Module Description, available in: EN, FR

Algorithms

General Information

Number of ECTS Credits

3

Module code

TSM_Alg

Valid for academic year

2019-20

Last modification

2017-04-05

Coordinator of the module

Éric Taillard (HES-SO, eric.taillard@heig-vd.ch)

Explanations regarding the language definitions for each location:

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Berne	Lausanne			Lugano	Zurich		
Instruction			**X** F 100%			**X** E 100%		
Documentation			**X** F 50%	**X** E 50%		**X** E 100%		
Examination			**X** F 100%			**X** E 100%		

Module Category

TSM Technical scientific module

Lessons

2 lecture periods and 1 tutorial period per week

Entry level competences

Prerequisites, previous knowledge

The student has working knowledge of:

- Geometry, linear algebra, algorithms (sorting, searching, hashing) and data structures (linear structures, trees)
- Basics of graph data structures and algorithms
- Algorithmic complexity, logic, probability theory.

These topics are generally contained in books introducing algorithms. For instance, chapters 1-12, 15-17, 22-26, 28-29, 34-35 of (Cormen 09) covers very well the prerequisites

Brief course description of module objectives and content

This module introduces students with different categories of advanced algorithms and typical application areas.
In the first part of the module, the students will have a sound understanding of data structures and algorithms for efficiently handling either very large, complex or dynamic data sets or combinations thereof. They will be able to evaluate suitable algorithms and to apply them to typical tasks such as efficiently indexing, searching, retrieving, inserting or updating data such as large volumes of hypertext or spatial data.
The students will be familiar with dynamic algorithms used, for example, in artificial intelligence or molecular sciences.
The second part of the module will present basic techniques for designing algorithms for hard combinatorial optimization problems. The combination of these basic components —problem modeling, problem decomposition, solution building, solution improvement, learning— lead to classical metaheuristics like genetic algorithms, ant colonies or tabu search. The students will be able to design new algorithms for hard combinatorial optimization problems and to apply them.

Aims, content, methods

Learning objectives and acquired competencies

This module gives the student an overview of frequently used algorithms classes. Based on this strong foundation, students can design and implement the most suitable and efficient algorithms for use in their own applications. The student:

- is familiar with different categories of advanced algorithms and with typical application areas;
- has a sound understanding of data structures and algorithms for efficiently handling either very large, complex or dynamic data sets or combinations thereof;
- is able to evaluate suitable algorithms and to apply them to typical tasks such as efficiently indexing, searching, retrieving, inserting or updating data, including data types such as large volumes of hypertext or spatial data;
- is familiar with dynamic algorithms used in robotics, artificial intelligence or molecular sciences.

Contents of module with emphasis on teaching content

Computational Geometry and Multi-dimensional Data Structures. Weight 50%

- geometric algorithms
- computational geometry algorithms
- multidimensional data structures and algorithms

Metaheuristic-based algorithms. Weight 50%

- Constructive methods
- Local searches
- Decomposition techniques
- Learning techniques
- Classical metaheuristics: GRASP, ant colonies, tabu search, simulated annealing, noising methods, genetic algorithms

Part I Computational geometry Reminder: • Asymptotical notation • Complexity of recursive algorithms • Basic data structures (table, list, stack, queue, binary tree, heap, hashing table) • Basic algorithms (sorting, 1D indexing and searching) • Graphs and networks, planar graphs, doubly-connected edge list (DCEL) • Linear algebra (point, vector, dot product, cross product)
Introduction to Computational Geometry • Introductory problem : visibility map, polygons and boolean operations, line intersection, numerical problems • Basic objects and primitives (points, segments, polygons, rays, lines) • Existing CG software and libraries.
Construction paradigms • Incremental construction (line arrangements, line segment intersection, overlay problem) • Divide and conquer (convex hull for points in the plane) • Plane sweep technique (closest pair problem, intersections detection in segments sets)
Range Searching • Kd and Range trees for orthogonal range searching (example of "multi-level structure") • Quadtrees
Windowing • Interval tree for horizontal line segments • Priority search tree
Voronoï Diagram, Delaunay triangulation • Computation of Voronoï Diagram • Terrain and randomized algorithm for Delaunay triangulation
Part II Metaheuristics

