

---

# PROGRAMME INGÉNIEUR

2024-2025

2e année / 3e année

---

## Option Disciplinaire Informatique pour l'Intelligence Artificielle

OD INFOIA

---

RESPONSABLE DU PROGRAMME

Didier LIME



INGÉNIEUR - OD INFOIA

# 1er Semestre

Unité d'Enseignement	Crédits ECTS	Parcours	Acronyme	Libellé
UE 73	12	Tronc commun	ALGOA INDUR PAPY STASC	Algorithmique avancée Informatique durable Programmation avancée en Python Introduction à l'apprentissage statistique avec Python
UE 74	13	Tronc commun	AGATH DEEP PAMEF PIIA1 QCM	Théorie et algorithmique des jeux Apprentissage profond Parallélisme et méthodes formelles Projet 1 Qualité, conception, modélisation

# 2e Semestre

---

Unité d'Enseignement	Crédits ECTS	Parcours	Acronyme	Libellé
UE 83	14	Tronc commun	GPGPU GRAAL MPAR PIA2 PRLOG	Programmation sur processeur graphique Graphes et algorithmes Modélisation probabiliste et apprentissage par renforcement Projet 2 Programmation logique

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 73 / 93

## Algorithmique avancée [ALGOA]

Responsable(s) du cours : Didier LIME

### Pré-requis

### Objectifs

À la fin de ce cours les étudiants et étudiantes sauront :

1. Analyser rigoureusement des algorithmes :
  - déterminer leur complexité
  - prouver leurs propriétés fonctionnelles et non-fonctionnelles
2. Concevoir des algorithmes efficaces en :
  - choisissant des structures de données adaptées
  - utilisant des méthodes génériques de résolution

### Plan de l'enseignement

Le cours est divisé en trois grandes parties :

1. Analyse de programmes
  - propriétés de base des algorithmes : terminaison, correction, complétude, complexité
  - preuves inductives et invariants
  - modèles de calcul de complexité
  - décidabilité et complexité des problèmes algorithmiques
  - complexité en pratique : pire cas, meilleur cas, algorithmes récursifs
  - complexité en moyenne et algorithmes probabilistes
2. Paradigmes de conception d'algorithmes
  - énumération exhaustive
  - backtracking
  - diviser pour régner
  - programmation dynamique
  - algorithmes gloutons
  - transformations de problèmes
3. Structures de données
  - tableaux et listes
  - complexité amortie
  - files et piles
  - files de priorités, tas
  - arbres binaires de recherche et arbres AVL
  - tables de hachage

Les TP permettront de mettre tout cela en pratique au travers de la conception et du développement d'un outil de compression de fichiers.

### Bibliographie

Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2009) [1990]. Introduction to Algorithms (3rd ed.).

MIT Press and McGraw-Hill. ISBN 0-262-03384-4. 1320 pp.

## Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	18 hrs	0 hrs	12 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 73 / 93

## Informatique durable [INDUR]

Responsable(s) du cours : Morgan MAGNIN

### Pré-requis

Connaissance de base de l'informatique (méthodologie et technologies)

### Objectifs

Objectifs en termes de connaissance :

1. Règlementation et éthique :
  - Cadre juridique en matière de traitement automatique de données
  - Éthique et responsabilité
    - Les biais possible de l'intelligence artificielle (IA)
    - Agents autonomes et éthique
    - Raisonnement moral et responsabilité éthique en IA
    - Enjeux sociaux
2. Informatique verte :
  - Identifier les principes de l'informatique verte et les leviers associés
  - Enjeux et impacts directs et indirects
  - Consommation énergétique matérielle et logicielle
  - Sobriété numérique
    - Algorithmique et logiciel vert : l'éco-conception logicielle
    - Centres de données verts

Objectifs en termes de compétences :

1. Savoir construire une analyse éthique et responsable des traitements automatiques de l'information (identification de l'impact de ces traitements, des biais possibles, etc.)
2. Savoir mettre en oeuvre un diagnostic de la consommation énergétique des applications informatiques

