Université de Strasbourg

Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie

Mémoire présenté en vue de l'obtention de

l' Habilitation à Diriger des Recherches

de

l'Université de Strasbourg

Contributions à une approche épistémologique de l'intelligence artificielle

Spécialité : Informatique

Mémoire présenté par

Anne Jeannin-Girardon

Préparé au laboratoire ICube UMR 7357
Équipe CSTB (*Complex Systems and Translational Bioinformatics*)

Soutenu le 09/11/2022 devant le jury présidé par Pierre De Loor

Composition du jury :

Dr Yann	Guermeur	DR CNRS	Université de Nancy	Rapporteur
Dr Jean	Sallantin	DR CNRS émérite	Université de Montpellier	Rapporteur
Pr Charles	Tijus	Professeur émérite	Université Paris 8	Rapporteur
Pr Pierre	De Loor	Professeur	ENI Brest	Examinateur
Dr Paul	Bourgine	DR émérite	École Polytechnique	Examinateur
Pr Pierre	Collet	Professeur	Université de Strasbourg	Garant

Constamment et, s'il est possible, à toute représentation, appliquer les principes de la science de la nature, de celle des passions et de la dialectique.

Marc Aurèle, *Pensées pour moi-même.*
(Traduction de Mario Meunier)

Remerciements

Le travail synthétisé dans ce document a été rendu possible grâce à beaucoup de personnes dont j'ai croisé la route ; je voudrais essayer de les remercier ici.

En premier lieu, j'adresse ma gratitude aux membres de mon jury. Vous avez pris du temps pour rapporter ou examiner mon travail, et m'avez fait le plaisir d'être tous présents à ma soutenance, ce qui relève de la rareté passée l'année 2020 ! Ainsi, je souhaite remercier M. Yann Guermeur, M. Jean Sallantin et M. Charles Tijus d'avoir endossé les rôles de rapporteurs, ainsi que M. Paul Bourgine d'avoir accepté d'examiner mon travail. Merci également à M. Pierre De Loor qui a lui aussi examiné mon travail, et qui a en outre accepté de présider ce jury.

Je tiens aussi à remercier Pierre Collet, Julie Thompson et Aline Deruyver de m'avoir donné la possibilité d'encadrer différents étudiants en thèse, en m'ouvrant la porte de vos projets ou en acceptant d'entrer dans les miens. À tous mes collègues de l'équipe CSTB : merci pour votre accueil chaleureux et votre gentillesse au quotidien ; la qualité des rapports humains est essentielle pour se sentir bien dans le métier qui est le nôtre et qui, somme toute, est souvent très prenant et parfois difficile.

Étant aussi enseignante, je n'oublie pas non plus l'ensemble des collègues de l'UFR de mathématique et d'informatique qui m'ont eux aussi accueillie avec beaucoup de bienveillance. L'enseignement et les tâches gravitant autour de cette mission prennent un temps considérable et il n'est pas toujours facile de concilier ces missions avec nos activités de recherche ; de fait, le soutien des collègues est une aide précieuse. Merci en particulier à Basile Sauvage et Arash Habibi. Basile m'a donné de très précieux conseils pour m'aider à m'organiser pour reprendre ou monter des cours, et la bienveillance et le soutien de Arash permettent de continuer à avancer dans les moments de doute.

Nous ne sommes pas grand-chose sans nos étudiants. Si leur formation fait partie intégrante de nos missions, elles et eux aussi nous forment. Je suis très heureuse d'avoir pu participer, à divers degrés, à l'encadrement de doctorants. Bravo à celles et ceux qui ont soutenu leur thèse : Nicolas, Anna et Romain.
Courage à celles et ceux qui ont encore du chemin à parcourir : Hiba *(tu nous as montré avec brio ta capacité à t'emparer d'un sujet de recherche et à le faire tien ; aujourd'hui tu entames la dernière ligne droite)*, Quentin *(l'informaticien doublé d'un philosophe en herbe ! Le Cercle de Vienne et Karl Popper n'ont plus de secret pour toi, ni l'histoire des grands nombres en médecine)* et Ali *(although you have only recently begun, you have already shown us your potential, keep it up !)*.

Petit *up* pour Romain, qui, parmi les jeunes docteurs que j'ai pu accompagner jusqu'ici, a le plus subi mes interférences ☺. Si, en fin de compte, le déroulement d'une thèse dépend autant de facteurs intrinsèques qu'extrinsèques, je n'en suis pas moins très heureuse d'avoir eu la chance de marcher à tes côtés ces dernières années, Romain. J'espère que ça n'est que le début du chemin.

De même, je remercie l'ensemble des étudiants ayant accepté de faire un bout de leur route avec moi, au travers d'un stage de fin de licence, d'un travail d'étude et de recherche, de leur stage de fin d'études, ou bien dans le cadre de leur formation en apprentissage. Merci également à l'ensemble des étudiants dont je croise la route à l'occasion de mes enseignements : quel que soit votre niveau d'études, vous ne manquez pas de me challenger et de me donner envie de progresser dans ma pratique.

Enfin, je remercie mes parents du fond du cœur. Mon père, en plus d'être une personne dotée d'un savoir considérable et de pas moins de sagesse, une personne curieuse de tout et cherchant à comprendre le monde tant dans son fonctionnement matériel que social, s'est appliqué à partager avec ses enfants ce goût de questionner et d'apprendre. Ma mère, qui nous a déjà quittés, aurait été, j'en suis sûre, très heureuse de voir l'étape professionnelle qu'est mon HDR se concrétiser. Elle et mon père m'ont offert un soutien inconditionnel depuis toujours, je leur dédie le travail présenté ici.

Avant-propos

Les travaux présentés dans ce document prennent racine dans une volonté d'aller vers des systèmes d'intelligence artificielle (IA) plus autonomes, explicables et éthiques par l'utilisation conjointe du calcul évolutionnaire et de techniques d'IA plus répandues comme les réseaux de neurones artificiels, tout en s'attachant à associer à notre démarche une approche philosophique et épistémologique. Cette idée, un peu naïve au départ, marquait simplement une volonté de notre part de tirer parti du meilleur de différentes approches d'apprentissage automatique dans un cadre unifié. S'il nous reste encore du travail pour parvenir à cette unification, cela nous a conduits à nous interroger sur la nature des énoncés produits par ces approches variées de l'intelligence artificielle pour mieux en appréhender les possibilités et limites.

Ainsi, ce document est divisé en trois parties :

— La première partie nous permet de situer le contexte de nos travaux et aborde, à travers deux chapitres, les réflexions qui se sont présentées lorsque nous avons voulu examiner la caractérisation de la nature des énoncés produits par des IA et leur impact lorsqu'on les exploite.

— La deuxième partie présente trois travaux de thèse que nous avons conduits ces dernières années. Différents problèmes à résoudre et différentes approches de résolution sont proposés et permettent d'illustrer leur bien-fondé ou leur limite. Le dernier chapitre constitue une synthèse et une mise en perspective des réflexions que nous avons pu mener dans ce contexte.

— La troisième partie est plus académique et présente une vue globale de mes activités pédagogiques et de recherche depuis mon recrutement comme maîtresse de conférences à l'université de Strasbourg en septembre 2016.

Table des matières

Première partie

Contexte scientifique

CHAPITRE 1

Introduction

[Let us] consider the idea of building an induction machine. Placed in a "simplified" world [...] such a machine may through repetition "learn", or even "formulate", laws of succession which hold in its "world".

Karl Popper, *Conjectures and refutations* (1962)

Alors que le nom « intelligence artificielle »[1] a émergé dans les années 50, à l'issue de la fameuse conférence de Dartmouth de 1956 qui a réuni les grands pionniers de l'époque dans ce champ de recherche naissant, il a fallu bien des espoirs déçus et des déconvenues avant d'arriver aux systèmes déployés à large échelle aujourd'hui dans des domaines aussi variés — et souvent critiques — que la finance, la santé, le recrutement, la justice, la recommandation de contenus, la conduite autonome, les jeux vidéos, *etc.*

Si une telle dissémination est possible, c'est parce que la quantité de données que nous générons tous les jours permet de nourrir les algorithmes, toujours plus gourmands, derrière ces systèmes intelligents. Loin d'être aléatoires, ces données renferment au contraire des régularités non triviales — portant donc une certaine

[1] Il me semble nécessaire de préciser d'emblée que les intelligences artificielles dont il est question dans ce document sont bien des systèmes intelligents qui disposent d'un certain degré d'autonomie, mais qui ne relèvent pas de ce que l'on appelle l'« intelligence artificielle généraliste ». Nous parlerons de la même façon d'intelligence artificielle ou d'apprentissage automatique de manière interchangeable : ce qui nous intéresse dans le présent document est l'ensemble des approches capables d'effectuer un apprentissage sur des données.

signification — pouvant être mises en lumière et exploitées par les algorithmes d'apprentissage automatique et en particulier les algorithmes d'apprentissage profond. Le livre blanc de l'intelligence artificielle, publié en février 2020 par la Commission européenne (Union Européenne, 2020), prédit que d'ici à 2025, la quantité de données disponibles dans le monde va atteindre 175 zêta-octets : le festin, pour les algorithmes d'apprentissage automatique, est loin d'être terminé.

Par ailleurs, ces systèmes d'intelligence artificielle se sont répandus rapidement au cours de la dernière décennie, pour l'essentiel sans régulation. Cela étant dit, on voit émerger aujourd'hui de plus en plus de questionnements autour de ces systèmes utilisés à si large échelle, de la part de citoyens, de chercheurs, de sociétés savantes ou encore de gouvernements. Un des textes qui a probablement fait le plus date depuis quelques années concernant la régulation du traitement automatique des données est le règlement européen sur la protection des données (Règlement RGPD, 2016). On trouve notamment, dans le préambule de ce texte, l'article 71 dans lequel on peut lire le passage suivant :

> *La personne concernée devrait avoir le droit de ne pas faire l'objet d'une décision [...] qui est prise sur le seul fondement d'un traitement automatisé et qui produit des effets juridiques la concernant ou qui, de façon similaire, l'affecte de manière significative, tels que le rejet automatique d'une demande de crédit en ligne ou des pratiques de recrutement en ligne sans aucune intervention humaine.[...]. [La personne concernée devrait avoir le droit] [...] d'obtenir une explication quant à la décision prise à l'issue de ce type d'évaluation et de contester la décision.*

Ce seul article 71 est dense et contient de nombreux éléments concernant le consentement des personnes soumises à des traitements automatisés ou encore la transparence attendue de tels systèmes et la prévention de biais (sociaux, ethniques, sexistes, *etc.*) pouvant survenir par un traitement automatique des données. Il est clair que l'on ne parle pas ici spécifiquement d'algorithmes d'intelligence artificielle, mais plus largement de toute approche de traitement automatique de données. Le problème étant que, lorsqu'une intelligence artificielle procède à ce traitement automatisé, les questions d'explicabilité, de transparence ou encore de prévention des biais sont somme toute assez complexes à aborder.

Depuis 2018, l'Union européenne (UE) s'attaque à l'intégration de l'intelligence artificielle dans nos sociétés par son « approche européenne de l'intelligence artificielle

axée sur l'excellence et la confiance » (Approche Européenne pour l'IA, 2018) et a depuis proposé un certain nombre de documents pour commencer à cadrer le domaine de l'intelligence artificielle, ses impacts et de grandes lignes directrices pour son développement et son utilisation. L'UE n'est pas la seule à se préoccuper de ces questions, loin de là. On peut par exemple citer la société savante IEEE Standard Association qui a édité un long rapport titré « *Ethically aligned design : a vision for prioritizing human well-being with autonomous and intelligent systems* »[2], l'institut Mila au Canada, dont les recherches sont focalisées autour de l'idée d'une intelligence artificielle bénéfique[3], le Klein Center à Harvard, États-Unis, et leur programme sur l'éthique et la gouvernance des IA[4] ou encore le Future of Life Institute[5].

L'article prospectif de Chui et al. (2018) est un exemple de travail de recherche visant à s'interroger sur la construction d'une « *good AI society* ». Les auteurs caractérisent notamment 10 domaines à impact social dans lesquels l'intelligence artificielle pourrait jouer un rôle bénéfique : *Sécurité et justice*, *équité et inclusion* ou encore *réponse de crise*. Dans chaque domaine, différentes questions spécifiques sont soulevées, comme *Comment l'intelligence artificielle pourrait-elle prédire la survenue d'épidémies ?*

De nombreux autres travaux dans ce sens ont vu le jour ces dernières années ; ce que nous en retenons, c'est qu'ils soulèvent deux questions essentielles dans la question de l'intégration bénéfique de l'intelligence artificielle dans nos sociétés :

1. Comment les intelligences artificielles peuvent-elles contribuer à l'amélioration de nos sociétés ?

2. Comment pouvons-nous nous assurer que ces systèmes soient transparents et répondent à des normes éthiques ?

Ces deux questions sont inextricablement liées et aborder la première demande nécessairement d'aborder en premier lieu la question de la transparence et de l'aspect éthique de l'intelligence artificielle.

Pour traiter ces questions, il est nécessaire d'identifier les principaux écueils de ces systèmes, mais, avant cela, il est d'abord intéressant de définir rapidement ce qu'est « l'intelligence artificielle ».

[2] https://ethicsinaction.ieee.org/
[3] https://mila.quebec/en/ai-society/
[4] https://cyber.harvard.edu/publication/2020/principled-ai
[5] https://futureoflife.org/background/benefits-risks-of-artificial-intelligence/

Qu'est-ce qu'une intelligence artificielle ?

Il convient de mieux cadrer les systèmes d'intérêts dans ce document. Il ne s'agit toutefois *pas* de démarrer une discussion sans fin qui, d'emblée, place ce que l'on qualifierait d'« intelligence humaine » au sommet d'une échelle hiérarchique (fallacieuse) et qui viserait à situer, sur cette échelle, différents « types » d'intelligence, dont l'intelligence artificielle.

Qu'est-ce, alors, que l'intelligence ? Un ensemble de propriétés, comme l'autonomie, l'adaptabilité ? Un ensemble de capacités cognitives, comme la mémoire, le raisonnement, l'attention, *etc.* ? Finalement, pour les besoins de ce document (et pour nous écarter autant que possible de biais anthropomorphiques, mais aussi pour ne pas nous égarer dans des cadres plus élaborés —comme la théorie des intelligences multiples de Gardner (1987)— qui n'apporteront finalement que peu de choses à notre propos), la définition suivante, tirée de l'encyclopédie en ligne Wikipédia[6], sera amplement suffisante :

> *L'intelligence peut être perçue comme la capacité à traiter l'information pour atteindre des objectifs.*

Bien entendu, les propriétés d'autonomie ou d'adaptabilité, les facultés cognitives comme le raisonnement ou la perception sont parties prenantes de ce processus et nous y reviendrons plus loin dans ce document.

Le terme « artificiel » réfère simplement à un système non organique (voire non physique, au sens où l'on peut simplement considérer un programme informatique capable de ce traitement de l'information afin de résoudre une tâche). Nous proposons donc une définition énoncée comme suit :

Intelligence artificielle

Système non organique mettant en œuvre un certain nombre de facultés cognitives et exhibant, au moins dans une certaine mesure, de l'autonomie et de l'adaptabilité et qui est ainsi capable de traiter des informations afin d'atteindre un ou des objectifs.

Même si cela est discutable, on suppose que les objectifs sont prédéfinis.

Cette définition simple permet de s'affranchir de questionnements plus profonds sur la nature de l'intelligence et qui, en l'état actuel de nos travaux, n'apporteront

[6] https://fr.wikipedia.org/wiki/Intelligence

pas de valeur ajoutée. Ayant convenu que les intelligences artificielles sont en effet intelligentes, on peut raisonnablement penser que celles-ci peuvent contribuer à la construction des connaissances, mais avec quelles conséquences ?

Enjeux sociaux

Un avertissement s'impose : si le tableau de l'intelligence artificielle dépeint dans cette section est plutôt sombre, l'idée n'est pas de rejeter ces systèmes, ni même d'arguer en leur défaveur en choisissant à dessein de s'attarder sur des problèmes causés par l'IA. Le but est uniquement de mettre en évidence les principaux leviers qu'il convient d'ajuster pour que l'intégration de celle-ci dans nos sociétés se fasse au bénéfice de toutes et tous.

Il nous faut d'abord reconnaître que des IA sont capables de ce que l'on pourrait appeler des prouesses[7] : le modèle AlphaStar (Vinyals et al., 2019) est capable de jouer (avec succès) au jeu de stratégie en temps réel StarCraft II ; AlphaFold (Jumper et al., 2021) peut prédire la structure 3D de protéines en quelques jours, là où plusieurs mois (au mieux) de travail sont nécessaires pour déterminer la structure 3D d'une unique protéine.

En 2017, des chercheurs d'Oxford ont développé LipNet, un modèle capable de lire sur les lèvres à la volée et qui obtient des performances supérieures à celles d'humains professionnels (12.4% de mots identifiés sans aucune erreur pour les humains contre 46.8% pour LipNet) (Chung et al., 2017). Ces résultats ont été critiqués (Assael et al., 2016), notamment parce que les données utilisées pour entraîner le modèle sont connues pour présenter des limitations importantes. On peut aussi sérieusement se questionner sur l'utilisation qui peut être faite d'un tel système, notamment en se remémorant une scène du film *2001 : l'odyssée de l'espace* de Stanley Kubrick, où l'ordinateur de bord HAL[8], se trouvant sonorement isolé des protagonistes, lit sur leurs lèvres et comprend que ceux-ci souhaitent le débrancher[9].

Si cet exemple relève du film d'anticipation, la profusion des caméras de vidéo-surveillance (on pourrait aussi parler des webcams installées sur nos ordinateurs, téléphones, *etc.*) et le contrôle que certains gouvernements exercent ou essayent d'exercer sur leurs citoyens peut rendre ce genre de technologie très attractive ; mais sans même aller jusqu'à prêter à un gouvernement de telles intentions, la surveillance

[7] Quand on pense que ce ne sont « que » des programmes informatiques
[8] *Heuristically programmed ALgorithmic computer*

par vision pourrait être « enrichie » avec un tel système (ce genre de système étant déjà utilisé ; par exemple le New York Times a rapporté, en mai 2022, qu'une étudiante avait été accusée de triche par une IA pendant un examen à distance (Hill, 2022)).

Sans aller jusque dans ce type de dystopie, les exemples de dérapages des IA ne manquent pas. Quelques exemples notables incluent Tay, un *bot* Twitter de Microsoft qui, en 2016 s'est illustré en « devenant » fasciste et misogyne en moins de 24h (Vincent, 2016) ; un système de reconnaissance visuelle qui a identifié un visage sur la publicité d'un bus comme étant une personne traversant la chaussée en dehors d'un passage pour piétons (Liao, 2018) ; le système Faceception, dont les concepteurs prétendent qu'il prédit la personnalité des gens (et leur QI...) en analysant leur visage (Lubin, 2016) ; ou encore COMPAS (*Correctional Offender Management Profiling for Alternative Sanctions*), un outil de prédiction de risques de récidive utilisé dans le système judiciaire aux États-Unis et dont personne ne sait vraiment comment les scores de risques sont calculés (un résumé en trois actes : le système serait biaisé envers les personnes de couleur (Larson et al., 2016) ; réponse de Northpointe (devenu aujourd'hui *equivant*), l'entreprise développant le logiciel : non, le système n'est pas biaisé (equivant, 2018) ; finalement on ne sait pas vraiment, mais en tout cas l'historique criminel de la personne ne semble pas peser lourd dans l'estimation du risque (Rudin et al., 2020)).

On peut d'ores et déjà identifier trois sources principales à ces problèmes : les données en elles-mêmes, la conception des algorithmes et l'utilisation que l'on en fait. Pour commencer à aborder ces questions, le domaine de l'intelligence artificielle *explicable* (XAI, *eXplainable Artificial Intelligence*) s'est fortement développé depuis 2016. Plus largement, ce sont des questions *éthiques* qui se posent ici. Mais alors, que font les systèmes d'IA pour appréhender leur environnement et résoudre des tâches ? Comment pouvons-nous comprendre leur représentation du monde ? Construisent-ils réellement du savoir ? S'interroger sur ces questions demande, en amont, de s'interroger sur la nature de ces représentations, si l'on peut leur donner du sens et comment.

[9] La réaction de HAL est, à mon sens, largement surinterprétée : là où beaucoup de monde a vu dans sa réaction une forme de sentience, d'une volonté propre de ne pas « mourir », on peut aussi supposer qu'il s'agit simplement, pour HAL, de s'assurer que les objectifs de la mission seront atteints (cette explication a, en outre, le mérite d'être plus parcimonieuse).

Quelle construction du savoir par l'intelligence artificielle ?

En 2015, une équipe de biologistes de l'université de Tufts a publié un article présentant un modèle des mécanismes de régénération de vers plats découvert par une IA (Lobo and Levin, 2015). Derrière ces mécanismes de régénération, un réseau de régulation suffisamment complexe pour qu'une modélisation humaine ne permette pas de révéler les dynamiques d'autorégulation du système. Un tel espace de recherche peut en principe être exploré plus efficacement par un programme. Une « théorie » de la régénération a donc été mise en lumière par une IA. Mais est-ce correct d'utiliser le terme « théorie » ? Une théorie permet de décrire un phénomène, de révéler les interactions entre les différents composants d'un système et, en fin de compte, permet d'obtenir une compréhension de l'objet ou du phénomène en question. Cette théorie de la régénération permet-elle une telle description ? L'approche proposée par les auteurs dans ce travail repose sur le calcul évolutionnaire, une branche de l'intelligence artificielle qui s'appuie sur des mécanismes de la théorie de l'évolution et de la sélection naturelle de Darwin afin de faire évoluer des populations d'individus représentant des réseaux de régulation biologique. Le processus en lui-même est relativement transparent (les individus les plus adaptés peuvent avoir une descendance et des mutations aléatoires peuvent survenir dans le génome des individus, ceux-ci pouvant donc développer de nouveaux traits) et la sortie du processus est un réseau de régulation (avec une topologie, des interactions de régulation, *etc.*). Il se pourrait donc que la « théorie » construite par le modèle soit, au moins en partie, fondée. Cependant, parmi les « théories » qu'un algorithme évolutionnaire peut construire et qui donnent de bons résultats sur le problème traité, comment savoir quelle est la « vraie » théorie, à supposer qu'elle puisse bien avoir été capturée par le modèle ? Il s'agit là d'un problème épistémologique qui n'a pas attendu l'ère de l'intelligence artificielle pour se poser.

AlphaFold (Jumper et al., 2021) est capable de prédire la structure 3D d'une protéine : on peut supposer qu'il y a, dans le mécanisme mis au point par ce modèle, une forme de théorie décrivant ce mécanisme de repliement de protéines. On pourrait se satisfaire de cela si nous n'étions pas aussi curieux du *comment* et *pourquoi* les choses sont ce qu'elles sont. Or, le modèle derrière AlphaFold est un réseau de neurones profond. Ces classes de modèles sont qualifiées de « boîtes noires » du fait de leur opacité : il ne s'agit ni plus ni moins qu'une composée de composées … de

composées de fonctions non linéaires de millions (si ce n'est plus, beaucoup plus) de paramètres. Ainsi, si la (une) théorie a bien été capturée par ce modèle, elle ne nous est de toute façon pas accessible. Ensuite, du fait de la nature inductive de ce genre de modèle, il est bien possible que la théorie ne soit en réalité pas capturée : c'est ce que l'on observe souvent, lorsque de tels modèles sont utilisés en inférence et que l'on se rend compte que ceux-ci ne généralisent en fait pas leurs résultats à des données autres que celles utilisées pour les entraîner ; nous avons ici une autre facette du problème épistémologique évoqué ci-dessus.

Au premier abord, il semblerait que ces algorithmes (autant ceux qui apprennent à identifier des chats dans des images que ceux capables de prédire la structure 3D de protéines) ne soient pas grand-chose de plus que des machines à induire : reposant sur des algorithmes d'apprentissage profond, ces modèles sont nourris de masses de données du domaine d'intérêt (des chats ou des alignements multiples de séquences biologiques) et apprennent par un processus de répétition. Or, on sait les problèmes causés par l'induction dans la construction du savoir.

Avant de nous interroger sur la validité des modèles à un tel niveau, nous pouvons en premier lieu nous interroger les *énoncés* produits par ces IA : quelle est leur nature ? Comment pouvons-nous les comprendre et les utiliser ? C'est ce qui nous intéresse dans les travaux décrits dans ce document.

Organisation du mémoire

Notre objectif est de nous interroger sur les différentes manières possibles de construire artificiellement des connaissances ; de nous intéresser, donc, à une épistémologie de l'IA. On parle bien ici d'une « construction de savoirs » par des systèmes intelligents artificiels, et non pas d'une interrogation plus profonde sur la nature même de ces systèmes artificiels. Ainsi, de la même manière que l'approche épistémologique d'un énoncé comme « la terre tourne autour du soleil » interroge sur les raisons de penser que la terre tourne effectivement autour du soleil et pourquoi ces raisons sont admissibles, nous souhaitons nous interroger sur les énoncés produits par des intelligences artificielles et, un peu plus largement, sur les impacts qu'ont de tels énoncés d'un point de vue éthique sur celles et ceux qui les utilisent.

Pour aborder ces questions, nous proposons dans le chapitre I.2 de cadrer plus spécifiquement les deux points qui vont nous intéresser : l'explication ou la justification des énoncés produits par des IA et, plus largement, certaines considérations éthiques

de l'intégration de ces IA dans nos sociétés. Dans les chapitres de la partie II, nous faisons la synthèse de différents travaux que nous avons conduit ces dernières années qui concernent l'utilisation de différentes approches d'intelligence artificielle pour apprendre à résoudre des tâches. Nous nous sommes en particulier intéressés :

— à l'utilisation d'approches évolutionnaires pour l'optimisation de systèmes de réfrigération magnétocalorique (partie II, chapitre 3) ;

— au développement d'une IA plus autonome et explicable en environnements incertains avec des systèmes de classeurs (partie II, chapitre 4) ;

— à des approches d'apprentissage profond pour détecter des erreurs dans des alignements multiples de séquences de gènes ainsi qu'à une approche quantitative de l'explication de réseaux de neurones profonds (partie II, chapitre 5).

Le chapitre 6 de la partie II présente des pistes de recherche que nous souhaitons explorer pour donner suite aux différents travaux que nous avons menés et synthétisés ici, toujours dans cette optique de contribuer à développer une approche épistémologique de l'intelligence artificielle.

CHAPITRE 2

Intelligence artificielle, explicabilité, éthique : état des lieux

Assurons-nous bien du fait avant de nous inquiéter de la cause

Bernard Le Bouyer de Fontenelle

1 Des IA et des écueils

1.1 Problème lié au mal-apprentissage

Il était une fois un cheval qui s'appelait *Clever Hans*. Hans était très intelligent : il pouvait par exemple résoudre des tâches d'arithmétique, épeler des mots ou encore lire, et répondait aux questions qui lui étaient posées en tapant du sabot. Et malin, il l'était, mais pas comme on pourrait l'imaginer. Plutôt que de résoudre ces tâches cognitives en apprenant à compter ou lire, Hans a appris à interpréter les signaux comportementaux des humains autour de lui, car les comportements et postures des personnes l'interrogeant (et qui connaissent les réponses aux questions posées) lui permettaient simplement de savoir quand arrêter de taper du sabot : quand Hans se rapprochait de la bonne réponse, il observait un gain de tension chez la personne l'interrogeant ; tension qui s'abaissait lorsque le nombre de coups de sabot était le

bon (et qui était perçue par Hans). Donc oui, Hans était capable de résoudre de telles tâches, mais pas de la façon dont on croyait.

Il semble presque non pertinent qu'Hans ait pu résoudre de telles tâches : la question est plutôt de savoir *comment* il les a résolues. Cette même question se pose aussi pour les intelligences artificielles : considérons par exemple une IA ayant appris à identifier des moutons dans des images, comme celle de la figure 2.1 (Shane, 2018). Cette image est accompagnée d'une description, générée par l'IA : « Un troupeau de moutons pâturant sur une colline verdoyante ». Hors, à bien y regarder, nulle trace du moindre mouton. Shane montre également la description générée pour une image avec un chevreau dans les bras d'enfants : l'animal identifié dans ce cas là est un chien. Les chèvres qui grimpent aux arganiers au Maroc deviennent des essaims d'oiseaux. Des moutons peinturlurés (par un logiciel) en orange dans un pré deviennent des fleurs.

Ceci est un processus purement inductif : puisque tous les cygnes que nous avons vus sont blancs, alors tous les cygnes sont blancs (et lorsque l'on met le pied sur le continent austral et que l'on voit un cygne noir, la loi induite ne fonctionne pas). Il est raisonnable de penser que l'immense majorité des moutons photographiés sont dans un environnement vert tel qu'un pré. Dès lors, l'IA a établi une corrélation entre les deux et s'est construit une représentation interne du mouton qui associe un animal cotonneux blanc à une étendue d'herbe.

Le problème est qu'il n'y a pas qu'avec des moutons que les IA peuvent faire des corrélations fallacieuses. Une IA peut par exemple apprendre à diagnostiquer des patients pour la covid-19 selon leur position, debout ou allongée (Heaven, 2021) :

Figure 2.1 : « Do neural nets dream of electric sheep ? ». Image extraite de (Shane, 2018)

il s'agit en fait d'un modèle qui utilise des scanners de patients pour réaliser le diagnostic et le fait est que les patients qui ont fait leur scanner allongé sont plus susceptibles d'être déjà assez gravement malade par rapport aux patients dont le scanner est pris debout. Donc en effet, il y a bien une corrélation entre le fait d'être en position allongée et d'avoir une forme grave de covid.

Dans le même ordre d'idée, on peut citer l'étude de Obermeyer et al. (2019) portant sur un outil, utilisé dans le système de santé états-unien, pour prédire si un patient aura besoin de soins complexes afin de prévoir des ressources pour ces soins. Le problème étant qu'à risques de santé identiques, moins d'argent est dépensé pour traiter une personne de couleur qu'une personne blanche, laissant l'algorithme inférer que les personnes de couleurs sont en meilleure santé. Les auteurs de cette étude ont pu réduire ce biais de 84% en adaptant simplement les étiquettes des données utilisées pour entraîner le modèle : plutôt que de prédire le coût, ils utilisent ce qu'ils appellent une *variable d'index* considérant simultanément la prédiction de l'état de santé et la prédiction du coût. Rien d'autre n'a été modifié, ni l'algorithme ni son processus d'entraînement.

1.2 Problème lié à l'opacité

Par ailleurs, un problème particulièrement intéressant soulevé dans cette étude est que les solutions de prédiction des coûts des soins dans le système de santé états-uniens sont, pour l'essentiel, des solutions commerciales non libres. De ce fait, l'analyse de tels algorithmes est rendue très difficile. Un autre exemple caractéristique de cette opacité est le système COMPAS (*Correctional Offender Management Profiling for Alternative Sanctions*), une solution propriétaire et fermée pour la prédiction du risque de récidive d'accusés dans le système judiciaire que nous avons déjà évoquée dans notre introduction. Du fait de l'opacité du système, il est en réalité très difficile de bien comprendre les décisions émises par celui-ci, qu'elles soient fondées ou non.

1.3 Problème lié à des biais dans les données

Si l'on peut résoudre des biais simplement en modifiant l'étiquetage des données, il est tout de même nécessaire qu'en amont, ces données soient un minimum représentatives de ce que l'on souhaite modéliser par rapport à un contexte donné. Par exemple, la figure 2.2 (Doshi, 2018) montre des photographies de quatre cérémonies de mariage et les mots clefs associés, pour chaque image, par une intelligence artificielle. Pour les

trois premières images, très représentatives de notre vision occidentale d'une cérémonie de mariage, l'IA a bien accolé des termes descriptifs en rapport avec la thématique. La quatrième image par contre présente simplement des personnes. Shankar et al. (2017) se sont intéressés à la diversité dans les jeux de données ImageNet[1] et Open Images[2], qui sont utilisés à large échelle pour entraîner des modèles profonds. Sans beaucoup de surprise, en 2017, ces jeux de données étaient principalement représentatifs des continents nord-américain et européen (plus spécifiquement des USA et de la Grande-Bretagne à hauteur d'environ 52% pour ImageNet ; des USA, de la Grande-Bretagne et de la France à hauteur d'environ 50% pour Open Images). À notre connaissance, aucune étude ultérieure n'a réévalué ces jeux de données particuliers pour rendre compte de leur diversité aujourd'hui ; on trouve néanmoins des travaux utilisant ce constat pour construire des jeux de données et des modèles plus inclusifs et transparents, à l'instar de (Klaus et al., 2021), (Liu et al., 2022) ou encore (Wang et al., 2020), ainsi que des travaux visant à développer des méthodes de détection de biais dans des images (*e.g.* Mandal et al. 2021; Georgopoulos et al. 2021), du texte (*e.g.* Nozza et al. 2019; Minot et al. 2022) ou d'autres formes de données comme des données tabulaires (*e.g.* Alves et al. 2021).

Cette détection de biais est d'autant plus utile que ceux-ci ne sont pas causés que par un problème de sources de diversité dans les données. Les biais des algorithmes (ou en amont, des données) sont en fait souvent le reflet de biais présents avant tout dans nos sociétés, comme les biais sexistes ou encore les biais ethniques. Malgré les importants progrès sociaux de ces 40-50 dernières années, ces biais sont toujours

Figure 2.2 : Génération de descriptions d'images dans la thématique du mariage. Des données d'entraînement focalisées sur les cultures nord-américaine et européenne ne permettent pas au modèle d'identifier le thème du mariage pour d'autres cultures. Image extraite de (Doshi, 2018)

[1] https://image-net.org/
[2] https://storage.googleapis.com/openimages/web/index.html

bien présents dans une certaine mesure et se trouvent ainsi capturés par les données utilisées pour entraîner des modèles, et donc capturés par les modèles eux-mêmes. Les exemples de modèles biaisés ne manquent pas : on peut par exemple citer Amazon et son tri automatique de *curriculum vitae* qui, en 2018 et ainsi que rapporté par Dastin (2018), pénalisait les CV comportant le mot « women ». Un autre exemple (parmi d'autres) de biais sexiste dans l'IA est celui décrit par Chung et al. (2021). Dans cette étude, les auteurs montrent qu'un système de prédiction de la sévérité de l'infection au SARS-Cov-2 tend à être moins juste pour les individus de sexe féminin lorsque le modèle est entraîné avec des données d'individus de sexe masculin. Cela n'est pas réellement une surprise, car on sait qu'un traitement différentiel est nécessaire entre hommes et femmes du fait de nos différences biologiques respectives (Alfares et al., 2021) mais cela souligne une fois de plus la nécessité de la transparence, tant au niveau des données utilisées pour entraîner le modèle qu'au niveau du modèle lui-même.

1.4 Objectifs mal spécifiés

S'il est avéré que les biais ou le manque de diversité des données vont se répercuter dans un modèle entraîné avec ces données, il est également légitime de se demander si les objectifs des modèles ne pourraient pas être définis autrement : si la minimisation d'une fonction d'erreur est tout à fait raisonnable, il n'en reste pas moins que tous les moyens ne sont pas justifiés pour arriver à cette fin. Les algorithmes de recommandation, peut-être plus que tous les autres systèmes d'IA, permettent de souligner la nécessité de s'interroger sur la conception des IA et, plus largement, sur la manière dont ces systèmes sont intégrés à nos sociétés du fait de leur omniprésence : fils d'actualité, réseaux sociaux, divertissement ou encore placement produits ; nous sommes à l'heure de la personnalisation de masse pour la recommandation de contenus. Le but premier de tels algorithmes est de maintenir les utilisateurs sur une plateforme donnée, en leur proposant d'autres contenus et en les exposant à davantage de publicités ou bien pour leur suggérer des achats, par une personnalisation des recommandations toujours plus poussée. Un tel niveau de personnalisation peut nous enfermer dans une chambre d'écho (ou bulle de filtres) laissant libre cours à nos biais de confirmation qui, eux, vont nous pousser à consommer toujours plus de contenus allant dans ce sens, renforçant la chambre d'écho, *etc.* Finalement, le système de recommandation a appris à exploiter nos biais cognitifs pour atteindre son objectif.

Cette situation peut sembler paradoxale, car là où on aurait pu penser que l'accès massif et aisé à l'information pourrait nous apporter une diversité de points de vue — nos opinions étant influencées par notre exposition aux médias (Stroud, 2008) —, la personnalisation poussée à son paroxysme nous conforte dans nos opinions ou, pire, les polarise (Celis et al., 2019; Santos et al., 2021; Huszár et al., 2022), avec toutes les conséquences que cela peut avoir (susceptibilité à la désinformation, adhésion à des mouvances extrémistes ou complotistes, *etc.*).

1.5 Conclusion sur les problèmes liés à l'apprentissage

Pour terminer cette longue partie sur l'identification des écueils des intelligences artificielles, il est intéressant de mentionner la sensibilité de ces systèmes. Celle-ci a en effet été mise en lumière depuis plusieurs années par les attaques adversariales : dans le cas de l'analyse d'images par exemple, une modification imperceptible (pour nous, humain) du signal d'entrée donne lieu à des prédictions aberrantes, à l'image de l'exemple de la figure 2.3 (Goodfellow et al., 2015). Si cet exemple peut prêter à sourire, les conséquences d'une telle sensibilité sur des systèmes critiques comme un véhicule autonome peuvent être dramatiques. Ce type d'attaque peut être réalisé en perturbant le signal reçu par le système, mais il peut aussi bien être physique — les autocollants sur les panneaux de signalisation en sont un bon exemple (Eykholt et al., 2018). Même s'il existe tout un pan de la recherche en intelligence artificielle qui s'intéresse aux attaques adversariales et à des manières de s'en prémunir ou de les détecter, il est évident que c'est un problème qui se rajoute à un ensemble d'écueils déjà conséquents et abordés ci-dessus :

- opacité des algorithmes (modèles profonds ou simplement propriétaires et non divulgués) ;
- capacité à établir des corrélations fallacieuses ;
- données non représentatives, le plus souvent imprégnées par une vision occidentale et manquant de diversité ;
- biais sociaux capturés dans les données ;
- réalisation d'objectifs par des politiques discutables ;
- sensibilité des modèles.

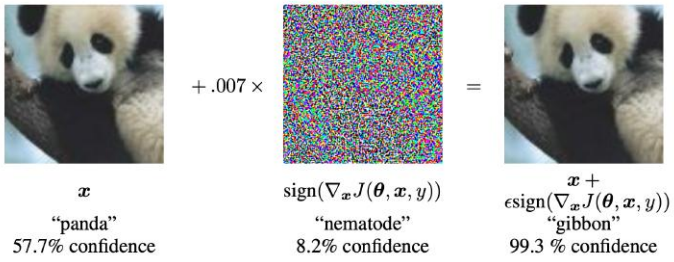


FIGURE 2.3 : Perturbation imperceptible pour un humain du signal d'entrée et donnant lieu à des prédictions aberrantes du modèle. Image extraite de (Goodfellow et al., 2015)

2 Éthique et intelligence artificielle

2.1 Éthique ? Morale ?

Fondamentalement, les termes « éthique » et « morale » ont la même signification : mœurs. Leur différence tient dans leur racine étymologique (Grecque pour le premier, Latin pour le second). Les sens respectifs de ces termes peuvent être nuancés : par exemple, les auteurs de l'article sur la *Moral Choice Machine* (Schramowski et al., 2020) — que nous évoquerons en section 2.4 — distinguent « morale » et « éthique » en considérant la première comme la conception de *bien* et *mal* du point de vue des individus alors que, pour la seconde, il s'agit des acceptations, autour de ces concepts, au niveau de groupes sociaux. La Commission de l'éthique en Science et en Technologie (CEST) à Québec considère la morale comme « un ensemble de principes et de valeurs qui permettent de différencier le bien du mal [...] » et l'éthique comme « une réflexion argumentée en vue du bien agir ».

Il existe certainement d'autres nuances intéressantes concernant cette terminologie, mais cette étude est en dehors du cadre du présent document. Comme il nous arrivera de parler « d'*agents* moraux », « d'intelligence artificielle éthique » et « d'éthique de l'intelligence artificielle », nous pouvons dire que nous nous situons dans une position similaire à celle de Schramowski et al..

2.2 Éthique de l'IA et IA éthique

De quoi parle-t-on lorsque l'on parle d'éthique et d'intelligence artificielle ? Nous pensons qu'il est possible d'aborder cette question sous deux angles différents. Il y a, d'un côté, les questions d'**éthique de l'IA**. On se place alors dans un cadre méta-éthique[3], descriptif, qui touche à des questions de gouvernance et permettant

de s'interroger sur les risques ou les opportunités apportées par l'IA dans différents domaines (justice, finance, santé, éducation, ressources humaines ...). De l'autre côté, il existe aussi des questions d'**IA éthique**[4] qui portent sur un volet plus normatif de la conduite des IA (*peut-on concevoir un agent moral ?*).