Introduction and reminder: • Basic problems and algorithms for graphs and networks • Optimal trees, paths and flows, linear assignment • Combinatorial optimization, complexity theory, problem modelling and utility function • Hard problems: Travelling salesman, Steiner tree, Quadratic assignment, Graph colouring, Scheduling
Constructive methods • Random building • Greedy construction • Beam search, Pilot method
Local searches • Neighbourhood structure, moves, 2-opt and 3-opt for the travelling salesman problem • Neighbourhood limitation • Neighbourhood extension, Lin-Kernighan neighbourhood for the travelling salesman problem, fan and filter
Randomized methods • Threshold accepting, great deluge and demon algorithms • Simulated annealing • Noising methods • GRASP • Variable neighbourhood search
Decomposition methods • Large neighbourhood search, POPMUSIC
Learning methods for solution building • Artificial ant systems • Vocabulary building
Learning methods for iterative local search • Tabu search
Methods with a population of solutions • Genetic algorithms • Scatter Search • GRASP with path relinking

Teaching and learning methods

- Ex-cathedra teaching
- Presentation and discussion of case studies
- Theory and programming exercises

Literature

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

T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms, third edition, MIT Press, 2009.

P. Siarry (ed.), Metaheuristics, ISBN 978-3-319-45403-0, Springer, 2016.

H. H. Hoos, Th. Stützle Stochastic Local Search: Foundations and Applications, Morgan Kaufmann / Elsevier, 2004.

É. Taillard Introductions aux méta-heuristiques, WWW, 2015-2018

Assessment

Certification requirements

Module does not use certification requirements

Basic principle for exams

As a rule, all the standard final exams for modules and also all resit exams are to be in written form

Standard final exam for a module and written resit exam

Kind of exam

written

Duration of exam

120 minutes

Permissible aids

Aids permitted as specified below:

Permissible electronic aids

None

Other permissible aids

Books, copy of slides (solutions to exercises excluded)

Special case: Resit exam as oral exam

Kind of exam

oral

Duration of exam

30 minutes

Permissible aids

No aids permitted

Description du module, disponible en: EN, FR

Algorithmique

Informations générales

Nombre de crédits ECTS

3

Code du module

TSM_Alg

Valable pour l'année académique

2019-20

Dernière modification

2017-04-05

Coordinateur/coordinatrice du module

Éric Taillard (HES-SO, eric.taillard@heig-vd.ch)

Explication des définitions de langue par lieu :

- Les cours se dérouleront dans la langue définie ci-dessous par lieu/exécution.
- Les documents sont disponibles dans les langues définies ci-dessous. Pour le multilinguisme, voir la répartition en pourcentage (100% = documents complets)
- L'examen est disponible à 100% dans chaque langue sélectionnée pour chaque lieu/exécution.

	Berne	Lausanne			Lugano	Zurich		
Leçons			**X** F 100%			**X** E 100%		
Documentation			**X** F 50%	**X** E 50%		**X** E 100%		
Examen			**X** F 100%			**X** E 100%		

Catégorie de module

TSM approfondissement technico-scientifique

Leçons

2 leçons et 1 leçon de pratique par semaine

Compétences préalables

Connaissances préalables, compétences initiales

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 (2009) couvrent parfaitement les pré-requis de ce cours.

Brève description du contenu et des objectifs

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 ou en intelligence artificielle.
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 métaheuristiques 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, contenus, méthodes

Objectifs d'apprentissage, compétences à acquérir

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. À l'issue du module, 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 des modules avec pondération du contenu des cours

Géométrie computationnelle et structures de données multidimensionnelles. Pondération: 50%

- algorithmes géométriques
- géométrie computationnelle
- structures de données et algorithmes multidimensionnels

Algorithmes basés sur les métaheuristiques. Pondération: 50%

- Méthodes constructives
- Recherches locales
- Techniques de décomposition
- Méthodes d'apprentissage
- Métaheuristiques classiques: GRASP, Colonies de fourmis artificielles, recherche avec tabous, recuit simulé, méthodes de bruitage, algorithmes génétiques, etc.