### Plan de l'enseignement

Plan de cours :

1. Introduction générale aux concepts
2. Mise en problématique sur quelques cas d'étude introduits au début du cours, et repris à la fin pour compléter les connaissances acquises dans le domaine de compétences dans l'analyse et mise en oeuvre de solutions plus responsables et durables
3. Série de cours/conférences/interventions sur les thèmes « informatique verte » et « éthiques et responsabilité »

### Bibliographie

- Cerna Collectif. Éthique de la recherche en apprentissage machine. [Rapport de recherche] CERNA; ALLISTENE. 2017, pp.51. hal-01643281 <https://hal.inria.fr/hal-01643281/document>
- Panorama de formations et de ressources pédagogiques existantes sur le thème « informatique verte » : <https://ecoinfo.cnrs.fr/2019/06/21/formations-abordant-les-aspects-environnementaux-du-numerique/>
- MOOC « Impacts environnementaux du numérique » <https://www.fun-mooc.fr/fr/cours/impacts-environnementaux-du-numerique/>
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6), 1-35. <https://arxiv.org/pdf/1908.09635>
- Acar, Hayri. Software development methodology in a Green IT environment. PhD thesis. Université de Lyon, 2017. <https://tel.archives-ouvertes.fr/tel-01724069/file/TH2017ACARHAYRI.pdf>

## Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	12 hrs	10 hrs	8 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 73 / 93

## Programmation avancée en Python [PAPY]

*Responsable(s) du cours : Lucas LESTANDI*

### Pré-requis

### Objectifs

1. Mettre en place un environnement python robuste ainsi qu'un environnement de développement adapté.
2. Ecrire des programmes python efficaces utilisant les bibliothèques et abstractions modernes.
3. Assurer la lisibilité et la maintenance en suivant les principes de structuration modulaire et de style cohérents avec celles de la communauté.
4. Créer un paquet pour la distribution sur d'autres machines y compris des dépendances.

### Plan de l'enseignement

Bien travailler avec python

- choisir un environnement de travail: IDE, notebooks jupyter, environnement python (conda,...)
- philosophie et syntaxe de python
- l'interpréteur python
- bonnes pratiques en programmation python (PEP8,...)

2. Structuration et types de données

- Variables, références et gestion de la mémoire
- Types de données et structures
- Programmation orienté objet (OOP) : les classes
- Ecrire du code robuste : architecture, introspection, exceptions, etc.

3. Programmer avec des modules

- les modules natifs : os, sys, subprocess,...
- les bibliothèques externes : avec pip ou conda
- quelques exemples communs : numpy, scipy, matplotlib
- créer ses propres modules

Pour aller plus loin

- les décorateurs
- communication avec d'autres langages (C++,...)
- tester et debugger
- distribution et portabilité des paquets

### Bibliographie

### Évaluation

Évaluation collective : EVC 1 (coefficient 0.5)

Évaluation individuelle : EVI 1 (coefficient 0.5)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	8 hrs	0 hrs	22 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 73 / 93

## Introduction à l'apprentissage statistique avec Python [STASC]

Responsable(s) du cours : Bertrand MICHEL

### Pré-requis

### Objectifs

Ce cours est une introduction à l'apprentissage statistique.

Objectifs :

- compréhension de la problématique de l'apprentissage statistique
- compréhension des méthodes standards
- pratique avec les libraires standards de Python

### Plan de l'enseignement

- introduction à l'apprentissage statistique
- rappels de statistique
- méthodes usuelles en classification
- méthodes CART, Forêts aléatoire et boosting
- méthodes non supervisées
- introduction des méthodes à noyau

### Bibliographie

- The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Trevor Hastie Robert Tibshirani Jerome Friedman, 2009 Springer.
- Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron, O'Reilly 2017.