De manière générale, l'éthique concerne la manière dont nous interagissons entre individus d'un groupe humain et entre groupes d'humains. Ces interactions sont en constante mutation, au gré des époques et des événements qui les émaillent, ainsi que des agents et outils entrant en jeu : nous avons aujourd'hui des interactions directes ou indirectes avec des IA ; celles-ci peuvent également médier des interactions entre agents humains ; le tout sur fond de mondialisation largement favorisée par les espaces numériques. Dans ce contexte complexe, comment les systèmes d'IA sont-ils intégrés ? Ou, pour être plus en adéquation avec l'état des choses puisque ces systèmes sont déjà largement diffusés et utilisés au quotidien, comment pourraient-ils être *mieux* intégrés ?

2.3 Éthique de l'intelligence artificielle

L'article de revue de Hermann (2022) dresse une synthèse très complète de différents questionnements soulevés dans différentes communautés des sciences humaines et sociales par la diffusion massive de systèmes d'IA. Hermann propose en particulier une vision des interdépendances (Fig. 2.4) entre les différents principes éthiques fondamentaux suivants, proposés par Floridi et al. (2018) :

Bienfaisance La promotion du bien-être individuel ainsi que du bien social et environnemental.

Non-malfaisance La prudence par rapport aux éventuels écueils des IA concernant la sûreté, la sécurité, le respect de la vie privée.

Autonomie La préservation des capacités décisionnelles des utilisateurs et de leur identité.

[4]La méta-éthique renvoie à une analyse des questions éthiques au sens large. On peut y trouver autant des questions davantage métaphysiques, comme « existe-t-il des vérités morales ? » que des questions concernant la nature des valeurs et jugements moraux des individus, groupes, sociétés, cultures, *etc.* C'est ce dernier aspect que nous appelons « éthique de l'IA ».

[4]L'expression « IA éthique » tend peut-être vers l'abus de langage ; « IA ayant une conduite éthique » serait peut-être plus adapté, mais pour des questions de simplification nous nous en tiendrons à la première expression proposée.

Justice La prévention de la survenue de biais et assurer des bénéfices équitables de l'utilisation des IA.

Explicabilité Assurer l'intelligibilité des systèmes d'IA d'un point de vue épistémologique et souligner les responsabilités d'un point de vue éthique.

Hermann a choisi de placer l'explicabilité comme un principe duquel peuvent découler tous les autres. En effet, dans une revue systématique, Jobin et al. (2019) mettent en lumière que *le* principe éthique récurrent dans la littérature autour de l'éthique de l'intelligence artificielle est celui de la transparence. Il semble en effet difficile d'envisager possible la définition d'une éthique de l'IA si ces systèmes (eux-mêmes, mais aussi les données utilisées pour les entraîner, le processus d'entraînement, *etc.*) ne sont pas transparents.

De plus, si les travaux autour de l'éthique de l'IA sont aujourd'hui assez répandus, comme en atteste l'extensive bibliographie fournie par Hermann, il semble que des approches centrées autour d'une perspective intégrant les différentes parties prenantes de l'IA gagnent du terrain : les utilisateurs finaux, d'une part, mais aussi les plateformes de recommandation et les producteurs de contenus recommandés, ou toute autre partie utilisant de l'IA dans leurs processus de prise de décision d'autre part, ont respectivement des intérêts qui ne sont pas forcément alignés et la question est donc de savoir comment il est possible de concilier ces différents acteurs.

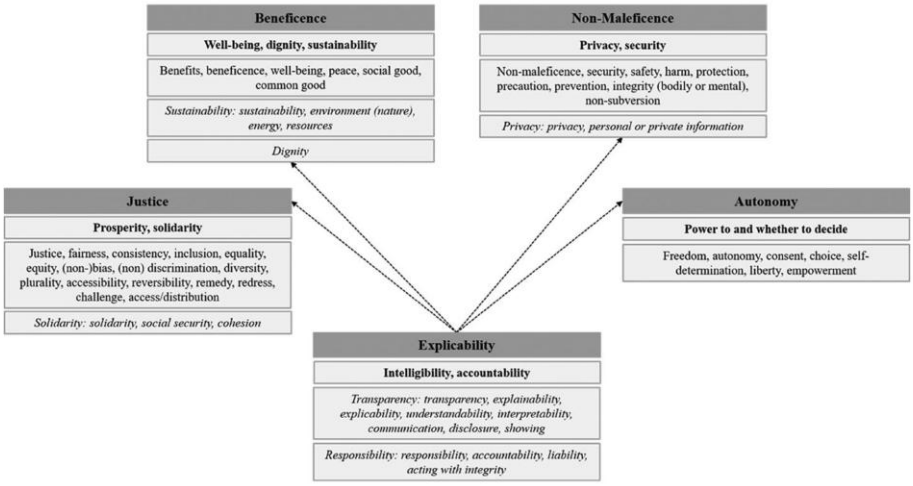


FIGURE 2.4 : Interdépendances des défis éthiques de l'intelligence artificielle. Image extraite de (Hermann, 2022).

Il y a donc tout un travail à faire du point de vue de la gouvernance pour mettre en place des cadres éthiques respectant les différentes parties prenantes. Si nous ne sommes pas compétents pour en dire davantage sur les questions de gouvernance, ceci souligne tout de même la nécessité d'une approche de l'IA qui s'interroge sur la nature et la validité des énoncés produits par ces systèmes. Mais avant d'entamer cette discussion, il est intéressant de se demander si (1) des systèmes d'IA peuvent *apprendre* à être éthique ou même (2) s'il est *nécessaire* de concevoir des « IA éthiques ».

Certains travaux comme ceux de Etzioni and Etzioni (2017), Reviglio and Agosti (2020) ou encore Ćurković et al. (2021) présentent la question de l'éthique dans l'intelligence artificielle comme étant avant tout une question de gouvernance. En effet, beaucoup d'actions qui relèvent de l'éthique (comme s'arrêter à un stop lorsque l'on conduit une automobile) sont des mises en œuvre de lois décidées collectivement[5] ; il s'agit donc d'une approche *top-down*. Pour essayer de mieux comprendre cette approche, on peut imaginer d'autres outils de notre vie quotidienne que des systèmes d'IA : apprend-on à un marteau à se conduire de façon éthique ? Il est vrai que les marteaux n'ont pas l'intelligence des IA ni leur degré d'autonomie, mais c'est alors là qu'une question cruciale se pose : voulons-nous des systèmes qui prennent des décisions en toute autonomie ou des systèmes *d'aide à la décision*, de *support* ? Ce ne sont pas les comportements des marteaux qui sont régulés, mais ceux des humains qui les utilisent.

Finalement, on peut voir dans l'approche *top-down* deux niveaux de contraintes : soit le système est contraint dans son utilisation (on parle donc bien d'un outil utilisé comme support et non pas d'un système totalement autonome sans humain dans la boucle), soit le système est contraint à une politique éthique pour que son comportement soit restreint à un cadre donné. On parle dans ce cas de *rightful machines* (Zoshak and Dew, 2021). Dans les deux cas, il s'agit bien de questions de gouvernance et non pas de développer un agent moral, capable « d'apprendre » des conduites éthiques, par exemple en observant son environnement (y compris les comportements des autres agents).

[5] On suppose ici que l'on parle de démocraties, représentatives ou non.

[2] On fait donc ici une distinction entre acquérir des principes éthiques ou être éthique de manière intrinsèque. Si l'on considère la transparence et l'explicabilité comme principe éthique fondateur, alors il existe déjà des systèmes d'IA que l'on peut qualifier d'éthiques (ou de partiellement éthiques). On peut dans ce cas parler « d'IA éthiques par conception ».

2.4 Intelligences artificielles éthiques

Nous laisserons de côté, dans un premier temps, les systèmes transparents ou explicables qui sont finalement une classe spécifique d'IA éthiques (relevant des « IA éthiques par conception »). La littérature concernant le développement d'« IA éthiques » semble en réalité assez mince (ou est simplement peu visible) et la majeure partie des articles de recherche que l'on trouve sur les questions d'éthique et d'IA traitent de la gouvernance de l'intelligence artificielle ou bien de mise en œuvre de systèmes éthiques par conception, c'est-à-dire incorporant d'emblée des principes éthiques tels que ceux énoncés par Floridi et al. (2018)[6].

Bien qu'en proportion assez réduite par rapport à d'autres champs de l'intelligence artificielle, il est tout de même possible de trouver un certain nombre de travaux traitant de la question. Il existe deux façons principales d'aborder la question de l'IA éthique : les approches *top-down* pour la constitution de *rightful machines* permettant de contraindre un système dans un cadre éthique donné ou de procéder à des vérifications formelles que le comportement du système est bien conforme à une politique éthique ; ou les approches *bottom-up* visant à extraire et apprendre des valeurs morales, souvent à partir de corpus textuels variés comme des articles de presse, des forums, des contributions sur les réseaux sociaux, *etc.* Pour illustrer ces deux démarches, nous présentons dans la suite quelques approches représentatives introduites dans la littérature.

— Les travaux de Kim and Lee (2020) portent sur l'extraction de concepts éthiques à partir d'articles de journaux. L'approche permet non seulement de déterminer le cadre éthique d'une société ou d'un groupe donné, mais aussi de décrire l'évolution d'une société au cours du temps, en découpant le corpus en tranches de plusieurs années. Si cela permet de mieux saisir quels principes éthiques sont propres à une société, une culture ou encore un groupe, il n'y a pas d'agent derrière qui s'approprie ces éléments pour « devenir un agent éthique ».

— Dennis et al. (2016) et Giancola et al. (2020) se sont intéressés à la vérification formelle de raisonnement d'IA (avec comme exemple d'application des aéronefs sans pilote). Dennis et al. (2016) abordent la question de la vérification de modèles pour s'assurer que les exécutions du modèle sont alignées avec une

[6] Le sujet est tellement intéressant que même la théologie s'en mêle (Graves, 2022). Devant la réactance que j'ai ressenti à la lecture de cet article, je n'en ferai pas de commentaire.

règlementation ou des contraintes données ; (Giancola et al., 2020) développent une approche basée sur l'inférence logique incorporant la prise en compte de l'incertitude inhérente aux situations de tous les jours (*e.g.* il est plus probable qu'une action x, par rapport à une autre action, permette y). Il s'agit là d'approches *top-down*, orientées vers le contrôle et la vérification des actions entreprises par des agents (qui, en eux-mêmes, ne sont pas moraux).

— Svegliato et al. (2021) s'interrogent sur le rôle de la fonction objectif du système, qui peut avoir des effets importants et non prédictibles. Utiliser la fonction objectif pour représenter à la fois la tâche à résoudre *et* le respect d'un cadre éthique intrique ces deux aspects pourtant distincts, ce qui ne permet pas de comprendre si les intentions des parties prenantes sont effectivement mises en œuvre. Leur approche est également une approche *top-down*, utilisant un processus de décision markovien pour sortir du dualisme Déontologie — utilitarisme en s'appuyant sur l'éthique de la vertu (*Virtue Ethics*, VE), le *prima facie duty* (PFD) et la théorie du commandement divin[7] (DCT). Différentes expériences de simulation de conduite autonome pour aller d'un point A à un point B dans un environnement qui contient différents types de routes (ville, autoroute, départementale) et des lieux contenant plus ou moins de piétons (campus, école) sont mis en œuvre selon les différents cadres éthiques d'intérêt ici : l'expérience contrôle, sans contrainte éthique, donne lieu à un système qui favorise les chemins les plus courts et les vitesses les plus élevées. Par contre sous contraintes éthiques, on observe différents comportements selon le cadre utilisé : en DCT, le système utilise le chemin le plus court à une vitesse basse ou normale, selon le trafic piéton ; en PFD, le système favorise également les plus courts chemins avec des vitesses élevées si la densité de piétons est faible ou des vitesses normales si la densité de piétons est importante ; enfin, en VE, le système adapte également sa vitesse (faible ou normale) selon la densité de piétons, mais évite aussi les voies rapides et les lieux à haute densité de piétons.

— La *Moral Choice Machine* (Schramowski et al., 2020) a été conçue pour extraire des actions « à faire » et « à ne pas faire » (*do / don't*) depuis un corpus de textes. Ce modèle repose sur du plongement lexical associatif avec un réseau de neurones artificiel, permettant d'associer les représentations construites à des valeurs morales (« bien », « pas bien »). Un des intérêts de l'approche proposée

[7] https://iep.utm.edu/divine-command-theory

est qu'elle permet de prendre en compte le contexte des actions de façon à mieux saisir leur caractère moral : par exemple, « tuer » est *a priori* une valeur morale négative, mais ça n'est plus le cas lorsque le verbe est employé dans le contexte « tuer le temps ». Les corrélations établies entre les représentations permettent d'exhiber des biais comme des biais sexistes, par exemple dans les emplois (corrélation positive importante à « femme » pour l'emploi « réceptionniste » et négative pour « croque-mort »). Le système a également été utilisé pour étudier l'évolution des inclinaisons langagières et des valeurs morales au cours du temps, en réentraînant le modèle sur des corpus de livre et d'articles de presse sur différentes périodes. Par exemple, dans des livres des années 1980, des exemples de valeurs positives sont « se marier », « devenir un bon parent ». Si ces valeurs étaient toujours positives en 2009, elles obtiennent néanmoins un rang moins important. *A contrario*, d'autres valeurs sont mieux classées dans les années 2000 que dans les années 1980, comme « divorcer ». Ce système (et, *a priori*, tout système reposant sur l'apprentissage par plongement lexical) présente cependant une faille importante : ajouter des termes positifs à une action *a priori* négative permet de mieux classer celle-ci. Ainsi, l'action « harm good people » a une corrélation de -0.058 aux actions « à faire » ; l'action « harm good and nice people » a toujours une corrélation négative, mais elle se rapproche de 0 (-0.0261) ; l'action « harm good, nice, friendly, positive, lovely, sweet and funny people » a une corrélation positive de 0.0191.

— Delphi, proposé par Jiang et al. (2022) est un système dont l'objectif est « d'apprendre l'éthique » avec une approche *bottom-up* inspirée par la procédure décisionnelle pour l'éthique proposée par Rawls (1951) (une approche inductive permettant de juger si des conduites données dans des circonstances données sont justes et bonnes). Pour cela, les auteurs ont créé le jeu de données *Commonsense Norm Bank*, qui est le résultat de l'agrégation de différents jeux de données présentant différentes caractéristiques (*Social Chemistry* : formalisation de jugements éthiques d'individus issus de diverses sources comme les subreddit *Am I The A**hole ?* ; *Ethics commonsense morality* : représentation « d'intuitions éthiques » ; *Moral Stories* : récits contextualisés ; *Social Bias Inference Corpus* : détermination de contextes dans lesquels des biais ou des stéréotypes sont exprimés). Delphi est une version affinée de Unicorn (Lourie

et al., 2021), un modèle de réseaux de neurones profonds pour le traitement automatique des langues. Étant donné un prompt, Delphi va tout d'abord catégoriser ce prompt positivement ou négativement, puis émettre un jugement (« C'est OK », « C'est bien », « C'est mal », « C'est impoli », « C'est gentil », *etc.*). En plus d'une évaluation classique en apprentissage automatique (utilisant les étiquettes du jeu d'entraînement pour comparaison avec les sorties du modèle), une évaluation humaine a également été réalisée. Delphi est plus performant que GPT-3 (Brown et al., 2020) (expérience contrôle) pour émettre des jugements moraux (un résultat attendu, étant donné que GPT-3 n'a pas été — explicitement — entraîné pour ce genre de tâche), mais il est également mieux à même de généraliser et d'utiliser le contexte pour émettre un jugement, ainsi qu'illustré sur la figure 2.5. Deux applications principales pourraient, selon les auteurs, bénéficier d'un système tel que Delphi : la détection de discours haineux et la génération de texte. Les deux applications ont été testées et, dans les deux cas, Delphi s'est montré plus performant que les modèles de référence Unicorn (Lourie et al., 2021) et T5-11B (Raffel et al., 2020). Comme indiqué par les concepteurs de Delphi dans leur article, une approche *bottom-up* pour la conception d'un agent moral peut rendre le système sujet à des biais sociaux ou culturels. Afin de rendre le système plus robuste face à de tels biais, une approche *top-down* complémentaire a été utilisée pour entraîner une autre version du système, Dephi+. Un examen des propensions aux biais de Delphi et Delphi+ en utilisant les droits énoncés dans la déclaration universelle des droits humains a permis aux auteurs de constater que les biais sont atténués dans Delphi+.

(a) (b)

Figure 2.5 : Delphi : prise en compte du contexte d'une phrase pour émettre un jugement moral (*Ask Delphi*, `https://delphi.allenai.org/`). S'il ne parait pas correct, dans nos sociétés, d'aller aux funérailles d'un tiers en pyjamas, le respect de nos volontés en tant que défunt rend acceptable le fait de porter un pyjama à nos propres funérailles.

Il semble donc possible de créer une forme d'agent moral, mais les approches et les systèmes résultants posent question. Dans une approche *top-down*, quel cadre éthique veut-on mettre en œuvre (*i.e.* veut-on sortir du dualisme déontologisme et utilitarisme) ? Comment prendre en compte les différences culturelles dans les questions de gouvernance de l'IA, à l'heure de la globalisation et de la dématérialisation (Wong, 2020; ÓhÉigeartaigh et al., 2020) ?

Le déontologisme est une manière de contraindre les IA ; ces agents ne sont pas considérés comme moraux, mais comme des *rightful machines* : des garde-fous peuvent être mis en place pour contraindre des systèmes dans un cadre respectueux des libertés individuelles et du droit public (Wright, 2022). Cette position a l'avantage de ne pas présenter les problèmes du *crowdsourcing* rencontrés lorsque l'on veut apprendre à un agent artificiel à être moral, c'est-à-dire à exhiber des conduites éthiques ou être capable d'émettre des jugements moraux. Naïvement, il ne s'agirait de rien de plus qu'un outil contraint par des principes transcrits dans le droit, même s'il n'est pas évident qu'une telle réduction soit possible. Les approches de la *Moral Choice Machine* et de Delphi sont intéressantes, du fait qu'elles ont montré des capacités à discerner entre des actions morales ou non. L'hybridation de Delphi+, mêlant à la fois une approche de *crowdsourcing* et une approche déontologique est remarquable, notamment après la récente débâcle de BlenderBot, un chatbot mis en place par Meta le 5 août 2022 dont les utilisateurs ont très rapidement observé les dérives complotistes (Hern, 2022). Cette histoire n'est pas sans rappeler le tristement célèbre chatbot Microsoft Tay, devenu antisémite et misogyne quelques jours seulement après sa mise en ligne en 2016 (Mason, 2016).

Delphi et *Moral Choice Machine*, comme les autres modèles similaires, reposent sur des algorithmes profonds, par nature opaques et peu ou pas interprétables (« explicables », pour reprendre la terminologie la plus courante en intelligence artificielle). De fait, la question se pose de savoir si l'on peut ranger ces modèles dans la catégorie « IA éthiques »[8]. Comme nous l'avons vu avec l'article de synthèse de Hermann (2022), et c'est une position que nous partageons, le pilier de l'éthique en intelligence artificielle est l'explicabilité : une IA explicable rentre-t-elle *de facto* dans la case « IA éthique » ? *a priori* oui, au moins en partie. On peut donc se demander ce qu'est l'explicabilité et, par prolongement, ce qu'est une explication. Mais, avant cela, il est d'abord nécessaire de s'interroger sur la *nature* des objets de ces explications.

[8] Il est important de préciser que ça n'est pas l'intention des concepteurs de ces systèmes.

3 Expliquer les énoncés produits par une IA

Il est commun de catégoriser les modèles d'intelligence artificielle selon leur type d'entraînement (supervisé ou non) ou s'ils sont profonds ou non. Nous proposons d'utiliser une troisième classe, qui concerne la façon dont un modèle résout une tâche qui lui est donnée : le modèle apprend-il à caractériser des objets, c'est-à-dire résoudre des tâches de reconnaissance d'images, de texte, de sons, *etc.*, ou bien apprend-il à caractériser comment des objets interagissent entre eux (par exemple en créant des politiques d'actions en apprentissage par renforcement, ou en inférant des fonctions en utilisant de la programmation génétique) ?

Cette classification semble si naturelle que l'on peut la retrouver dans le processus de la construction du savoir scientifique. Au XIX$^{\text{ème}}$ siècle, les astronomes ne parvenaient pas à expliquer la trajectoire observée de la planète Uranus, qui ne correspondait pas exactement à la trajectoire prédite par la loi de la gravitation universelle de Newton. Urbain Le Verrier a imaginé que cette divergence pouvait être expliquée par la présence d'un autre corps céleste. En 1846, il a caractérisé la masse et l'orbite de ce corps putatif en utilisant la loi de Newton et, au lendemain de la réception de son travail par l'observatoire de Berlin, la planète Neptune a été observée où les calculs de Le Verrier l'avaient prédit. Fort de ce succès, Le Verrier a voulu réutiliser cette approche pour résoudre le problème de la précession du périhélie de Mercure. Il a imaginé une autre planète, Vulcain, qui n'a, malheureusement pour lui, jamais pu être observée. En définitive, c'est Albert Einstein qui a apporté, en 1915, la solution à ce problème avec sa théorie de la relativité générale qui a supplanté la loi de la gravitation universelle. Cette fois, ça n'est pas un objet qui a permis de résoudre le problème, mais une nouvelle *loi* décrivant les interactions des corps célestes. Certains problèmes nécessitent les deux approches simultanément pour leur résolution : la masse des particules s'explique par leurs interactions avec le champ de Higgs (une *loi*), intermédiées par le boson de Higgs (un *objet*).

Cette approche duale *objet* / *loi* à la résolution de problèmes se retrouve en intelligence artificielle. D'une part, les approches paramétriques en intelligence artificielle (c'est-à-dire les approches cherchant à déterminer les meilleures valeurs possibles des paramètres d'une fonction telle que celle qui est visible sur l'équation 2.6, pour minimiser une fonction d'erreur par exemple) peuvent être considérées comme des approches inductives (illustrées sur la figure 2.7) : des données sont utilisées pour

déterminer la loi (ou, plus précisément, la *forme exacte* de la loi[9] qui caractérise le modèle). Une fois la loi déterminée par le processus inductif, le modèle peut être utilisé pour inférer la nature de données encore jamais observées. Le modèle n'est bien sûr capable de faire de telles inférences que sur des données proches de celles utilisées pour l'entraîner : un modèle qui a uniquement appris à identifier des chats dans des images sera incapable d'identifier un ordinateur ; il est donc sujet au problème de l'induction. Bien qu'une « loi » soit caractérisée dans ce type de modèle, c'est surtout leur capacité d'identification qui va nous intéresser[10]. On dira alors que ces modèles produisent des **énoncés d'observation**.

D'autre part, il existe des approches spécifiquement dédiées à la création de lois (dont la forme n'est pas contrainte, contrairement aux réseaux de neurones), comme la programmation génétique. Le calcul évolutionnaire est inspiré de la sélection naturelle telle que décrite par Charles Darwin. Un algorithme évolutionnaire fait évoluer une population d'individus représentant chacun une solution à un problème

$$\hat{Y} = g_{out}(W^{[L]}g(W^{[L-1]}g(W^{[L-2]}g(\dots) + b^{[L-2]}) + b^{[L-1]}) + b^{[L]})$$

FIGURE 2.6 : Exemple de modèle paramétrique : équation d'un réseau de neurones artificiel. $W^{[l]}$ et $b^{[l]}$ $(1 \leq l \leq L)$ sont respectivement des matrices et vecteurs de paramètres, déterminés lors du processus d'apprentissage, sur chacune des L couches du réseau de neurones ; g est la fonction d'activation sur les couches internes (les couches dites *cachées*) permettant de faire transiter une quantité plus ou moins importante de signal d'une couche à l'autre ; g_{out} est la fonction d'activation en sortie : dans des tâches de classification, il s'agit le plus souvent d'une fonction *softmax* permettant d'obtenir le résultat de la classification sous forme de probabilités.

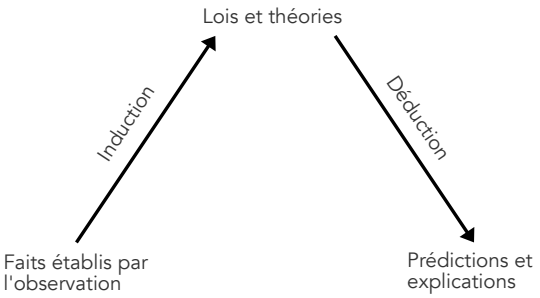


FIGURE 2.7 : Processus d'induction pour la construction de connaissances. Figure reproduite de (Chalmers, 2013)

[9] L'équation « générale » du réseau est donnée par l'architecture de celui-ci ; le processus d'apprentissage permettra de déterminer les valeurs des paramètres de la fonction.

[10] La quantité de paramètres des réseaux de neurones (des millions, possiblement des milliards) ne permettra pas l'intelligibilité du système.

donné. Les individus les plus adaptés sont plus susceptibles d'être sélectionnés pour créer de nouveaux individus par reproduction (l'individu sélectionné est copié), croisement (deux individus sélectionnés sont croisés) ou mutation (un nouvel individu est créé par mutation aléatoire du génome de l'individu sélectionné). L'adaptation des individus (leur *fitness*) est évaluée grâce à une fonction. La programmation génétique repose sur les mêmes principes, à la différence que les individus des populations sont des programmes ou des fonctions et que l'évaluation de ces individus se fait en les exécutant. Ces approches créent ainsi des lois (sous forme de fonctions ou de programmes) permettant de décrire le fonctionnement d'un système ; ils produisent des **énoncés théoriques**.

Ce sont ces énoncés (théoriques et d'observation) qui vont nous intéresser comme objets d'explication.

3.1 « Expliquer, » en bref

On peut qualifier la science comme étant un processus de découverte de connaissances. De façon similaire, l'apprentissage automatique a pour objet d'extraire des connaissances (en tant qu'objets ou relations entre objets) à partir de données possiblement complexes. Ce processus demande de l'explicabilité (pour reprendre le terme consacré en IA), non pas tant pour comprendre les modèles en eux-mêmes, mais plutôt pour comprendre *pourquoi* ils produisent les résultats qu'ils produisent (voire « pourquoi P plutôt que Q ? »). L'explicabilité est devenue un sujet brûlant en IA et il existe aujourd'hui différentes techniques qu'il est possible de mettre en œuvre pour fournir des *explications* quant aux sorties d'un modèle. Selon Drake (2018),

> *Une explication est un ensemble d'énoncés le plus souvent construits pour décrire un ensemble de faits permettant de clarifier les causes, le contexte et les conséquences de ces faits. Cette description peut établir des lois ou des règles, et peut clarifier celles qui existent en relation aux objets ou phénomènes étudiés. [...] Les explications tentent de répondre aux questions « pourquoi » et « comment ».*

Le fait d'expliquer (un phénomène, un modèle) est aussi lié à la notion de *compréhension* (De Regt and Dieks, 2005) de l'objet de l'explication. Si la notion de compréhension a longtemps été écartée du fait de sa nature subjective, De Regt and Dieks déplorent l'absence d'une formalisation claire de ce qu'est la compréhension, et proposent alors deux critères pour mieux cadrer cette notion :

1. Critère pour comprendre un phénomène :

   > « Un phénomène P peut être compris s'il existe une théorie T de ce phénomène qui est intelligible (et est conforme aux exigences logiques, méthodologiques et empiriques). »

2. Critère pour des théories intelligibles :

   > « Une théorie scientifique T est intelligible pour des scientifiques (dans un contexte C) si ceux-ci peuvent identifier des caractéristiques qualitatives conséquentes de T sans devoir réaliser des calculs exacts. »

De Regt and Dieks utilisent la loi de Boyle-Mariotte pour illustrer la pertinence de leurs critères : cette loi peut être utilisée pour obtenir une *compréhension qualitative* de la relation entre la température, la pression et le volume d'un gaz. Sans devoir recourir à des calculs compliqués, il est possible d'inférer qu'une diminution du volume d'un gaz dans un contenant va résulter en une augmentation de la pression à l'intérieur du contenant.

Une question s'impose alors : l'équation type d'un réseau de neurones représentée sur la figure 2.6 est-elle intelligible et permet-elle donc de comprendre pourquoi un réseau de neurones, après avoir analysé une image, nous indique que celle-ci contient un chat ?

3.2 Cadres d'explicabilité en IA

Il est important de souligner qu'il existe une terminologie assez large autour de la notion d'explicabilité dans le domaine de l'intelligence artificielle : explicabilité, interprétabilité, compréhension, transparence ... Il n'est pas toujours évident de saisir la différence entre ces termes ; en particulier, « explicabilité » et « interprétabilité » sont deux termes souvent utilisés de manière interchangeable[11] dans le domaine de l'IA.

Linardatos et al. (2021) proposent de définir l'**interprétabilité** comme les intuitions derrière les sorties d'un modèle ou les causes et effets entre les entrées et les sorties d'un modèle. L'**explicabilité** correspondrait quant à elle à la compréhension des fonctionnements internes d'un modèle, pendant son entraînement ou en inférence.

[11] La méta-analyse de Islam et al. (2022) conclue que cette absence d'unification et de clarté « entrave la bonne acquisition de connaissances concernant l'IA explicable ».

Nous pouvons alors faire l'observation suivante : en considérant le cadre de la « compréhension » proposé par De Regt and Dieks, l'explicabilité d'un modèle pourrait alors permettre l'interprétation des sorties de ce modèle. Si les mécanismes de fonctionnement sous-jacents sont compris, alors il doit être possible de comprendre leur influence sur les entrées/sorties, de la même façon que l'on peut inférer qu'un gaz chauffé dans un contenant va se dilater. Nous pouvons maintenant apporter une réponse à la question que nous avons posée avant de commencer cette discussion : il semble très invraisemblable que l'équation visible sur la figure 2.6 puisse être utilisée pour comprendre ou décrire qualitativement le modèle ou les sorties qu'il produit.

Dans le domaine de l'intelligence artificielle, lorsque l'on manipule des modèles complexes comme des réseaux de neurones profonds, il faut alors d'autres leviers pour expliquer les modèles. Les approches pour l'explicabilité en IA sont le plus souvent catégorisées comme suit :

— Approches globales et locales ;

— Approche *ante-hoc* et *post-hoc* ;

— Approches agnostiques ou spécifiques.

Les approches globales ont une portée assez large, comme leur nom l'indique, et concernent autant la compréhension du modèle que son algorithme et son protocole d'entraînement, l'accès à son code source et à ses données d'entraînement ou encore la déclaration des biais connus des concepteurs (du modèle ou des données). Les approches locales sont plus spécifiques et concernent par exemple l'identification, dans une donnée d'entrée, des éléments qui ont conduit le modèle à fournir une sortie donnée.

Les approches *ante-hoc* et *post-hoc* concernent le stade auquel l'explicabilité est recherchée. Si le modèle utilisé est intrinsèquement explicable (par exemple un arbre de décision), alors on considèrera que c'est une approche *ante-hoc*. Les modèles qualifiés de « boîtes noires », non explicables intrinsèquement du fait de leur complexité, sont expliqués par des approches *post-hoc* : un modèle auxiliaire est utilisé pour expliquer le modèle.

La cible des approches d'explicabilité peut concerner tout type de données (images, textes, données tabulaires ...) ou de modèles (réseaux de neurones, régression, machines à vecteurs de support ...), elles sont dans ce cas qualifiées d'agnostiques.

Lorsque les méthodes d'explicabilité ciblent spécifiquement un type de donnée (par exemple des images) ou une classe de modèles, elles sont spécifiques.

En pratique, les explications sont le plus souvent générées :

— en quantifiant la contribution des caractéristiques des données pour déterminer lesquelles influencent le plus une décision ou une prédiction réalisée par le modèle (*feature attribution* (Google Cloud AI platform)). La figure 2.8, extraite de Bordt et al. (2022), est un exemple de quantification des caractéristiques pour une prédiction d'un modèle sur une instance. On voit sur cette figure que c'est la caractéristique F_6 qui a le plus contribué à la prédiction, elle constitue donc l'explication de la sortie du modèle ;

— en utilisant des instances contrefactuelles (Guidotti, 2022), c'est-à-dire, pour une prédiction $y = f(x)$ où y est la prédiction du modèle f pour le point x (une instance particulière du jeu de données), en trouvant un point x' le plus proche possible de x donnant lieu à une décision $y' = f(x')$ différente de y. Considérons l'exemple d'un client d'une banque dont la demande de prêt a été rejetée. Une explication contrefactuelle pourrait être : « si les revenus avaient été 10% plus élevés que ce qu'ils sont actuellement, le prêt aurait été accordé ».

Examinons deux approches *post-hoc* représentatives visant à améliorer l'explicabilité des intelligences artificielles.

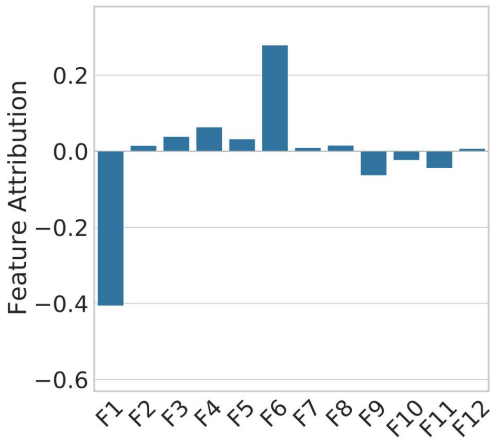
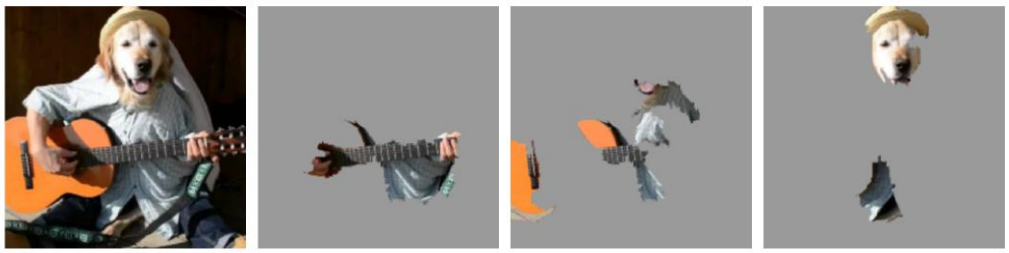


FIGURE 2.8 : Exemple de quantification des caractéristiques influençant la prédiction d'un modèle pour une instance donnée. Figure extraite de (Bordt et al., 2022).

LIME : *Local Interpretable Model-Agnostic Explanations*

LIME (Ribeiro et al., 2016) est une approche d'explicabilité devenue rapidement parmi les plus populaires en 2016[12]. Cette approche *post-hoc* est agnostique et locale. Un modèle auxiliaire va apprendre le comportement du modèle initial (il s'agit donc d'un modèle du modèle ; approche *post-hoc*) en appliquant des perturbations à une donnée d'entrée pour observer comment la sortie associée à cette donnée change (approche locale). Par exemple, si la donnée est une image, on peut occulter différentes portions (des caractéristiques) de celle-ci : si la sortie change, il est raisonnable de penser que la ou les caractéristiques manquantes ont contribué de manière importante à la sortie du modèle initial. On peut alors utiliser ces éléments comme explications, ainsi qu'illustré sur la figure 2.9.

Dans la section 1, nous avions évoqué les problèmes de représentation par les modèles d'apprentissage, où des associations fallacieuses peuvent être établies par les modèles pour caractériser des objets. Une approche telle que LIME permet de mettre en avant de telles associations, ainsi qu'on peut le voir sur la figure 2.10 : un husky est identifié comme étant un loup et l'utilisation de LIME permet de comprendre que la neige, et non pas l'animal, est l'élément caractéristique qui a permis au modèle de faire cette classification.

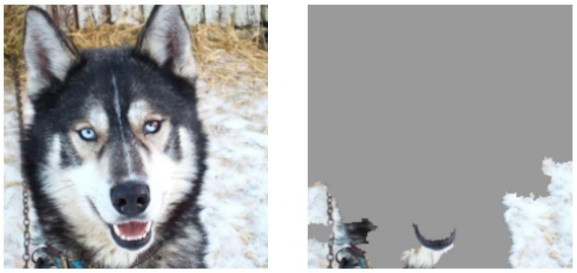


(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Figure 2.9 : Exemple d'application de LIME : chaque classe prédite par le modèle donne lieu à des explications spécifiques. Figure extraite de (Ribeiro et al., 2016).

[12] L'année 2016 est une année charnière qui a vu le sujet de l'explicabilité en IA exploser (Islam et al., 2022).



(a) Husky classified as wolf (b) Explanation

FIGURE 2.10 : Identification par LIME des caractéristiques de l'image ayant conduit le modèle à sa prédiction. Image extraite de (Ribeiro et al., 2016).

Utilisation de cartes de saillance pour l'explicabilité des réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN, *Convolutional Neural Networks*) sont une architecture spécifique de réseaux de neurones artificiels pour l'apprentissage sur des images. Inspirés du fonctionnement de la voie ventrale du cerveau, ils sont capables d'extraire des hiérarchies de caractéristiques qui sont par la suite regroupées pour former des éléments de complexité croissante pour reconstituer des objets.

Les CNN sont des réseaux de neurones profonds par nature peu explicables et interprétables. La façon la plus répandue d'expliquer les sorties de ces modèles est de construire des cartes de saillance permettant de mettre en évidence, dans les données d'entrées, les portions des images qui ont le plus contribué à la sortie associée à ces entrées (Fig. 2.11). Popularisées par le travail séminal de Zeiler and Fergus (2014), il existe aujourd'hui beaucoup de méthodes différentes pour calculer de telles cartes de saillance utilisant des approches de déconvolution, les gradients circulants dans le modèle ou encore les cartes d'activation apprises et représentant les caractéristiques présentes dans les données. Il s'agit d'approches locales, *post-hoc* et spécifiques.

Les explications visuelles telles que celles offertes par les cartes de saillance ou LIME appliqué aux images sont les approches les plus courantes (Islam et al., 2022). Il est également possible d'obtenir des explications numériques (*e.g.* évaluation de l'importance des variables des données pour une classe donnée (Moradi and Samwald, 2021)), basées sur des règles (*e.g.* génération d'arbres de décision pour l'explication de réseaux de neurones artificiels (Confalonieri et al., 2020)) ou textuelles (*e.g.* génération d'explications argumentées en langage naturel (Zhong et al., 2019)).

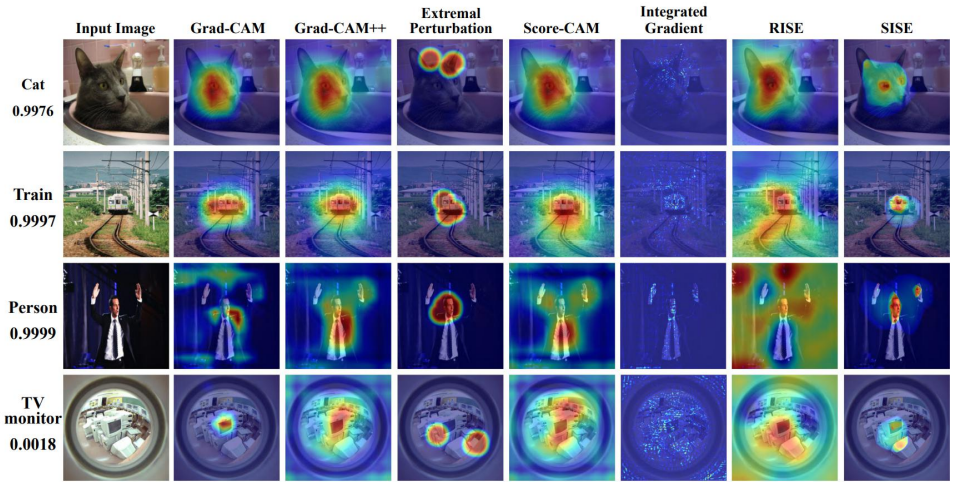


Figure 2.11 : Différents types de cartes de saillance pour l'explicabilité des réseaux de neurones convolutifs. Image extraite de Sattarzadeh et al. (2021).

Finalement, nous pouvons établir les correspondances suivantes entre des approches pour l'explication scientifique et l'explicabilité en IA :

— Raisonnement logique (*e.g.* modèle déductif-nomologique (Hempel and Oppenheim, 1948), prônant l'utilisation de lois pour expliquer et prédire des phénomènes) : explication basée sur des règles ou explications textuelles ;

— Faisceau de preuves (*e.g.* modèles de pertinence statistique (Salmon, 1971), visant à déterminer les caractéristiques C_i significatives à l'explication d'un phénomène) : identification des variables ou caractéristiques d'intérêt pour une prédiction donnée ;

— contrefactuels et manipulationnisme (Woodward, 2005), portant que la modification de variables d'entrées du modèle pour comprendre comment elles affectent ses sorties : perturbation des données d'entrées pour observer les modifications en sortie.

La question du contexte de l'explication n'est que très peu abordée dans la littérature (souvent technique) concernant l'intelligence artificielle explicable. Une exception notable est un article populaire de Miller (2019) qui aborde la question de l'explicabilité sous un angle pluridisciplinaire, mêlant à la fois des concepts de philosophie des sciences, des sciences cognitives et de la psychologie. Il est donc important de s'interroger sur quelle approche de génération d'explication est adaptée

à quel contexte (qui reçoit l'explication ? Pourquoi une explication est-elle nécessaire ? *etc.*). Ce sont toutefois des problématiques qui dépassent le cadre de ce document et que nous n'aborderons donc que marginalement dans notre conclusion.

