradigmes d'algorithmes et applications)</li> </ul>
5	<ul style="list-style-type: none"> <li>• Arbre de requête d'énumération pour recherches orthogonales, arbre Kd (exemple de structure multi-niveaux)</li> <li>• Arbre de quadrants</li> </ul>
6	<ul style="list-style-type: none"> <li>• Arbre d'intervalles pour segments de lignes horizontaux</li> <li>• Liste d'arcs doublement chaînés (DCEL) et application au problème de superposition de cartes</li> </ul>
7	<ul style="list-style-type: none"> <li>• Triangulation de polygones (survol et algorithme pour les polygones monotones)</li> <li>• Définition de diagrammes de Voronoï, triangulation de Delaunay, applications</li> </ul>
8	<b>Partie II Métaheuristiques</b> Introduction et rappels <ul style="list-style-type: none"> <li>• Problèmes de base en graphes et réseaux</li> <li>• Arbres, chemins et flots optimaux, affectation linéaire</li> <li>• Modélisation de problèmes d'optimisation combinatoire</li> <li>• Problèmes difficiles : voyageur de commerce, arbre de Steiner, affectation quadratique, coloration de graphes, ordonnancement</li> </ul>
9	Méthodes constructives <ul style="list-style-type: none"> <li>• Construction aléatoire</li> <li>• Construction gloutonne</li> <li>• Recherche en faisceaux, méthode pilote</li> </ul>
10	Recherches locales <ul style="list-style-type: none"> <li>• Structure de voisinage, mouvements</li> <li>• Limitation du voisinage</li> <li>• Extension du voisinage, recherche en éventail, chaîne d'éjections</li> </ul>
11	Méthodes randomisées <ul style="list-style-type: none"> <li>• Acceptation à seuil, algorithmes du grand déluge et du démon</li> <li>• Recuit simulé</li> <li>• Méthodes de bruitage</li> <li>• GRASP</li> <li>• Recherche à voisinage variable</li> </ul>
12	Méthodes de décomposition <ul style="list-style-type: none"> <li>• Recherche dans de grands voisinages</li> <li>• Matheuristiques</li> <li>• POPMUSIC</li> </ul> Méthodes d'apprentissage pour la construction de solutions <ul style="list-style-type: none"> <li>• Colonies de fourmis artificielles</li> <li>• Construction de vocabulaire</li> </ul>
13	Méthodes d'apprentissage pour la modification de solutions <ul style="list-style-type: none"> <li>• Recherche avec tabous</li> </ul>

14	Méthodes à population de solutions <ul style="list-style-type: none"><li>• Algorithmes génétiques</li><li>• Recherche par dispersion</li><li>• GRASP avec chemin de liaison</li><li>• Essaims particulaires</li></ul>
----	---

Le module sera divisé en deux thèmes (géométrie computationnelle et méta-heuristiques) présentés par le même professeur.

#### Méthodes d'enseignement et d'apprentissage

- Enseignement ex-cathedra
- Présentation et discussion d'études de cas
- Exercices théoriques et travaux de programmation à la maison

#### Connaissances et compétences prérequis

L'étudiant dispose de connaissances pratiques en :

- géométrie, algèbre linéaire, algorithmes (classification, recherche, hachage) et structures de données (structures linéaires, structures arborescence)
- connaissances de base de la théorie des graphes (structures de données et algorithmes)
- complexité algorithmique, logique, théorie des probabilités

Ces connaissances font partie de tout bon livre d'introduction sur les algorithmes. Par exemple, les chapitres 1-12, 15-17, 22-26, 28-29, 34-35 de [Cormen 09] couvrent parfaitement les pré-requis de ce cours.

#### Bibliographie

[de Berg 08] : M. de Berg, G. Cheong, M. van Kreveld, M. Overmars. *Computational Geometry : Algorithms and Applications*, Springer, Third Edition 2008.

[Cormen 09] : Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms*, third edition, MIT Press, 2009.

Traduction française : *Introduction à l'algorithmique*, 2<sup>e</sup> édition, Dunod, 2002.

[Dréo 03] : J. Dréo, A. Pérowski, P. Siarry, É. D. Taillard, [Métaheuristiques pour l'optimisation difficile](#), ISBN 2-212-11368-4, Eyrolles, 2003

[Holger 04] : Holger H. Hoos & Thomas Stützle *Stochastic Local Search: Foundations and Applications*, Morgan Kaufmann / Elsevier, 2004.

[Taillard 15] Introduction aux méta-heuristiques, à paraître, 2015.

#### Mode d'évaluation

##### Conditions d'admission aux examens de fin de module (tests exigés)

##### Examen écrit de fin de module

Durée de l'examen: 120 minutes  
Moyens autorisés: Livres, transparents (théorie)