### Évaluation

Évaluation collective : EVC 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	15 hrs	17 hrs	0 hrs	0 hrs	0 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 74 / 94

## Théorie et algorithmique des jeux [AGATH]

*Responsable(s) du cours : Didier LIME*

### Pré-requis

### Objectifs

À la fin de ce cours les étudiants et étudiantes sauront :

- modéliser les problèmes de décision impliquant plusieurs agents dans des environnements variés sous forme de jeux stratégiques,
- déterminer algorithmiquement les stratégies permettant le meilleur gain personnel ou le meilleur équilibre global.
- programmer des intelligences artificielles basées sur ces concepts

### Plan de l'enseignement

Le cours est divisé en trois grandes parties :

1. Jeux en forme normale
  - jeux, gains, et stratégies
  - concepts de solutions : stratégies dominées, équilibres de Nash, minimisation du regret maximum, équilibres corrélés
  - jeux à deux joueurs à somme nulle
2. Jeux en forme extensive
  - jeux séquentiels
  - équilibres parfaits en sous-jeux
  - induction en arrière,
  - approximations: Monte Carlo et évaluation statique
3. Jeux répétés
  - jeux répétés de manière finie ou infinie
  - apprentissage basé sur le regret

### Bibliographie

- Ken Binmore. *Playing for Real : A Text on Game Theory*, OUP USA, 2007.
- *Algorithmic Game Theory*, Nisan, Roughgarden, Tardos, and Vazirani, Cambridge University Press, 2007.
- *Multiagent Systems*, Y. Shoham, K. Leyton-Brown, Cambridge University Press, 2009.
- Michael Maschler, Eilon Solan, Shmuel Zamir. *Game Theory*, Cambridge University Press, 2013.

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	16 hrs	0 hrs	14 hrs	0 hrs	2 hrs

## INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 74 / 94

# Apprentissage profond [DEEP]

Responsable(s) du cours : Bertrand MICHEL / Didier LIME

## Pré-requis

## Objectifs

À la fin de ce cours les étudiants et étudiantes connaîtront le fonctionnement théorique et algorithmique des réseaux de neurones feed-forward, ainsi que les principaux éléments de la théorie de différents représentants de la famille des réseaux profonds (réseaux récurrents, auto-encodeurs, réseaux génératifs).

Ils et elles seront également familiarisé.es avec leur utilisation en pratique à l'aide de bibliothèques dédiées.

Enfin, ils et elles verront en comment ces modèles peuvent être appliqués pour la résolution de problèmes réels, via plusieurs cas d'études.

## Plan de l'enseignement

Le cours est divisé en quatre grandes parties :

### 1. Réseaux feed-forward:

- Calcul en avant
- Optimisation et apprentissage
- Modèles convolutionnels
- Principes du transfer learning et du fine tuning

### 2. Modèles récurrents

- Modèles de type Long term Short Term Memory (LSTM)
- Modèles pour le traitement du langage naturel: embeddings, attention, et transformers

### 3. Auto-encodeurs et modèles génératifs

- Auto-encodeurs
- Réseaux antagonistes génératifs (GAN)

### 4. Cas d'étude

- Véhicule autonome;
- Applications biomédicales.

## Bibliographie

- The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Trevor Hastie Robert Tibshirani Jerome Friedman, 2009 Springer.

- Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Géron, O'Reilly 2017.

## Évaluation

Évaluation collective : EVC 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	17 hrs	7 hrs	8 hrs	0 hrs	0 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 74 / 94

## Parallélisme et méthodes formelles [PAMEF]

Responsable(s) du cours : Olivier ROUX

### Pré-requis

### Objectifs

À la fin de cet enseignement, les étudiants connaîtront et sauront appliquer les principes de la programmation sur processeurs graphiques (GPU) pour accélérer et optimiser des calculs. Ils seront capable d'implémenter des algorithmes parallèles en prenant en considération la localité de la mémoire et des données ainsi que les spécificités matérielles du flot de contrôle sur les GPU. La partie pratique permettra aux étudiants d'implémenter, de valider et d'évaluer les algorithmes classiques de la programmation parallèle afin d'en optimiser les performances.