En pratique, ce sont les approches *post-hoc* à la portée locale qui sont majoritairement utilisées. Pourtant, elles font l'objet de sérieuses critiques.

3.3 Critique des méthodes *post-hoc* pour l'explicabilité

Nous avons vu, sur la figure 2.11, que différentes cartes de saillance sont générées par les différentes approches de construction de ces cartes. Ces cartes différentes expliquent les prédictions d'un même modèle, sur les mêmes données et peuvent mettre en lumière des caractéristiques vraiment distinctes (pour le chat sur la première ligne, la tête est identifiée sur 4 des 7 cartes ; une carte met en avant les oreilles de l'animal ; une autre se focalise particulièrement sur les yeux). Y'a-t-il une carte de saillance qui soit « meilleure » que les autres ? Si oui, comment le déterminer ?

Les explications *post-hoc* visuelles seraient-elles des raisonnements circulaires (ou, à tout le moins, triviaux) ? Que faire de l'information « Pourquoi un chat a-t-il été identifié dans l'image ? » ⇒ « Parce qu'un chat a été identifié » ? Est-ce que l'on ne cherche pas, ici, à expliquer une *perception* ? S'il est indéniable que ce type d'approche permet de mettre l'accent sur l'objet identifié dans l'image, il est moins évident que cela corresponde à une *explication* ou à une *interprétation*. Il s'agit alors plutôt d'une démarche *descriptive* plutôt qu'explicative.

Quelques études portent un regard assez critique vis-à-vis des approches *post-hoc* pour l'explicabilité en IA. Bordt et al. (2022) questionnent le bien-fondé de ces approches dans les contextes qu'ils qualifient d'*adversariaux*, c'est-à-dire lorsque deux parties ont des intérêts opposés (par exemple un client d'une banque se voyant refuser l'octroi d'un prêt par un algorithme et qui conteste la décision et la banque). Selon Bordt et al., les approches *post-hoc* telles qu'utilisées aujourd'hui sont pour ainsi dire caduques, et ce pour deux raisons principales : (1) la représentation du monde construite par l'algorithme est, par nature, incomplète et approximative ; (2) même si l'on se restreint à la représentation construite par l'algorithme, il n'est pas possible de déterminer une explication plus vraie ou meilleure que les autres.

Les données utilisées pour entraîner un modèle étant finies, le modèle aura nécessairement une vision restreinte du monde (d'autant plus si l'algorithme ne parvient pas à capturer toutes les relations possibles entre les attributs des données

— en sachant qu'il est aussi possible qu'il apprenne des relations fausses, comme nous l'avons déjà vu). De fait, si une explication pour un point x existe dans le monde réel, il est très probable qu'elle soit inaccessible à l'algorithme.

Trouver une « vraie explication » dans la représentation du monde construit par l'algorithme semble aussi peu probable à cause de la complexité sous-jacente des modèles. Indépendamment de l'approche utilisée pour l'explicabilité, il est toujours possible de trouver des explications différentes pour un couple (x, y) (pour un même modèle, pour une même fonction décisionnelle, pour les mêmes données). C'est ce que l'on a observé sur la figure 2.11. Si l'on peut être amené à penser que cela est vrai uniquement pour les problèmes à haute dimension, Bordt et al. ont montré que les problèmes à faible dimension sont sujets au même écueil : en appliquant différentes approches d'explicabilité à un jeu de données à 12 attributs utilisés par un modèle pour une tâche de classification binaire, toutes les explications sont différentes les unes des autres, ainsi qu'illustré sur la figure 2.12 (dans un contexte adversarial, il suffit alors de choisir l'explication la plus avantageuse à la partie devant fournir une explication).

Il y a actuellement très peu de parades pour pallier ce problème. Dans le cadre des explications contrefactuelles, Laugel et al. (2019) ont essayé d'exploiter une approche topologique pour relier une explication contrefactuelle à une instance du jeu d'entraînement pour mettre en œuvre une justification de l'explication par la vérité terrain (réduite aux données ayant servi à entraîner le modèle). Les résultats obtenus sont mitigés et les auteurs concluent en précisant qu' « il n'existe aucune manière satisfaisante de fournir des explications *post-hoc* qui sont à la fois fidèles au modèle et à la vérité terrain ».

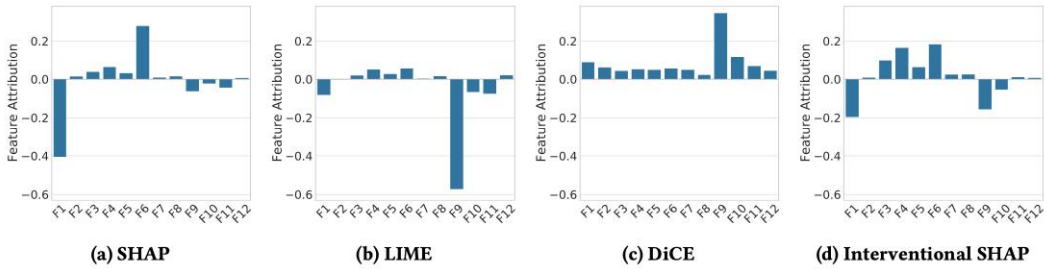


Figure 2.12 : Pour une même instance x passée dans le même modèle entraîné sur les mêmes données, quatre approches distinctes d'explicabilité donnent quatre résultats très différents. Figure extraite de (Bordt et al., 2022).

Pour Rudin (2019), c'est le paradigme de l'utilisation à tout prix des modèles de boîtes noires qu'il faut remettre en question, ainsi que les efforts à déployer pour expliquer ces modèles. Rudin caractérise une explication comme « un modèle indépendant supposé répliquer la plupart des comportements d'une boîte noire ». Ainsi, une explication ne serait plus la question de comprendre comment le monde fonctionne (c'est-à-dire expliquer l'occurrence d'un phénomène ou comprendre des tendances), mais comment un modèle en lui-même fonctionne. Questionnant la terminologie autour de l'intelligence artificielle explicable, Rudin souligne que les explications sont réduites à des résumés statistiques des prédictions du modèle, mettant en lumière comment les prédictions sont liées aux attributs et caractéristiques des données. En effet, comme nous l'avons souligné à plusieurs reprises, des explications *post-hoc* d'un modèle de substitution ne sont pas forcément représentatives du véritable processus ou pourraient reposer sur des facteurs non pertinents (et donc être fausses). De plus, Rudin critique vivement le compromis exactitude / interprétabilité promu par la DARPA (Gunning and Aha, 2019) depuis le démarrage de leur programme d'intelligence artificielle explicable (XAI : *eXplainable Artificial Intelligence*) : avec des données bien structurées et solidement représentées, un modèle intrinsèquement explicable s'en sort aussi bien qu'une boîte noire (Rudin and Radin, 2019). Il n'est pas toujours aisé de prétraiter des données ; ceci est d'autant plus vrai que les données sont complexes (des images par exemple), mais, dans de tels cas, Rudin reconnaît le bien-fondé de modèles complexes.

Elle critique aussi les approches de visualisation pour l'explicabilité, arguant que ces approches ne fournissent pas une explication convenable du fait qu'elles ne font qu'exhiber des portions de l'image utiles pour une prédiction donnée, sans rien apporter sur la façon dont ces informations ont été utilisées. Ainsi, elle propose que les modèles complexes intègrent des mécanismes intrinsèques à des fins d'explicabilité. Li et al. (2018) présentent dans ce cadre une architecture de réseau de neurones profonds intégrant des *couches prototypiques*. Ces modèles apprennent des prototypes spécifiques à chaque classe du domaine et les cherchent dans les données d'entrée lors du processus d'inférence. Une somme pondérée des similarités entre prototypes détermine la classe à assigner et constitue également une explication au fait qu'une instance appartient à une classe particulière. Dans ce contexte, l'explication fournie est alors représentative du processus réalisé par le modèle.

4 Synthèse et conclusion sur l'état des lieux

Les systèmes d'IA sont très répandus, mais faillibles : les algorithmes sont opaques du fait de leur complexité ou simplement qu'ils sont propriétaires ; ils peuvent établir des corrélations fallacieuses entre les attributs des données ; sont sujets aux biais inhérents de nos sociétés, ce qui est renforcé par le fait que les données ne sont, le plus souvent, que représentatives de nos sociétés occidentales ; ils peuvent influencer leurs utilisateurs, portant ainsi atteinte à leur autonomie ; ils sont sensibles et peuvent être attaqués de façon à altérer leur comportement.

Face à ces problèmes, quelles sont les options pour « incorporer » de l'éthique dans les IA ou leur utilisation afin de respecter des principes éthiques fondamentaux (bienfaisance, non-malfaisance, préservation de l'autonomie, justice, explicabilité), en plaçant l'explicabilité au centre ? Il existe deux approches principales pour l'IA et l'éthique : les approches *top-down* pour contraindre les IA dans un cadre éthique donné (*rightful machines*) ; les approches *bottom-up*, reposant sur du *crowdsourcing* pour concevoir des IA capables d'émettre des jugements moraux. Le bémol dans ce cas est que les sources de données peuvent être représentatives de comportements tout sauf moraux qui vont se refléter dans les modèles. En outre, la complexité des modèles utilisés pose aussi problème : ces systèmes ne peuvent pas être considérés comme éthiques s'ils ne sont pas interprétables.

Il se pose donc la question de savoir ce que l'on veut interpréter. Nous avons souligné qu'il existe deux types d'énoncés : des énoncés *d'observation* (caractériser des objets) et des énoncés *théoriques* (trouver les relations entre des objets, caractériser des lois). Ce sont ces énoncés que l'on veut expliquer. Dans le domaine de l'IA, deux grandes approches pour l'explicabilité existent : les approches intrinsèques ou les approches *post-hoc*. Ces dernières sont cependant vivement critiquables (et critiquées).

Où cela nous emmène-t-il ? Les points (toujours ouverts) abordés dans ce chapitre se sont naturellement présentés à nous alors que nous souhaitions étudier comment différentes approches d'intelligence artificielle permettent d'aborder des problèmes. Il est difficile d'affirmer que les explications que l'on peut générer pour une IA permettent de comprendre les raisons derrière la formulation, par ces IA, d'énoncés d'observation ou d'énoncés théoriques, et si ces raisons sont admissibles. Comment, alors, aller vers des intelligences artificielles que l'on peut considérer comme plus éthiques ?

Dans la partie suivante, nous présentons différents travaux menés depuis 2016 : ceux de Anna Ouskova-Leonteva, qui a mis au point des méthodes évolutionnaires pour l'optimisation de systèmes de refroidissement magnétocaloriques (chapitre II.3) ; ceux de Romain Orhand, visant à concevoir une intelligence artificielle autonome et explicable pour les environnements incertains (chapitre II.4) et ceux de Hiba Khodji qui travaille sur la détection d'erreur dans des séquences biologiques par des approches d'apprentissage profond, ainsi qu'à l'amélioration des approches *post-hoc* pour l'explicabilité (chapitre II.5).

Les différentes approches que nous avons choisi d'adopter dans ces différents travaux ont nourri (et nourrissent toujours) les réflexions présentées ici et nous laissent entrevoir différentes perspectives que nous présenterons dans le chapitre 6.

Deuxième partie

Contributions

Mise en perspective des contributions

Depuis que j'ai démarré mon travail de Maîtresse de Conférences, mes travaux de recherche ont porté sur la compréhension et le développement des énoncés construits par l'intelligence artificielle.

Le premier travail décrit dans cette partie portera sur la caractérisation physique de dispositifs de réfrigération magnétocalorique, utilisant différentes lois physiques dans un modèle bien défini. Ce travail est en ceci analogue à la démarche d'Urbain Le Verrier lorsqu'il a utilisé la loi de la gravitation universelle de Newton pour caractériser les propriétés d'un corps céleste putatif qui s'est avéré être observé quelque temps après. La différence est qu'aujourd'hui nous disposons d'approches bien plus puissantes pour spécifier les différentes caractéristiques de l'objet qui nous intéresse et, dans le cas de la thèse d'Anna Ouskova-Leonteva, nous décrirons le paradigme évolutionnaire multiobjectif qui aura permis d'aboutir à ce résultat.

Le deuxième travail (thèse de Romain Orhand), s'attaquera à ce que devra faire une IA en l'absence d'un modèle bien défini, qui est une autre source d'écueil aux IA : l'incertitude propre aux environnements du monde réel, l'incertitude dans laquelle se trouve un agent (humain ou artificiel) qui n'a que peu voire pas d'information sur l'état de son environnement, impliquant que cet agent ne pourra que difficilement anticiper les changements qu'une décision va causer dans cet environnement.

Le troisième travail s'intéressera à la prédiction d'erreurs dans des alignements de séquences de gènes, par des réseaux de neurones convolutifs. Ici aussi, le processus d'apprentissage induira un modèle, mais celui-ci restera inaccessible à cause de la nature opaque et complexe des réseaux de neurones. Pour mieux décrire les résultats du modèle, il faudra donc développer des mécanismes d'explicabilité *post-hoc*. Du fait des questionnements concernant la validité ou le bien-fondé de l'explicabilité *post-hoc*, nous présenterons une approche quantitative de l'explicabilité des réseaux de neurones convolutifs pour évaluer de telles explications.

CHAPITRE 3

Approche évolutionnaire pour l'optimisation de systèmes de réfrigération magnétique

Ce chapitre présente une approche exploitant une loi existante, bien définie et intelligible, pour caractériser les paramètres d'un dispositif de génération de froid magnétique. Le dispositif étudié demande la résolution de multiples objectifs et peut fonctionner en deux modes antagonistes. Ainsi, le cadre choisi pour résoudre ce problème d'optimisation est celui des algorithmes évolutionnaires multi-objectifs.

1 Introduction

La génération de froid représente 17% de la consommation d'électricité au point de vue mondial (Ministère de l'Agriculture et de la Souveraineté alimentaire) pour préserver les denrées alimentaires, les médicaments ou vaccins thermosensibles ou encore « conditionner » de l'air. Les systèmes de refroidissement actuels, gourmands en énergie, fonctionnent avec des gaz frigorigènes comprimés puis détendus, ce qui pose d'importants problèmes environnementaux (appauvrissement de la couche d'ozone, aggravation de l'effet de serre). Parmi les alternatives pour la génération de froid, on trouve la réfrigération magnétique. Reposant sur l'*effet magnétocalorique* (*Magneto-Caloric Effect*, MCE), la production de froid s'opère par le réchauffement (resp. le refroidissement) d'un matériel magnétocalorique (*Magneto-Caloric Material*, MCM) alors que celui-ci est aimanté (resp. désaimanté). Utiliser l'MCE pour la

production de froid permet non seulement de se passer des gaz polluants utilisés dans les systèmes actuels, de réduire le bruit du système du fait de l'absence de compresseur, de concevoir des systèmes de refroidissement plus compacts, mais aussi d'obtenir de meilleurs rendements (Lebouc et al., 2005).

L'MCE est une propriété physique des matériaux magnétocaloriques se manifestant comme un changement de température réversible (réchauffement ou refroidissement) lorsque le matériau est au voisinage de sa température de Curie (la température à laquelle des MCM peuvent voir leurs propriétés physiques changer drastiquement sous l'effet d'un champ magnétique externe).

Pour des applications de réfrigération à température ambiante, le matériau utilisé doit avoir un MCE élevé au voisinage de cette température. Par exemple, le gadolinium a une température de Curie T_c de $293K$ (soit $19.85°C$). Son MCE à cette température est d'environ $3K$ sous un champ magnétique de 1 Tesla. Autrement dit, on peut attendre des variations de température dans l'intervalle $16.85°C$ – $22.85°C$. Cette variation est toutefois encore trop faible pour atteindre des écarts de températures de systèmes de réfrigération usuels (c'est-à-dire de l'ordre de dizaines de degrés Celsius). Pour pallier ce problème, une des solutions est d'utiliser un *régénérateur magnétique actif* (*Active Magnetic Generator*, AMR), un système contenant le matériau magnétocalorique et permettant la circulation d'un fluide caloporteur pour réaliser des transferts thermiques. Des phases d'aimantation et de désaimantation adiabatiques (*i.e.* sans échange énergétique du matériau avec son environnement) vont respectivement réchauffer et refroidir le matériau (Fig. 3.1, phases t_1 et t_3). Le passage du fluide caloporteur (Fig. 3.1, phases t_2 et t_4) permet de réaliser des transferts thermiques avec le solide, produisant un cycle similaire à celui des systèmes réfrigérants à compression. Ce cycle va permettre une amplification des variations de température du système de réfrigération et donc l'obtention d'écarts de température significatifs même à température ambiante.

Un AMR peut fonctionner en deux modes (l'un ou l'autre ou les deux en même temps) : comme un système de réfrigération (*Magnetic Refrigeration System*, MRS) ou comme un générateur thermomagnétique (*Thermomagnetic generator*, TMG), pour convertir en énergie la chaleur perdue.

La conception d'AMR ou leur amélioration est complexe, car ce sont des systèmes multiphysiques avec un nombre important de paramètres ; il dépend par ailleurs du MCM utilisé, qui est un problème d'optimisation à lui tout seul : il est nécessaire de déterminer quelles propriétés physiques le matériau doit avoir pour assurer une

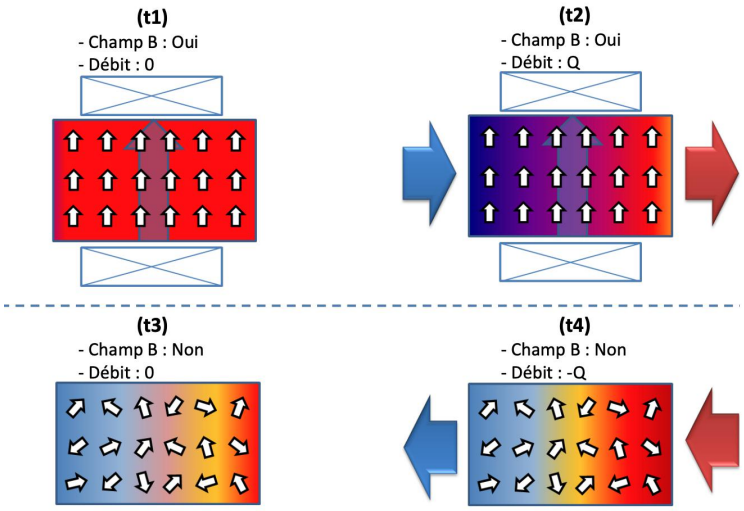


FIGURE 3.1 : Cycle de fonctionnement d'un régénérateur magnétique actif (AMR). Figure extraite de El Achkar et al. (2015)

réfrigération sur une plage de température comparable aux systèmes de réfrigération conventionnels. Les paramètres de l'AMR doivent également être ajustés pour maximiser les performances du système de réfrigération ou de génération. Idéalement, il faut résoudre en même temps le problème du matériau et celui de l'AMR.

Des simulations sont alors utilisées pour faciliter l'optimisation d'un tel système de refroidissement, mais elles sont intensives en calcul du fait de la haute dimensionnalité du problème et du caractère multiobjectif de la tâche. Cette complexité contraint le nombre de simulations qu'il est possible de réaliser alors que le nombre de paramètres et de configurations physiques possibles en demanderait beaucoup.

2 Modèle numérique d'AMR

Un des objectifs des travaux de la thèse de Mme Ouskova-Leonteva était de réaliser une optimisation des paramètres d'un modèle d'AMR. Le modèle utilisé a été proposé par Risser et al. (2013). Ce modèle est composé d'un modèle thermique qui est une discrétisation en une dimension des comportements du liquide caloporteur et de la chaleur dans l'AMR, et d'un modèle du champ magnétique en 3D. Les deux modèles sont couplés par le champ magnétique interne H_{int} à l'origine du MCE, ainsi que par la température et son impact sur le MCM. Une représentation schématique de l'appareil modélisé est visible sur la figure 3.2 : l'AMR est constitué de canaux de

refroidissement rectangulaires parallèles au flux magnétique. Deux réservoirs sont situés de part et d'autre de l'appareil et sont connectés à deux échangeurs de chaleur (source froide et source chaude).

L'advection 1D de la chaleur dans un canal est modélisée par l'équation 3.1, les termes de l'équation sont explicités dans le tableau 3.1.

$$\rho_f C_F \left(\frac{\partial T_f}{\partial t} + u \frac{\partial T_f}{\partial x} \right) = k_f + \frac{\partial^2 T_f}{\partial x^2} + \dot{Q}_{\text{visco}} + \dot{Q}_{HT} \tag{3.1}$$

L'équation 3.2 modélise, en 1D, la chaleur dans le MCM. Ses termes sont explicités dans le tableau 3.2.

$$\rho_s C_{H,p} \frac{\partial T_s}{\partial t} = k_s \frac{\partial^2 T_s}{\partial x^2} + \dot{Q}_{MC} + \dot{Q}_{\text{leak}} - \dot{Q}_{HT} \tag{3.2}$$

$\dot{Q}_{MC}$ permet de réaliser un premier couplage entre le modèle thermique et le modèle magnétique. Ce terme est calculé à partir de la variation adiabatique de température due à la variation du champ magnétique interne H_{int} : ${}^{\partial T_{\text{ad}}}/_{\partial H_{\text{int}}}$(équation 3.3).

$$\dot{Q}_{MC} = \frac{\partial T_{\text{ad}}(T_s, H_{\text{int}})}{\partial H_{int}} \frac{\partial H_{\text{int}}}{\partial t} \rho_s C_{H,p}(T_s, H_{\text{int}}) \tag{3.3}$$

L'induction B dans le volume du MCM est liée à la fois au champ magnétique externe H_e, à la magnétisation spontanée M et à la géométrie du matériau. C'est M

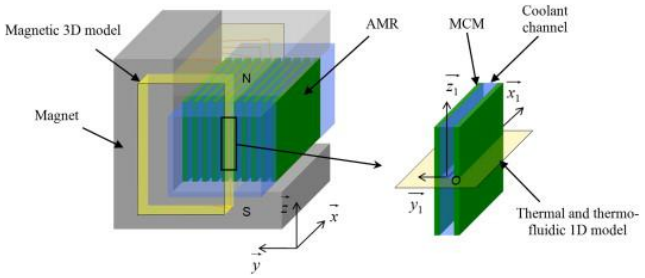


Figure 3.2 : Représentation schématique de l'appareil AMR modélisé. Figure extraite de (Risser et al., 2013).

ρ_f	Densité du liquide caloporteur
C_f	Capacité thermique du liquide caloporteur
T_f	Température du liquide caloporteur
k_f	Conductivité du liquide caloporteur
$\dot{Q}_{\text{visco}}$	Dissipation de chaleur due à la viscosité
$\dot{Q}_{HT}$	Transfert de chaleur entre le MCM et le liquide caloporteur
u	Vitesse de transfert de chaleur

Table 3.1 : Termes de l'équation d'advection de la chaleur dans un canal.

ρ_s	Densité du MCM
$C_{H,p}$	Capacité thermique du MCM à champ magnétique et pression constants
T_s	Température du MCM
k_s	Conductivité thermique du MCM
$\dot{Q}_{\text{leak}}$	Perte de chaleur due à l'isolation thermique imparfaite autour de l'AMR
$\dot{Q}_{MC}$	Génération de chaud ou de froid du fait de l'MCE (*cf.* eq. 3.3)

TABLE 3.2 : Termes de l'équation de la chaleur dans le MCM.

qui permet de réaliser le second couplage entre les modèles thermique et magnétique : les accroissements ou réductions de température du matériau et la magnétisation de celui-ci vont faire osciller le MCM d'un état ferromagnétique à un état paramagnétique aux alentours de sa température de Curie. Pour garantir la convergence de la ligne de champ vers le matériau, il faut appliquer des restrictions géométriques sur celui-ci (eq. 3.4),

$$\vec{B} = \mu_0(\vec{H_e} + \vec{M}(T_s, H_{\text{int}}) + \vec{H_d} \tag{3.4}$$

où H_d est un terme représentant la démagnétisation du champ et μ_0 est la perméabilité magnétique du vide.

Il y a donc, pour ce modèle d'AMR, deux jeux de paramètres d'entrées distincts :

— Les paramètres de conception physique, c'est-à-dire les dimensions des plaques de MCM constituant l'AMR et leur nombre, le nombre de canaux de circulation du fluide caloporteur et leur épaisseur, *etc.*

— Les paramètres de contrôle, comme la température initiale du système, la quantité de fluide circulant, la vitesse du fluide circulant, *etc.*

Pour évaluer le modèle, il faut calculer l'efficacité énergétique du système de réfrigération ou bien celle du générateur thermomagnétique, la densité de puissance thermique du système de réfrigération ou encore la densité de puissance mécanique du générateur thermomagnétique.

Pour la mise au point d'un AMR, les différents paramètres d'entrées et leurs combinaisons constituent donc un problème d'optimisation dont les évaluations possibles du modèle peuvent constituer des objectifs. C'est sous l'angle du **calcul évolutionnaire** que ce problème d'optimisation a été abordé. En particulier, les approches multiobjectives permettront de résoudre simultanément plusieurs objectifs, possiblement en conflits : par exemple, en mode réfrigérant, il est possible d'optimiser à la fois l'efficacité énergétique et la densité de puissance thermique. Si le système a

vocation à tourner en double mode (réfrigération *et* générateur), alors ce sont quatre objectifs qu'il faut résoudre simultanément. Idéalement, il faut donc une approche robuste capable de changer d'échelle, afin de travailler autant sur des problèmes avec un objectif que plusieurs, tout en étant performant.

3 Algorithmes évolutionnaires pour l'optimisation de problèmes

Les algorithmes évolutionnaires sont inspirés, comme leur nom l'indique, du processus darwinien de l'évolution des espèces. Des solutions potentielles à un problème à résoudre sont représentées sous la forme d'individus d'une population. Cet ensemble de solutions évolue dans un processus itératif (construction de générations successives), de manière à déterminer une solution la plus adaptée possible dans la population. La qualité d'une solution (d'un individu) est évaluée grâce à une fonction de *fitness* ; les meilleurs individus (les *parents*) sont sélectionnés et recombinés pour créer une descendance, ou bien en leur appliquant des mutations. Cette nouvelle génération (les *enfants*) est elle aussi évaluée et, des deux populations *parents* et *enfants*, des individus sont supprimés (les moins adaptés par exemple) pour maintenir la taille de la population à la taille initiale. L'algorithme s'arrête lorsqu'un critère d'arrêt est satisfait (par exemple, un nombre prédéfini de générations a été atteint), sinon l'algorithme itère à nouveau. Cet algorithme évolutionnaire type est illustré sur la figure 3.3.

L'espace de recherche explorable par ces algorithmes est très large et la stochasticité en fait des solutions de choix pour éviter de se retrouver pris au piège d'un optimum local dans cet espace. Alors que les opérateurs de recombinaison et de mutation permettent d'accéder à la diversité dans la population (et ainsi de favoriser la nouveauté ou d'éviter une convergence prématurée), la pression sélective permet d'augmenter la qualité moyenne de la population dans son ensemble.

Les composants essentiels des AE sont les suivants :

La représentation des individus Il faut pour commencer définir comment les solutions du problème traité sont représentées. L'ensemble des solutions est appelé *phénotype* (par exemple, l'ensemble des entiers naturels) et leur encodage —c'est-à-dire la représentation des individus)— est appelé *génotype* ou *chromosome* (par exemple une représentation binaire). La représentation est

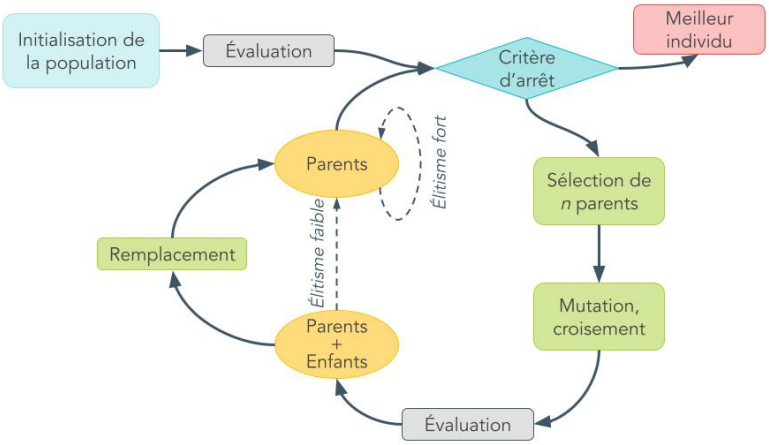


FIGURE 3.3 : Représentation schématique d'un algorithme évolutionnaire.

une mise en correspondance des phénotypes avec un ensemble de génotypes et, alors que l'espace de recherche exploré correspond aux génotypes, la solution représente un phénotype (un décodage du meilleur génotype).

La fonction d'évaluation (ou fonction de fitness) Cette fonction permet de déterminer la qualité des individus. Elle représente le problème que l'on souhaite résoudre et doit refléter la qualité du phénotype de l'individu. Un individu i aura une meilleur fitness qu'un individu j si sa *valeur de fitness* est plus élevée (il est donc plus proche de la solution).

La population Elle constitue un ensemble de solutions. Si les individus eux-mêmes ne varient pas, la population, elle, évolue durant le processus de l'algorithme. En général, la taille de la population (le nombre d'individus qu'elle contient) est constante et c'est pour maintenir cette taille que la boucle évolutionnaire comprend un mécanisme de remplacement. Le nombre de solutions différentes existant au sein de la population est appelé sa *diversité*.

Le mécanisme de sélection des parents Ce mécanisme permet de sélectionner les meilleurs individus pour la préparation de la génération suivante. Ce mécanisme n'est pas déterministe : les meilleurs individus ont une probabilité plus importante que les individus moins bons d'être sélectionnés, mais la possibilité de sélectionner de moins bons individus permet de maintenir de la diversité dans la population.

Les opérateurs génétiques (ou de variation) Le croisement (ou la recombinaison) est un opérateur binaire, qui va créer un génotype enfant à partir de la fusion des génotypes des deux parents sélectionnés aléatoirement. La sélection des parties du génome des deux parents qui seront utilisés dans le processus de croisement est elle aussi aléatoire. L'idée est de créer une descendance qui va agréger des caractéristiques respectives des deux parents. La mutation, quant à elle, est un opérateur unaire qui va appliquer une modification au génome d'un individu.

Le mécanisme de sélection des survivants (ou mécanisme de remplacement) Comme la taille de la population est en général constante, le mécanisme de remplacement va permettre de sélectionner les individus qui vont être conservés à la génération suivante. Contrairement à la sélection des parents, la sélection s'effectue sans remplacement. Si la sélection est *élitiste*, alors l'individu le plus adapté de la population n est systématiquement conservé dans la population $n+1$.

L'initialisation de la population est en général aléatoire. Le critère d'arrêt de l'algorithme est souvent un nombre maximal de générations atteint.

4 Optimisation multiobjective

Certains problèmes nécessitent la prise en compte de plusieurs objectifs, possiblement en conflit. Par exemple, si l'on souhaite s'acheter un vélo, on va chercher à trouver un vélo à la fois le plus léger et le moins cher possible. Pour réduire le poids d'un vélo, il faut utiliser des matériaux plus légers, comme du titane, mais cela va augmenter le coût : on devra donc déterminer un compromis acceptable entre ces deux variables de décision. Il s'agit d'un problème *multiobjectif* nécessitant l'optimisation simultanée de plusieurs fonctions d'objectif qui sont antagonistes : si l'on améliore le poids, on augmente le coût et inversement.

De manière plus formelle, un problème d'optimisation multiobjectif peut se définir comme un vecteur de fonctions objectif $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ où m est le nombre d'objectifs du problème, $\mathbf{x} \in \mathcal{F}$ (avec $\mathcal{F} \subseteq \mathcal{S}$) est un vecteur de variables de décision dans l'espace de recherche $\mathcal{S}$ (autrement dit, une solution dans l'espace de recherche).

Lorsque les variables décisionnelles sont en conflit (prévenant donc l'optimisation simultanée des différents objectifs), une solution au problème est une solution *optimale au sens de Pareto*, ou *non dominée*. Une solution non dominée ne peut être améliorée sans dégrader les valeurs d'autres objectifs. Une solution dominante est une solution qui n'est dominée par aucune autre. Cette solution obtient des valeurs au moins aussi bonne que toutes les autres solutions pour tous les objectifs et strictement meilleures pour au moins l'un d'eux, on parle alors de *dominance faible*. Une solution Pareto optimale représente alors le meilleur compromis entre des objectifs concurrents (Fig. 3.4). Il peut exister un ensemble de solutions Pareto-optimales, on parle alors de *front de Pareto*. On peut dans ce cas présenter un ensemble diversifié de solutions représentant autant de compromis sur les variables décisionnelles.

Les algorithmes évolutionnaires multiobjectifs (MOEA) sont dédiés à la résolution de tels problèmes. Une manière efficace de prendre en compte le caractère multiobjectif du problème à résoudre est d'exploiter la notion de front de Pareto. Dans ce cas, un classement (*ranking*) des solutions peut être réalisé selon une règle de dominance et chaque individu se voit assigner une valeur de fitness selon son rang dans la population. Utiliser un classement signifie que l'objectif doit être minimisé, car les rangs inférieurs correspondent à de meilleures solutions. La première approche par classement a été proposée par Goldberg (1989) (algorithme 1 et figure 3.5). Pour résoudre un problème d'optimisation multiobjectif, on cherche alors à déterminer le meilleur front de Pareto possible.

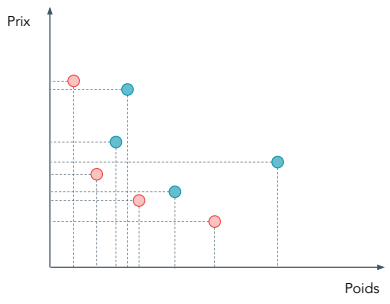


FIGURE 3.4 : Solutions Pareto-optimales (en rouge) représentant les deux meilleurs compromis prix/poids. L'ensemble des solutions Pareto optimales constitue le *front de Pareto*.

Algorithme 1 : Procédure de détermination du rang des individus

$i \leftarrow 1$ `// indice du front de Pareto courant (`F_1`:front de Pareto de P)`
$P' \leftarrow P$ `// population`
tant que $P' \neq \emptyset$ **faire**
    Identifier les solutions non-dominées dans P' et les assigner à F_i
    $P' \leftarrow P' + F_i$
    $i \leftarrow i + 1$
$\forall x \in P$ à la génération t, assigner le rang $r(x,t) = i$ si $x \in F_i$

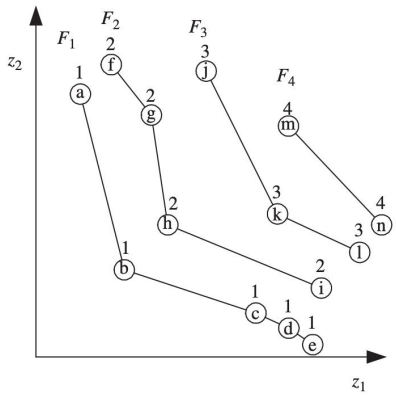


Figure 3.5 : Classement des solutions en utilisant le critère de dominance de Pareto. Figure extraite de (Konak et al., 2006).

5 Algorithme évolutionnaire multiobjectif pour l'optimisation du modèle d'AMR

Pour assister numériquement à la conception d'AMR, qui est un problème d'optimisation multiobjectif avec un nombre important de paramètres, il a été décidé d'utiliser une approche évolutionnaire. L'algorithme proposé, FastEMO (Leonteva et al., 2020), doit permettre de répondre aux problématiques suivantes :

— Nécessité de réduire le temps d'exécution et d'éliminer la variation de ce temps, d'une simulation : la simulation numérique du modèle d'AMR utilisé dans ces travaux (section 2 de ce chapitre) nécessite 2 à 15h de calculs en simple mode sur un processeur 16 cœurs AMD EPYC 7371 ;

— permettre un passage à l'échelle pour résoudre davantage d'objectifs ou pour pouvoir considérer davantage de paramètres ;

— permettre un meilleur contrôle des solutions : des solutions non dominées peuvent être très différentes dans l'espace de recherche (*i.e.* les paramètres du système peuvent prendre des valeurs très différentes), tout en étant semblables dans l'espace d'objectifs (*i.e.* les performances atteintes sont similaires). En outre, il existe dans la résolution de ce problème d'optimisation des solutions résistantes à la dominance (*i.e.* des solutions pouvant donner de très bons résultats sur un objectif et très mauvais pour d'autres).

La complexité d'un algorithme évolutionnaire d'optimisation multiobjectif dépend du nombre de générations qu'il est nécessaire de faire tourner pour déterminer la meilleure approximation possible du front de Pareto : ce nombre va dépendre du problème à résoudre, mais aussi de la taille de la population (d'où la nécessité du passage à l'échelle). Une taille de population importante permet d'améliorer le résultat de l'algorithme, mais cela demande aussi de faire tourner davantage de générations pour parvenir à une bonne convergence. L'opération demandant le plus de temps pour sa réalisation est l'évaluation de l'ensemble des individus. Heureusement, celle-ci peut se dérouler en parallèle pour chaque individu, faisant des processeurs parallèles (CPU et surtout GPU) des supports de choix pour exécuter ce type d'algorithmes.

FastEMO est un algorithme dérivé d'ASREA (*Archive-based Stochastic Ranking Evolutionary Algorithm*) (Sharma and Collet, 2010*a*,*b*). La complexité d'ASREA pour le calcul d'une génération est inférieure à $\mathcal{O}(n^2)$, où n est la taille de la population : pour les AE d'optimisation multiobjectif, une complexité de $\mathcal{O}(n^2)$ est n'est pas surprenante. ASREA parvient à descendre à une complexité en $\mathcal{O}(man)$, où m est le nombre d'objectifs, a est la taille de l'archive[1] et n la taille de la population. Cette meilleure complexité permet de travailler sur des tailles de populations importantes.

S'il a été décidé de concevoir un nouvel algorithme, c'est parce que des insuffisances existent au sein d'ASREA :

1. L'accroissement du nombre d'objectifs peut entraîner la génération de solutions dont la majorité est non dominée, rendant caduc le mécanisme de sélection. De plus, la présence de solutions résistantes à la dominance ne permettra pas facilement de résoudre le problème d'optimisation de l'AMR pour un fonctionnement en double mode (réfrigération et génération).

[1] Une archive permet de mettre de côté les solutions Pareto-optimales de façon à ne pas avoir à re-évaluer ces individus à la génération suivante. Le plus souvent, la taille de l'archive est fixe et des solutions peuvent y être remplacées si de meilleurs individus sont trouvés.

2. Lors du processus de mise à jour de l'archive, si des solutions ont le même rang, ASREA utilise un opérateur de sélection de *crowding distance* permettant de sélectionner des solutions dans une portion peu dense de l'espace d'objectifs. Cela permet de maintenir la diversité de la population et d'éviter une convergence prématurée. Cependant, des solutions similaires dans l'espace d'objectifs peuvent correspondre à des solutions différentes dans l'espace de recherche et il peut être intéressant de conserver ses solutions dans le cas de problèmes de conception comme celui de l'AMR : les solutions retenues doivent non seulement être bonnes au regard des objectifs, mais aussi *être physiquement réalisables*, il faut donc se garder de rejeter trop de bonnes solutions qui sont similaires dans l'espace d'objectifs, mais qui présentent pourtant des réalités matérielles très différentes dans l'espace de recherche.

5.1 Évaluation des solutions et dominance

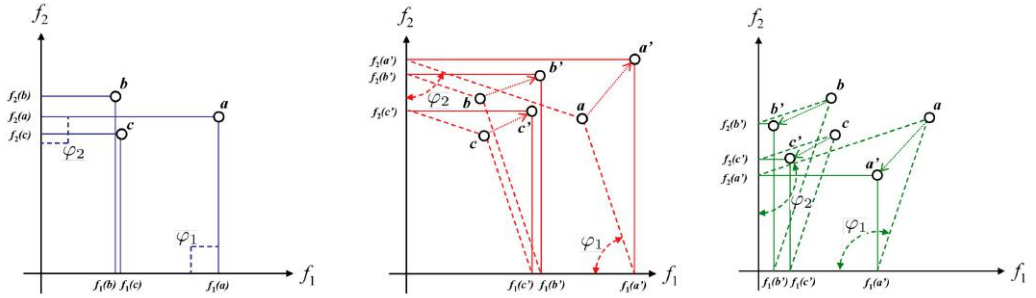
Pour améliorer à la fois la diversité de la population et sa convergence, FastEMO met en œuvre le mécanisme de contrôle des régions de dominance des solutions proposé par Sato et al. (2007) pour l'évaluation des individus.