Partie I : Géométrie computationnelle Rappels : • Notation asymptotique • Complexité d'algorithmes récursifs • Structures de données de base (tableau, liste, pile, queue, arbre binaire, tas, table de hachage) • Algorithmes de base (tri, indexation et recherche monodimensionnelles) • Graphes et réseaux, graphes planaires, liste d'arcs doublement connectés (DCEL) • Algèbre linéaire (point, vecteur, produit scalaire, produit vectoriel)
Introduction à la géométrie computationnelle : • Problème introductif : visibilité (carte de visibilité, polygones, opérations booléennes, intersection de segments, problèmes numériques) • Objets élémentaires et primitives (points, segments, polygones, produits scalaire et vectoriel, intersection) • Bibliothèques logicielles de géométrie computationnelles
Paradigmes de construction : • Construction incrémentale (arrangement de ligne, intersection de segments, problème de superposition) • Diviser pour régner (enveloppe convexe de points du plan) • Technique de balayage (paire de points les plus proches, détection des intersections de segments)
Recherches énumératives • Arbre de requête d'énumération et arbre KD pour recherches orthogonales (exemple de structure multi-niveaux) • Arbre de quadrants
Fenêtrage • Arbre d'intervalles pour segments de lignes horizontaux • Arbre de recherche de priorité
Diagramme de Voronoï, triangulation de Delaunay • Calcul du diagramme de Voronoï • Terrain et algorithme randomisé pour le calcul d'une triangulation de Delaunay

Partie II Métaheuristiques

Introduction et rappels
- Problèmes de base en graphes et réseaux
- Arbres, chemins et flots optimaux, affectation linéaire
- Optimisation combinatoire, théorie de la complexité, modélisation et fonction-utilité
- Problèmes difficiles : voyageur de commerce, arbre de Steiner, affectation quadratique, coloration de graphes, ordonnancement

Méthodes constructives
- Construction aléatoire
- Construction gloutonne
- Recherche en faisceaux, méthode pilote

Recherches locales
- Structure de voisinage, mouvements, 2-opt et 3-opt pour le voyageur de commerce
- Limitation du voisinage
- Extension du voisinage, recherche en éventail, chaîne d'éjections, voisinage de Lin-Kernighan pour le voyageur de commerce

Méthodes randomisées
- Acceptation à seuil, algorithmes du grand déluge et du démon
- Recuit simulé
- Méthodes de bruitage
- GRASP
- Recherche à voisinage variable

Méthodes de décomposition
- Recherche dans de grands voisinages
- POPMUSIC

Méthodes d'apprentissage pour la construction de solutions
- Colonies de fourmis artificielles
- Construction de vocabulaire

Méthodes d'apprentissage pour la modification de solutions
- Recherche avec tabous

Méthodes à population de solutions
- Algorithmes génétiques
- Recherche par dispersion
- GRASP avec chemin de liaison

Méthodes d'enseignement et d'apprentissage

- Enseignement ex-cathedra
- Présentation et discussion d'études de cas
- Exercices théoriques et de programmation

Bibliographie

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

T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms, third edition, MIT Press, 2009.

Traduction française : Introduction à l'algorithmique, 3e édition, Dunod, 2010.

P. Siarry (ed.), Métaheuristiques, EAN13 : 9782212139297, Eyrolles 2014.

H. H. Hoos, Th. Stützle Stochastic Local Search: Foundations and Applications, Morgan Kaufmann / Elsevier, 2004.

É. Taillard Introductions aux méta-heuristiques, WWW, 2015-2018

Evaluation

Conditions d'admission

Le module n'utilise pas de conditions d'admission.

Principe pour les examens

En règle générale, tous les examens de fin de module réguliers et les examens de rattrapage sont organisés sous la forme écrite

Examen de fin de module régulier et examen écrit de répétition

Type de l'examen

écrit

Durée de l'examen

120 minutes

Aides autorisées

Les aides suivantes sont autorisées:

Aides électroniques autorisées

Aucun

Autres aides autorisées

Livres, copie des transparents (sauf solutions aux exercices)

Cas spécial: examen de répétition oral

Type de l'examen

oral

Durée de l'examen

30 minutes

Aides autorisées

Sans aides