### Plan de l'enseignement

Le cours sera décomposé en quatre grandes parties :

1. Généralités sur la programmation sur GPU : — programmation concurrente vs parallèle — principe et architecture des GPU
2. Opérateurs, mémoires et structurations des données : — allocation des données et flot de contrôle — types de mémoire et localité des données — opérations atomiques et synchronisation
3. Etude de patterns d'algorithme pour la programmation parallèles
4. Mise en oeuvre de leurs connaissance sur l'optimisation d'un calcul de réseau de neurones

### Bibliographie

- Programming massively parallel processors : a hands-on approach (3rd Edition), David Kirk et Wen-mei W. Hwu, Morgan Kaufmann, 2017
- CUDA par l'exemple : une introduction à la programmation parallèle de GPU, Jason Sanders et Edward Kandrot, Pearson, 2011
- Nvidia Online training <https://developer.nvidia.com/cuda-education>

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	20 hrs	0 hrs	10 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 74 / 94

## Projet 1 [PIIA1]

*Responsable(s) du cours : Didier LIME*

### Pré-requis

### Objectifs

Mettre en œuvre les connaissances et compétences acquises tout au long de l'année.

### Plan de l'enseignement

Début fin septembre, fin début janvier. Peut éventuellement être fusionné avec le 2e projet (PIIA2)

### Bibliographie

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	1	0 hrs	0 hrs	0 hrs	32 hrs	0 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 1er Semestre - UE 74 / 94

## Qualité, conception, modélisation [QCM]

*Responsable(s) du cours : Myriam SERVIÈRES*

### Pré-requis

### Objectifs

Le cours couvre trois thématiques clés du développement informatique en entreprise : le Génie Logiciel, les Bases de Données, et les Méthodes du Développement Informatique. Les objectifs selon ses trois thématiques sont, respectivement :

- Acquérir les bases du Génie Logiciel et de la Gestion de Projet informatique.
- Maîtriser les éléments de conception et d'utilisation des bases de données relationnelles.
- Développer en maîtrisant des outils et méthodes qui permettent de construire des applications logicielles de qualité industrielle.

À la suite de ce cours, les compétences acquises devront permettre de :

- Concevoir et modéliser un logiciel et rédiger un cahier des charges.
- Maîtriser les éléments de conception et d'utilisation des bases de données relationnelles.
- Collaborer pour le développement du logiciel, automatiser de suites de tests, et garantir la qualité du code.

### Plan de l'enseignement

1. Le génie logiciel. Nous aborderons ici :

- Cycles de développement logiciel : cahier des charges, cycle de vie, planning, qualité, spécifications, production, recette.
- Modèles de conception UML : cas d'utilisation, diagrammes de classes, diagrammes de séquence, diagrammes d'états-transitions, et diagrammes d'activité.

2. Les bases de données

- Modélisation Conceptuelle et Physique, Modèle Relationnel
- Requêtes SQL
- Conception de programmes d'interrogation d'une base de données

3. Les méthodes du développement informatique.

Nous aborderons ici des outils de gestion de version, les tests unitaires, et les métriques de code.

### Bibliographie

- Modélisation objet avec UML, Pierre-Alain Muller, Best of Eyrolles, 2005.
- UML2 et les design patterns, Craig Larman, Pearson Education, 2005.
- Software Engineering 8, Ian Sommerville, Addison Wesley, 2007.
- Le génie logiciel et ses applications, Ian Sommerville, InterEdition, 1988.
- Processus d'ingénieries du logiciel, méthodes et qualité, Claude Pinet, Pearson Education, 2002.
- UML2, Benoit Charroux, Aomar Osmani, Yann Thierry-Mieg, Pearson Education, 2005.