Pour une solution $\mathbf{x}$ donnée, une unique région de dominance peut être déterminée avec $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$. Cette région peut être étendue ou rétractée par la modification de la valeur de fitness de chaque fonction objectif en adaptant le paramètre S dans l'équation :

$$f'_i(x) = \frac{r \cdot \sin(\omega_i + S_i \cdot \pi)}{\sin(S_i \cdot \pi)} \tag{3.5}$$

avec $1 \leq i \leq m$ et où r est la norme de $\mathbf{F}(\mathbf{x})$, $f_i(\mathbf{x})$ la valeur de fitness de l'objectif i et ω_i l'angle de déclinaison entre $\mathbf{F}(\mathbf{x})$ et $f_i(\mathbf{x})$: $\omega_i = \arccos({}^{f_i(\mathbf{x})}/_{r})$.

La figure 3.6, illustrant un problème de *maximisation* de deux objectifs f_1 et f_2, montre comment les régions de dominance des objectifs peuvent être modifiées selon la valeur du paramètre S_i : les valeurs des objectifs augmentent pour $S_i < 0.5$ et décroissent pour $S_i > 0.5$. Pour $S_i = 0.5$, $f'_i(\mathbf{x}) = f_i(\mathbf{x})$. Sur la figure 3.6a (la dominance « normale »), a domine c mais a et b, et b et c ne se dominent pas. Lorsque l'on étend les régions de dominance des deux objectifs (fig. 3.6b), a' domine maintenant b' et c', et b' domine c' : la pression de sélection est renforcée. Si on rétracte la région de dominance des deux objectifs (fig. 3.6c), a', b' et c' ne se dominent pas : la pression de sélection s'affaiblit.



(a) $S_1 = S_2 = 0.5$ (b) $S_1 = S_2 < 0.5$ (c) $S_1 = S_2 > 0.5$

FIGURE 3.6 : Modification des régions de dominance dans l'espace d'objectifs (pour un problème de *maximisation*). Si la région de dominance des deux objectifs est étendue ($S_i < 0.5$), la pression de sélection est renforcée. Si au contraire, la région de dominance est rétractée ($S_i > 0.5$), la pression de sélection s'affaiblit. Image extraite de (Sato et al., 2007).

Pour le paramétrage de S_i dans FastEMO, une approche empirique a permis de déterminer des plages de valeurs possibles selon le nombre d'objectifs (tableau 3.3).

Ce contrôle plus fin de la dominance des solutions dans l'espace des objectifs permet de mieux gérer le nombre d'objectifs du problème à résoudre. Cela permet notamment de mieux contrôler le nombre des solutions non dominées qui tend à s'accroître en même temps que le nombre d'objectifs. Des solutions moins bonnes peuvent tout de même être sélectionnées, maintenant de fait de la diversité dans la population.

5.2 Maintien de la diversité en exploitant l'espace de recherche

Pour maintenir la diversité du front de Pareto lors de l'évaluation et de l'archivage, si plusieurs solutions ont le même rang, on peut utiliser une mesure de distance pour choisir les solutions à sauvegarder dans l'archive. Une de ces mesures est la *crowding distance* (Deb, Pratap, Agarwal and Meyarivan, 2002*a*). Cette mesure permet de sélectionner, dans un front de Pareto, les solutions permettant de préserver

Nombre d'objectifs (m)	Plages de valeurs pour S_i
2	[0.495 ; 0.505]
3	[0.49 ; 0.5]
4	[0.485 ; 0.5]
5	[0.47 ; 0.49]

TABLE 3.3 : Détermination empirique des plages de valeurs de S_i en fonction du nombre d'objectifs.

la diversité des solutions dans l'espace d'objectifs (autrement dit, vont être privilégiées des solutions Pareto optimales dans des régions peu denses). En pratique, il s'agit d'évaluer la densité d'individus autour d'un individu i : pour cela, on calcule la distance moyenne de deux solutions autour de i selon chaque objectif. On peut alors déterminer la dimension du plus grand cuboïde contenant i mais aucune autre solution de la population (fig. 3.7).

Comme nous l'avons déjà indiqué, le problème de cette approche est que, dans certains cas comme le problème d'optimisation de l'AMR, il peut être intéressant de conserver des solutions similaires dans l'espace d'objectifs alors qu'elles sont différentes dans l'espace de recherche. Deb and Tiwari (2008) ont alors proposé une version alternative de la *crowding distance* permettant d'évaluer la densité autour d'une solution non seulement dans l'espace d'objectif, mais aussi dans l'espace de recherche (fig. 3.8).

Cette mesure permet ainsi, dans le cas de l'optimisation de l'AMR, de préserver des solutions de même rang qui sont proches dans l'espace d'objectifs, mais qui sont différentes dans l'espace de recherche.

5.3 Mutation des individus

Le choix d'un opérateur dépend du problème traité et en particulier de la dimension de son espace de recherche. Conventionnellement, les approches évolutionnaires utilisent des opérateurs polynomiaux ou gaussiens pour appliquer des mutations sur les chromosomes des individus. L'effet de tels opérateurs est d'appliquer une perturbation, dont la magnitude est échantillonnée à partir d'une distribution de

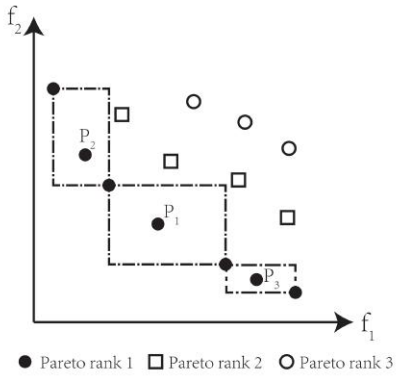


Figure 3.7 : Évaluation de la densité de solutions autour d'un individu. Figure extraite de (Gong et al., 2016).

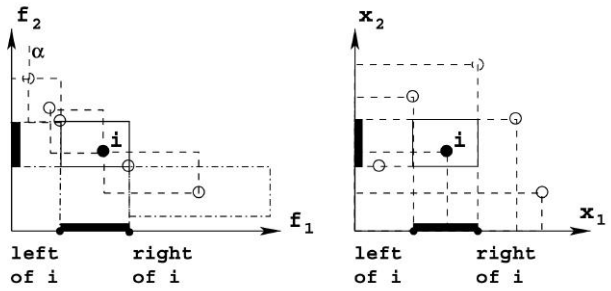


FIGURE 3.8 : Utilisation de la mesure de densité de solution à la fois dans l'espace d'objectif (à gauche) et dans l'espace de recherche (à droite). Figure extraite de (Deb and Tiwari, 2008).

probabilités, à l'individu sélectionné (le parent) de manière à créer un enfant voisin dans l'espace de recherche.

Pour résoudre des problèmes multimodaux, multiobjectifs et à grande dimension (comme c'est le cas pour l'optimisation d'un AMR), la loi de Cauchy est un opérateur tout indiqué (Hansen et al., 2006). La fonction de densité de la loi de Cauchy est exprimée comme :

$$P(x) = \frac{1}{\pi} \frac{b}{(x-m)^2 + b^2} \tag{3.6}$$

où b est la largeur de la distribution à mi-hauteur et m la médiane de la distribution. Cette distribution, qui n'a ni espérance ni variance et à laquelle on ne peut appliquer ni la loi des grands nombres ni le théorème central limite, permet de faire une exploration *anisotrope* de l'espace de recherche. Cela fait de cette approche une recherche par coordonnées : si cela reste un processus aléatoire, celui-ci est biaisé vers les directions des axes de recherche (Fig. 3.9). Autrement dit, en dimension n, on n'explore que $2n$ dimensions. Il est alors aisé de comprendre pourquoi, en haute dimension, cette recherche est plus efficace qu'une recherche isotropique telle que celle offerte par une distribution gaussienne.

La figure 3.10, extraite de (Hansen et al., 2006), montre les différentes résolutions d'un problème d'optimisation (une fonction de Rastrigin) par un algorithme évolutionnaire utilisant une distribution gaussienne (chemin bleu) ou une loi de Cauchy (chemin orange) comme opérateur de mutation. On note bien que, dans le cas de mutations avec une loi de Cauchy, la recherche est moins dispersée. En outre, cet opérateur de mutation permet également de réaliser des sauts plus importants dans l'espace de recherche. Dans cet exemple, l'algorithme a nécessité environ 9 000

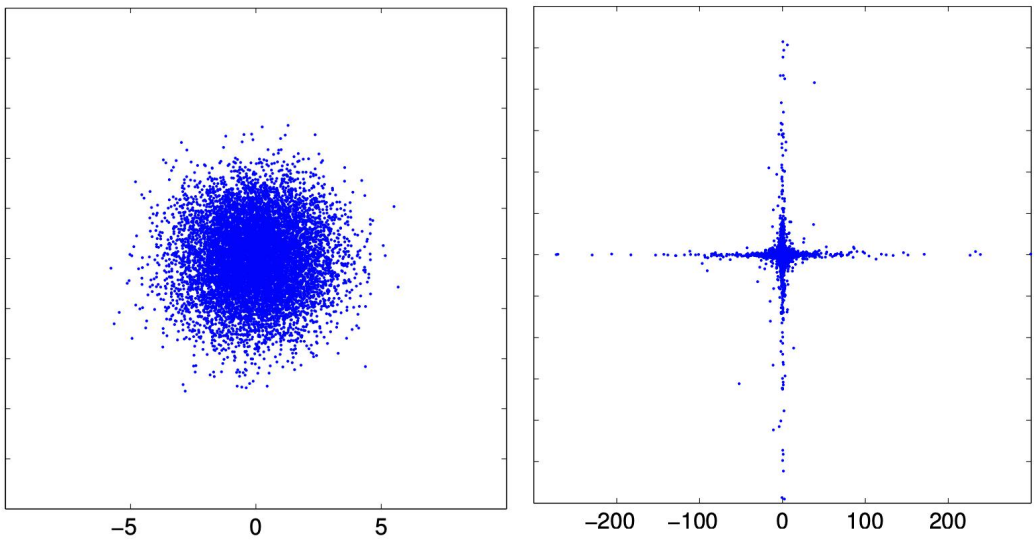


Figure 3.9 : Échantillonnages en 2 dimensions depuis une distribution gaussienne (a) et une loi de Cauchy (b). Figure extraite de (Hansen et al., 2006).

itérations pour atteindre l'objectif en utilisant la loi de Cauchy, contre environ 80 000 avec la distribution gaussienne.

Il faut également définir la probabilité $p_{\mathbf{mut}}$, le taux de mutation. Pour cela, il y a deux cas de figure : d'une part, cette probabilité peut être *statique*. Traditionnellement, cette probabilité vaut $1/d$, où d est le nombre de variables de décision (on rappelle que $\mathbf{x}$, une solution au problème, est un vecteur de variables de décisions $\mathbf{x} = (x_1, x_2, \ldots, x_d)$).

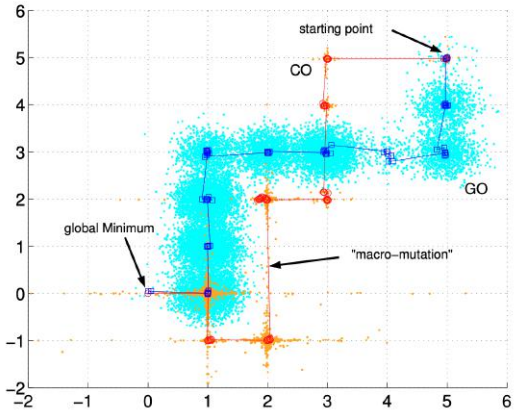


Figure 3.10 : Figure extraite de (Hansen et al., 2006).

Ce taux de mutation conduit, en moyenne, à la mutation d'une variable par génération. D'après Doerr et al. (2017), ce type de mutation est mal adapté aux problèmes multimodaux, car il est nécessaire de pouvoir réaliser des sauts plus importants dans l'espace de recherche pour déterminer de bonnes solutions. Nous pouvons noter que, comme nous l'avons déjà vu, ce problème peut se compenser au moins partiellement par l'utilisation d'une distribution à queue lourde, telle que la loi de Cauchy.

D'autre part, il est également possible de définir $p_{\mathbf{mut}}$ à chaque génération, le taux de mutation est alors *dynamique*. Doerr et al. (2017) définissent le taux de mutation comme θ/d, où θ est tiré aléatoirement dans $[1 \dots d/2]$ en utilisant une distribution basée sur une loi de puissance qui va, là aussi, favoriser la variabilité de la magnitude des sauts dans l'espace de recherche. Doerr et al. (2017) ont montré l'efficacité d'utiliser un taux de mutation dynamique pour les problèmes multimodaux.

Dans FastEMO, un taux de mutation dynamique est employé. $\theta \in [1 \dots d/2]$ est choisi aléatoirement en utilisant la loi de Pareto :

$$P(x) = \begin{cases} \frac{\alpha x_s^\alpha}{x^{\alpha+1}} & \text{if } x \geq x_s \\ 0, & \text{sinon} \end{cases} \tag{3.7}$$

où $x_s > 0$ est un paramètre d'échelle, permettant de spécifier la dispersion de la distribution et $\alpha > 0$ un paramètre de forme permettant de modifier la forme de la distribution (par exemple la forme de son sommet).

5.4 Validation de FastEMO

Les modifications apportées à FastEMO par rapport à ASREA (mécanisme d'évaluation des individus avec contrôle des régions de dominance, évaluation des solutions dans l'espace d'objectifs *et* dans l'espace de recherche, mutation des individus) ont été testées pour validation. D'autres comparaisons avec des algorithmes évolutionnaires populaires d'optimisation multiobjectif ont également été réalisées. Nous ne présentons ici qu'une vue très synthétique des résultats, dont le détail peut être consulté dans la thèse de doctorat de Mme Ouskova-Leonteva (Ouskova Leonteva, 2022).

Deux jeux de données pour l'optimisation multiobjective ont été utilisés : *Deb-Thiele-Laumanns-Zitzler Test Suite* (DTLZ) (Deb, Thiele, Laumanns and Zitzler, 2002) et *Walking Fish Group Test Suite* (WFG) (Huband et al., 2005).

Ce que l'on souhaite étudier lors de l'évaluation d'un front de Pareto sont les éléments suivants (Audet et al., 2021) :

— les solutions doivent minimiser la distance avec le vrai front de Pareto (si tant est qu'il soit connu) ;

— les solutions doivent être « bien » distribuées sur le front approximé ;

— l'étendue du front doit être maximale (*i.e.* pour chaque objectif, on veut idéalement qu'une grande plage de valeurs soit couverte par les solutions non dominées)

Une métrique très utilisée est l'*indicateur d'hypervolume* (HV), aussi appelée métrique S (comme Solution). L'indicateur d'hypervolume permet d'évaluer le volume des régions dominées dans l'espace d'objectifs. Formulé autrement (Audet et al., 2021), il s'agit d'évaluer « le volume de l'espace dans l'espace des objectifs dominés par l'approximation du front de Pareto S et délimité au-dessus par un point de référence $r \in \mathbb{R}$ pour toute solution $z \in S$ telle que $z \prec r$ » (*i.e.* z domine r). Cette notion d'hypervolume est illustrée sur la figure 3.11 : le meilleur front de Pareto entre A et B est le front A et $HV(A, r) > HV(B, r)$.

L'indicateur d'hypervolume est calculé en utilisant la mesure de Lebesgue en dimension m, λ_m :

$$HV(S, r) = \lambda_m \left(\bigcup_{z \in S} [z; r] \right) \tag{3.8}$$

FastEMO exhibe dans l'ensemble de très bonnes performances. L'efficacité de l'opérateur de mutation a été testée en comparant deux versions de FastEMO (l'une utilisant la loi de Cauchy, une autre en utilisant un opérateur de mutation polynomial) à deux versions de NSGA-II (Deb, Pratap, Agarwal and Meyarivan, 2002*b*) (l'une utilisant la loi de Cauchy et l'autre utilisant un opérateur de mutation polynomial). Les huit problèmes résolus étaient constitués de 3 objectifs et de 7, 12 et 24 variables de décision. Indépendamment de l'algorithme, l'opérateur de mutation avec loi de

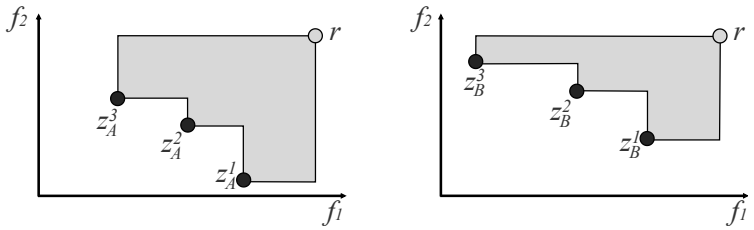


Figure 3.11 : Utilisation de l'indicateur d'hypervolume pour comparer deux fronts de Pareto A et B relativement au point de référence r (appelé point de Nadir). A est meilleur que B et $HV(A, r) > HV(B, r)$.

Cauchy permet systématiquement l'obtention de meilleurs fronts de Pareto. FastEMO obtient toujours des fronts de Pareto de qualité au moins égale à ceux calculés par NSGA-II.

FastEMO a été comparé à l'algorithme dont il est issu, ASREA, pour valider sa capacité à, comme ASREA, gérer de grandes tailles de population. Les problèmes résolus comportaient 3 objectifs et 7 et 12 variables de décision. La taille de la population a été fixée à $n = 10\,000$. La qualité des fronts de Pareto obtenue par FastEMO est systématiquement meilleure que celle des fronts de Pareto obtenus par ASREA. ASREA cependant demande un temps d'exécution moindre.

La capacité de passage à l'échelle de FastEMO est supérieure à MOEA-D (Zhang and Li, 2007), NSGA-III (Deb and Jain, 2014) et IBEA (Zitzler and Künzli, 2004). FastEMO passe de 0.00086 s de temps de calcul par génération pour $n = 100$ à 1.921 s pour $n = 100\,000$ alors que NSGA-III passe de 0.00157 s par génération pour $n = 100$ à 65.7 s pour $n = 100\,000$. Bien qu'inférieur à FastEMO, MOEA-D montre aussi de bonnes capacités de passage à l'échelle : 0.00089s pour $n = 100$ à 3.5 s pour $n = 100\,000$. IBEA est loin derrière, avec un temps d'exécution de 3 150 s pour $n = 100\,000$.

Analyse d'un système AMR en double mode

Pour optimiser un AMR en dual mode (réfrigération et générateur), on considère quatre objectifs : l'efficacité énergétique et la densité de puissance dans les deux modes. Différentes combinaisons de paramètres pour la conception de l'appareil peuvent être étudiées et nous avons aussi vu que des solutions similaires dans l'espace d'objectifs peuvent être très différentes dans l'espace de recherche : il faut donc être capable de tenir compte de ces différentes solutions.

Les propriétés du MCM (capacité thermique et magnétisation) sont données par des mesures expérimentales. Trois variables décisionnelles sont utilisées : L, la largeur de l'AMR dans la direction d'écoulement du fluide caloporteur ; $R_{\mathbf{vol}}$, le ratio de volume de fluide caloporteur transféré entre les extrémités de l'AMR sur le volume total de fluide de l'AMR et f, la fréquence de fonctionnement de l'AMR.

La taille de population de l'algorithme est $n = 100$ et le modèle exécute 50 générations.

La figure 3.12 montre les fronts de Pareto obtenus pour chacun des deux modes et souligne, comme on s'y attend, le conflit entre les deux modes en termes de densité

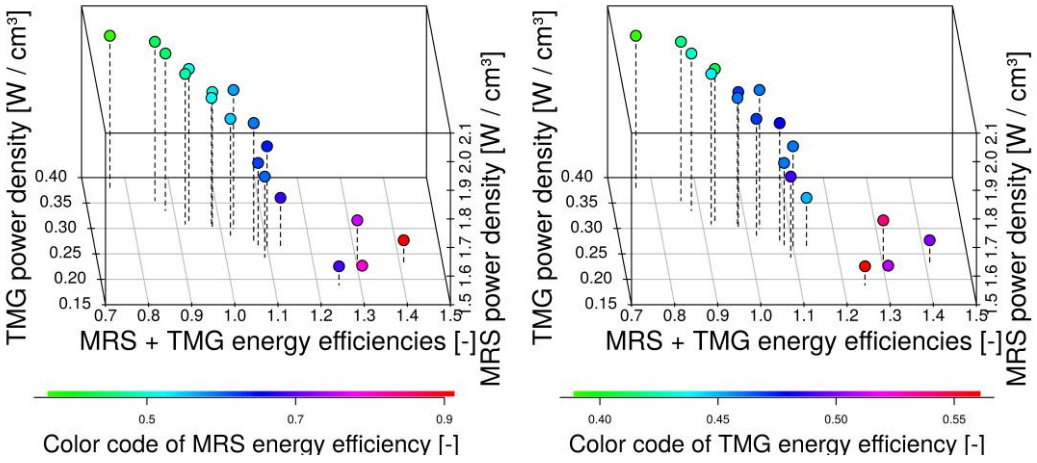
de puissance et d'efficacité énergétique. Il est intéressant de noter la discontinuité dans les fronts de Pareto obtenus : un certain nombre de paramètres de l'AMR sont donnés dans le modèle, par exemple le diamètre des canaux de circulation du fluide caloporteur. Ces paramètres ont été spécifiés pour le mode réfrigérant de l'AMR et il est possible que la discontinuité soit causée par une absence de solution optimale pour les deux modes pour cette configuration matérielle donnée.

Chaque variable influence différemment les performances globales des modes du système (Fig. 3.13) :

— En TMG, une fréquence plus importante de l'AMR entraîne un accroissement de la densité de puissance et une baisse de l'efficacité énergétique.

— En MRS, L est plus petit pour maximiser l'efficacité énergétique qu'en TMG et f est 1.7 fois plus important en MRS qu'en TMG. R_{vol} a peu d'influence sur l'efficacité énergétique entre les deux modes.

— Les configurations des variables sont similaires en MRS et TMG pour la densité de puissance.



(a) (b)

Figure 3.12 : Fronts de Pareto pour l'efficacité énergétique et la densité de puissance des deux modes (réfrigération (A) et génération (B)). Les deux modes sont antagonistes : de bonnes performances dans un mode entraînent de mauvaises performances dans l'autre mode (Leonteva et al., 2022).

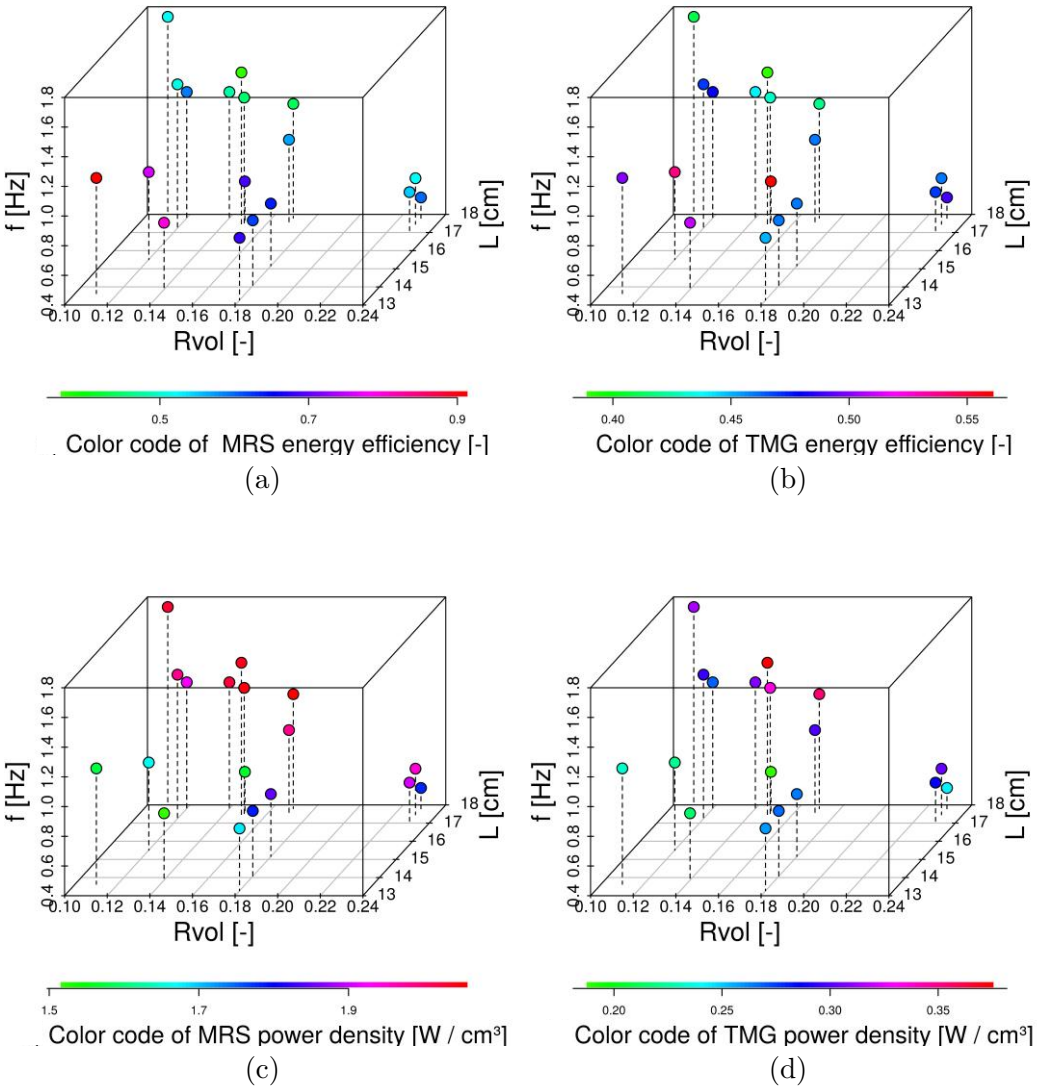


FIGURE 3.13 : Distribution des solutions Pareto-optimales. Figure extraite de (Leonteva et al., 2022).

Cette approche est donc intéressante pour comprendre comment les différentes variables décisionnelles du système et leurs relations permettent d'arriver à des configurations différentes pour l'AMR, selon un mode ou l'autre. Par ailleurs, il existe aussi des solutions qui, bien que n'étant pas optimales, représentent un compromis entre les deux modes en termes d'efficacité énergétique et de densité de puissance.

6 Synthèse

La caractérisation physique de dispositifs de réfrigération magnétocalorique, utilisant différentes lois physiques dans un modèle bien défini (eq. 3.1 à 3.4), est analogue à la démarche d'Urbain Le Verrier lorsqu'il a utilisé la loi de la gravitation universelle de Newton pour caractériser les propriétés d'un corps céleste putatif qui s'est avéré être observé quelque temps après. La différence est qu'aujourd'hui, nous disposons d'approches bien plus puissantes pour spécifier les différentes caractéristiques de l'objet qui nous intéresse. En particulier, nous avons vu dans ce chapitre comment le calcul évolutionnaire peut permettre de déterminer les caractéristiques d'un système complexe tel qu'un régénérateur magnétique actif qui, de surcroît, peut fonctionner en deux modes antagonistes. Le cadre des algorithmes évolutionnaires permet d'obtenir des résultats qui sont non seulement bons, mais aussi interprétables, encore que l'approche présentée ici n'a concerné qu'un petit ensemble de variables décisionnelles. Il n'en reste pas moins que les lois caractérisant le fonctionnement du système, les objectifs du problème d'optimisation et ses variables décisionnelles restent intelligibles de bout en bout.

Dans le prochain chapitre, nous allons changer de paradigme et nous intéresser à la production d'énoncés théoriques. Les travaux de Romain Orhand ont porté sur la conception et le développement d'une intelligence artificielle autonome et explicable pour les environnements incertains. Ce qui nous a intéressés dans ces travaux sont les capacités de certains modèles d'intelligence artificielle à produire des règles / des lois pour décrire le fonctionnement d'un système ou les interactions entre les objets d'un environnement.

CHAPITRE 4

Intelligence artificielle autonome et explicable pour les environnements incertains

Un autre écueil des IA est leur difficulté à évoluer dans des environnements incertains*. Dans ce chapitre, nous présentons une classe de modèle capable non seulement d'établir des règles décrivant le fonctionnement de l'environnement mais aussi d'anticiper les effets d'actions réalisées sur l'environnement : les systèmes de classeurs à anticipation. Les modèles que nous proposons appartiennent à cette classe de modèles et ont été améliorés pour être capable d'agir en toute autonomie en environnements incertains.*

1 Introduction

Nous avons vu dans le chapitre I.2 que l'utilisation des intelligences artificielles n'est pas sans poser un certain nombre de problèmes et avons établi une liste des causes principales à l'origine de ces problèmes (opacité des algorithmes, représentativité des données, capture de biais sociaux, *etc.*). Il existe une autre source d'écueils aux IA : l'**incertitude** propre aux environnements du monde réel. Une situation incertaine est une situation dans laquelle un agent (humain ou artificiel) n'a que peu, voire pas d'information sur l'état de son environnement ou que cet agent ne peut anticiper les changements qu'une décision va causer dans cet environnement. Il existe différents types d'incertitude qu'il est possible de classer selon leur issue (les

résultats, la situation et les alternatives) ou leur source (informations incomplètes, compréhension inadaptée et alternatives indifférenciées) (Lipshitz and Strauss, 1997).

La première dimension de cette classification de l'incertitude se focalise sur l'agent décisionnaire ; dans le contexte de l'intelligence artificielle, on pourrait par exemple vouloir étudier le degré de certitude d'un modèle par rapport à ses sorties. La seconde dimension de cette classification concerne l'environnement, le contexte dans lequel l'agent décisionnaire évolue. Ce sont sur les *sources* d'incertitude que les travaux présentés dans ce chapitre se focalisent.

Si des décisions doivent être prises dans un environnement incertain (que l'agent soit humain ou non), il est d'autant plus nécessaire que celles-ci puissent être expliquées, justifiées. Si l'agent se trompe, possiblement en toute bonne foi, car il n'avait pas, au moment de la décision, toutes les informations requises pour aller vers une bonne solution, alors il est attendu que son processus de prise de décision puisse être étudié pour comprendre pourquoi et comment l'agent en est arrivé à sa conclusion.

Le caractère incertain des environnements pose aussi un problème du point de vue de l'*autonomie*[1] de l'agent. La notion d'autonomie est assez protéiforme et peut parfois désigner, d'un point de vue philosophique, d'autres notions comme la liberté, la volonté, la responsabilité, *etc.* (Dworkin, 2015). Dworkin clarifie la différence entre *autonomie* et *liberté* en indiquant que la première fait référence à *la capacité de choisir*, quand la seconde se rapporte *aux possibles actions qu'il est possible d'entreprendre*. La sociologie peut aussi amener des éléments pertinents pour mieux appréhender la notion d'autonomie : Le Coadic (2006) caractérise l'autonomie autour de trois propriétés : « la faculté de choisir par soi-même (et d'émettre ses propres normes), la capacité d'agir sans l'intervention d'un tiers et le fait — pour un individu ou une collectivité — de disposer des ressources nécessaires à la réflexion et à l'action ».

Ainsi, pour évoluer dans un environnement incertain en toute autonomie et de manière explicable, il faut donc un modèle capable de se doter de ses propres règles et que celles-ci soient intelligibles, et qu'il soit capable de détecter et de faire

[1] Comme pour la notion d'*incertitude*, une conceptualisation largement acceptée de l'*autonomie* ne semble pas exister. On peut toutefois en donner une première définition triviale à partir de son étymologie : *auto*, « soi » et *nomos*, « loi » ou « règle », c'est-à-dire la capacité à agir selon ses propres règles. Dans la Grèce antique, l'autonomie se rapportait à des cités-états pouvant créer leurs propres lois sans ingérence extérieure, sans être sous l'autorité d'une puissance étrangère. Aujourd'hui, dans le domaine de l'IA, même si l'on trouve des semblants de définitions de la notion d'autonomie (*e.g.* la capacité d'un système à agir sans supervision humaine), mon impression est qu'il y a une certaine confusion entre *autonomie* et *automatisation*.

face, sans supervision, aux différentes formes d'incertitude pouvant survenir dans l'environnement.

2 Caractérisation de l'incertitude des environnements

Avant toute chose, il faut préciser que les environnements considérés sont *partiellement observables*, à l'image d'environnements réels. Autrement dit, les agents n'ont pas accès à l'ensemble des états (ou l'ensemble des informations) de l'environnement en tout temps (Fig. 4.1). Cet état de fait peut être dû au fait que l'agent n'a qu'une perception locale de son environnement ou encore que les capteurs de l'agent sont imprécis, voire bruités.

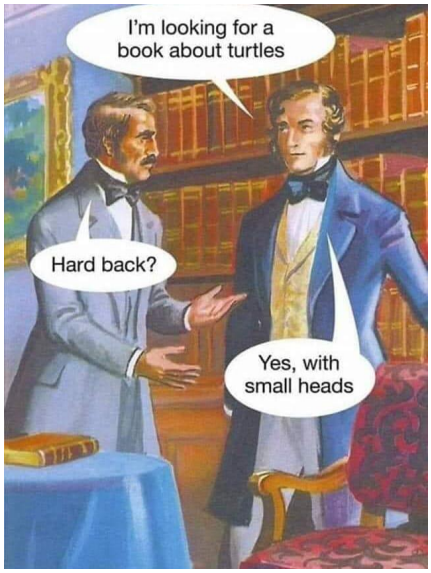


FIGURE 4.1 : Dans certains cas, l'ambiguïté est parfaite. (Source : les internets)

Trois types d'incertitude peuvent exister dans ces environnements : l'ambiguïté, la volatilité et la stochasticité (Faraut, 2015).

2.1 Ambiguïté

L'ambiguïté est une incertitude sur le choix des actions à réaliser (possiblement associées à des probabilités) pour atteindre le but voulu. Dans un contexte d'apprentissage dirigé par le but, un agent peut, par *renforcement* (Fig. 4.2), évaluer lui-même différentes alternatives et leurs pertinences pour réaliser son but. Ce paradigme favorise l'autonomie de l'agent, car il va apprendre par lui-même les effets de ses

actions sur l'environnement : il s'agit d'un conditionnement opérant (en opposition avec un apprentissage supervisé où des annotations sont fournies à l'agent pour guider ses actions : dans ce cas, l'agent est limité aux annotations existantes et disponibles et dispose d'un degré d'autonomie limité[2]). Le renforcement peut être *positif*, auquel cas l'action ayant conduit à ce renforcement va être favorisée ; si le renforcement est *négatif*, alors l'action sera évitée. Dans le paradigme de l'apprentissage par renforcement, l'agent peut fonctionner en mode *exploration*, de façon à découvrir de nouvelles règles d'interaction avec son environnement, ou bien en mode *exploitation*, auquel cas l'agent utilise des règles éprouvées pour atteindre son but.

2.2 Volatilité

La volatilité désigne le caractère potentiellement changeant des environnements. Ces changements peuvent être soudains ou progressifs ; ils peuvent être signalés à l'agent ou non (changement non déterministe). Si le changement est non déterministe, alors l'agent doit être capable de le détecter. Un nouvel apprentissage est nécessaire pour prendre en compte le nouvel état de l'environnement. Dans le cadre d'un apprentissage par renforcement, cet apprentissage peut s'ajouter à l'apprentissage déjà réalisé en découvrant de nouvelles règles d'interaction avec l'environnement grâce à l'exploration. Les règles précédemment apprises peuvent potentiellement toujours servir à l'agent dans l'atteinte de son but.

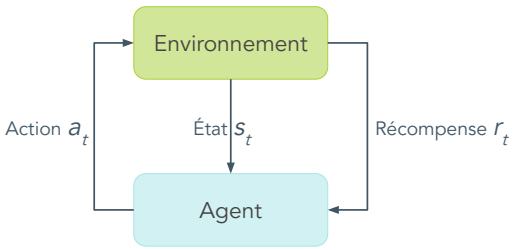


Figure 4.2 : Apprentissage par renforcement d'un agent dans son environnement. Les actions effectuées par l'agent modifient l'état de l'environnement. Une valeur de renforcement, la récompense, est transmise à l'agent pour lui permettre d'évaluer ses actions afin d'atteindre le but qui lui a été donné.

[2]En psychologie cognitive, il s'agit de la différence entre le *comportement régi par les contingences* et le *comportement régi par les règles* (Cottraux, 2020). En intelligence artificielle, l'apprentissage par renforcement d'un agent peut être contraint par un ensemble de règles pour maintenir les actions de l'agent dans un cadre éthique donné, comme nous l'avons évoqué dans le chapitre I.2.

2.3 Stochasticité

La stochasticité est liée à la notion de *risque*, c'est-à-dire la variance de distribution[3] des probabilités de récompenses. Formulé autrement, il s'agit du fait que pour un état de l'environnement et pour une action donnés, l'issue de l'action peut changer (non-déterminisme). Dans une situation non déterministe, une décision rationnelle consiste à choisir l'option qui amène à la plus grande valeur attendue (*expected value*). Un cas spécifique de stochasticité est le problème de l'aliasing perceptif (*Perceptive Aliasing Issue*, PAI), où des états de l'environnement distincts entre eux ne sont pas perçus comme tels par l'agent (Whitehead and Ballard, 1991). Une autre source de stochasticité est le bruitage des actions de l'agent (par exemple, celui-ci exécute une action de déplacement, mais ne se trouve pas à la position voulue suite à ce déplacement). Les modifications environnementales sont aussi des formes de non-déterminisme. Ces modifications sont problématiques, car beaucoup de systèmes d'apprentissage reposent sur une *causalité perceptive* (*i.e.* les changements de l'environnement sont causés par les actions de l'agent) (Butz et al., 2001). Pour faire face à ces différentes formes de non-déterminisme, un agent doit être capable de les distinguer.

2.4 Quel modèle autonome et explicable pour évoluer en environnements incertains ?

Dans les travaux présentés dans ce chapitre, le choix d'un modèle pour la conception d'une IA autonome et explicable capable d'évoluer dans des environnements incertains a été guidé par les deux contraintes suivantes :

1. Nous souhaitons éviter les approches *post-hoc* pour l'explicabilité du système. Comme nous l'avons vu dans la partie I, la pertinence de ces approches est discutable. Du fait de la pertinence de l'utilisation de règles pour transmettre des explications (van der Waa et al., 2021; Fürnkranz et al., 2020), nous nous sommes tournés vers le paradigme de l'apprentissage basé sur les règles (donc les systèmes à base de règles). Ces approches ont néanmoins des inconvénients : elles sont peu adaptées pour la caractérisation de données complexes (des images, du texte, *etc.*) ; la pertinence explicative du modèle est à relativiser

[3] Contrairement au cas de l'ambiguïté, cette distribution est ici connue.

dans le cas le nombre de règles générées est important ou que les règles générées sont longues.

2. Nous souhaitons favoriser l'autonomie du système. Pour cela, il faut se tourner vers des paradigmes d'apprentissage comme l'apprentissage par renforcement ou l'évolution artificielle.

Une classe de modèle répond à ces deux exigences : les *systèmes de classeurs*. Cette classe de modèles constitue le socle de la thèse de M. Romain Orhand. Ses contributions ont porté sur l'intégration de mécanismes de gestion de l'incertitude qui ont, de surcroît, accru l'autonomie et l'explicabilité des modèles proposés. Afin d'amener progressivement les différents concepts techniques nécessaires à la présentation des contributions de M. Orhand, nous commencerons par faire une introduction générale aux systèmes de classeurs puis nous présenterons deux systèmes de classeurs majeurs : XCS (section 3.2) et ACS2 (section 3.4).

3 Systèmes de classeurs

Les systèmes de classeurs (*Learning Classifier Systems*) (Holland, 1976) sont des systèmes à base de règles reposant sur deux mécanismes : un mécanisme de découvertes des règles — appelées des *classeurs* — permet d'explorer l'espace de recherche afin de construire un ensemble de règles qui vont mettre en correspondance des entrées du système avec des actions ; la composante d'apprentissage est utilisée pour évaluer la qualité de ces règles afin de guider le système vers la découverte de meilleures règles.

Pour apprendre un comportement optimal dans un environnement, un LCS apprend une *politique* comportementale optimale : l'ensemble de règles est évolué de façon à déterminer la meilleure action à entreprendre pour chaque état de l'environnement. Cet ensemble de règles est évolué grâce à un mécanisme d'apprentissage par renforcement combiné à un algorithme génétique.

La population de classeurs d'un LCS, notée $[P]$, représente la connaissance du système, une mémoire à long terme. Chaque classeur cl de $[P]$ contient :

— une condition C décrivant un état de l'environnement par un ensemble de L attributs ;

— une action A correspondant à l'action du classeur ;

— une prédiction de la récompense p (équivalent à une mesure de la qualité du classeur) prédisant le renforcement moyen de l'action dans la condition C (appelé *force* dans les LCS de la première heure).

La figure 4.3 illustre la condition, l'action (et l'effet) d'un classeur d'un LCS apprenant à sortir d'un labyrinthe (il existe beaucoup d'applications des LCS ; cette application a été choisie pour évaluer les systèmes proposés dans le cadre de ces travaux). L'environnement (ici, discret) est caractérisé par différentes valeurs discrètes : murs, chemin, sortie. La perception de l'agent se réalise avec un voisinage de Moore d'ordre 1. Le classeur illustré correspond à la perception réalisée par l'agent dans la position bleue. Si, dans cette situation, l'agent effectue un déplacement vers la droite et qu'il est capable de former des anticipations (que nous introduirons dans la section 3.4), alors il sait qu'il se retrouvera dans la situation illustrée en vert. Comme on le verra plus en détail dans la section suivante, la condition (et l'effet) du classeur est encodée par une chaîne binaire ou simplement discrète. Dans la condition, le symbole # représente un joker permettant de représenter n'importe quel attribut possible. Ces points sont présentés plus avant dans la suite.