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	9 hrs	8 hrs	13 hrs	0 hrs	2 hrs

## INGÉNIEUR - OD INFOIA

2e année / 3e année - 2e Semestre - UE 103 / 83

# Programmation sur processeur graphique [GPGPU]

Responsable(s) du cours : Pierre-Emmanuel HLADIK

### Pré-requis

### Objectifs

Intégrer les concepts de l'informatique parallèle (communication et synchronisation),  
Comprendre comment ces principes peuvent être mis en œuvre,  
Entendre des mécanismes de plus haut niveau,  
Pratiquer pour comprendre les spécificités de la concurrence,  
S'initier aux preuves de propriétés des systèmes concurrents.

### Plan de l'enseignement

- 1 Généralités
- 2 Introduction : pourquoi des programmes parallèles ?
- 3 La synchronisation par variables partagées
- 4 La synchronisation par la communication
- 5 Vérification de propriétés des systèmes informatiques (répartis)
- 6 Logiques temporelles, vérifications sur modèle
- 7 Vérifications temporelles et vérifications temporisées
- 8 Conclusion
- 9 Bibliographie

### Bibliographie

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	18 hrs	0 hrs	23 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 2e Semestre - UE 103 / 83

## Graphes et algorithmes [GRAAL]

*Responsable(s) du cours : Didier LIME / Loig JEZEQUEL*

### Pré-requis

### Objectifs

Sous forme de séminaires ciblés sur des sujets variés ayant pour point commun de traiter de graphes, ce cours vise à donner un aperçu de l'utilité pratique de ces modèles et des algorithmes, parfois complexes, mis en œuvre pour les analyser.

### Plan de l'enseignement

Dans un premier temps, quelques notions et algorithmes de base sur les graphes seront présentés.

Ensuite, un ensemble de sujets sera proposé et chaque étudiant (en groupe) devra préparer un séminaire sur l'un de ces sujets et le présenter au reste de la classe. Voici une liste non exhaustive des sujets qui pourront être abordés : représenter graphiquement les grands graphes ; systèmes multi-agents : exploration collaborative d'un graphe ; recherche de chemins, algorithme A ; construire le graphe d'Internet ; marches aléatoires sur les graphes ; graphes planaires et mineurs exclus ; graphes petit-mondes ; réécriture de graphes. L'un des sujets sera traité par l'enseignant du cours pour donner un exemple de ce qui est attendu.

### Bibliographie

- E. Goodaire, M. Parmenter. Discrete Mathematics with Graph Theory. 2018.
- D. Beauquier, J. Berstel, P. Chrétienne. Eléments d'algorithmique. 1992. (disponible en ligne)
- M. Gondran, M. Minoux. Graphes et algorithmes. 2009.
- J.-C. Fournier. Théorie des graphes et applications. 2006.

### Évaluation

Évaluation collective : EVC 1 (coefficient 0.5)

Évaluation individuelle : EVI 1 (coefficient 0.5)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	16 hrs	16 hrs	0 hrs	0 hrs	0 hrs

## INGÉNIEUR - OD INFOIA

2e année / 3e année - 2e Semestre - UE 103 / 83

# Modélisation probabiliste et apprentissage par renforcement [MPAR]

Responsable(s) du cours : Benoit DELAHAYE / Didier LIME

## Pré-requis

---

## Objectifs

---

Dans une première partie du cours, les étudiantes et étudiants apprendront à maîtriser les techniques de modélisation probabilistes à états tels que les chaînes de Markov, les processus de décision Markoviens et leurs extensions temporisées et à information imparfaite, ainsi que les techniques formelles de vérification qui leur sont dédiées.

Dans une deuxième partie, les étudiantes et étudiants découvriront les principes de l'apprentissage par renforcement pour ces modèles, ainsi que plusieurs techniques permettant de s'y attaquer.

Les méthodes et techniques de ce cours seront illustrées et évaluées lors de travaux dirigés sur machine en langage Python.

## Plan de l'enseignement

---

Plan de cours :

Le cours est divisé en trois grandes parties :

- I. Modélisation probabiliste
  1. Chaîne de Markov discrètes (DTMC)
  2. Processus de décision Markoviens (MDP)
  3. Extensions à information imparfaite (HMC et PO-MDP)
- II. Vérification de modèles probabilistes
  1. Model-Checking probabiliste
  2. Model-Checking statistique
- III. Apprentissage par Renforcement
  1. Bandit Manchot et MDP
  2. Programmation dynamique (DP)
  3. Méthodes de Monte-Carlo
  4. Différence temporelle (TD)

## Bibliographie

---

- Christel Baier and Joost-Pieter Katoen. Principles of model checking. MIT press, 2008.
- Pedro R. D'Argenio, Arnd Hartmanns, and Sean Sedwards. Lightweight statistical model checking in nondeterministic continuous time. In Tiziana Margaria and Bernhard Steffen, editors, ISoLA'18, volume 11245 of LNCS, pages 336–353. Springer, 2018.
- David Henriques, Joao G Martins, Paolo Zuliani, André Platzer, and Edmund M Clarke. Statistical model-checking for markov decision processes. In 2012 Ninth international conference on quantitative evaluation of systems, pages 84–93. IEEE, 2012.
- Vikram Krishnamurthy. Partially observed Markov decision processes. Cambridge university press, 2016.
- Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking : An overview. In International conference on runtime verification, pages 122–135. Springer, 2010.
- Martin L Puterman. Markov decision processes. Handbooks in operations research and management science, 2 :331–434, 1990.
- Richard S Sutton and Andrew G Barto. Reinforcement learning : An introduction. MIT press, 2018.

- Koushik Sen, Mahesh Viswanathan, and Gul Agha. On statistical model checking of stochastic systems. In International Conference on Computer Aided Verification, pages 266–280. Springer, 2005.
- Marco A Wiering and Martijn Van Otterlo. Reinforcement learning, volume 12. Springer, 2012.

## Évaluation

Évaluation collective : EVC 1 (coefficient 0.5)

Évaluation individuelle : EVI 1 (coefficient 0.5)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	16 hrs	6 hrs	8 hrs	0 hrs	2 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 2e Semestre - UE 103 / 83

## Projet 2 [PIIA2]

Responsable(s) du cours : Didier LIME

### Pré-requis

### Objectifs

Mettre en œuvre les connaissances et compétences acquises tout au long de l'année.

### Plan de l'enseignement

De janvier à fin mars. Peut éventuellement être fusionné avec le 1er projet (PIIA1)

### Bibliographie

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	2	0 hrs	0 hrs	0 hrs	48 hrs	0 hrs

# INGÉNIEUR - OD INFOIA

2e année / 3e année - 2e Semestre - UE 103 / 83

## Programmation logique [PRLOG]

Responsable(s) du cours : Carito GUZIOLOWSKI

### Pré-requis

### Objectifs

Ce cours consiste à proposer de solutions à un problème (notamment de recherche combinatoire) en modélisant ce problème, au lieu de coder sa solution. La modélisation se fait sous forme de règles logiques qui ont une syntaxe et sémantique définie, nous utiliserons pour cela la programmation par ensemble réponses (ou Answer Set Programming, ASP). La résolution se fait avec des solveurs très puissantes, nous utiliserons gringo et clasp.

### Plan de l'enseignement

Le plan à suivre pour illustrer les principales notions de ce paradigme de programmation est :

- Programmation Déclarative
- Syntaxe de l'ASP
- Sémantique de l'ASP
- Modélisation d'un problème

Les TDs et TP seront articulés autour de quelques exemples de problèmes de recherche combinatoire classiques.

### Bibliographie

Knowledge representation, Reasoning, and Declarative Problem Solving. Chitta Baral. Cambridge University Press New York, NY, USA, 2003.

Gelfond, M., & Kahl, Y. (2014). Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139342124

### Évaluation

Évaluation individuelle : EVI 1 (coefficient 1)

LANGUE DU COURS	CRÉDITS ECTS	COURS MAGISTRAUX	TRAVAUX DIRIGÉS	TRAVAUX PRATIQUES	PROJET	DEVOIRS SURVEILLÉS
Français	3	20 hrs	0 hrs	10 hrs	0 hrs	2 hrs