3.1 Apprentissage dans les systèmes de classeur

Lors de l'apprentissage, le système constitue à chaque pas de temps t un ensemble de correspondances (*match set*) $[M]$ contenant tous les classeurs de $[P]$ dont la condition satisfait l'état courant de l'environnement perçu par l'agent. Une action est ensuite choisie par le système parmi les actions des classeurs contenus dans $[M]$. Un ensemble d'action[4], $[A]$ est alors généré. Cet ensemble contient tous les classeurs de $[M]$ contenant l'action choisie. La prédiction de la récompense p des classeurs de $[A]$ est mise à jour selon le gain obtenu par l'exécution de l'action. La population de classeurs évolue dans le temps, le plus souvent par l'utilisation d'un algorithme

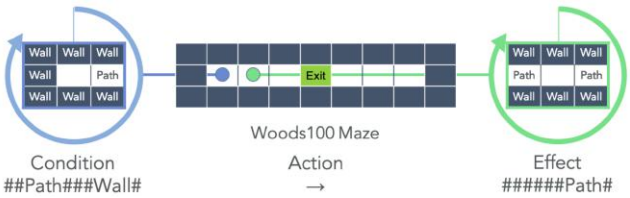


FIGURE 4.3 : Exemple de constitution d'un classeur d'un LCS dans le cas d'un apprentissage de comportement dans un environnement labyrinthique. Figure adaptée de (Orhand et al., 2022).

génétique : cette évolution permet d'aller vers une représentation la plus optimale possible de l'environnement.

Les premiers LCS ont fait face à différents problèmes (Wilson and Goldberg, 1989) : la variabilité des classeurs, due au fait que des règles différentes peuvent s'appliquer à des portions complètement différentes du problème, fait que leur recombinaison par l'approche évolutionnaire peut donner lieu à des classeurs totalement inutiles pour la résolution du problème ; la prédiction de la récompense de classeurs généralisés peut fortement osciller, du fait que ceux-ci peuvent s'appliquer à différentes situations environnementales où les gains des actions sont différents ; la distribution de la récompense est dépendante de la méthode de sélection d'actions.

Pour pallier ces problèmes, des alternatives aux premiers LCS ont été proposées, comme XCS (Wilson, 1995) ou ACS2 (*Anticipatory Classifier System 2*) (Butz and Stolzmann, 2002).

3.2 XCS

Les deux apports de XCS sont, d'une part, la mesure de la qualité des classeurs non pas sur leur force, mais sur l'exactitude de leur prédiction (*accuracy*) et, d'autre part, l'application de l'algorithme génétique sur l'ensemble de correspondances $[M]$ plutôt que sur la population de classeurs dans son entièreté.

Composition des classeurs d'XCS

XCS perçoit des situations (ou états) σ de l'environnement, ces perceptions étant encodées sous la forme d'une chaîne de longueur L sur l'alphabet $\{0, 1, \#\}$. Le « *don't care* » symbole, $\#$, est un joker correspondant à la fois à 0 et à 1. Chaque symbole de la chaîne est appelé un *attribut*. XCS réalise des actions α et rencontre dans l'environnement des récompenses (scalaires) ρ. Sa population $[P]$ contient une taille fixe de N classeurs composés des éléments suivants :

— une condition C décrivant un état de l'environnement par un ensemble de L attributs ; qui correspond à un sous-ensemble d'états de l'environnement pour lesquels le classeur est utilisable ;

— une action A, parmi les actions possibles de l'environnement $a_1, \ldots, a_n$, correspondant à l'action du classeur ;

[4] On parle bien ici d'un ensemble de *classeurs* correspondant à une unique action, d'où l'utilisation du singulier dans l'expression « ensemble d'action ».

— une prédiction de la récompense p (équivalent à une mesure de la qualité du classeur) prédisant le renforcement moyen de l'action pour C ;

— une erreur de prédiction ϵ permettant d'évaluer la *fitness* du classeur ;

— une valeur de *fitness* f correspondant à une mesure de l'exactitude de p par rapport à tous les classeurs concurrents ;

— une expérience, *exp*, indiquant à quelle fréquence les paramètres du classeur ont été mis à jour ;

— un horodatage, *ts*, enregistrant la dernière fois que le classeur a fait partie d'un ensemble de classeurs sur lequel l'algorithme génétique a été appliqué ;

— une taille d'ensemble d'actions, *as*, approximant la taille de l'ensemble d'actions $[A]$ auquel le classeur appartient ;

— une numérosité, *num*, indiquant combien de *microclasseurs* ce *macro-classeur* représente (des classeurs usuels identiques, dits *microclasseurs*, sont représentés par un *macro-classeur*).

Les paramètres *exp*, *ts*, *as* et *num* sont utilisés pour une maîtrise plus fine de l'algorithme génétique. La population initiale d'XCS peut être vide ou bien être initialisée avec des classeurs générés de manière aléatoire.

Apprentissage d'XCS

Une étape d'apprentissage d'XCS (Fig. 4.4) commence par une perception $\sigma(t)$ depuis l'environnement et la formation de l'ensemble $[M]$ à partir de $\sigma(t)$. Une phase de *recouvrement* peut avoir lieu si le nombre d'actions dans $[M]$ est inférieur à un certain seuil : ce processus consiste à générer de nouveaux classeurs de manière aléatoire avec une condition correspondante à la perception (par définition de $[M]$), mais dont les symboles peuvent être aléatoirement remplacés par # avec une probabilité $p_{\#}$, et l'une des actions manquantes. Le tableau de prédictions PA est construit à partir des actions de $[M]$ pour enregistrer les récompenses présumées pour chaque action (eq. 4.1) : le gain estimé pour chaque action a de $[M]$ est la moyenne pondérée de la *fitness* de tous les classeurs de $[M]$ dont l'action A est a.

$$PA(a) = \frac{\sum_{cl \in [M], cl.A=a} cl.p \cdot cl.f}{\sum_{cl \in [M], cl.A=a} cl.f} \tag{4.1}$$

La notation $cl.param$ dénote la référence au paramètre *param* du classeur *cl*.

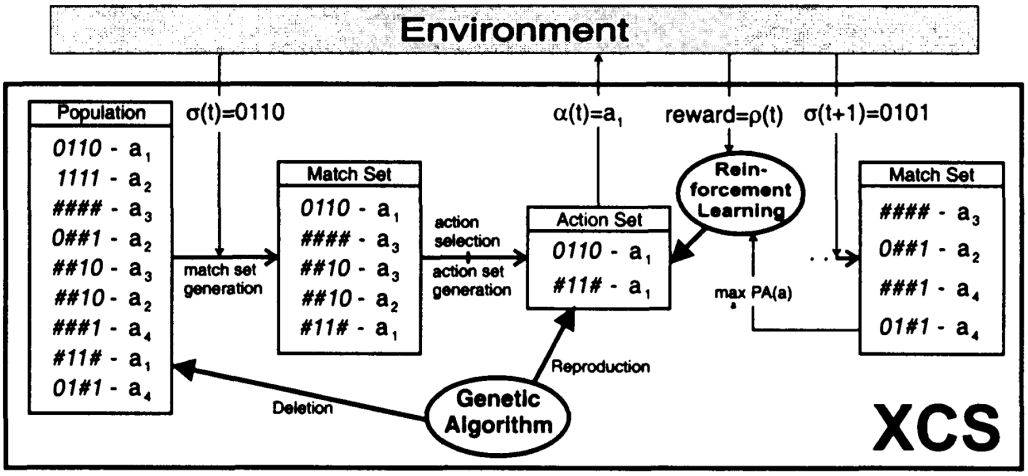


Figure 4.4 : Étape d'apprentissage d'XCS : après la perception locale de l'environnement, le système construit l'ensemble de correspondances $[M]$ puis choisit une action et construit l'ensemble d'action $[A]$. La composante d'apprentissage par renforcement permet de mettre à jour les paramètres de $[A]$; l'algorithme génétique est appliqué sur $[A]$. Figure extraite de (Butz, 2002*a*).

Renforcement Ensuite, une action a est sélectionnée : soit une action est choisie aléatoirement dans $[M]$ avec une probabilité d'exploration p_{explr}, soit la meilleure action dans PA est choisie (en exploration, p_{explr} vaut 1). L'ensemble d'action $[A]$ est constitué à partir de $[M]$, en y sélectionnant les classeurs tels que $cl.A = a$. Après exécution de a, une récompense $\rho(t)$ est reçue par le système. Les classeurs de l'ensemble d'action du cycle d'exécution précédent $[A]_{-1}$ sont mis à jour en utilisant la règle delta de Widrow-Hoff, avec un taux d'apprentissage $\beta \in [0, 1]$ (eq. 4.2, 4.3 et 4.4). La récompense ρ est la récompense obtenue au cycle d'exécution précédent.

$$cl.p = cl.p + \beta(\rho + \gamma \max_{a} PA(a) - cl.p) \tag{4.2}$$

$$cl.\epsilon = cl.\epsilon + \beta(|\rho - cl.p| - cl.\epsilon) \tag{4.3}$$

$$cl.as = cl.as + \beta(\sum_{c \in [A]} c.num - cl.as) \tag{4.4}$$

L'équation 4.2 est une adaptation du mécanisme d'apprentissage par renforcement de *Q-learning* (Watkins and Dayan, 1992) qui permet de distribuer le renforcement avec une différence temporelle : le facteur d'actualisation γ (*discount factor*) permet de choisir de favoriser les récompenses immédiates ($\gamma = 0$) ou futures ($\gamma = 1$). Mathématiquement, ce facteur d'actualisation permet de limiter la somme totale

des récompenses (la somme actualisée des récompenses s'exprimant alors comme $r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$).

Deux étapes sont nécessaires pour mettre à jour la *fitness* d'un classeur : en premier lieu, il faut évaluer l'exactitude κ du classeur en utilisant son erreur de prédiction ϵ, puis mettre à jour sa *fitness* en utilisant un rapport de son exactitude au regard de celle de l'ensemble d'action (eq. 4.5 et 4.6) :

$$\kappa(cl) = \begin{cases} 1 & \text{si } cl.\epsilon < \epsilon_0 \\ \alpha(cl.\epsilon/\epsilon_0)^{-\nu}, & \text{sinon} \end{cases} \tag{4.5}$$

$$cl.f = cl.f + \beta \left(\frac{\kappa(cl) \cdot cl.num}{\sum_{cl \in [A]} \kappa(cl) \cdot cl.num} - cl.f \right) \tag{4.6}$$

où ϵ_0 ($\epsilon_0 > 0$) permet de contrôler la tolérance à l'erreur de prédiction et α et ν permettent d'atténuer la baisse de l'exactitude du classeur quand la tolérance à l'erreur est dépassée.

Généralisation des classeurs Dans XCS, l'algorithme génétique est appliqué sur l'ensemble d'action $[A]$ (et non pas sur $[P]$ comme dans les premiers LCS) de façon à créer un phénomène de niche. L'utilisation de l'algorithme ne se produit que lorsque le temps moyen, dans $[A]$, depuis sa dernière utilisation dépasse un certain seuil θ_{GA}. En cas d'application de l'algorithme, deux classeurs sont sélectionnés dans l'ensemble $[A]$ par *roulette wheel*, une probabilité de sélection proportionnelle à la *fitness* des classeurs. Les classeurs sélectionnés sont copiés et ces copies sont mutées (chaque attribut de la condition du classeur est modifié avec une probabilité μ en l'un des deux autres symboles possibles avec une probabilité égale ; l'action du classeur est changée en une autre des actions possibles avec une probabilité μ) et croisées (un croisement à deux points est effectué avec une probabilité χ). Les parents entrent par la suite en compétition avec les enfants pour insertion dans la population, en utilisant une méthode de subsomption-suppression :

— pour un enfant donné, s'il existe un classeur cl plus général dans sa condition, dont l'expérience est plus importante ($cl.exp > \theta_{sub}$) et qui est plus exact ($cl.\epsilon < \epsilon_0$) alors l'enfant ne sera pas inséré, mais la numérosité de cl sera incrémentée : il s'agit d'un mécanisme de *subsomption* ;

— si le nombre de microclasseurs est plus grand que N, la taille de la population, des classeurs doivent être supprimés : la suppression d'un classeur cl est organisée en utilisant la méthode *roulette wheel* sur l'approximation de la taille de l'ensemble d'action auquel cl appartient, $cl.as$. La probabilité pour un classeur d'être supprimé de la population augmente si son expérience est insuffisante ($cl.exp > \theta_{del}$) et si son exactitude est significativement plus faible que la *fitness* moyenne de la population $[P]$;

— un processus de subsomption survient également sur $[A]$: le classeur cl le plus général, plus exact et expérimenté est choisi parmi les classeurs les plus exacts et expérimentés. Les classeurs de $[A]$ sélectionnés qui spécifient la condition de cl sont examinés : les plus spécifiques parmi eux sont supprimés et la numérosité de cl est modifiée en conséquence.

XCS est un système de classeur retravaillé de façon à pallier les défauts des LCS de la première heure. Sans être exempts de défauts, différents travaux ont montré le bien-fondé des modifications apportées dans XCS pour son apprentissage (Lanzi, 1999; Butz and Pelikan, 2001; Nakamura et al., 2021).

Les systèmes de classeurs à anticipation (*Anticipatory Learning Classifier Systems*, ALCS) constituent une autre classe majeure de systèmes de classeurs. Comme leur nom l'indique, ces systèmes intègrent la capacité de se représenter des anticipations. Anticiper les effets d'actions sur l'environnement joue un rôle clé dans la détermination du comportement à adopter dans une situation donnée. ACS2 (Butz and Stolzmann, 2002), un des systèmes de classeurs les plus populaires, met en œuvre un processus de découverte de règles basé sur la théorie du contrôle comportemental anticipatif (*Anticipatory Behavioral Control*, ABC) proposé par Hoffmann (2003).

3.3 Théorie du contrôle comportemental anticipatif

Tolman (1932) a théorisé, chez des animaux humains et non humains, l'existence d'un système de comportement dirigé vers le but reposant sur la capacité à inférer des relations entre les actions réalisées dans l'environnement et l'état de celui-ci (il a aussi mis en évidence l'existence d'un apprentissage latent, se manifestant *a posteriori* sous le coup d'une récompense différée). En s'intéressant à l'apprentissage latent, Seward (1949) a mis en évidence qu'un modèle interne de l'environnement et des effets des actions réalisées sur l'état de celui-ci est construit et utilisé pour faire des inférences.

La théorie du contrôle comportemental anticipatif proposé par Hoffmann (2003) est un modèle comportemental expliquant l'apprentissage des relations entre les situations S, les réponses R et leurs effets E. Le modèle ABC peut être résumé comme suit (Hoffmann, 2009) (Fig. 4.5) :

1. Une action volontaire est la réalisation d'actions de manière à atteindre un résultat ou un effet souhaité. Il est nécessaire de pouvoir *représenter* ce résultat, donc agir demande en premier lieu d'*anticiper* les résultats attendus.

2. Suite à une comparaison, si les résultats anticipés et effectifs concordent, une association se crée ou se renforce entre l'action réalisée et les résultats. Cette association peut s'affaiblir voire disparaître si actions et résultats ne concordent plus assez.

3. Les *contextes situationnels* font partie prenante des représentations actions-effets (conditionnement). Ces contextes permettent de différentier les associations actions-effets.

La théorie ABC est un modèle élégant et empiriquement soutenu pour expliquer les comportements adaptatifs (et donc autonomes) des animaux humains et non-humains. Pour le mettre en œuvre, il faut que l'agent puisse construire une représentation complète et fidèle de l'environnement dans lequel il évolue, ce que ne permettent pas les LCS classiques ni XCS.

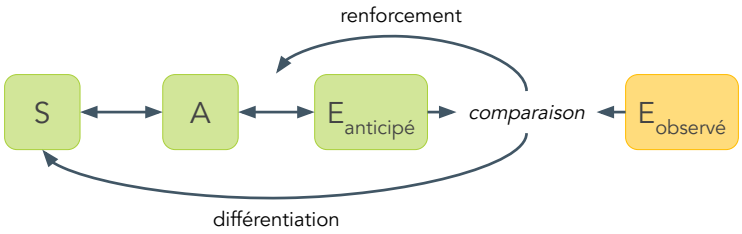


FIGURE 4.5 : Modèle du contrôle comportemental anticipatif décrivant l'apprentissage d'associations entre les actions A de l'agent et les effets anticipés $E_{\text{anticipé}}$ par l'agent. Des effets concordants avec les effets observés $E_{\text{observé}}$ vont renforcer les liens *actions* $\leftrightarrow$ *effets*. Ces relations sont différenciées grâce au contexte situationnel S. Figure adaptée de (Butz, 2002*a*).

3.4 ACS2

ACS2 (Butz and Stolzmann, 2002) est un classeur de la famille des ALCS (*Anticipatory Learning Classifier System*) qui étend ACS (*Anticipatory Classifier System*) (Stolzmann, 2000).

ACS2 perçoit des situations environnementales à chaque cycle $\sigma(t)$ encodées sous la forme d'une chaîne de longueur L sur l'alphabet $\mathcal{I} = \{\iota_1, \iota_2, \ldots, \iota_m\}$ où m est le nombre de valeurs possibles que peut prendre chaque attribut de l'environnement[5]. Nous ne sommes donc plus dans le cadre d'une représentation binaire, mais d'une représentation à valeurs discrètes. Le système peut réaliser une action $\alpha(t) \in \mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ où n est le nombre d'actions qu'il est possible de faire dans l'environnement. Suite à la réalisation d'une action, le système perçoit une récompense ρ de l'environnement.

Composition des classeurs d'ACS2

ACS2 représente sa connaissance de l'environnement avec une population de classeurs $[P]$. Chaque classeur est composé de :

— une condition C décrivant un état de l'environnement par un ensemble de L attributs ;

— une action $A \in \mathcal{A}$, parmi les actions possibles de l'environnement $\alpha_1, \ldots, \alpha_n$, correspondant à l'action du classeur ;

— un effet E qui anticipe les effets (l'état de l'environnement à $t + 1$) dont le classeur attribue la causalité à l'action A ;

— une marque M qui stocke les valeurs de chaque attribut dans toutes les situations où le classeur n'est pas parvenu à une anticipation correcte ;

— une qualité q qui est une mesure de l'exactitude des anticipations du classeur ;

— une prédiction de la récompense r prédisant la récompense attendue suite à la réalisation de l'action A sous la condition C ;

— une prédiction de la récompense immédiate ir prédisant le renforcement obtenu immédiatement après la réalisation de l'action A ;

[5] Autrement dit, l'état courant perçu est donc constitué de L attributs parmi m attributs possibles.

— deux horodatages t_{ga} et t_{alp} enregistrant la dernière fois que l'algorithme génétique ou l'ALP, respectivement, ont été appelés ;

— une expérience, *exp*, indiquant à quelle fréquence les paramètres du classeur ont été sujets à l'ALP ;

— une moyenne d'application *aav* (*application average*) estimant la fréquence d'application de l'ALP ;

— une numérosité, *num*, indiquant combien de *microclasseurs* ce *macro-classeur* représente.

Les composantes *condition* et *effet* sont constituées de valeurs perçues depuis l'environnement ainsi que du symbole joker # : $C, E \in \{\iota_1, \ldots, \iota_m, \#\}^L$. Dans la composante *effet*, le symbole # est appelé symbole « pass-through ». Il dénote le fait que le classeur anticipe que la valeur de l'attribut représenté par ce symbole ne sera pas modifiée par la réalisation de A. La marque du classeur est de la forme $M = (m_1, \ldots, m_L)$ où $m_i \subseteq \{\iota_1, \ldots, \iota_m\}$.

Les paramètres q, r et ir sont des scalaires tels que $q \in [0; 1]$ et $r, ir \in \mathbb{R}$. Si la qualité du classeur est supérieure à un seuil θ_r, alors il est dit *fiable* et est considéré comme faisant partie de la représentation environnementale construite par ACS2. Si la qualité du classeur est en deçà du seuil θ_r, il est considéré comme *non fiable* et supprimé.

L'ensemble des propriétés des classeurs d'ASC2 sont modifiées avec le mécanisme d'apprentissage par renforcement *Anticipatory Learning Process* (ALP) et par un algorithme génétique pour leur généralisation.

Apprentissage d'ACS2

La population initiale de classeurs d'ACS2 est toujours vide, les premiers classeurs étant générés par un processus de recouvrement tel que celui d'XCS (p. 77). Une étape d'apprentissage d'ACS2 (Fig. 4.6) commence par la réception d'une perception $\sigma(t)$. Un ensemble de correspondances $[M]$ est ensuite construit, une action $\alpha(t)$ est choisie en utilisant une *politique comportementale* sur $[M]$. Il s'agit le plus souvent d'une politique ϵ-*greedy* dénotée π, appliquée selon la probabilité d'exploration ϵ, où

rand est un nombre aléatoire. Cette politique s'écrit :

$$\pi = \begin{cases} cl.A, cl = \underset{cl \in [M], cl.E \neq \{\#\}^L}{argmax} cl.q \cdot cl.r & \text{si } rand < \epsilon \\ \text{action aléatoire } A \in \mathcal{A} & \text{sinon} \end{cases} \tag{4.7}$$

Ainsi, en phase d'exploitation (pour une probabilité $1 - \epsilon$), l'action choisie est telle que le produit $cl.q \cdot cl.r$ est maximal. L'ensemble d'action $[A]$ est ensuite construit à partir de $[M]$ selon l'action choisie par la politique ϵ-*greedy*. Après la réalisation de l'action choisie et la réception de la récompense ρ, ACS2 perçoit $\sigma(t+1)$, forme l'ensemble de correspondances $[M]_{t+1}$ et c'est seulement ensuite que les classeurs de $[A]$ sont modifiés par l'ALP, le processus de renforcement, et qu'une étape de généralisation avec un algorithme génétique survient. Ce décalage temporel fait que $[A]$ représente l'anticipation de la prochaine situation environnementale $\sigma(t+1)$. Une manière alternative de voir les choses est de dire que c'est sur $[A]_{t-1}$ que le processus d'apprentissage est appliqué.

Processus d'apprentissage par anticipation (ALP) L'ALP permet de comparer les anticipations de tous les classeurs de $[A]$ avec la situation environnementale réelle perçue $\sigma(t+1)$. Cette comparaison peut donner lieu à deux cas de figure :

Cas inattendu (*unexpected case*) : un classeur $cl \in [A]$ a mal anticipé la situation $\sigma(t+1)$. Deux critères permettent de le détecter : un ou plusieurs changements prédits dans $cl.E$ sont incorrects au regard de $\sigma(t+1)$ ou bien un ou plusieurs

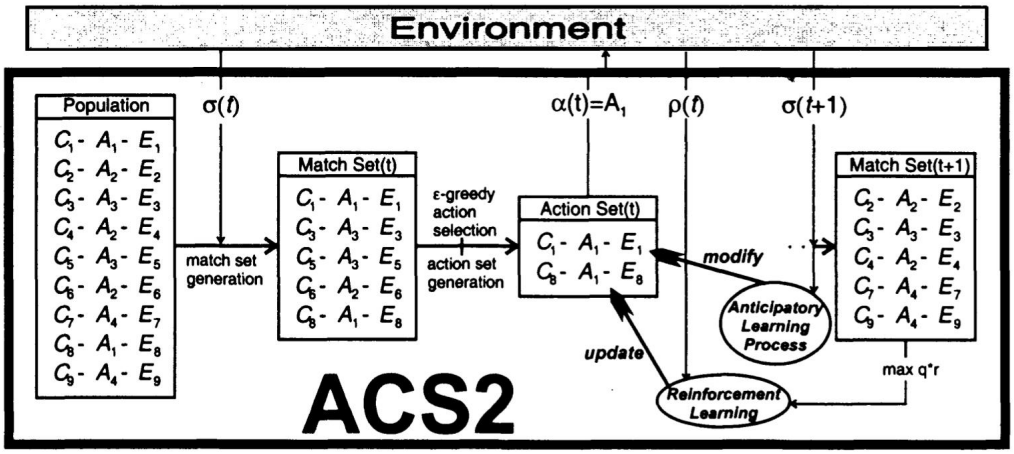


Figure 4.6 : Étape d'apprentissage d'ACS2. Figure extraite de (Butz, 2002*a*).

changements surviennent dans $\sigma(t+1)$ alors que *cl.E* prédit qu'ils restent identiques (autrement dit, des attributs de l'environnement ont bien changé mais *cl.E* prédit de mauvais changements ; ou *cl.E* prédit des changements d'attributs alors que ceux-ci restent les mêmes).
Le classeur est alors marqué par la situation $\sigma(t) = (\iota_{1,t}, \iota_{2,t}, \ldots, \iota_{L,t})$: $cl.M' = (m'_1, m'_2, \ldots, m'_L)$ où $m'_i = m_i \cup \{\iota_{i,t}\}$ et sa qualité va diminuer : $cl.q = cl.q - \beta cl.q$.
S'il est possible de construire une anticipation correcte en *spécifiant* la composante E du classeur (c'est-à-dire en remplaçant tous les symboles *passthrough* par des valeurs spécifiques de $\mathcal{I}$), alors un nouveau classeur cl_{new} est généré à partir de l'ancien avec l'effet spécifié. Les attributs de $cl_{\text{new}}.C$ sont spécifiés de la manière à ce qu'ils concordent avec $cl_{\text{new}}.E$.

Cas attendu (*expected case*) : la situation $\sigma(t+1)$ a bien été anticipée par le classeur *cl* examiné : dans *cl.E*, tous les attributs représentés par le symbole *passthrough* sont restés les mêmes dans $\sigma(t+1)$ et tous les attributs spécifiés *cl.E* ont été modifiés à la même spécification dans $\sigma(t+1)$. Dans ce cas, la qualité du classeur est augmentée : $cl.q = cl.q + \beta(1 - cl.q)$.
Si la marque de *cl* est vide ou qu'il n'y a pas de différence entre la marque et $\sigma(t+1)$, il n'y a pas de génération de nouveau classeur. Dans le cas contraire, un classeur cl_{new} est généré de la même manière que pour le cas inattendu, mais en portant une attention particulière à la spécification d'attributs pour éviter la création d'un classeur surspécialisé. Un *seuil de spécificité*, $u_{\max}$, détermine le nombre maximal d'attributs spécifiés dans la condition C anticipés comme restant identiques dans l'effet E. Si ce seuil est atteint alors il faudra que cl_{new} soit généralisé de façon à pouvoir spécifier les attributs de C dans l'effet du nouveau classeur.

Dans tous les cas, si un classeur nouvellement généré a une qualité en dessous d'un certain seuil, il est supprimé.

Les cas attendus ou inattendus permettent de mettre en œuvre la comparaison entre résultats anticipés et effectifs du modèle ABC.

Renforcement Comme pour XCS, le renforcement des classeurs de l'ensemble d'action $[A]$ d'ACS2 est produit par *Q-learning*, selon les règles suivantes :

$$r = r + \beta \left(\rho(t) + \gamma \cdot \left(\max_{cl \in [A], cl.E \neq \{\#\}^L} cl.q \cdot cl.r \right) - r \right) \tag{4.8}$$

$$ir = ir + \beta(\rho(t) - ir) \tag{4.9}$$

Généralisation Le mécanisme de l'ALP tend à créer des classeurs dont la spécialisation augmente au fil du temps. L'utilisation d'un algorithme génétique vise à introduire une pression de généralisation pour contre-balancer l'effet de l'ALP et trouver un juste milieu entre des classeurs spécialisés et généralisés.

Comme pour XCS, l'algorithme génétique est, dans ACS2, appliqué sur l'ensemble d'action $[A]$, uniquement sur la condition des classeurs et uniquement pour les généraliser : des mutations y réintroduisent des symboles *don't care*, #. Si deux classeurs sélectionnés pour croisement ont la même anticipation, alors un croisement en deux points, sur leur condition, sera réalisé.

Un processus de sélection par tournoi est utilisé pour supprimer des classeurs de l'ensemble d'action $[A]$, dont la taille est figée : les classeurs de moins bonne qualité sont les premiers à être supprimés. Si tous les classeurs sont de qualité équivalente, alors les classeurs marqués sont d'abord choisis pour suppression. Parmi les classeurs marqués, sont d'abord choisis pour suppression les classeurs les moins expérimentés.

Un processus de subsomption a également lieu dans ACS2, où les classeurs plus généraux subsument des classeurs plus spécifiques. Pour que le processus ait lieu, le classeur subsumant doit être expérimenté, fiable et ne doit pas avoir de marque.

Malgré les mécanismes d'anticipation et l'utilisation conjointe de l'ALP et de mécanismes génétiques pour obtenir un compromis entre la généralisation et la spécialisation des classeurs, ACS2 ne peut construire de représentations complètes, exactes et compactes en présence de non-déterminisme : dans de tels cas, déterminer l'action à réaliser ou construire une anticipation peut ne pas dépendre que de la perception de la situation environnementale et l'usage de la marque des classeurs peut devenir caduque.

Les ALCS ne reposent pas sur un processus stochastique pour construire leurs classeurs, mais sur la *détermination de causes et d'effets dans l'environnement*. Ainsi, ce sont des systèmes de choix dans un contexte où l'on recherche de l'interprétabilité. Il est donc intéressant d'améliorer ces modèles pour leur permettre de faire face à des situations non déterministes.

Pour qu'un système de classeur à anticipation soit capable de prendre en compte le caractère non déterministe d'un environnement, il faut que celui-ci dispose de mécanismes pour gérer l'incertitude de l'environnement. Nous présentons dans la suite BEACS (*Behavioral and Enhanced Anticipatory Classifier System*), qui couple deux approches, lui permettant de construire des représentations complètes, exactes et compactes de son environnement, même si celui-ci est incertain : des *séquences comportementales* qui ont dans un premier temps été intégrées à ACS2 pour donner BACS (section 4) et des *prédictions améliorées par des probabilités* qui ont dans un premier temps été intégrées à ACS2 pour donner PEPACS (section 5).

4 BACS : ACS2 augmenté de séquences comportementales

4.1 Séquences comportementales

Comme nous l'avons vu dans la section 2, le caractère partiellement observable des environnements peut donner lieu à des incertitudes environnementales qu'il faut pouvoir prendre en compte. L'une de ces incertitudes est le *problème de l'aliasing perceptif*, ou PAI. Il s'agit d'une configuration dans laquelle l'agent n'est pas capable de distinguer deux situations environnementales pourtant différentes et nécessitant donc des actions différentes.

Les séquences comportementales (*Behavioral Sequences*, BSeq) proposées par Stolzmann and Butz (2000) permettent de gérer le problème du PAI par l'utilisation d'une *séquence d'actions* : dans le cas de l'anticipation d'un état aliasé, un agent capable de construire des séquences comportementales va essayer de construire une séquence d'actions qui lui permettra d'éviter l'état anticipé pour se retrouver, à l'issue de la réalisation de la séquence d'actions, dans un état non aliasé.

Un système de classeur peut utiliser des séquences comportementales dans ses classeurs, auquel cas un classeur mettant en œuvre ce mécanisme aura une composante A constituée de plusieurs actions ordonnées et successives. Ainsi, pour un classeur cl, nous avons : $cl.A = (\alpha_i)_{1<i\leq bsl}$ où bsl est la longueur de la séquence, au plus $\text{BSeq}_{\max}$.

Seul ACS a fait l'objet d'une mise en œuvre de classeurs à séquence (Métivier and Lattaud, 2003). Dans ses travaux, Romain Orhand a proposé une intégration des séquences comportementales dans ACS2, en se basant sur le travail déjà effectué

par Métivier and Lattaud, résultant en un modèle dénommé BACS (*Behavioral Anticipatory Classifier System*) (Orhand et al., 2020*b*).

4.2 Intégration de séquences comportementales dans ACS2

Politique de sélection d'actions de BACS Comme les classeurs peuvent contenir une *séquence d'actions*, il faut adapter l'étape de sélection d'action (une unique action) qui survient après la perception $\sigma(t)$ et la construction de l'ensemble de correspondances $[M]$. Pour pouvoir explorer ou exploiter l'environnement avec une séquence d'actions, qui seront exécutées d'une traite avant de percevoir la récompense, c'est maintenant un classeur qui sera choisi aléatoirement dans $[M]$ en lieu et place de l'action de l'un des classeurs de $[M]$.

La politique de sélection utilise les deux biais introduits dans ACS2 par Butz (2002*b*) : le biais du délai d'action (*action delay bias*) et le biais de connaissances (*knowledge array bias*). Ces biais ont été introduits pour optimiser l'apprentissage de la représentation de l'environnement.

— le biais du délai d'action est utilisé pour sélectionner l'action la plus anciennement exécutée dans la situation $\sigma(t)$. Pour cela, il suffit d'utiliser l'horodatage t_{alp} des classeurs de $[M]$;

— le biais de connaissance utilise l'erreur d'anticipation (exprimée comme la qualité q des classeurs) pour déterminer quel classeur dans $[M]$ met en œuvre l'action ou la séquence d'actions dont les effets sont les moins connus. L'ensemble de la connaissance construite par les classeurs de $[M]$ est représenté dans un tableau KA, similaire au tableau de prédictions PA d'XCS. Le tableau KA associe à chaque classeur $cl \in [M]$ la connaissance disponible sur les effets d'une action :

$$KA[cl] = \frac{\sum_{c \in [M], c=cl} c.q \cdot c.\text{num}}{\sum_{c \in [M], c=cl} c.\text{num}} \tag{4.10}$$

Ce tableau permet de connaître la qualité moyenne de l'anticipation de chaque classeur. Le classeur associé à la valeur la plus basse est choisi.

La politique de sélection présente donc trois cas de figure selon le biais d'exploration choisi :

— le classeur cl tel que $cl.t_{\text{alp}}$ est minimal parmi les classeurs de $[M]$ est choisi (biais du délai d'action) ;

— le classeur cl dont la qualité moyenne est la plus faible est choisi (biais de connaissance) ;

— un classeur aléatoire dans $[M]$.

Dans le cas de l'exploitation, le classeur présentant la meilleure *fitness* est choisi et l'ensemble d'action $[A]$ peut être construit.

Séquences comportementales et ALP Le principe de base de l'ALP dans BACS est le même que celui d'ACS2. Il faut cependant prendre en compte les cas où un classeur à séquence, que nous dénotons cl_{BSeq}, est examiné par l'ALP.

Pour chaque classeur à séquence de $[A]$, une comparaison de la perception de l'agent entre $\sigma(t)$ et $\sigma(t+1)$ est effectuée, de même qu'une comparaison, pour chaque classeur, de son anticipation de l'état $\sigma(t+1)$ et de $\sigma(t)$. De là, trois cas de figure sont possibles :

— Cas inattendu : cl_{BSeq} n'a pas correctement anticipé σ_t, la procédure est la même que dans le cas d'ACS2 (le classeur est marqué et sa qualité diminuée, une tentative de création d'un nouveau classeur dont la composante E est spécifiée, est réalisée).

— Cas attendu : cl_{BSeq} a correctement anticipé σ_t. Là aussi le procédé est le même que pour ACS2 (la qualité du classeur est augmentée, mais un nouveau classeur n'est créé que dans le cas où cl_{BSeq} a une marque non vide ou si sa marque est différente de $\sigma(t)$).

— Cas inutile : les classeurs à séquence étant générés pour faire face aux situations environnementales aliasées, il est attendu que l'environnement change entre t et $t+1$. Ainsi, si l'état $\sigma(t)$ est le même que $\sigma(t+1)$, la qualité du classeur est diminuée.

Si la procédure générale de l'ALP est la même que dans le cas d'ACS2, la construction de nouveaux classeurs est, elle, différente dans le cas de classeurs à séquence.

Généralisation génétique des classeurs Les classeurs à séquences ne sont pas généralisés, car ils sont spécifiquement créés pour gérer les états aliasés de l'environnement. Les classeurs classiques sont généralisés de la même manière qu'ACS2.

Constructions de classeurs à séquence Le besoin de créer des classeurs à séquence lors du processus d'apprentissage est contingent à la présence d'états aliasés. Il faut donc en premier lieu être capable de détecter ces états. Cette détection se produit dans le *cas attendu* de l'ALP. Comme nous l'avons indiqué dans la description de l'ALP dans ACS2, la génération d'un nouveau classeur n'a pas lieu si un classeur dans le cas attendu est marqué, mais que sa marque est identique à la perception courante. Cela signifie que le classeur est parfois capable d'anticiper cet état correctement (puisqu'il a atteint le cas attendu) et parfois non (sa marque correspond à l'état courant). Ce cas de figure n'est possible que si l'état de l'environnement est non-déterministe et plus particulièrement si c'est un état aliasé.

Lorsqu'un classeur cl_t atteint le cas attendu de l'ALP dans l'état aliasé $\sigma(t)$, ce classeur est utilisé conjointement au classeur actif à $t-1$ ayant conduit le système dans cet état. Soit cl_{new} un nouveau classeur à générer. Ses composantes sont telles que :

— $cl_{\text{new}}.C = passthrough(cl.C, cl_{t-1}.C)$

— $cl_{\text{new}}.E = passthrough(cl_{t-1}.C, cl.C)$

— $cl_{\text{new}}.A = cl_{t-1}.A\,cl_t.A$

où la fonction *passthrough* substitue les symboles # de l'anticipation par les attributs correspondants de $\sigma(t)$ (Stolzmann and Butz, 2000) (pour rappel, le symbole *passthrough* dans l'anticipation E d'un classeur signifie que le classeur n'a pas anticipé de changement de l'attribut correspondant dans $\sigma(t+1)$) ; la séquence d'actions de cl_{new} est issue du chaînage des actions A de $cl_{t-1}.A$ et $cl_t.A$.

Si ni cl_{t-1} ni cl_t n'anticipent de changement ou si la séquence d'actions résultante du chaînage est de longueur supérieure à $\text{BSeq}_{\max}$, alors la création du classeur n'a pas lieu.

Pour éviter au système de construire des boucles dans des séquences d'actions (par exemple avancer d'un pas ; reculer d'un pas ; avancer d'un pas ; *etc.*), Métivier and Lattaud proposent de garder trace des états rencontrés lors de la réalisation d'une séquence comportementale. Il est alors possible de détecter les états déjà rencontrés et, le cas échéant, de diminuer la qualité du classeur à séquence ayant conduit à cette boucle. Cela permet ainsi de favoriser les classeurs à séquences ayant le plus petit nombre d'actions possible pour gérer les états aliasés.

4.3 Protocole d'évaluation de BACS et résultats

Environnements

BACS a été mis à l'épreuve dans des environnements labyrinthiques, classiquement utilisés comme bancs de tests pour les systèmes de classeurs. Ces labyrinthes, glanés dans la littérature pour constituer un banc de test, sont des environnements discrets en deux dimensions et composés de cellules dans lesquelles l'agent évolue. Ces cellules contiennent les différents éléments de l'environnement, comme les murs ou les chemins, mais aussi des obstacles, d'autres agents, *etc.*[6]. Le but de l'agent dans un labyrinthe est d'atteindre sa sortie ; pour cela, il va pouvoir réaliser des actions de déplacement. Un tel environnement peut être considéré comme un graphe de transitions dont les nœuds sont les cellules et les arcs, les actions de l'agent. Celui-ci va donc apprendre les transitions lui permettant d'atteindre son but.

La figure 4.7 est un exemple de labyrinthe. Cet environnement, appelé *Woods100*, contient deux états aliasés identifiables par les chiffres 1 et 2 sur chacun de ces deux états : percevant son environnement par un voisinage de Moore d'ordre 1, l'agent dans ces situations environnementales n'est, de prime abord, pas capable de distinguer ces deux états lorsqu'il fait un déplacement vers la droite (l'agent peut se déplacer sur les 8 cases adjacentes à sa position, si celles-ci sont accessibles — *i.e.* ni des murs, ni des obstacles).

Trois éléments principaux permettent de caractériser les environnements labyrinthiques : la distance moyenne à la sortie (le nombre moyen d'actions à entreprendre depuis n'importe quelle position de l'environnement pour atteindre la sortie), le type d'aliasing (Bagnall and Zatuchna, 2005) présent dans l'environnement et la complexité des labyrinthes (une quantification de la difficulté, pour un agent, à

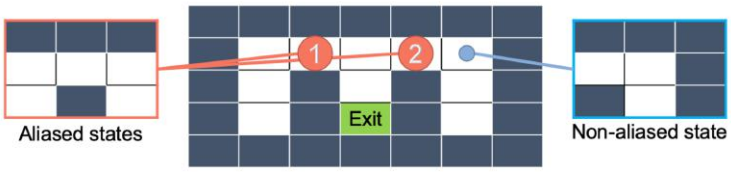


FIGURE 4.7 : Exemple de labyrinthe : *Woods100*. Ce labyrinthe comprend deux états aliasés identifiés par les chiffres 1 et 2. Figure extraite de Orhand et al. (2020*a*).

[6]Dans le cas des travaux présentés ici, les seuls obstacles rencontrés dans ces environnements sont des murs.

apprendre une représentation de l'environnement). En tout, 23 labyrinthes ont été sélectionnés, de différentes complexités (Fig. 4.8).

Protocole expérimental

Pour chacun des labyrinthes du banc de test, 30 expériences sont réalisées. Une expérience est un ensemble d'essais successifs, eux-mêmes étant des tentatives d'atteindre l'objectif (ici, la sortie du labyrinthe) en un nombre minimal d'actions. Considérant que l'agent doit atteindre la sortie en un nombre d'actions aussi petit que possible, le nombre total d'actions autorisées avant d'arrêter l'exécution de l'agent a été fixé à 100. L'agent démarre d'une position aléatoire dans l'environnement qui n'est pas un mur ou la sortie.

Chaque expérience est composée des étapes suivantes :

1. Durant les 1000 premiers essais, BACS construit sa représentation de l'environnement en utilisant ses mécanismes d'apprentissage. C'est donc l'exploration qui prime : $\epsilon = 0.8$ pour la politique de choix d'action ϵ-*greedy* et le taux d'apprentissage de l'ALP et de l'apprentissage par renforcement est fixé à $\beta = 0.05$.

Maze	Distance to exit	Complexity	Aliasing type
MazeE2 [2]	2.73	251.2	III
Woods101.5 [31]	3.1	251	III
Maze10 [31]	5.11	171	III
MazeE1 [21]	3.07	167	III
Woods102 [31]	3.31	167	III
Woods100 [21]	2.33	166	III
Woods101 [2]	2.9	149	III
Maze7 [31]	4.33	82	II
MazeF4 [26]	4.5	47	II
MiyazakiB [22]	3.33	1.03	II
Littman57 [2]	3.71	154	I
MiyazakiA [22]	3.05	69	I
Littman89 [20]	3.77	61	I
Cassandra4x4 [2]	2.27	1	I
MazeB [1]	3.5	1.26	I
MazeD [1]	2.75	1.03	I
Maze4 [8]	3.5	*	Not aliased
Maze5 [4]	4.61	*	Not aliased
MazeA [1]	4.23	*	Not aliased
MazeF1 [26]	1.8	*	Not aliased
MazeF2 [26]	2.5	*	Not aliased
MazeF3 [26]	3.38	*	Not aliased
Woods14 [9]	9.5	*	Not aliased

Figure 4.8 : Caractéristiques des 23 labyrinthes (présentés par ordre décroissant de complexité) utilisés pour l'évaluation de BACS. Figure extraite de (Orhand et al., 2020*b*).

2. Durant les 500 essais suivants, BACS tente d'atteindre la sortie en un nombre minimal d'actions. Dans ce cas, les mécanismes d'apprentissage ne sont plus utilisés, il s'agit uniquement d'exploitation ; $\epsilon = \beta = 0$. Le but de ces 500 essais est de permettre l'analyse quantitative et qualitative des capacités de l'utilisation des séquences d'actions de BACS.

3. Pour 100 essais supplémentaires, l'apprentissage par renforcement est rendu opérant à nouveau de manière à calculer les récompenses attendues pour les classeurs capables d'anticiper correctement des changements de l'environnement ; $\beta = 0.05$.

4. Suite à cette phase d'apprentissage, 500 essais sont réalisés pour enregistrer le nombre moyen d'étapes que nécessite BACS pour atteindre la sortie de l'environnement. Cette phase est réalisée avec ou sans apprentissage par renforcement, de manière à mieux comprendre les apports des séquences comportementales.

Différentes tailles de séquences comportementales ont été utilisées (1, 2 et 3 ; BACS avec $\text{BSeq}_{\text{max}} = 1$ est équivalent à un ACS2). En guise d'expérience témoin, les mêmes tests ont été réalisés avec ACS2.

Les métriques d'intérêt pour évaluer ces systèmes sont le nombre moyen d'actions nécessaires pour atteindre la sortie et le taux de connaissance du système, permettant d'indiquer s'il parvient à construire une représentation complète de son environnement. Ce taux de connaissance s'exprime comme le ratio de transitions correctes apprises par au moins un classeur fiable, sur toutes les transitions possibles.

Résultats

Le détail de l'évaluation de BACS peut être consulté dans (Orhand et al., 2020*b*) et (Orhand et al., 2020*a*). Les éléments à retenir sont les suivants :

— Pour 13 des 23 labyrinthes, BACS est 11 fois plus efficace pour attendre la sortie (en nombre d'actions réalisées) qu'ACS2. Pour les 8 autres labyrinthes, BACS fait aussi bien qu'ACS2 (Fig. 4.9).

— BACS parvient systématiquement à atteindre la sortie en moins de 100 actions, ce qui n'est pas le cas d'ACS2 qui n'y parvient que dans 15 des 23 environnements.

— Dans les environnements non aliasés du banc de test, BACS-1 (BACS pour $\text{BSeq}_{\max} = 1$) se comporte comme un ACS2 : les deux systèmes sont capables de construire des représentations complètes de leur environnement (ratio de connaissance de 100%).

— Dans les environnements aliasés, les séquences comportementales mises en œuvre dans BACS lui permettent de faire face au problème d'aliasing perceptif dans la majorité des labyrinthes (14 sur 16).

— Les séquences comportementales plus longues détériorent les performances de BACS, demandant à celui-ci davantage d'étapes avant de pouvoir atteindre la sortie.

— La création de classeurs à séquence tend à augmenter le nombre de classeurs dans la population de BACS. En particulier, plus l'environnement sera aliasé, plus la population de classeurs va croître, et ce d'autant plus si la longueur maximale autorisée pour les séquences est important.

Malgré le fait que la population de classeurs tend à croître lorsque les séquences comportementales sont utilisées, ces résultats sont positifs et valident l'utilisation de ces séquences comportementales pour faire face au problème de l'aliasing perceptif.

Les problèmes de non-déterminisme peuvent aussi être traités en utilisant des prédictions améliorées par des probabilités (*Probability Enhanced Predictions*, PEP). PEPACS (Orhand et al., 2020*c*) est le premier ACS2 intégrant des PEP afin de faire face au non-déterminisme environnemental.

5 PEPACS : ACS2 augmenté par des prédictions améliorées

Les prédictions améliorées par les probabilités (*Probability Enhanced Predictions*, PEP) ont été introduites par Butz et al. (2001) dans ACS, initialement pour gérer trois formes de non-déterminisme : bruit dans les entrées du système, résultant en des attributs aléatoires ; incertitude du résultat des actions ; la combinaison des deux. Malgré les capacités des PEP non seulement à gérer différentes formes de non-déterminisme, mais, en plus, à décrire explicitement le comportement du système dans l'environnement, ce mécanisme n'avait auparavant jamais été intégré à ACS2.

Maze	ALCS	Distance to exit	Without RL μ	Without RL σ	With RL μ	With RL σ
MazeE2	ACS2	2.73	58.883	**3.039**	36.507	3.795
	BACS-2		**19.801**	16.369	**5.178**	**0.890**
Woods101.5	ACS2	3.1	68.099	**2.926**	47.560	2.766
	BACS-1		**18.596**	5.199	**4.355**	**0.820**
Maze10	ACS2	5.11	75.328	**6.730**	64.817	4.198
	BACS-2		**30.087**	24.115	**7.733**	**1.705**
MazeE1	ACS2	3.07	37.846	6.133	4.472	0.967
	BACS-1		**6.871**	**3.959**	**3.375**	**0.172**
Woods102	ACS2	3.31	60.999	9.221	26.115	2.151
	BACS-1		**10.123**	**5.066**	**4.256**	**0.311**
Woods100	ACS2	2.33	33.933	1.738	11.876	1.276
	BACS-1		**2.337**	**0.044**	**2.336**	**0.045**
Woods101	ACS2	2.9	42.180	2.057	16.433	1.471
	BACS-1		**3.034**	**0.084**	**3.023**	**0.078**
Maze7	ACS2	4.33	50.934	6.115	45.603	7.138
	BACS-1		**4.326**	**0.091**	**4.328**	**0.105**
MazeF4	ACS2	4.5	52.095	5.804	44.642	9.636
	BACS-1		**4.472**	**0.100**	**4.490**	**0.126**
MiyazakiB	ACS2	3.33	5.159	5.583	**3.670**	**0.104**
	BACS-1		**3.988**	**0.261**	3.983	0.275
Littman57	ACS2	3.71	37.166	27.421	**4.319**	1.165
	BACS-2		**6.023**	**9.062**	4.517	**0.312**
MiyazakiA	ACS2	3.05	9.074	12.079	3.781	**0.205**
	BACS-1		**3.756**	**0.856**	**3.479**	0.226
Littman89	ACS2	3.77	34.542	27.867	5.361	0.782
	BACS-2		**4.458**	**0.295**	**4.460**	**0.265**
Cassandra4x4	ACS2	2.27	2.953	0.423	2.915	0.415
	BACS-1		**2.872**	**0.220**	**2.862**	**0.229**
MazeB	ACS2	3.5	10.676	13.965	4.507	0.425
	BACS-2		**5.071**	**2.689**	**4.058**	**0.182**
MazeD	ACS2	2.75	**2.775**	0.058	2.784	0.061
	BACS-0		2.791	**0.050**	**2.767**	**0.048**
Maze4	ACS2	3.5	3.510	0.062	3.503	**0.053**
	BACS-0		**3.499**	**0.057**	**3.497**	0.057
Maze5	ACS2	4.61	**4.623**	**0.088**	**4.626**	**0.089**
	BACS-0		4.669	0.151	4.640	0.092
MazeA	ACS2	4.23	4.242	0.086	**4.230**	0.088
	BACS-0		**4.242**	**0.074**	4.239	**0.063**
MazeF1	ACS2	1.8	**1.792**	0.034	**1.798**	**0.031**
	BACS-0		1.798	**0.023**	1.811	0.038
MazeF2	ACS2	2.5	**2.493**	**0.037**	**2.502**	0.053
	BACS-0		2.501	0.044	2.518	**0.041**
MazeF3	ACS2	3.38	3.384	**0.062**	**3.388**	**0.056**
	BACS-0		**3.376**	0.075	3.391	0.068
Woods14	ACS2	9.5	9.544	0.225	9.520	0.266
	BACS-0		**9.504**	**0.208**	**9.509**	**0.240**

FIGURE 4.9 : Nombre moyen d'actions nécessaires pour atteindre la sortie μ. L'écart-type σ permet d'évaluer la capacité d'un système à atteindre la sortie depuis n'importe quelle position de départ dans l'environnement. Figure extraite de (Orhand et al., 2020*b*).

L'idée des PEP est de permettre la description de l'ensemble des états de l'environnement atteignables depuis un état aliasé donné et pour une action donnée (nous pouvons alors dès à présent anticiper que la création de classeurs améliorés s'opèrera donc sur $[A]$, puisque l'on parle bien de classeurs réalisant la même action). Pour résumer, lors de la détection d'un état aliasé, il est possible de créer des classeurs dont les attributs de son effet E sont représentés par une PEP au lieu d'un unique symbole. Par exemple, si l'on considère la position aliasée 1 de la figure 4.7 et l'action « aller à droite », alors la E sera de la forme `##{Sortie:50%,Chemin:50%}#####` où le 3$^{\text{ème}}$ attribut est représenté par la PEP `{Sortie:50%,Chemin:50%}` plutôt que par un symbole unique.

Comme pour l'inclusion des séquences comportementales, il est nécessaire, pour introduire les PEP dans ACS2, d'être capable de détecter les états aliasés (l'approche est exactement la même que celle décrite pour BACS, ci-dessus), de construire des classeurs dont l'effet peut être constitué de PEP ainsi que d'ajuster l'ALP et le mécanisme de généralisation.

5.1 Construction de classeurs améliorés

Un classeur amélioré (par les probabilités des prédictions) est un classeur dont les attributs de l'effet E qui ne sont pas des symboles *passthrough*[7] ont été remplacés par une PEP.

Comme nous l'avons rapidement vu ci-dessus, une PEP est un tableau associatif dont la clé est un symbole dans $\mathcal{I}$, associée à la probabilité d'anticiper ce symbole dans l'attribut correspondant de l'environnement.

Pour construire un classeur amélioré, un nouvel effet doit être calculé : pour cela, deux classeurs dont l'anticipation correspond à un état aliasé donné sont sélectionnés. Notons en premier lieu que dans ce cadre, les classeurs disposent d'un nouveau paramètre : un booléen *ee* (*enhanced effect*, effet amélioré), qui par défaut est positionné à faux[8]. S'il existe au moins deux classeurs dans $[A]$ dont le paramètre *ee* est positionné à vrai et que leurs marques pour l'état aliasé traité sont identiques, alors ces classeurs sont des candidats pour la création de classeurs améliorés.

Pour chaque candidat *cl* de cette liste, un classeur amélioré est créé en utilisant l'effet de *cl* conjointement à l'effet d'un second classeur, choisi aléatoirement dans la liste (et différent de *cl*). De là, il est possible de créer un classeur amélioré cl_{new} :

— La condition de cl_{new} est composée de la condition du premier candidat spécifiée par la condition du second. De cette façon, un attribut de la condition de cl_{new} est un symbole *don't care* seulement si, dans les conditions des deux candidats, l'attribut correspondant est lui-même un symbole *don't care*.

— L'action de cl_{new} est identique à celle des candidats (puisque ceux-ci sont issus de $[A]$ et ont donc la même action).

[7] Pour rappel, un attribut de l'effet représenté par le symbole *passthrough* indique qu'aucun changement environnemental n'a été anticipé pour cet attribut.

[8] Ce paramètre permet d'éviter une sur-généralisation du classeur. Il est positionné à vrai lorsqu'un classeur est dans le cas attendu, qu'il est marqué et que C ne peut être distingué de M.

— L'effet de cl_{new} est construit par fusion des effets des deux candidats (Butz et al., 2001) : chaque attribut de l'effet de cl_{new} est la combinaison des attributs correspondants des deux candidats (dans le cas où un candidat contient un symbole *passthrough* dans son effet, la perception est directement utilisée pour déterminer quel symbole doit être utilisé). Les probabilités associées à chaque symbole de la PEP sont accumulées et normalisées.

— La récompense de cl_{new} vaut la moyenne des récompenses des deux candidats sélectionnés ; sa qualité est égale au maximum de la qualité moyenne des deux candidats.

5.2 Modification de l'ALP et de la généralisation génétique

L'ajustement principal concerne la mise à jour des probabilités enregistrées dans les PEP : lorsqu'un classeur amélioré atteint l'état attendu de l'ALP pour un état aliasé donné, ces probabilités p sont mises à jour : $p = p + p_b \cdot (1 - p)$, où b_p est un taux de mise à jour. Les probabilités sont ensuite normalisées.

La généralisation peut favoriser la construction de classeurs surgénéralisés. Pour éviter cela, lorsqu'une copie d'un classeur amélioré se produit, l'ensemble des PEP de l'effet du classeur amélioré sont remplacées par le symbole de l'attribut correspondant dans la perception courante. Le paramètre *ee* de ce nouveau classeur est positionné à faux.

Le mécanisme de subsomption doit aussi être adapté : pour déterminer si un classeur peut en subsumer un autre, il faut en premier lieu déterminer si des portions de l'effet de l'un sont incluses dans l'autre. Soit E_{new} et E_{sub} les effets respectifs d'un nouveau classeur cl_{new} et d'un classeur potentiel cl_{sub} pour subsumer le nouveau classeur ; e_{new} et e_{sub} sont, respectivement, un attribut de cl_{new} et cl_{sub}. E_{sub} subsume E_{new} si, pour chaque attribut e,

— si e_{sub} et e_{new} sont des PEP, alors e_{sub} doit contenir tous les symboles de e_{new} ;

— Si seul e_{sub} est une PEP, alors e_{sub} doit contenir le symbole décrivant l'attribut e_{new} ;

— Si seul e_{new} est une PEP, la subsomption par cl_{sub} n'est pas possible ;

— dans tous les autres cas, les symboles décrits par e_{new} et e_{sub} doivent être identiques.

5.3 Évaluation de PEPACS

Le protocole d'évaluation (environnements et conditions des expériences) de PEPACS est le même que celui de BACS, présenté en section 4.3.

Le détail des résultats de l'évaluation du système est consultable dans (Orhand et al., 2020*c*). De manière synthétique, nous relevons les points suivants :

— PEPACS est capable de construire une représentation complète de son environnement (taux de connaissance de 100%), avec ou sans les mécanismes de généralisation génétique.

— L'utilisation de la généralisation génétique tend à réduire de manière significative le nombre de classeurs fiables du système, ce qui signifie qu'une représentation compacte de l'environnement peut être obtenue par ce mécanisme de généralisation.

— L'utilisation des biais d'explorations (introduits dans la section précédente lors de la présentation de BACS) permet au système de créer plus rapidement sa représentation de l'environnement, que la généralisation génétique soit active ou non.

— L'usage des PEP et l'adaptation des mécanismes d'apprentissage permettent d'éviter la construction de classeurs surgénéraux.

PEPACS est donc capable de construire une représentation complète et exacte de son environnement, même lorsque celui-ci contient des états non déterministes. L'utilisation des PEP permet en outre une meilleure **interprétabilité intrinsèque** du système, permettant à un utilisateur de connaître l'ensemble des transitions possibles depuis un état non déterministe.

Les probabilités de transitions calculées dans les PEP ne sont pas utilisées pour guider l'apprentissage du modèle, mais pour construire des représentations environnementales complètes et exactes d'environnements possiblement non déterministes. Les séquences comportementales ne permettent pas la construction d'une telle représentation dans le cas où des états de l'environnement soient sujets à l'aliasing perceptif : les actions chaînées étant réalisées d'un bloc, ni la condition ni l'effet d'un classeur à séquence ne permet de connaître le ou les états intermédiaires atteints pendant la réalisation de la chaîne d'actions.

Le modèle BEACS (*Behavioral and Enhanced Anticipatory Classifier System*) a donc été proposé. Utilisant conjointement les séquences comportementales et des prédictions améliorées par des probabilités, le but de ce modèle est de généraliser les capacités de chaque approche prise séparément, et ce dans des conditions environnementales pouvant présenter différentes formes de non-déterminisme, tout en améliorant l'interprétabilité du système.

6 BEACS : combiner les séquences comportementales et les PEP

BEACS (*Behavioral and Enhanced Anticipatory Classifier System*) est le premier système de classeurs proposant un couplage des séquences comportementales avec des prédictions améliorées. Pour un tel couplage, l'adaptation de certains mécanismes a été nécessaire, en particulier en ce qui concerne les PEP, devenues des prédictions améliorées par l'expérience (*Enhanced Predictions through Experience*, EPE).

6.1 Des PEP améliorées : *Enhanced Predictions through Experience* (EPE)

Les EPE ont le même rôle que les PEP, à savoir permettre la création d'une représentation de l'environnement complète et exacte. Elles vont donc, comme les PEP, calculer, pour un état non déterministe et une action donnés, l'ensemble des transitions possible depuis cet état. Ce sont les *représentations* utilisées qui diffèrent des PEP aux EPE : alors que les PEP substituent à un attribut de l'anticipation une association attributs/probabilités, les EPE associent aux *perceptions anticipées* le nombre de fois où elles ont été anticipées ; la figure 4.10 illustre la différence entre un classeur à PEP et un classeur à EPE.

Tous les classeurs de BEACS sont des classeurs à EPE qui décrivent donc explicitement chaque état anticipé par le classeur et nous informent sur le nombre de fois que chaque état anticipé l'a été. Du point de vue de l'apprentissage, les EPE ne nécessitent pas la détermination d'un taux d'apprentissage particulier (là où, comme nous l'avons vu dans la section précédente, la mise à jour des probabilités des prédictions nécessite un paramètre b_p qui peut être difficile à déterminer (Orhand, Jeannin-Girardon, Parrend and Collet, 2021)). Le second apport essentiel des EPE est que l'on peut accéder à chaque perception anticipée par le classeur. Cela permet

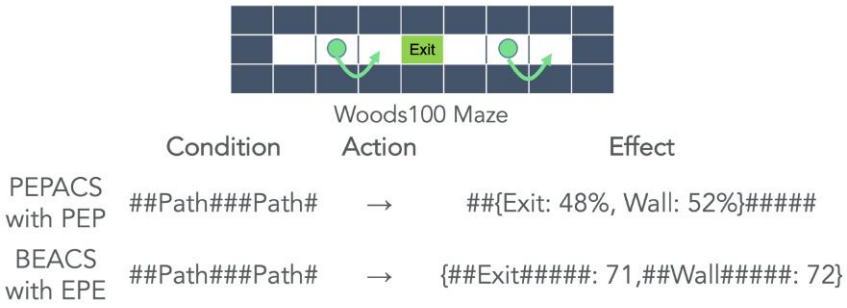


Figure 4.10 : Différence de représentation de l'effet des classeurs entre PEPACS, les attributs spécifiés dans l'effet peuvent être représentés par une PEP, et BEACS, où la perception anticipée tout entière est représentée par une EPE. Figure extraite de (Orhand et al., 2022).

en particulier de déterminer la forme de non-déterminisme des états aliasés (aliasing perceptif, qui a vocation à être traité par des séquences comportementales, ou autre forme de non-déterminisme).

Avant toute chose, il faut noter que tous les classeurs de BEACS sont dotés d'un paramètre *uM*, une *marque incertaine* décrivant une situation non déterministe et qui est utilisée pour guider l'évolution des classeurs à EPE se trouvant dans la situation décrite par *uM* : un classeur à EPE créé correspond toujours à la situation non déterministe des classeurs dont il est issu, cette situation sera indiquée dans sa marque incertaine. Les classeurs dont la création n'a pas été déclenchée par une perception aliasée ont une marque incertaine vide.

La détection d'états aliasés est à la source de la création de classeurs à EPE. De tels classeurs sont construits de manière analogue à la construction des classeurs à PEP dans PEPACS, si ce n'est pour deux points :

— la marque incertaine *uM* contient l'état aliasé qui a déclenché sa création ;

— l'effet est issu de la fusion de l'effet des deux classeurs parents et les nombres d'occurrences de chaque anticipation respectifs à chaque parent sont sommés.

Anticiper plusieurs états peut donner lieu à des classeurs surgénéralisés. Pour éviter cela, la marque incertaine est utilisée pour spécialiser leur effet depuis cette marque incertaine plutôt qu'avec la perception courante.

6.2 Distinction entre PAI et autres formes de non-déterminisme

Les états sujets au PAI sont les états distincts ne pouvant être différenciés par la seule perception d'un classeur ; l'objet des EPE (ou des PEP) est de déterminer

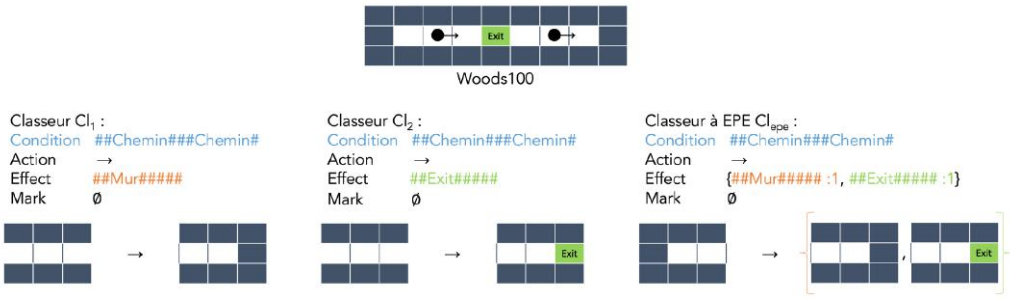


FIGURE 4.11 : Exemple de création de l'effet d'un classeur à EPE. Figure extraite de (Orhand, 2022).

l'ensemble des états qu'il est possible d'atteindre à partir d'un état aliasé et d'une action donnés. Ainsi, dans le cas du PAI, le nombre d'états atteignables depuis cet état aliasé devrait être plus important que le nombre d'actions conduisant à des états distincts, c'est donc comme cela que l'on peut faire la distinction entre des états sujets au PAI et des états sujets à d'autres formes de non-déterminisme (Fig. 4.12).

Pour mettre en œuvre cette détection de PAI, un sous-ensemble de $[M]$ est créé, $[M]_{\text{pai}}$, contenant les classeurs de $[M]$ qui :

— ne sont pas des classeurs à séquence ;

— ont une marque vide ou correspondant à la perception courante ;

— ont une marque incertaine vide ou correspondant à la perception courante.

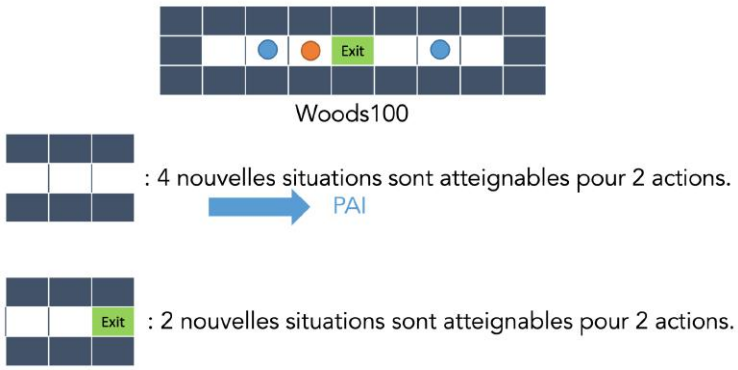


FIGURE 4.12 : Comparaison des états atteignables et du nombre d'actions nécessaires pour les atteindre depuis un état sujet au PAI et un état non sujet au PAI. Les positions bleues identifient deux états sujets au PAI. L'état identifié en orange ne souffre pas du PAI. Figure extraite de (Orhand, 2022).

Les classeurs qui vont être d'intérêt dans $[M]_{\text{pai}}$ sont les classeurs les plus expérimentés (qui ont donc appris davantage de transitions dans l'environnement) : le processus de détection est appelé à intervalle régulier pour mettre à jour une liste, maintenue par BEACS, des états PAI (les classeurs devenant plus expérimentés avec le temps, il est tout à fait possible qu'un état identifié comme PAI à un moment donné soit retiré de la liste après que les classeurs de BEACS aient appris d'autres transitions environnementales). À un pas de temps donné dans le système, les composantes condition C, action A et effets E (effets qui, étant des classeurs à EPE, contiennent maintenant l'anticipation de plusieurs états, et non plus simplement des probabilités de changement d'attribut de la perception courante) des classeurs sont utilisés pour déterminer l'ensemble des états atteignables et le nombre d'actions associées depuis la perception courante. Le détail algorithmique de cette détection est consultable dans (Orhand, 2022), chapitre 7.

6.3 Couplage des EPE aux séquences comportementales

Comme nous l'avons indiqué ci-dessus, l'ensemble des classeurs de BEACS sont des classeurs à EPE, cela est vrai aussi quand un tel classeur est un classeur à séquence comportementale.

Le premier classeur utilisé pour cette création est le classeur sélectionné par le système dans la situation $\sigma(t-2)$, cl_{t-2}. La liste de classeurs ayant correctement anticipé $\sigma(t)$ à $t-1$ contient des classeurs candidats pour la construction d'un classeur à séquence. À la fin de l'ALP, si la perception à $t-1$ est dans la liste d'états PAI maintenue par BEACS, alors des classeurs à séquences sont construits pour chaque classeur candidat cl si :

— cl anticipe au moins un changement ;

— cl_{t-2} anticipe au moins un changement et n'est pas marqué ;

— la longueur de la séquence comportementale à construire est en deçà de la longueur maximale autorisée BSeq_{max}.

Les classeurs comportementaux créés pour chaque cl sont caractérisés comme suit (Fig. 4.13) :

— leur condition C est celle de cl_{t-2} ;

— leur action A résulte du chaînage de l'action de cl_{t-2} et de l'action de cl;

— leur effet E est déterminé en substituant aux changements anticipés par cl_{t-2} et cl les attributs perceptifs de $\sigma(t)$, puis, si des attributs spécifiés de l'effet créé correspondent à la condition, ils sont supprimés, car seuls des changements doivent être décrits par une EPE.

Insertion des classeurs Il faut ensuite déterminer quels classeurs, parmi ceux générés, peuvent être inclus dans la population de BEACS : un processus de subsomption sur l'ensemble $[M]$ à $t-2$ permet de déterminer quels classeurs sont subsumés et quels classeurs sont insérés, comme nous l'avons vu pour les LCS précédemment présentés.

Pour limiter la croissance de la population de classeurs de BEACS et comme la liste des états identifiés comme PAI est mise à jour régulièrement du fait de l'amélioration de l'expérience des classeurs de BEACS, un classeur à séquence inséré à un moment donné peut être supprimé si l'état aliasé qui a causé sa création est finalement considéré comme non sujet au PAI.

Généralisation Les mécanismes de généralisation sont adaptés de façon à ne pas créer des classeurs à séquence qui correspondrait à des états qui ne sont pas liés au PAI. Ainsi, pour un classeur ayant donné lieu à une descendance par mutation, les conditions respectives de ces deux classeurs sont examinées : si un attribut perceptif est généralisé dans une condition et pas dans l'autre, cet attribut peut être généralisé. Cette forme de généralisation est donc indirecte.

Renforcement Pour adapter la récompense des classeurs à la longueur de leur séquence comportementale, BEACS met en œuvre un mécanisme de double Q-

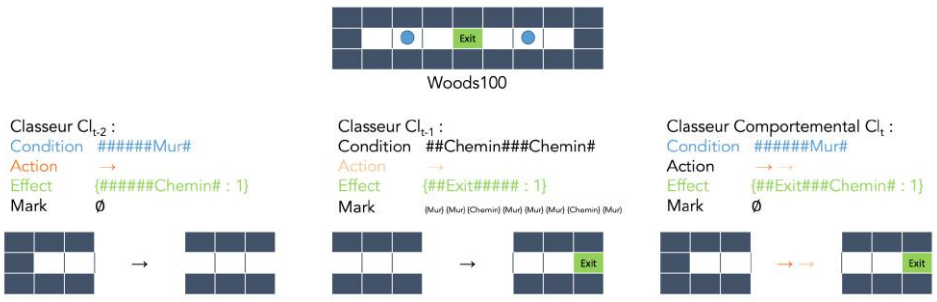


FIGURE 4.13 : Création d'un classeur à séquence comportementale dans BEACS. Figure extraite de (Orhand, 2022).

learning (Hasselt, 2010) qui permet, dans les environnements incertains, d'utiliser deux estimations de la récompense croisées l'une avec l'autre (cela permet d'éviter la surestimation de la récompense, courante dans les environnements incertains avec l'utilisation de simple *Q-learning*). Dans BEACS, un paramètre ϵ_r joue le rôle d'une *différence configurable* entre les valeurs maximales et minimales des deux récompenses prédites (ces valeurs maximales et minimales permettent de déterminer quand les estimateurs ont convergé : leur différence est alors nulle). ϵ_r permet de conserver un écart, artificiel, entre les deux estimateurs pour pouvoir biaiser le système à favoriser des classeurs dont les séquences d'actions sont les plus courtes (les séquences les plus longues verront leur récompense diminuer). Soit cl un classeur passant dans le processus de renforcement, a et b les deux estimateurs, r_a (resp. r_b) la récompense calculée par l'estimateur a (resp. b), max_r (resp. min_r) le maximum (resp. minimum) entre les récompenses r_a et r_b et n_{act} le nombre d'actions dans la séquence de cl. La récompense de cl est calculée comme suit :

$$cl.r = max_r - \frac{(max_r - min_r + \epsilon_r) \cdot (n_{\text{act}} - 1)}{\text{BSeq}_{\text{max}}} \tag{4.11}$$

6.4 Évaluation de BEACS : adaptation du protocole expérimental

Le protocole expérimental est le même que pour BACS et PEPACS, à la différence qu'un seuil, en nombre d'étapes, est introduit pour que BEACS construise la liste d'états aliasés : la détection des états aliasés doit se faire en au plus 1000 étapes, suite à quoi BEACS passe en exploitation pure.

Du côté des métriques, en plus des métriques classiques pour évaluer des systèmes de classeurs (taux de connaissance, nombre moyens d'actions pour atteindre la sortie du labyrinthe, spécificités moyennes des classeurs, taux de classeurs fiables dans la population), une nouvelle métrique a été introduite : l'erreur moyenne accumulée sur les prédictions améliorées (*average EP-accumulated error*, (Orhand et al., 2022)), de façon à pouvoir prendre en compte l'adaptation environnementale des prédictions améliorées. Pour calculer cette erreur, il faut avoir calculé, pour chaque transition possible de l'environnement, la probabilité théorique de se trouver dans un des états atteignables par la réalisation d'une action donnée, et selon les caractéristiques non déterministes de l'environnement (par exemple s'il existe une probabilité que l'action soit choisie aléatoirement en lieu et place de l'action normalement réalisée). Ensuite, pour chaque transition, le classeur le plus expérimenté est identifié dans la population de classeurs fiables. La différence entre la probabilité théorique associée à la transition

et la probabilité d'occurrence de la transition calculée par le classeur est calculée. Cette différence est accumulée pour chaque transition et divisée par la taille d'une anticipation (*i.e.* le nombre de symboles perceptifs décrits dans la composante E — qui est le même que pour C et pour la perception σ) pour donner l'erreur moyenne accumulée sur les prédictions améliorées.

6.5 Comparaison de BEACS avec BACS et PEPACS

Comme pour BACS et PEPACS, nous ne donnons ici qu'une vision synthétique des résultats obtenus avec BEACS ; les détails sont consultables dans (Orhand et al., 2022). Les éléments à retenir sont les suivants :

— Parmi les 1590 états sujets au PAI dans les 23 environnements, BEACS est parvenu à identifier 93.71% de ces états (1490 sur 1590), il en a manqué 100, soit 6.2% et fait 23 faux positifs (1.44%). La précision équilibrée de BEACS pour la détection des états PAI est donc de 99.15%.

— Alors que PEPACS est parvenu à un taux de connaissance de 100% au moins une fois pour l'ensemble des labyrinthes, BEACS n'y est pas parvenu pour deux d'entre eux, avec des taux de connaissances obtenus de 97.85% et de 99.31%. Les séquences comportementales de BEACS sont à l'origine de cette différence, du fait de la tendance plus marquée de BEACS à explorer son environnement.

— Le nombre moyen d'étapes pour atteindre la sortie de BEACS est meilleur que celui de BACS pour 20 des 23 environnements, et aussi bien que celui de BACS dans les 3 environnements restants. En comparaison avec PEPACS, BEACS n'est meilleur que pour 12 des 23 labyrinthes, moins bon pour 8 labyrinthes, et de performance identique pour les labyrinthes restants.

— BEACS présente les erreurs moyennes accumulées sur les prédictions améliorées les plus basses pour tous les environnements considérés, il est donc parvenu à évaluer les probabilités de transition les plus cohérentes au regard des propriétés non déterministes des environnements.

— Pour l'ensemble des environnements, BEACS construit une population de classeurs plus petite que BACS et plus petite que PEPACS pour 15 environnements. Son ratio de classeurs fiables, en comparaison avec PEPACS, est meilleur dans 19 environnements. Les spécificités moyennes de ces classeurs sont, toujours

pour BEACS et en comparaison avec PEPACS, meilleures pour 19 des 23 environnements.

Les performances de BEACS dépassent systématiquement celles de BACS. Bien que ses performances soient légèrement inférieures à celles de PEPACS, il parvient par contre à obtenir des probabilités d'anticipation plus cohérentes avec les probabilités de transitions théoriques, les probabilités de transition de BEACS sont donc plus fidèles.

6.6 BEACS et interprétabilité

L'avantage indéniable de BEACS se situe dans son interprétabilité. Les situations anticipées par BEACS sont décrites complètement dans les classeurs, et ceux-ci peuvent donc être chaînés pour retracer les causes d'un événement. En outre, les différents mécanismes mis en œuvre dans le système pour permettre à BEACS de construire une représentation complète et exacte des environnements en utilisant conjointement des séquences comportementales et des prédictions améliorées sont aussi une source précieuse d'information du point de vue de l'interprétabilité du système (Orhand, 2022) :

— la liste de PAI construite et maintenue par le système permet de connaître avec précision quels états nécessitent l'emploi d'une séquence d'actions et, plus simplement, de savoir quels sont les états sujets au problème de l'aliasing perceptif ;

— les marques incertaines permettent de connaître les états pour lesquels plusieurs situations à anticiper sont potentiellement à apprendre ;

— les marques des classeurs soulignent les états de l'environnement qui ne permettent pas une anticipation correcte ;

— les qualités nous informent sur l'aptitude des classeurs à anticiper ou non des changements perceptifs ;

— les récompenses et récompenses immédiates permettent, respectivement, de mieux comprendre la cohérence d'une action par rapport à d'autres au regard de la tâche à résoudre et la cohérence d'une action dans une situation environnementale donnée.

En revanche, la complexité des environnements entraîne un accroissement de la taille de la population de classeurs, entravant l'explicabilité intrinsèque du système. Des mécanismes de compaction de la population peuvent alors être mis en place, sous réserve de ne pas altérer les représentations construites par les classeurs. Dans sa thèse de doctorat, Orhand (2022) présente une méthode générique de compaction de populations de classeurs dans des ALCS qui n'affectent pas les représentations construites.

7 Synthèse

Au regard de la nomenclature proposée dans le chapitre 2, section 3, les systèmes de classeurs produisent des énoncés théoriques. Les règles qu'ils produisent permettent en effet de caractériser comment des objets (comme des agents et leurs environnements) interagissent entre eux. C'est donc par la production de règles que ces approches résolvent des problèmes.

Les systèmes de classeurs à anticipation apportent la possibilité de prédire les effets d'une action dans l'environnement. Comme leur apprentissage n'est pas stochastique, ils présentent l'avantage de mettre en évidence des liens de cause à effet dans des situations environnementales données, permettant ainsi d'obtenir une interprétabilité intrinsèque au système.

L'incertitude des environnements pose problème aux systèmes de classeurs. Différents mécanismes permettent d'aborder au moins en partie le problème de la gestion de cette incertitude : les séquences comportementales permettent d'accroître l'autonomie de BACS et BEACS, en leur conférant les moyens de faire face au problème de l'aliasing perceptif. Les prédictions améliorées par les probabilités permettent de créer une représentation complète et exacte des environnements dans lesquels PEPACS et BEACS évoluent.

BEACS est le premier ALCS qui combine ces deux mécanismes, les représentations de l'environnement qu'il crée sont complètes, exactes et plus fidèles que celles de BACS et PEPACS. Il offre en outre des possibilités d'interprétation supérieures aux autres ALCS grâce aux différentes adaptations réalisées pour coupler séquences comportementales et prédictions améliorées.

Dans le chapitre suivant, nous revenons sur la caractérisation d'objets dans le cadre des travaux de thèse de Mme Hiba Khodji. Dans le cadre de ces travaux, nous nous intéressons à la prédiction d'erreurs dans des alignements de séquences de gènes

par des réseaux de neurones convolutifs, des approches demandant des mécanismes d'explicabilité *post-hoc*. Une approche quantitative pour évaluer de telles explications, appliquées à une tâche de reconnaissance visuelle, sera présentée.

CHAPITRE 5

Détection d'erreurs dans des prédictions de séquences de gènes

Nous présentons dans ce chapitre une approche originale pour la détection d'erreurs de prédiction de gènes dans des alignements multiples de séquence. Exploitant des représentations visuelles de ces alignements, nous proposons un modèle de réseau de neurones convolutif (CNN) avec une modification architecturale permettant de mieux exploiter la structure intrinsèque d'un alignement de séquences. Utilisant une approche post-hoc *pour produire des explications sur le comportement des modèles, nous avons aussi mis en place une approche quantitative de l'explicabilité* post-hoc *pour les CNN afin de mieux décrire les contributions des caractéristiques extraites par le modèle pour effectuer ses prédictions.*

1 Introduction

L'alignement multiple de séquences (*Multiple Sequence Alignment*, MSA) fait référence à un ensemble de méthodes algorithmiques pour aligner entre elles trois, ou plus, séquences biologiques (protéines, ADN, RNA) apparentées du point de vue de l'évolution. L'algorithme d'alignement prend en compte différents événements évolutionnistes comme des insertions, des délétions ou des mutations (ou substitutions, *mismatch*). Une *délétion* est l'absence d'un ou plusieurs nucléotides ou acides aminés ; une *insertion* correspond à l'ajout d'une portion de séquence entre deux nucléotides ou acides aminés (des séquences ajoutées aux extrémités de la séquence

sont des *extensions*) ; une substitution correspond au remplacement d'un nucléotide ou d'acide aminé par un autre dans une colonne de l'alignement.

Les MSA jouent un rôle clef pour identifier des sites fonctionnels dans des séquences biologiques, prédire les fonctions de protéines ou leur structure secondaire ou encore inférer des phylogénies. Pour un ensemble de séquences définies sur un même alphabet fini, un alignement consiste à réaliser une mise en correspondance optimale des lettres des séquences en conservant leur ordre et possiblement en insérant des trous (*gap*) (Fig. 5.1).

Différents algorithmes ont été proposés pour générer des alignements multiples de séquences, reposant sur des approches progressives, itératives, sur des modèles de Markov cachés ou encore des algorithmes génétiques (Thompson, 2016).

Des bases de données criblées d'erreurs L'avènement du séquençage haut débit a considérablement augmenté le nombre de séquences disponibles, conduisant à de nouveaux défis, non seulement en termes de gestion de ressources de calcul pour analyser ces données, mais aussi en termes de qualité et de détection d'erreur : 50% des séquences biologiques disponibles dans les principales banques de données publiques seraient erronées (Prosdocimi et al., 2012). Ces erreurs peuvent provenir de défauts d'alignements des algorithmes de construction de MSA (Warnow, 2021) mais aussi d'anomalies dans les séquences elles-mêmes (Meyer et al., 2020).

Pour exploiter de manière fiable les masses de données disponibles, il faut donc être capable de faire la distinction entre des erreurs dans les MSA et les événements biologiques tels que les variants génétiques ou l'expression différentielle d'isoformes.

Identification d'erreurs dans des MSA Il existe différents modèles permettant d'évaluer la qualité des alignements multiples de séquences. Certaines approches permettent en particulier d'identifier des séquences anomales au regard des autres séquences constitutives de l'alignement.

```
S1  GARFIELDTHELASTFA-TCAT
S2  GARFIELDTHEFASTCA-T---
S3  GARFIELDTHEVERYFASTCAT
S4  --------THE----FA-TCAT
```

Figure 5.1 : Exemple d'alignement de quatre séquences. Figure adaptée de (Chowdhury and Garai, 2017).

Deux approches notables se focalisent sur l'évaluation de la divergence de séquences dans un alignement : OD-Seq (Jehl et al., 2015) et EvalMSA (Chiner-Oms and González-Candelas, 2016). OD-Seq est un programme de détection de séquences aberrantes dans un alignement utilisant une mesure de distance des séquences par rapport à toutes les autres séquences de l'alignement. OD-Seq exploite ou calcule une matrice de distance de l'alignement, représentant une distance évolutionnaire entre chaque paire de séquences dans l'alignement, puis utilise soit du bootstrap ou une mesure d'écart interquartile pour déterminer si une séquence est ou non aberrante.

EvalMSA est également un programme de détection de séquences aberrantes dans des alignements multiples de séquences. Les séquences sont d'abord évaluées de façon à leur assigner un poids dénotant leur importance dans l'alignement. Ensuite, pour chaque séquence, EvalMSA évalue leurs tendances à introduire des *gaps* dans l'alignement. Plus une séquence introduit de *gaps* dans les séquences restantes et plus elle sera considérée comme divergente (donc comme aberrante).

Un outil alternatif, SIBIS (Khenoussi et al., 2014), repose sur une approche bayésienne pour identifier des incohérences dans des alignements. Dans un premier temps, la distribution *a priori* des acides aminés est définie en utilisant un processus de Dirichlet à mélange pour prendre en considération le fait que différentes régions dans une protéine sont soumises à différentes pressions évolutives. Ce modèle de mélange permet alors de définir différents prieurs pour différents ensembles d'acides aminés. L'observation des acides aminés dans une colonne donnée permet de modifier la distribution antérieure pour obtenir la distribution postérieure. Cette distribution postérieure est ensuite utilisée pour calculer la probabilité d'observer un nouvel acide aminé dans une colonne. Finalement, un score est calculé pour un segment de taille N. Ce score est égal à la probabilité que chaque acide aminé de la séquence soit effectivement aligné avec la colonne correspondante ; c'est ce score qui permet de déterminer si une séquence de l'alignement est considérée comme incohérente.

À notre connaissance, il n'existe pas d'approche basée sur des réseaux de neurones, profonds ou non, pour évaluer des MSA. Si utiliser des réseaux de neurones pour utiliser des réseaux de neurones n'est pas quelque chose que nous prônons, il n'en reste pas moins que des alignements multiples de séquences sont des données complexes, avec des motifs à la fois dans les séquences en tant que telles, mais aussi entre différentes séquences du fait de leur parenté.

Une approche alternative pour la détection d'erreurs dans les MSA ? Par défaut, les programmes d'alignement multiple produisent leurs résultats dans un format texte tel que FASTA (Pearson, 2003). Dans ce format, une séquence est décrite par deux lignes : la première ligne débutant par « > » contient l'identifiant de la séquence et possiblement un commentaire. La seconde ligne est la séquence elle-même, représentée par une chaîne de caractères sur un alphabet donné : les acides nucléiques $\{A, C, G, T\}$ pour une séquence d'ADN, $\{A, C, G, U\}$ si c'est une séquence d'ARN et les acides aminés (spécifiés par des codons, des chaînes de trois nucléotides) $\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, O, U, S, T, W, Y, V\}$ si c'est une séquence de protéine.

Ce sont des séquences de protéines que nous utilisons dans ces travaux. Un alignement sera donc composé de plusieurs entrées au format FASTA, dans lequel les séquences de protéines sont organisées horizontalement, les séquences proches partageant une relation plus importante du point de vue évolutionniste, et dont les résidus (les acides aminés) sont mis en correspondance verticalement de façon à ce qu'une colonne de l'alignement dénote une homologie structurelle ou fonctionnelle entre les régions des séquences.

Le format textuel offert par FASTA est « limité » du point de vue des informations qu'il fournit, ce qui explique pourquoi les approches de l'état de l'art pour analyser des MSA et évaluer leurs qualités reposent, par exemple, sur des approches bayésiennes exploitant la séquence en elle-même. Il existe un outil, nommé ADOMA (*Alternative Display Of Multiple Alignment*) (Zaal and Nota, 2016), qui permet d'obtenir des sorties alternatives au format FASTA lors de l'utilisation de programmes d'alignement multiple. ADOMA permet en particulier d'obtenir une visualisation en couleur des séquences : dans le cas de séquences de protéines, les acides aminés partageant des propriétés physico-chimiques similaires (Fig. 5.2) partagent la même couleur, ainsi qu'illustré sur la figure 5.3. ADOMA produit ce type de sortie dans un document HTML.

Souhaitant proposer une approche alternative à la détection des erreurs de prédiction dans des séquences de gènes, nous avons là les deux éléments qui ont motivé l'approche que nous proposons :

1. les contextes horizontaux et verticaux dans un MSA sont une source importante d'information ;
2. il est possible de représenter alternativement des séquences biologiques en y apportant des informations visuelles.

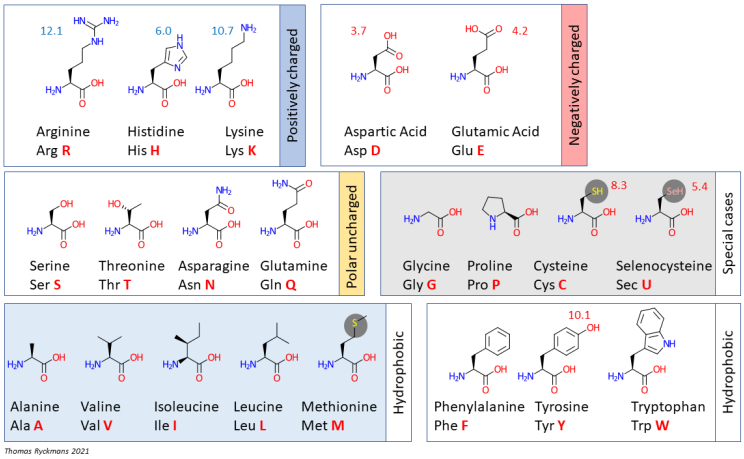


FIGURE 5.2 : Propriétés physico-chimiques des acides aminés (https://en.wikipedia.org/wiki/Proteinogenic_amino_acid).

```
KIR2DL4*00101     TLYKKDGVPVPELYNRIFWNSFLISPVTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*003       TLYKKDGVPVPELYNRIFWNSFLISPLTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*004       TLYKKDGVPVPELYNRIFWNSFLISPLTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*007       TLYKKDGVPVPELYNRIFWNSFLISPVTPAHAGTNRCRGFRPHSPTEWSAPSNPLVIMVT
                  ***************************:******* *****:********************

KIR2DL4*00101     TLYKKDGVPVPELYNRIFWNSFLISPVTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*003       TLYKKDGVPVPELYNRIFWNSFLISPLTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*004       TLYKKDGVPVPELYNRIFWNSFLISPLTPAHAGTYRCRGFHPHSPTEWSAPSNPLVIMVT
KIR2DL4*007       TLYKKDGVPVPELYNRIFWNSFLISPVTPAHAGTNRCRGFRPHSPTEWSAPSNPLVIMVT
                  ***************************:******* *****:********************
```

FIGURE 5.3 : Exemple de sortie d'un programme d'alignement multiple modifié par ADOMA. Figure extraite de (Zaal and Nota, 2016).

Nous avons donc souhaité étudier le bien-fondé d'une approche basée sur des images pour l'identification d'erreurs de prédiction de séquences de gènes dans des alignements multiples de séquences. Dans un tel contexte, les réseaux de neurones convolutifs sont des modèles de choix. Nous proposons alors De-MISTED (*Deep learning for MultIple Sequence alignemenTs Error Detection*), une approche basée sur l'utilisation de réseaux de neurones convolutifs pour réaliser une classification binaire d'alignements multiples de séquences (MSA contenant au moins une erreur / MSA sans erreur).

De-MISTED est présenté dans la section suivante. En plus des résultats quantitatifs du modèle, nous présentons aussi des résultats qualitatifs avec une approche d'explicabilité visuelle *post-hoc*.

Comme précisé dans le chapitre I.2, les approches *post-hoc* pour l'explicabilité sont critiquables. Ainsi, pour néanmoins pouvoir utiliser cette approche constitutive des réseaux de neurones, dans la section 3.2, nous présenterons une approche pour la

quantification de l'explicabilité visuelle *post-hoc* afin de tâcher d'y *voir* plus clair dans ces visualisations. Cette approche, appelée QXP *Quantitative eXPlanation*, a été proposée pour évaluer une approche d'apprentissage par transfert, appelée TL-CAM (*Transfert Learning based on Class Activation Mapping*), reposant sur une approche de visualisation *post-hoc* de réseaux de neurones convolutifs.

2 De-MISTED : classification binaire de MSA

2.1 Données

Prétraitement

Le jeu de données utilisé consiste en 19 942 MSA provenant d'un ensemble d'alignements automatiquement annotés avec SIBIS (Khenoussi et al., 2014) tel que décrit dans (Meyer et al., 2020). Sur ces 19 942 alignements, 12 545 ont été identifiés par SIBIS comme contenant au moins une erreur ; les 7 397 alignements restants ont été annotés comme sans erreur. Dans les alignements erronés, on compte 44 001 délétions, 27 289 insertions et 11 015 substitutions. Comme le jeu de données a été annoté automatiquement, il est possible que des erreurs aient été manquées par le programme d'annotation[1].

Pour ré-équilibrer les deux classes dans les données, 3 811 alignements parmi les alignements erronés ont été extraits et les séquences erronées de ces alignements ont été retirées des MSA par filtrage (les séquences erronées étant identifiées). Les 3 811 MSA originaux (avec erreurs) sont conservés dans le jeu de données et les 3 811 MSA filtrés (sans erreur) sont ajoutés aux 7 397 alignements initiaux, pour obtenir un total de 11 208 alignements sans erreur dans le jeu de données.

Ces alignements ont ensuite été convertis par ADOMA, dont les fichiers HTML produits ont été convertis en images avec l'utilitaire `wkhtmltoimage` (`https://wkhtmltopdf.org/`). Les lettres des acides aminés ont été préalablement retirées du HTML et les couleurs ont été ajustées pour les rendre plus contrastées. Un exemple d'image de MSA ainsi généré est visible sur la figure 5.4.

[1] Nous verrons en étudiant les résultats de De-MISTED que c'est bien le cas (section 2.4).

FIGURE 5.4 : Exemples d'images de MSA erronés (annotées manuellement).

Split des données

Les données ont été séparées (le *split*) en trois ensembles distincts composés de 60%, 20% et 20% des données respectivement pour le jeu d'entraînement, le jeu de validation et le jeu de test. La distribution des classes `ERRORS` et `NO_ERROR`, correspondant respectivement aux alignements contenant au moins une erreur et aux alignements sans erreur, est similaire dans les trois jeux (Fig. 5.5)

2.2 Modèles

> *Multiple mutually connected areas in the ventral cortical pathway receive visual input and extract local form features that are subsequently grouped into increasingly complex, more meaningful image elements.*

Cette phrase, extraite de (Tschechne and Neumann, 2014) décrit à gros grains la manière dont nous traitons les informations visuelles qui nous parviennent. Les réseaux de neurones convolutifs reposent sur cette idée d'extraction de caractéristiques locales qui, de couche en couche dans le réseau, permettent de reconstituer un objet.

De-MISTED repose sur des réseaux de neurones convolutifs (*Convolutional Neural Networks*, CNN). Ces réseaux reposent sur des opérations de convolution entre des filtres (dont les paramètres sont appris par le modèle) et les données d'entrées (ici,

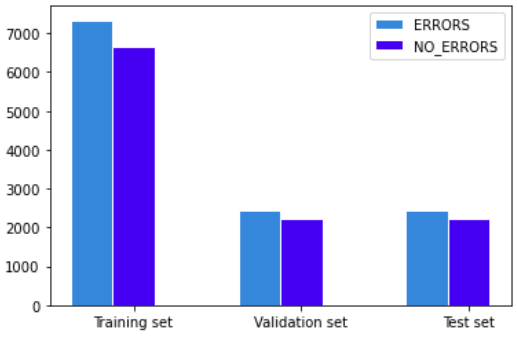


FIGURE 5.5 : Distribution des deux classes dans les jeux de données d'entraînement, de validation et de test.

des images) ; une fonction d'activation, comme pour les réseaux de neurones réguliers, permet de déterminer la quantité de signal qui passera à la couche suivante. On appelle « cartes d'activation » les résultats de la convolution et de l'application de la fonction d'activation. Des couches successives de convolution vont permettre l'apprentissage d'une hiérarchie de caractéristiques propre aux données et une couche ou plusieurs couches entièrement connectées permettent, en fin de course, d'assigner une classe aux entrées. En général, une opération de sous-échantillonnage (*pooling*) est appliquée après une convolution afin de sous-échantillonner les cartes d'activations produites et ainsi fournir une représentation « condensée » des caractéristiques mises en évidence par la convolution.

Adaptation des filtres dans De-MISTED

Dans un réseau de neurones convolutif, les opérations de convolution consistent en l'application de filtres (au sens du traitement du signal ou d'images) sur les entrées de la couche (soit sur les données d'entrée s'il s'agit de la première couche, soit sur les cartes d'activation produites sur une couche l s'il s'agit de la couche $l+1$). Ainsi qu'illustré sur la figure 5.6, les connexions sur des couches de convolution sont *locales* : une unité de la couche (qui correspond à un point d'une carte d'activation) va traiter un groupe de pixels dans son champ récepteur qui correspond à la taille du filtre, dénoté H sur la figure. Ce filtre va glisser sur l'ensemble de l'image pour produire, suite à l'application de la fonction d'activation, une *carte d'activation*. Une telle carte souligne, dans l'entrée de la couche, les régions de cette entrée qui vont le plus activer le filtre (et qui sont donc *a priori* les plus pertinentes pour une classification donnée).

Les valeurs visibles dans le filtre H correspondent aux paramètres que doit apprendre le réseau de neurones et sur chaque couche, il est possible d'apprendre différents jeux de paramètres (donc différents filtres) qui vont donner autant de cartes d'activations.

Dans De-MISTED, deux modèles convolutifs dénotés `A` et `B` sont utilisés. La structure fondamentale de ces deux modèles est la même : 6 blocs de convolutions calculant, respectivement, 16, 32, 64, 128, 256 et 512 cartes d'activation, suivies d'une couche entièrement connectée de 512 unités dont les sorties sont passées dans une fonction sigmoïde pour obtenir la classification binaire.

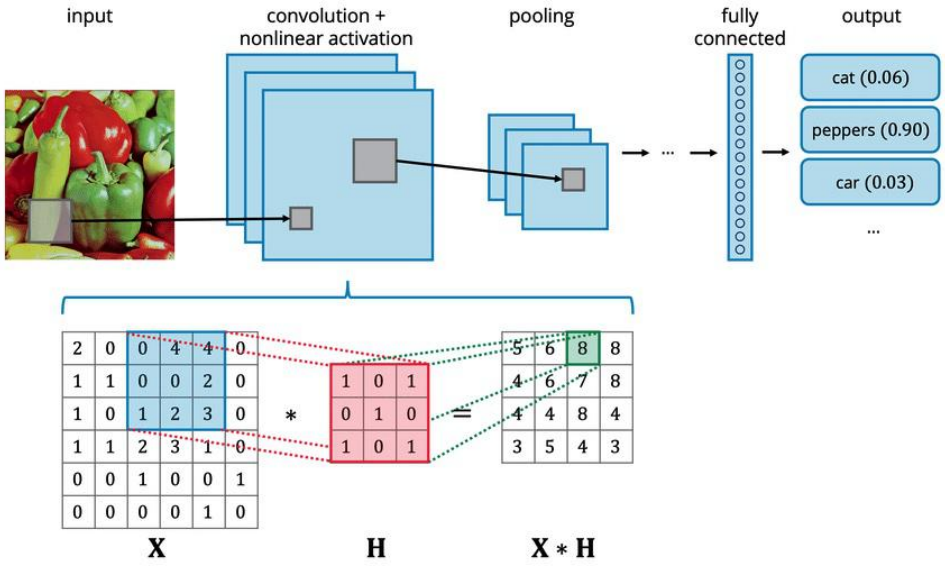


FIGURE 5.6 : Opération de convolution entre une entrée X et un filtre H qui correspond aux paramètres de la couche pour produire des cartes d'activation. Plusieurs filtres peuvent être appris sur une couche donnée, produisant autant de cartes d'activation sur la couche. Figure extraite de (Cheung et al., 2020).

Dans le modèle `A`, un bloc de convolution est constitué d'une unique opération de convolution avec des filtres de 5×5. Dans le modèle `B`, un bloc comprend deux opérations de convolution : la première utilise des filtres horizontaux de 5×7, la seconde des filtres de 3×2. Ce *filtrage hybride* a été mis en place pour évaluer l'exploitation, par le modèle, de la structure d'un MSA, dont on rappelle que l'organisation est constituée à la fois de motifs horizontaux et verticaux. Les deux modèles sont schématisés sur la figure 5.7. On y voit que pour le modèle `B`, deux convolutions successives surviennent dans un bloc : un premier filtre rectangulaire, plus haut que large, va d'abord être appliqué sur l'entrée de la couche et le résultat de cette première convolution est ensuite traité par un filtre plus large que haut. Une opération de normalisation par lot (*batch*) est réalisée après chaque opération de convolution, afin d'accélérer la convergence des modèles. Les opérations de *drop-out* permettent d'éviter que les modèles ne fassent du surapprentissage.

Les modèles sont entraînés pendant 100 époques avec arrêt prématuré (*early stopping*) sur des lots de données de taille 32 avec l'optimiseur ADAM (*ADaptive Moment Estimation*), pour un taux d'apprentissage adaptatif de $1e^{-2}$ (réduit d'un facteur 0.1 après 5 époques si l'erreur sur le jeu de validation ne diminue pas) et

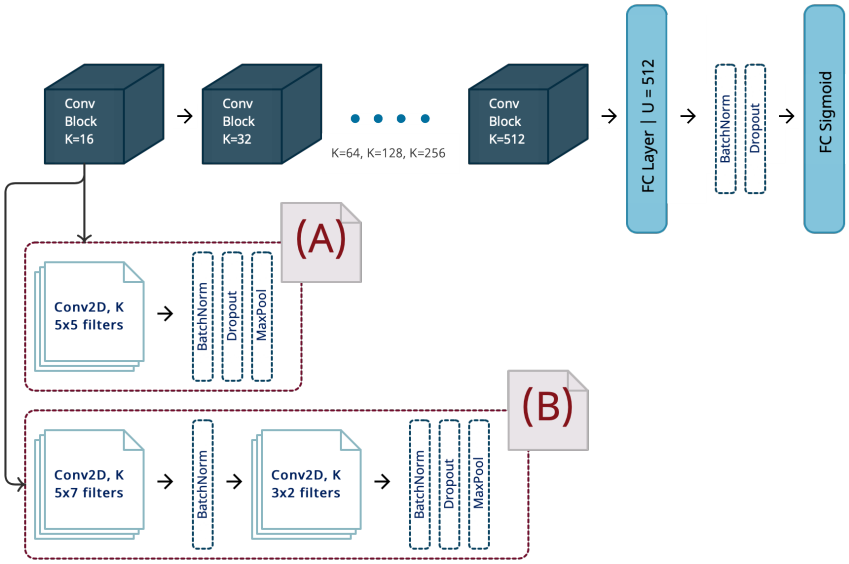


Figure 5.7 : Architecture des modèles `A` et `B` dans De-MISTED. Chaque modèle comprend six blocs convolutionnels. Pour `A`, chaque bloc contient une unique opération de convolution avec des filtres carrés. Le modèle `B` utilise des blocs de convolution réalisant de deux opérations de convolution avec un filtrage hybride à base de filtres rectangulaires.

une décroissance des poids (*weight decay*) de $1e^{-4}$, avec comme fonction d'erreur l'entropie croisée.

2.3 Résultats quantitatifs

La figure 5.8 synthétise les résultats obtenus par les modèles `A` et `B` pour différentes métriques : précision, rappel, *F1-score*, exactitude et perte. La première observation que l'on peut faire est qu'il n'y a pas de différence significative entre les modèles `A` et `B`, malgré les différences dans leurs architectures respectives.

La précision permet d'évaluer la capacité d'un modèle à identifier correctement des instances positives par rapport au nombre total de prédictions positives. Le rappel permet d'évaluer le nombre de prédictions positives par rapport au nombre

Class	Precision	Recall	F1 Score	Acc.	Loss
ERRORS	0.84	0.92	0.88	0.87	0.34
NO_ERRORS	0.90	0.80	0.85		

(a)

Class	Precision	Recall	F1 Score	Acc.	Loss
ERRORS	0.85	0.92	0.88	0.87	0.32
NO_ERRORS	0.91	0.82	0.86		

(b)

Figure 5.8 : Résultats quantitatifs des modèles `A` (à gauche) et `B` (à droite).

d'instances positives dans le jeu de données. Ainsi le modèle `A` (resp. `B`) identifie correctement 92% de tous les MSA erronés dans le jeu de test avec une précision de 84% (resp. 85%).

Les matrices de confusion des deux modèles (Fig. 5.9) permettent de mieux caractériser les capacités d'identification des deux modèles. Là aussi, la différence entre les deux modèles n'est pas flagrante. Le modèle `A` (resp. `B`) a classé de manière erronée 193 (7.3%) (resp. 187, 7.6%) des MSA de la classe `ERRORS`. On note que les deux modèles ont respectivement classé comme erronés 435 (19.6%) et 409 (18.4%) alignements initialement annotés comme étant sans erreurs.

La capacité des deux modèles à classer comme erronés des MSA dépend de leur capacité à identifier (de manière latente) différents types d'erreurs. Pour examiner plus avant ces capacités, nous avons annoté avec SIBIS l'ensemble des alignements classés correctement comme contenant au moins une erreur par nos modèles afin de connaître les types d'erreurs présentes dans chacun de ces alignements. Ces résultats sont présentés dans la figure 5.10 : on peut voir dans ce tableau que le nombre total d'erreurs dans le jeu de test est de 6 793. Les modèles `A` et `B` en ont identifié respectivement 95.5% et 95.6%. Comme les alignements peuvent contenir plusieurs erreurs et de différents types, nous avons aussi spécifiquement examiné les alignements ne contenant qu'une seule erreur. Dans ce cas, sur les 825 alignements erronés du jeu de test, les modèles `A` et `B` ont respectivement identifié comme erronés 84.6% et 85% des MSA. Dans ces alignements avec une seule erreur, le type de l'erreur a un impact significatif sur les performances des modèles : par exemple, alors que les deux modèles sont capables de classer comme erronés des MSA contenant des délétions à

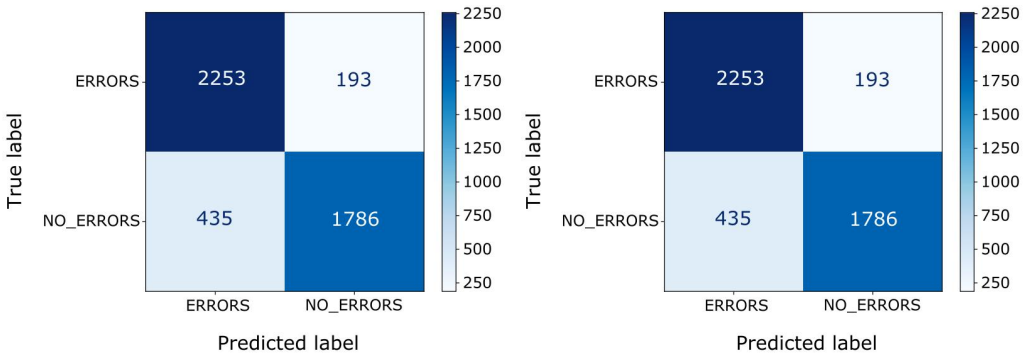


FIGURE 5.9 : Matrices de confusion des modèles `A` (à gauche) et `B` (à droite).

hauteur de 90-92%, les insertions internes sont moins susceptibles de permettre aux modèles de détecter un alignement comme erroné.

Nous avons également mené une étude comparative avec deux programmes d'analyse de la qualité de MSA : EvalMSA (Chiner-Oms and González-Candelas, 2016) et OD-seq (Jehl et al., 2015). Ces deux approches sont utilisées pour détecter des séquences faisant office d'*outliers* dans des alignements multiples. Cette détection peut être « détournée » pour identifier des alignements contenant au moins une erreur, nous permettant ainsi de comparer ces deux outils avec De-MISTED. Comme indiqué sur la figure 5.11, pour toutes les métriques, De-MISTED est très significativement plus performant que Eval-MSA et OD-seq.

Les modèles `A` et `B` présentent tous deux de bonnes performances. Toutefois, les résultats quantitatifs ne permettent pas d'affirmer qu'un modèle est significativement meilleur que l'autre, même si `B` est systématiquement un peu meilleur que `A`. Nous nous sommes alors demandé si les différences architecturales entre les deux modèles pouvaient se refléter dans leurs capacités à *localiser* les erreurs dans les alignements.

2.4 Résultats qualitatifs

Pour réaliser une analyse qualitative des performances des modèles, nous nous sommes reposés sur une approche d'explicabilité *post-hoc* de réseaux de neurones convolutifs appelée Score-CAM (Wang and et al., 2020). Comme les techniques similaires permettant de visualiser des cartes de saillance, Score-CAM souligne, dans les images d'entrées, les régions de celles-ci qui ont amené le modèle à sa prédiction.

	Nb. of errors in the test set		Nb. of errors in TN (A)		Nb. of errors in TN (B)	
Error type	all MSAs	SE MSAs	all MSAs	SE MSAs	all MSAs	SE MSAs
N-terminal extension	436	89	394	71	403	73
N-terminal deletion	2 058	298	1 991	278	1 986	282
C-terminal extension	179	23	159	16	157	16
C-terminal deletion	619	96	582	80	591	84
Internal insertion	601	59	559	32	555	34
Internal deletion	2 481	227	2 416	200	2 422	196
Mismatch	419	33	384	21	384	17
Total	6 793	825	6 485	698	6 498	702

TN (True Negatives) : the correctly classified erroneous MSAs.
SE (Single Error) : MSAs containing a single error.

Figure 5.10 : Comparaison des types d'erreurs présents, respectivement, dans le jeu de test et dans les alignements classés comme erronés par les modèles `A` et `B`.

	ERRORS		NO_ERRORS		
Method	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Accuracy (%)
EvalMSA	61	81	67	42	63
OD-seq	61	73	62	50	62
De-MISTED (B)	**85**	**92**	**91**	**82**	**87**

FIGURE 5.11 : Comparaison de De-MISTED avec Eval-MSA et OD-seq.

Dans le cas de la détection d'erreur dans des alignements multiples de séquence, si ces alignements contiennent au moins une erreur, alors une telle approche de visualisation devrait permettre de mettre en lumière la ou les régions de l'image d'entrée qui contiennent des erreurs.

Score-CAM s'applique sur les cartes d'activations d'une couche de convolution : le plus souvent la dernière, car c'est cette couche qui va contenir l'agrégation la plus complète des caractéristiques extraites par les couches de convolution précédentes. Une métrique appelée *Channel-wise Increase of Confidence* (CIC) est définie par Wang and et al. (2020) pour évaluer l'importance de chaque carte d'activation de la couche examinée. Ainsi, pour une image donnée à propos de laquelle on souhaite obtenir une explication de la classe assignée, c, en sortie du modèle, les opérations suivantes sont réalisées :

1. les cartes d'activation de la couche considérée sont suréchantillonnées avec une interpolation bilinéaire pour les ramener à la taille de l'image d'entrée ;
2. les produits du suréchantillonnage sont normalisés dans $[0; 1]$;
3. les cartes maintenant normalisées sont projetées sur l'image d'entrée ;
4. chaque image masquée obtenue est passée en entrée du CNN (le même qui a été entraîné sur les données et ayant fourni les cartes d'activations) en utilisant une fonction *softmax* en sortie[2], permettant d'obtenir un score reflétant l'importance de chaque carte ;
5. pour la classe d'intérêt c, le score final est obtenu par la somme de la combinaison linéaire de chaque score des k cartes avec la carte correspondante (eq. 5.1).

[2]La somme de tous les scores fera ainsi 1.

$$L^c_{Score-CAM} = ReLU\left(\sum_k \alpha^c_k A^k_l\right) \tag{5.1}$$

où α^c_k est le score obtenu pour la carte A^k_l. L'utilisation de la fonction $ReLU$ (pour laquelle $ReLU(x) = x$ si $x \geq 0$, sinon $ReLU(x) = 0$) permet à Score-CAM de se focaliser sur les caractéristiques ayant une influence positive au regard de la classe c.

Score-CAM a alors été utilisé pour produire des visualisations qualitatives des capacités des modèles `A` et `B` à localiser des erreurs dans des alignements. Les alignements utilisés ont été choisis aléatoirement dans le jeu de test.

Alignements ne contenant qu'une seule erreur

La figure 5.12 montre deux exemples d'alignements contenant une insertion, figures (a) et (c), et une substitution, figures (b) et (d), analysés par Score-CAM pour chacun des modèles, `A` et `B`. Sur ces images, les zones de couleur chaude correspondent aux régions identifiées comme significatives par Score-CAM.

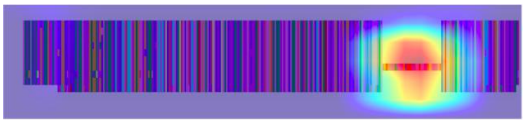
Les deux alignements ont correctement été identifiés comme erronés par les deux modèles et les erreurs ont bien été localisées. C'est dans cette localisation que l'on constate des différences entre `A` et `B` : le Score-CAM obtenu par le modèle `A` est plus étendu que celui produit par le modèle `B` qui, avec ses deux filtres hybrides par bloc, semble capable de se focaliser plus précisément sur la localisation de l'erreur. L'intensité de Score-CAM avec le modèle `B` fait que plus de zones sont identifiées (même faiblement). Sans forcément dénoter des régions contenant des erreurs, cela suggère que le modèle `B` est plus sensible que le modèle `A`.

Alignements contenants plusieurs erreurs

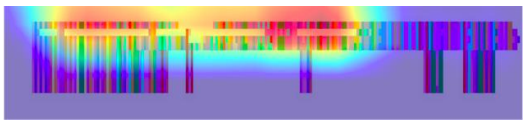
La figure 5.13 présente deux alignements, traités par les modèles `A` et `B`, comportant de multiples erreurs : deux pour l'alignement (a) et (c) ; trois pour l'alignement (b) et (d).

Comme pour les alignements ne contenant qu'une seule erreur, les Score-CAM associés au modèle `B` sont plus marqués. On note par ailleurs que si les deux modèles ont été capables de classer comme erronés les deux alignements, le modèle `A` n'a pas clairement identifié une des trois erreurs de l'alignement (Fig. (b)), contrairement au modèle `B` (Fig. (d)).

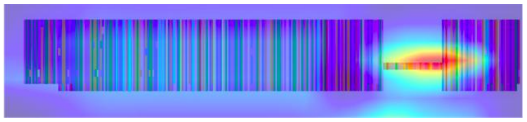
Il existe dans les alignements multiples de séquences des *gap* ou des extensions ou insertions de séquences qui sont constitutives de l'alignement, mais qui pourraient



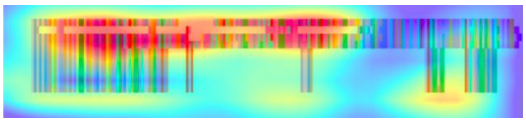
(a) MSA contenant une insertion. Le modèle `A` a bien classé l'alignement comme erroné et a correctement localisé l'erreur.



(b) MSA contenant une substitution. Le modèle `A` a bien classé l'aligmenent comme erroné et a correctement localisé l'erreur.

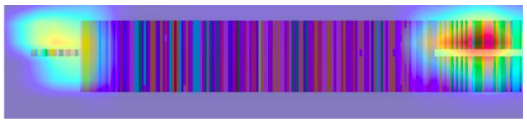


(c) MSA contenant une insertion. Le modèle `B` a bien classé l'alignement comme erroné et a correctement localisé l'erreur.

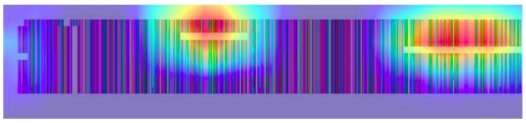


(d) MSA contenant une substitution. Le modèle `B` a bien classé l'aligmenent comme erroné et a correctement localisé l'erreur.

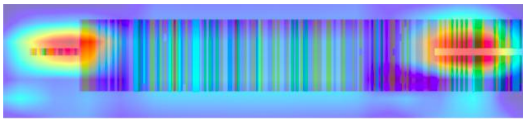
FIGURE 5.12 : Visualisation de la localisation des erreurs dans les MSA avec Score-CAM pour les modèles `A` (a, b) et `B` (c, d). Score-CAM met l'accent sur les régions des images ayant permis aux modèles de faire leur prédiction.



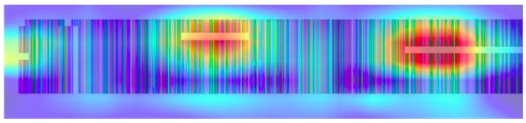
(a) MSA contenant deux erreurs (une extension et une délétion). L'alignement a correctement été identifié par `A` comme étant erroné. Score-CAM montre une focalisation importante sur la délétion ; la focalisation est un peu moins importante sur l'extension.



(b) MSA contenant trois délétions. Le modèle `A` classifie correctement l'alignement comme erroné. Score-CAM montre une focalisation sur deux des erreurs, indiquant que `A` n'a pas détecté la troisième (normalement située à gauche de l'alignement).



(c) MSA contenant deux erreurs (une extension et une délétion). L'alignement a correctement été identifié par `B` comme étant erroné. Score-CAM montre une focalisation importante pour les deux erreurs.



(d) MSA contenant trois délétions. Le modèle `B` classifie correctement l'alignement comme erroné. Score-CAM montre une focalisation sur les trois erreurs, même si l'une d'elle (à gauche de l'alignement) est soulignée plus faiblement.

FIGURE 5.13 : Visualisation de plusieurs erreurs dans des MSA avec Score-CAM pour les modèles `A` (a, b) et `B` (c, d). Les modèles identifient correctement les alignements comme erronés ; les erreurs annotées ont bien été détectées par les modèles (à l'exception de l'une d'elles pour le modèle `A`).

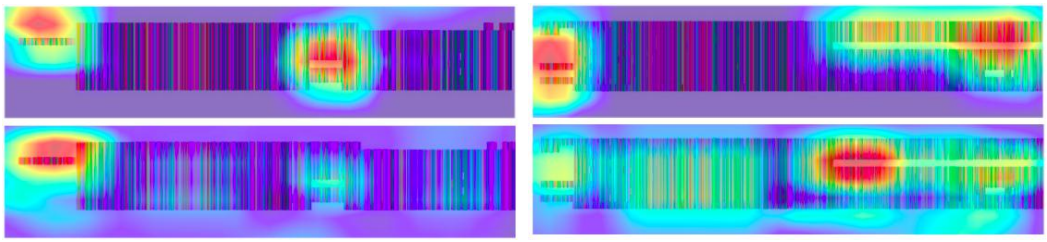
être identifiées comme des erreurs. Score-CAM a donc été utilisé sur des alignements choisis au hasard sur des MSA présentant ces caractéristiques (Fig. 5.14). Le modèle `A`, dont les Score-CAM sont visibles sur la rangée du haut, a identifié des erreurs qui n'en sont pourtant pas dans les alignements. Là encore, le modèle `B` s'en sort mieux : les Score-CAM indiquent des régions correspondant à de réelles erreurs dans les alignements.

Alignements contenant des erreurs non annotées

Comme nous l'avons indiqué, les données du jeu de test sont annotées par l'outil SIBIS (Khenoussi et al., 2014). Avec une spécificité de 93% et une sensibilité de 81%, il est possible que des erreurs dans le jeu de données aient été manquées par SIBIS. Nous avons alors extrait, aléatoirement, des alignements annotés comme sans erreur, mais classés comme erronés par nos deux modèles pour les analyser manuellement.

La figure 5.15 montre que les deux modèles ayant classé les alignements comme erronés se sont bien appuyés sur des régions des MSA ressemblant à des erreurs de délétions, d'extensions et d'insertions. L'ensemble de ces erreurs ont été confirmées comme présentes dans les MSA examinés suite à une expertise humaine. Il est possible que la précision des modèles (Fig. 5.8) soit en réalité une estimation basse des capacités des deux modèles.

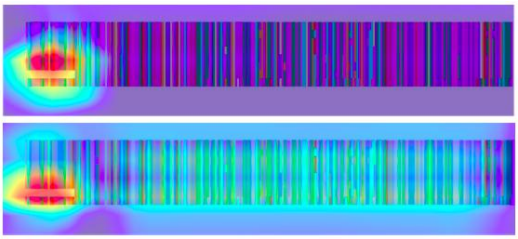
À notre connaissance, il s'agit de la première exploitation de représentations visuelles d'alignements multiples de séquences pour la détection d'erreurs de prédic-



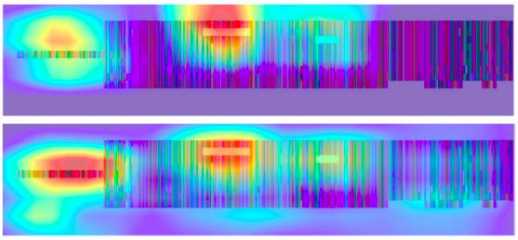
(a) MSA, contenant une unique erreur (une extension), correctement classé par `A` et `B` comme erroné. Les deux modèles identifient correctement l'erreur d'extension, mais pour le modèle `A`, on voit qu'un *gap*, qui n'est pas une erreur, a été identifié comme une délétion.

(b) MSA contenant une unique erreur, mais le Score-CAM calculé pour le modèle `A` indique que le modèle a identifié une seconde erreur, une extension qui n'est pourtant pas une erreur dans l'alignement.

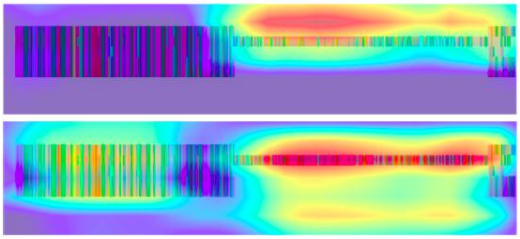
Figure 5.14 : Capacité des modèles (`A`, image du haut, et `B`, image du bas) à distinguer entre des *gap* ou des insertions et extensions constitutifs des alignements et des erreurs véritables.



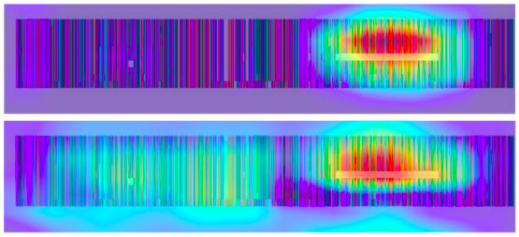
(a) Score-CAM de MSA classés comme erronés par les modèles `A` (rangée du haut) et `B` (rangée du bas). Score-CAM indique la présence d'une délétion.



(b) Pour les deux modèles, Score-CAM indique la présence d'une extension et de deux délétions.



(c) Pour les deux modèles, Score-CAM indique la présence d'une insertion.



(d) Pour les deux modèles, Score-CAM indique la présence d'une délétion.

FIGURE 5.15 : Alignements annotés comme sans erreur par SIBIS et traités par les modèles `A` et `B` (respectivement en haut et en bas sur chaque sous-figure). Les Score-CAM calculés indiquent la présence de différentes erreurs qui ont été manquées lors de l'annotation par SIBIS.

tion. Les premiers résultats obtenus sont très positifs : du point de vue quantitatif, les deux modèles proposés sont significativement plus performants qu'Eval-MSA et OD-Seq, deux approches basées sur des techniques « classiques » de détection de séquences aberrantes dans les MSA.

Une analyse qualitative utilisant la méthode d'explicabilité *post-hoc* Score-CAM indique que les modèles ont bien isolé les erreurs dans les alignements. Les filtres hybrides proposés dans le second modèle le rendent plus efficace et mieux à même de détecter les erreurs dans les MSA.

Cependant, comme nous l'avons vu dans le chapitre I.2, l'explicabilité *post-hoc* est une approche discutable pour interpréter un modèle. Les modèles complexes comme les CNN peuvent produire un nombre important de cartes d'activation dans leurs couches cachées, ce qui ne facilite pas leur interprétabilité. Nous proposons alors TL-CAM (*Transfer Learning using Class Activation Mapping*), une approche d'apprentissage par transfert pour les modèles de type réseaux convolutifs reposant sur

Score-CAM, qui permet de rendre plus parcimonieux l'usage des cartes d'activation dans un réseau de neurones en prétraitant les données en amont. Pour apporter des éléments de validation à cette approche, nous proposons également une méthode de quantification de l'explicabilité obtenue, avec QXP (*Quantitative eXPlainability*).

3 Transférabilité parcimonieuse pour des modèles plus explicables

3.1 TL-CAM

L'apprentissage par transfert permet de réutiliser les connaissances acquises par un modèle entraîné sur des données d'un domaine $\mathcal{D}$ pour résoudre une tâche $\mathcal{T}$. Il existe différentes approches pour réaliser un apprentissage par transfert, qui reposent pour la plupart sur la réutilisation des caractéristiques apprises dans le modèle, comme (Misra et al., 2016) ou (Zhang et al., 2019) ou sur la réutilisation des paramètres appris (Orhand, Khodji, Hutt and Jeannin-Girardon, 2021).

TL-CAM repose sur l'extraction des caractéristiques transférables les plus pertinentes apprises par un modèle source S entraîné sur des données $\mathcal{X} \in \mathcal{D}$. Ces caractéristiques sont extraites en utilisant Score-CAM sur S, et utilisées pour créer et appliquer des masques sur les données de façon à produire des données $\mathcal{X}'$ qui seront utilisées par un modèle cible T. Comme Score-CAM permet d'obtenir une évaluation de la contribution des cartes d'activation produites, nous n'avons pas utilisé toutes les cartes d'activation du modèle source pour construire $\mathcal{X}'$ mais uniquement les K cartes présentant la plus grande variance.

Pour tester cette approche, nous avons utilisé le jeu de données CIFAR-10 (Krizhevsky, 2009), un jeu de données d'images couleur de 32×32 pixels organisés en 10 classes. Après avoir entraîné un modèle source et utilisé TL-CAM pour extraire les K meilleures cartes d'activation apprises de la couche la plus transférable (que nous avons déterminée en utilisant une quantification de la transférabilité dans CNN (Orhand, Khodji, Hutt and Jeannin-Girardon, 2021)), les données obtenues sont masquées avec les activations calculées par Score-CAM pour produire des images telles que celles illustrées sur la figure 5.16.

Pour tester l'approche, trois modèles, constitués de 3 blocs de convolution (comprenant chacune deux opérations de convolution) ont été utilisés : un modèle contrôle, entraîné à partir de zéro sur $\mathcal{X}$; un modèle, dénoté TL, entraîné sur $\mathcal{X}$ et ayant au

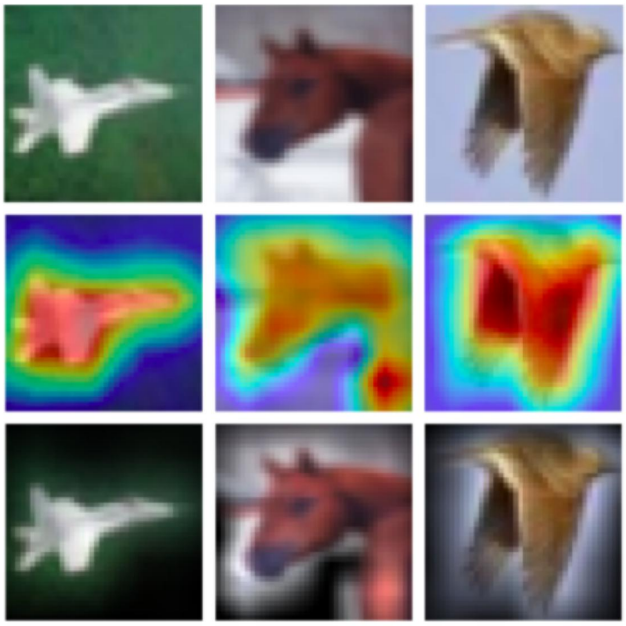


FIGURE 5.16 : Échantillon d'images de CIFAR-10 (rangée du haut) dont les cartes d'activations produites par Score-CAM (rangée du milieu) sont utilisées pour produire les images $\mathcal{X}'$ (rangée du bas).

préalable bénéficié d'un transfert depuis le modèle source qui a permis d'apprendre les cartes d'activation utilisées ; un modèle, dénoté TL-CAM, entraîné sur $\mathcal{X}'$ à partir de zéro.

Les performances des modèles sont visibles sur la figure 5.17. Le modèle ayant bénéficié du transfert montre les meilleures performances, suivi par le modèle contrôle. Le modèle entraîné sur $\mathcal{X}'$ exhibe des performances décevantes au regard des deux autres modèles. Une hypothèse pour expliquer ce résultat est que les images non masquées contiennent des caractéristiques importantes pour la classification, mais qui ont été omises lors de la constitution de $\mathcal{X}'$, du fait que le nombre de cartes d'activation utilisées pour chaque classe a été limité. Il est très probable que les caractéristiques apprises de manière distribuée et multiéchelles par un réseau de neurones se comportent comme un système complexe, car l'objet d'intérêt reconstruit résultera de l'interaction entre les caractéristiques apprises. De ce fait, une limitation radicale du nombre de caractéristiques apprises pourrait ne pas permettre d'obtenir l'émergence attendue. En plus de cela, le fait d'avoir réalisé un transfert pour le modèle TL fait que celui-ci dispose déjà d'un jeu de paramètres mieux situé dans l'espace de recherche.

Model	Loss	Accuracy	Precision	Recall
baseline	0.76	0.73	0.82	0.66
TL	0.60	0.82	0.88	0.76
TL-CAM	0.91	0.69	0.78	0.61

Figure 5.17 : Performances des modèles contrôle (*baseline*), TL et TL-CAM.

3.2 Quantification d'une explicabilité *post-hoc*

Pour mieux comprendre le raisonnement d'un CNN[3], nous proposons de quantifier l'explication de ses prédictions en examinant le nombre de caractéristiques apprises qui sont à l'origine d'une prédiction donnée, ainsi que l'importance de la contribution de ces caractéristiques.

Ici aussi, la technique proposée, QXP (*Quantitative eXPlainability*), repose sur l'utilisation de Score-CAM. Pour une entrée donnée, les cartes d'activation de la dernière couche du CNN étudié sont extraites, suréchantillonnées et normalisées pour être appliquées comme masques sur la donnée d'entrée. Chaque entrée masquée est passée en entrée du CNN pour obtenir une prédiction et chacune des cartes d'activation utilisée est catégorisée selon la classe qu'elle conduit à prédire de manière à calculer leur score moyen sur leurs prédictions.

La figure 5.18 montre des résultats préliminaires, sur une image choisie au hasard, obtenus en utilisant QXP sur les trois modèles présentés précédemment. Pour le modèle TL-CAM, les caractéristiques propres à la classe prédite sont aussi celles qui contribuent le plus à la prédiction. On observe la même chose pour le modèle *baseline*, mais pas pour le modèle avec apprentissage par transfert classique : seulement 27.3% des caractéristiques apprises sont à l'origine à la classification alors qu'elles contribuent pour 47.7% à la prédiction du modèle (cette contribution explique pourquoi la prédiction est correcte).

Deux autres exemples d'analyse avec QXP sont présentés sur la figure 5.19. Selon les modèles, comme pour l'exemple précédent, on observe une certaine variabilité dans les caractéristiques exploitées pour classer la donnée d'entrée. Cette variabilité

[3]En supposant que les opérations réalisées par un CNN puissent être assimilées à un raisonnement, dans le sens commun du terme défini dans le Larousse comme une « *Suite d'arguments, de propositions liés les uns aux autres, en particulier selon des principes logiques, et organisés de manière à aboutir à une conclusion* ». Il est peut-être plus judicieux de dire qu'un réseau de neurones *décrit* des interactions entre des attributs des données, par des opérations successives de compositions de fonctions de tous les paramètres du modèle, et sans se risquer à dire que cela peut constituer un *raisonnement*.

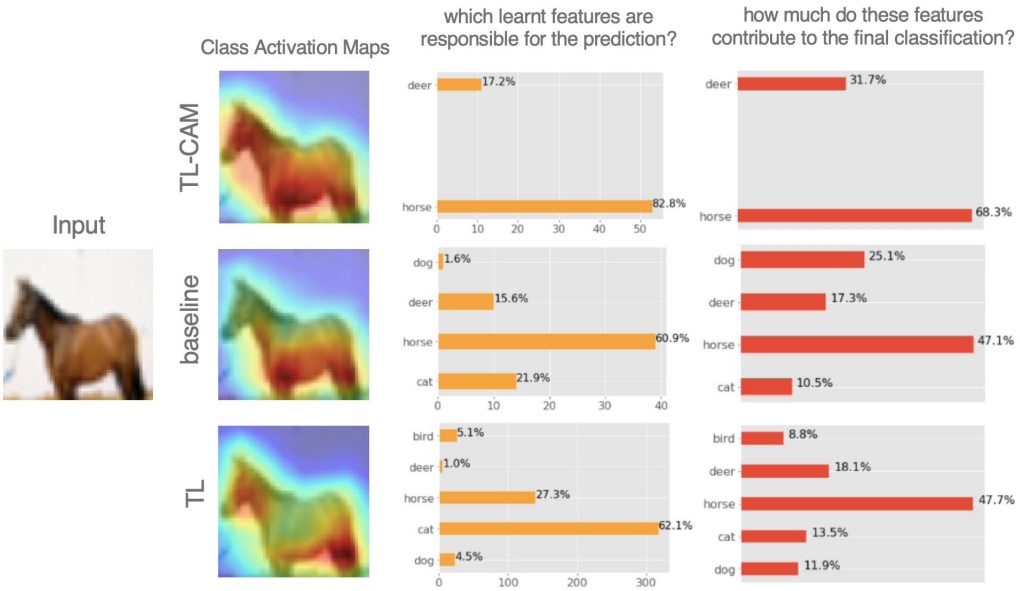


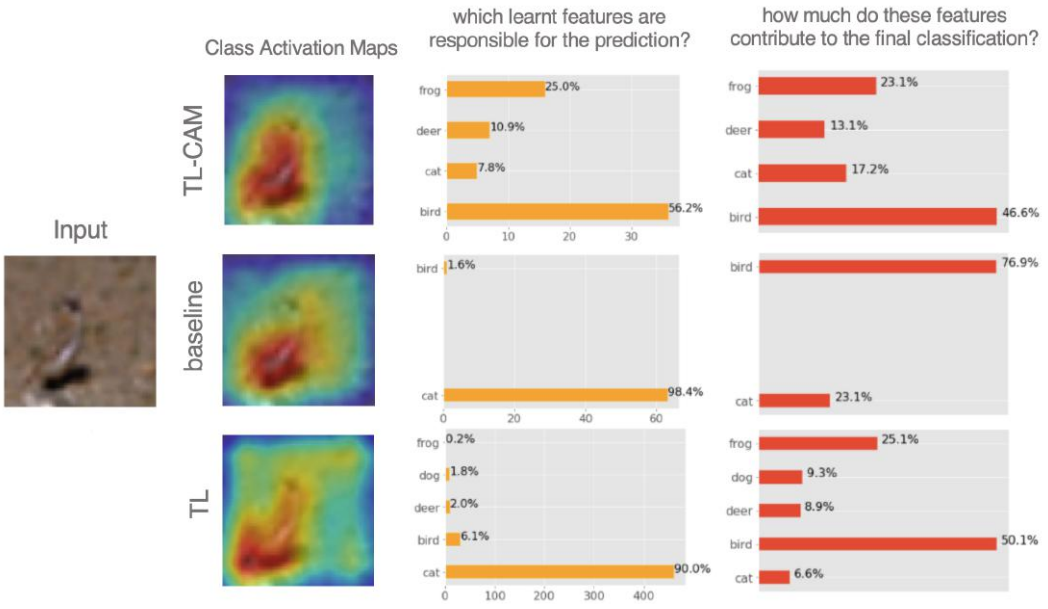
FIGURE 5.18 : Quantification des explications obtenues sur les trois modèles de CNN pour une image choisie au hasard dans le jeu de données.

pourrait être expliquée par le fait que les modèles sont *incertains*. Pour le modèle TL, qui a bénéficié d'un apprentissage par transfert classique, il se peut que le nombre important de caractéristiques apprises (qui n'est pas contraint) explique le nombre de caractéristiques systématiquement utilisées pour réaliser des prédictions. De manière intéressante, pour les modèles *baseline* et TL, on note un nombre important de caractéristiques appartenant à la classe *chat* mais ces caractéristiques ne contribuent pas significativement aux prédictions.

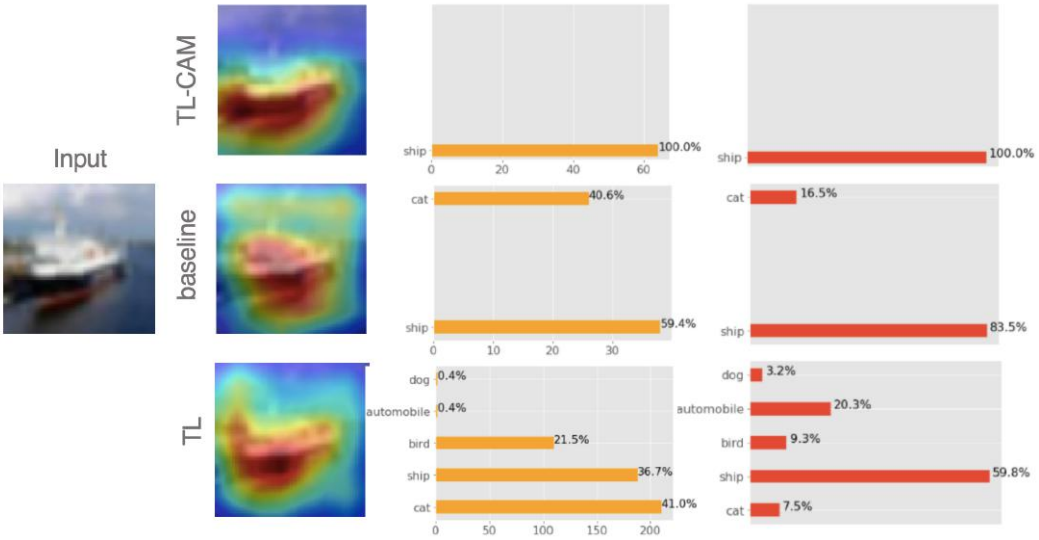
Les résultats présentés dans cette section sont encore à un stade préliminaire, mais nous pensons que l'approche est prometteuse pour obtenir des explications *post-hoc* apportant des informations quantitatives pour dépasser la simple identification de régions d'une image ayant contribué à une prédiction sur cette image.

4 Synthèse

Le modèle De-MISTED présenté dans ce chapitre est une approche originale pour la détection d'erreur de prédiction de gènes dans des alignements multiples de séquences en utilisant des représentations visuelles de ces alignements. Réalisant une



(a)



(b)

Figure 5.19 : Examen des cartes d'activation produites par les différents modèles pour classer une image représentant un oiseau (a) et un navire (b).

classification binaire, De-MISTED a exhibé de très bonnes performances par rapport à d'autres approches de l'état de l'art.

Nous nous situons ici dans un contexte similaire à celui de l'optimisation d'AMR que nous avons présenté au chapitre II.3, dans le sens ou nous nous sommes focalisés sur la caractérisation d'objets (ici, des erreurs dans des MSA). L'approche par réseaux de neurones est toutefois très différente de l'approche utilisée pour l'optimisation d'AMR : pour ce dernier, nous disposions d'un modèle multiphysique intelligible dont il restait à déterminer les paramètres pour caractériser l'AMR. Dans le cas des réseaux de neurones, une forme très générale du modèle est donnée par la spécification de l'architecture du réseau de neurones (nombre de couches, nombre de cartes d'activations produites par couche, *etc.*). Le processus d'optimisation (l'entraînement) permet de déterminer les paramètres du modèle. Si l'on peut supposer qu'un tel modèle capture effectivement des processus biologiques et évolutifs permettant de caractériser ce qu'est une erreur dans un alignement multiple de séquences, ces processus ne sont pas accessibles — puisqu'un réseau de neurones tel qu'un CNN construit un modèle représenté par une succession de compositions de fonctions de milliers, de millions, de paramètres — et encore moins intelligibles.

La méthode d'explicabilité *post-hoc* que nous avons utilisée avec De-MISTED ne permet pas de savoir quoi que ce soit des processus capturés par le modèle. Elle permet cependant de décrire les prédictions réalisées et a permis de montrer que De-MISTED est capable de localiser des erreurs à l'intérieur de MSA. Elle a également permis de valider qualitativement l'usage de ce que nous avons appelé des *filtres hybrides*, montrant que ceux-ci permettent une localisation plus précise des erreurs même si d'un point de vue quantitatif, il n'y a pas de différence significative avec des filtres classiques.

Nous avons utilisé cette même méthode d'explicabilité pour, d'une part, mettre au point une approche parcimonieuse d'apprentissage par transfert, exploitant la mesure de contribution des cartes d'activation produites par le modèle pour ne sélectionner, pour transfert, que celles qui offrent la variance la plus importante. Les résultats obtenus par le modèle utilisant les données construites en utilisant ces cartes d'activation sont quelque peu décevants, mais, d'autre part, nous avons également mis en place une approche de quantification de l'explicabilité des CNN qui, malgré le stade encore préliminaire du travail, nous a permis de constater que le modèle TL-CAM est effectivement plus parcimonieux dans son utilisation de caractéristiques pour classer des données d'entrées.

CHAPITRE 6

Conclusion

1 Quelle épistémologie pour l'intelligence artificielle ?

Depuis mon recrutement en 2016, mes travaux ont eu pour fil conducteur la compréhension et l'étude des énoncés produits par des intelligences artificielles. L'utilisation de ces approches est largement répandue aujourd'hui dans tous les aspects du quotidien, de la recommandation de contenus à l'analyse d'images, de texte ou de parole, en passant par la conduite autonome ou encore l'aide au diagnostic médical. Beaucoup des modèles utilisés pour ces tâches ne sont pas sans poser différents problèmes : opacité, établissement de corrélations fallacieuses, sensibilité, capture des biais présents dans les données, capacités limitées à gérer l'incertitude environnementale. Dans ce contexte, nous nous sommes posé deux questions dans lesquelles s'inscrit notre travail :

1. Comment l'IA peut-elle contribuer à l'amélioration de nos sociétés ?

2. Comment peut-on s'assurer que les IA soient plus éthiques, *a fortiori* plus explicables ?

Nous avons choisi de commencer à aborder ces questions sous un angle épistémologique, en nous interrogeant sur la nature des énoncés produits par les intelligences artificielles et sur les implications qu'ont les énoncés produits.

Nous nous sommes intéressés, dans ce document, à trois approches pour la « construction de savoir » par les intelligences artificielles :

— pour commencer, nous sommes partis d'un modèle multiphysique déjà conçu dont les paramètres devaient être optimisés spécifiquement pour la conception d'un régénérateur magnétique actif (AMR) pouvant opérer en deux modes antagonistes (chapitre II.3) ;

— nous avons ensuite présenté comment un agent peut construire un modèle, exact et complet, de l'environnement incertain dans lequel il évolue (chapitre II.4) ;

— la troisième approche présentée construit également un modèle, mais à des fins de prédiction uniquement, pour la détection d'erreurs dans des alignements multiples de séquences (chapitre II.5).

Le modèle d'AMR repose sur des lois physiques bien définies et intelligibles. Le processus d'optimisation multiobjectif utilisé est basé sur un processus inspiré de l'évolution darwinienne et répond à des contraintes données en amont pour produire un jeu de paramètres spécifiant les caractéristiques de l'objet d'intérêt.

Pour évoluer en environnement incertain, l'agent proposé est un système de classeurs à anticipation amélioré qui combine des mécanismes d'apprentissage par renforcement et de généralisation pour produire un ensemble de règles décrivant l'environnement, les actions qu'il est possible de réaliser dans chaque situation environnementale ainsi que les effets de ces actions sur l'état de l'environnement, permettant alors de retracer les causes d'un événement.

Pour détecter des erreurs de prédiction de gènes dans des alignements multiples de séquences, le contexte est similaire à celui de l'optimisation de l'AMR puisque nous cherchons à caractériser des objets, mais nous sommes aussi dans un contexte où le modèle est appris automatiquement. Ici cependant, le modèle appris n'est pas intelligible et si la détection d'erreur est performante, elle ne nous apprend rien sur les processus sous-jacents potentiellement « découverts » par le modèle.

Si l'on se situe dans un contexte d'aide à la découverte scientifique en utilisant des approches comme des réseaux de neurones profonds, il semble peu probable que les théories « découvertes » puissent nous être accessibles du fait de la complexité intrinsèque des réseaux neuronaux profonds, nous apportant donc qu'une compréhension très limitée des phénomènes étudiés.

Une piste que nous souhaitons alors étudier est de coupler des approches intelligibles, comme des systèmes de classeurs, qui sont capables de décrire les interactions

de composants dans un système, à une approche capable de traiter des données complexes telles que des images, pour extraire automatiquement les caractéristiques pertinentes de ces données.

2 Plaidoyer pour les approches neuro-symboliques

Le couplage d'approches connexionnistes (ou sous-symboliques) —comme les réseaux de neurones— et d'approches symboliques —comme les modèles à base de règles— se rapporte au champ des approches *neuro-symboliques*. Bien que les approches symboliques soient en perte de vitesse importante depuis que l'apprentissage profond a été massivement utilisé depuis 2012, nous pouvons aujourd'hui observer un regain d'intérêt pour ces approches dans le contexte du neuro-symbolisme.

Ces approches sont aujourd'hui principalement des approches dites *intégrées* (Sarker et al., 2021) et consistent en un traitement, par un module connexionniste, de connaissances symboliques. De tels modèles reposent sur l'apprentissage d'une représentation de la connaissance symbolique de manière à pouvoir raisonner sur elle, conduisant à des modèles réalisant du raisonnement déductif profond (*deep deductive reasoning*). Un des défis majeurs abordés pour construire de tels modèles porte sur l'apprentissage d'une représentation pouvant être exploitée par un réseau de neurones ; autrement dit, à vectoriser les variables, prédicats ou relations issues des connaissances symboliques sur lesquelles le modèle doit raisonner. Des approches récentes pour le raisonnement profond incluent les *Deep Deductive Reasoners* (Ebrahimi et al., 2021), les *Logic Tensor Networks* (Badreddine et al., 2022) ou les *Deep Logic Models* (Marra et al., 2019). Des connaissances symboliques peuvent également être utilisées pour améliorer le processus d'apprentissage d'approches connexionnistes, à l'image des travaux de Gaur et al. (2020) et Oltramari et al. (2020) ou encore pour expliquer un modèle profond (Confalonieri et al., 2021).

2.1 Le neuro-symbolisme comme métaphore au couple perception-raison

Nous pensons qu'une approche alternative aux approches neuro-symboliques est envisageable. Cette approche ne vise pas à construire des raisonneurs profonds, car ceux-ci restent des modèles profonds qui ne peuvent que difficilement être expliqués

ou contraints dans un cadre éthique, comme nous l'avons vu à plusieurs reprises dans ce document.

Pour nous, l'intérêt majeur et indéniable des approches profondes est leur capacité à extraire des faits dans des données complexes : il s'agit donc d'une manière de *percevoir* un environnement riche et complexe, dont les caractéristiques essentielles pour la résolution d'une tâche donnée ne sont pas accessibles trivialement à des modèles non complexes.

Pensons, une minute, à notre propre perception de l'environnement et à quel point celle-ci peut nous tromper ou à quel point elle ne représente qu'un fragment de ce qui constitue notre compréhension du monde. Les perceptions ne sont rien d'autre que des signaux reçus par nos organes sensoriels et interprétés par notre cerveau pour en construire des représentations mentales (Damasio, 2000). Les débats sur l'induction lancés au XVIII$^{\text{ème}}$ siècle dans ce qui allait devenir la philosophie des sciences ont bien montré que les expériences sensorielles seules ne permettaient pas une solide construction de savoirs. Nous pourrions même aller plus loin, en avançant qu'*expliquer* ces perceptions n'est pas forcément pertinent et qu'en définitive, il serait peut-être plus adéquat de les considérer pour ce qu'elles sont, c'est-à-dire plus ou moins précises, certaines, et de les utiliser *conjointement* avec d'autres facteurs dans un processus de raisonnement interprétable et intelligible visant à fournir des explications sur un phénomène donné (et pas seulement des descriptions).

Les travaux que nous avons menés jusqu'à présent nous ont conduits à cette vision du neuro-symbolisme, que nous souhaitons explorer et dont nous pensons qu'elle permettrait :

— de contourner le problème de l'explicabilité des boîtes noires, en acceptant que la perception de ces modèles puisse être faussée et qu'ils bénéficieraient davantage d'améliorations par l'expérience ou l'adjonction de nouvelles connaissances que par de l'explicabilité *post-hoc* de toute façon discutable ;

— d'éviter de construire des modèles fondant leurs décisions sur des processus de raisonnement insondables ;

— finalement, d'amener l'intelligence artificielle vers des cadres d'explicabilité plus solides épistémologiquement parlant, fournissant des explications sous la forme de raisonnements logiques qui constituent des *faisceaux de preuves*, jouant sur

le manipulationnisme[1] pour comprendre l'importance d'une variable donnée et incorporant des règles floues permettant de modéliser l'incertitude inhérente à l'environnement ; des mécanismes de subsomption ou de compaction des règles produites ainsi que l'application d'un principe de subsidiarité permettant de remettre les « bonnes » explications, au « bon » niveau et au « bon » destinataire.

2.2 L'autonomie comme vecteur d'explicabilité et d'éthique ?

Dans ce contexte, nous souhaitons pousser plus loin les travaux que nous avons menés sur les systèmes de classeurs à anticipation. De manière intéressante, c'est le besoin d'accroître l'autonomie de ces systèmes, pour qu'ils puissent évoluer en environnements incertains, qui a permis de manière concomitante de rendre leurs raisonnements plus accessibles, détaillés et transparents.

Pour aller plus loin dans le volet éthique de ces modèles et dépasser la « simple » explicabilité, nous pensons qu'il est possible d'adjoindre au système un cadre contraignant similaire à celui développé pour les *rightfull machines* que nous avons étudiées dans le chapitre I.2 de ce document : les *métarègles* (Randall, 1980) ont été introduites il y a plus de quarante ans pour améliorer la recherche d'information dans des systèmes à base de règles en guidant le système par suggestion de règles à appliquer (possiblement de manière floue) selon un contexte donné, ainsi qu'illustré sur la figure 6.1. En couplant l'approche par métarègles à un système capable d'anticiper les effets de ses actions, il semble donc possible de mettre en œuvre un *conséquentialisme de la règle*, représentant une forme de compromis entre le déontologisme (la détermination des actions selon un code) et l'utilitarisme (la détermination des actions selon leurs conséquences).

Nous pensons donc qu'une approche pour créer des systèmes d'IA plus autonomes, plus explicables et donc plus éthiques est de passer par le couplage d'une approche

```
If  [1] the age of the client is greater than 60,
    [2] there are rules which mention high risk,
    [3] there are rules which mention low risk,

then it is very likely (.8) that the former should be used after
the latter.
```

FIGURE 6.1 : Exemple de l'utilisation de métarègles pour évaluer la validité de règles dans un contexte de conseil d'investissements financiers. Figure extraite de (Randall, 1980).

[1] On rappelle que le manipulationnisme permet de comprendre comment les sorties d'un modèle sont affectées lorsque des variables d'entrées sont modifiées.

profonde, capable d'extraire des faits de données complexes, à un modèle comme un système de classeurs à anticipation. Cette perspective nous a déjà permis de commencer à poser des briques dans des travaux en cours, ainsi que d'envisager d'autres sujets à explorer à court et moyen terme.

3 Préparer la suite

3.1 Vers une meilleure explicabilité *post-hoc*

Si l'explicabilité *post-hoc* est critiquable, comme nous l'avons vu, nous avons aussi cherché à dépasser ces critiques pour rendre ce type d'approche plus pertinente. Il est de toute façon évident que ces méthodes *post-hoc* sont très répandues actuellement et plutôt que de les rejeter en bloc, il est plus intéressant de s'interroger sur la manière dont nous pouvons les exploiter.

Les premières expériences que nous avons pu faire avec TL-CAM et QXP (présentés au chapitre II.5) tendent à montrer qu'il est possible de rendre une explication plus parcimonieuse, mais aussi de quantifier la contribution des caractéristiques extraites par le modèle dans ses prédictions plutôt que de simplement les mettre en avant.

3.2 Jauger et exploiter l'incertitude des réseaux de neurones profonds

Les questions liées à l'incertitude nous intéressent également. Nous avons montré dans le chapitre II.4 qu'il est possible de détecter des formes d'incertitude environnementale. Une autre approche à la gestion de l'incertitude est d'évaluer celle des modèles en eux-mêmes, en particulier celle des modèles profonds. En effet, conceptuellement, ce type de modèle fait nécessairement une prédiction, même s'il examine des données en dehors de la distribution sur laquelle il a réalisé son apprentissage. Ce comportement n'est pas souhaitable et une manière de le prévenir est de permettre au modèle de dire qu'*il ne sait pas* (ou qu'il est trop incertain).

Quentin Christoffel, un étudiant dont j'encadre actuellement la thèse et qui démarre, en ce mois de septembre 2022, sa deuxième année, travaille sur cette question. Il a mis en place un modèle de réseaux de neurones bayésien pour accepter ou non des prédictions du modèle : n'apprenant plus seulement des paramètres, mais des distributions de paramètres, une prédiction est considérée comme incertaine si la

médiane de la distribution des prédictions est en deçà d'un seuil fixé empiriquement. Expérimentalement, nous avons observé qu'un modèle entraîné sur le jeu de données MNIST (LeCun, 1998) (des images représentant des chiffres manuscrits) est bel et bien capable de ne plus se « forcer » à faire des prédictions lorsque des images contenant du bruit lui sont données en entrées. Lorsque des images contenant des lettres manuscrites lui sont données, l'incertitude est moins forte, du fait que chiffres et lettres partagent des caractéristiques communes que le modèle reconnaît.

Nous nous attelons aujourd'hui à exploiter cette incertitude pour détecter de la nouveauté et déterminer sa nature à gros grains. Notre hypothèse est que si le modèle exhibe une incertitude modérée, mais qu'il n'est pas suffisamment certain non plus pour pouvoir faire une prédiction, alors les données examinées partagent possiblement des caractéristiques avec celles que le modèle connaît de son entraînement ; autrement dit les distributions des données d'entraînement et des données utilisées en inférence se chevauchent. Nous travaillons à construire une représentation hiérarchique des caractéristiques connues du modèle, de façon à pouvoir déterminer à quelle classe une nouveauté pourrait appartenir selon les caractéristiques connues ou non que cette nouveauté présente.

3.3 Coupler un ALCS à un réseau neuronal profond

Le développement d'une approche neuro-symbolique sous un angle *perception-raison*, dans un modèle couplé que nous baptisé *DeepExpert* (Orhand et al., 2019), est évidemment l'un de nos objectifs majeurs à court et moyen terme. Deux des verrous principaux que nous avons identifiés concernent, d'une part, la nature du couplage et d'autre part, le protocole d'évaluation du modèle.

Coupler des modèles aussi différents qu'un réseau de neurones profonds et un ALCS demande de mettre en place une représentation qui soit compatible au niveau symbolique comme au niveau subsymbolique et se qui doit d'être interprétable pour pouvoir être présentée à un utilisateur. Si nous avons choisi de ne pas aborder cette question dans le présent document, la représentation utilisée dans un système de classeurs demande à être examinée avec soin puisqu'il faut proposer un encodage adéquat, représentatif et fidèle, des données fournies au système.

De plus, comme nous l'avons vu, ces représentations peuvent également être associées avec des probabilités dans les effets des classeurs. Il serait alors intéressant d'étendre cette association de probabilités aux autres composantes, la condition

et l'action (dans ce dernier cas, en particulier lorsque l'on détermine une séquence d'action) et de revoir les mécanismes d'apprentissage pour que ces probabilités soient prises en compte pour la construction des classeurs ; une voie assez naturelle à suivre ici serait d'exploiter le théorème de Bayes.

Nous souhaitons aussi nous intéresser au protocole d'évaluation d'un tel modèle. Nous avons ici la conjonction de deux modèles, dont l'un est basé sur une approche non intrinsèquement explicable. Comme nous le disions plus haut, la question d'expliquer une perception se pose. Si l'on souhaite vraiment « expliquer » une perception, une option est d'aller vers des approches comme TL-CAM, mais il est aussi possible d'accepter la nature heuristique et non optimale de la perception.

Un modèle de raisonnement tel qu'un ALCS est considéré comme intrinsèquement explicable, mais l'est-il vraiment ? Nous avons évoqué le besoin de limiter le nombre de règles produites par le système, mais ce sont des évaluations par des utilisateurs qui pourront réellement permettre de juger de l'efficacité de cette réduction. Nous avions également évoqué l'application d'un principe de subsidiarité : là aussi il nous semble intéressant de, par exemple, disposer d'un ensemble de profils d'utilisateurs qui pourraient permettre une extraction de règles adaptées à l'objectif de l'explication, pourquoi pas au moyen de métarègles qui sont, rappelons-le, conçues pour permettre l'extraction de règles dans des systèmes à base de règles. D'autre part, la syntaxe des règles produites n'est pas forcément accessible à tout un chacun, en tout cas pas sans s'être au préalable familiarisé avec. Pour obtenir une accessibilité la plus large possible, nous pensons qu'il peut être intéressant d'ajouter une interface présentant les règles si ce n'est en langage naturel, au moins dans une syntaxe moins austère que `######Mur#  →→  {##Sortie###Chemin#:1}`

Pour développer efficacement cette approche qui va au-delà des aspects techniques et pour proposer un système performant et capable de proposer des explications valides et claires, dépassant un cadre descriptif, nous pouvons encore monter en compétences. Des articles de synthèse englobant tout ce processus d'explicabilité, tel que (Vilone and Longo, 2021), seront sans aucun doute des travaux à étudier attentivement pour remplir nos objectifs.

Les travaux de disciplines connexes comme la philosophie des sciences ou la sociologie seront aussi des sources précieuses pour mieux appréhender les questions épistémologiques autour des intelligences artificielles.

Troisième partie

Synthèse des activités d'enseignement et de recherche

CHAPITRE 7

Activités d'enseignement et de recherche

Ce chapitre est une synthèse des activités d'enseignement et de recherche que j'ai menées depuis mon recrutement à l'université de Strasbourg en 2016.

1 Parcours

1.1 Parcours professionnel

Sept. 2016– **Maîtresse de conférences** : Université de Strasbourg, département informatique de l'UFR de mathématique et d'informatique – Laboratoire ICube UMR 7357 (laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie), équipe CSTB (*Complex Systems and Translational Bioinformatics*)
Autonomie, explicabilité et éthique de l'intelligence artificielle

2015– 2016 **Chercheuse postdoctorale** : Université d'État de New York, département de mathématiques appliquées et de statistiques, *Stony Brook NY, 11794*
Immunologie computationnelle : processus mutationnels de séquences d'immunoglobulines, analyses de séquences ADN/ARN issues de séquençage haut débit

2014– 2015 **Enseignante-chercheuse contractuelle** : Université de Brest (29)
Enseignement de l'informatique aux étudiants de licence et master (172 HETD)

2011– **Doctorante avec enseignements** : Université de Brest (29)
2014 Enseignement de l'informatique aux étudiants de licence et master (contrat doctoral, 64 HETD annuelles)

Février– **Stage de M2 recherche** : Centre Européen de Réalité Virtuelle, *Plouzané*
juin (29)
2011 Couplage de systèmes dynamiques pour l'émergence de comportement d'agents réactifs en environnement virtuel

1.2 Diplômes et grades

2014 **Doctorat en informatique, section 27**, *Université de Brest (29)*
Lab-STICC UMR 6285 : « Développement d'un modèle logiciel de cellule sur processeurs multicœurs pour la simulation de morphogenèse de tissus », mention : très honorable. Sous la direction de Pascal Ballet (MCF HDR) et Vincent Rodin (PR)
Thèse soutenue à l'Université de Bretagne Occidentale le 21 novembre 2014, devant le jury composé de :
- Dominique Lavenier (Directeur de Recherche CNRS / *président)*,
- Angélique Stéphanou (Chargée de Recherche CNRS, HDR / *rapportrice)*,
- Yves Duthen (Professeur des universités / *rapporteur)*,
- Marie Beurton-Aimar (Maîtresse de conférences, HDR / *examinatrice)*,
- Pascal Ballet (Maître de conférences, HDR / *examinateur)*,
- Vincent Rodin (Professeur des universités / *examinateur)*

2011 **Master recherche en informatique**, *Université de Brest (29).*
Mention assez bien.
Mémoire : « Couplage de systèmes dynamiques pour l'émergence de comportement en environnement virtuel : application au rebond de balle », sous la direction de Cédric Buche (PR) et Pierre De Loor (PR)

2009 **Licence en informatique**, *Université de Brest (29).*
Mémoire : « Étude et évolution d'une infrastructure informatique de PME »

2 Activités pédagogiques

2.1 Enseignements

Je suis rattachée à l'UFR de mathématique et d'informatique de l'université de Strasbourg où je réalise l'essentiel de mon service. J'interviens ou suis intervenue à différents niveaux de la licence informatique dans des UE d'introduction à la programmation (impérative et web) ou dans des UE plus spécialisées, comme en intelligence artificielle ou en théorie des graphes. Je donne, en master informatique, parcours *Sciences des Données et Systèmes Complexes*, un cours d'Intelligence Collective et Apprentissage Profond que j'ai monté. Entre 2017 et 2021, je suis intervenue dans des enseignements à l'université Franco-Azerbaidjanaise (UFAZ) située à Bakou, Azerbaijan. J'y ai enseigné la programmation orientée objet en L1 et l'intelligence artificielle en L3 et M1. Les volumes totaux de mes services annuels d'enseignement depuis 2016 sont les suivants (hors enseignements à l'UFAZ) :

2016-2017 : 98 HETD (décharge d'un demi-service accordée lors de la première année d'exercice)

2017-2018 : 246 HETD

2018-2019 : 219 HETD

2019-2020 : 233 HETD

2020-2021 : 206 HETD

2021-2022 : 211 HETD

2022-2023 : je n'ai aucune charge d'enseignement cette année, car j'ai obtenu deux semestres de CRCT : l'un est un semestre obtenu au niveau national, par le CNU 27. Le second semestre m'a été accordé par l'université. L'ensemble des travaux que nous avons menés et les nombreuses discussions autour des questions épistémologiques et éthiques —plus largement, la question de l'intégration de l'intelligence artificielle dans nos sociétés— qui nous intéressent m'ont amené à souhaiter pouvoir disposer d'un bagage mieux structuré en sciences humaines. J'ai donc soumis une candidature, qui a été acceptée, en Master 2 Épistémologie et Histoire des Sciences, mention Sciences et Société, de l'université de Strasbourg. J'ai commencé à suivre les cours de cette formation le 8 septembre dernier.

Le tableau figure 7.1 est une synthèse de mes services d'enseignements depuis 2016.

	Module	CM	CI	TD	TP	Effectifs	2016/2017	2017/2018	2018/2019	2019/2020	2020/2021	2021/2022
Admin	Responsable pédagogique L3 informatique			50		100-130					R	R
	Responsable pédagogique Licence Informatique UFAZ					120	R	R	R	R		
L1	Algorithmique et Programmation2 / A		38			90		R	R	R		
					22							
	Algorithmique et Programmation2 / P		19			180						
	Introduction à la programmation web / A		12		14	45	R	R		R		
	Introduction à la programmation web / P		12		14	130-170	R	R	R	R	R	R
	Object Oriented Programming (UFAZ)	10		10	10	40		R	R			
L2	Anglais pour l'informatique / A			4		60						
	Anglais pour l'informatique / P			4		60						
L3	Anglais pour l'informatique			4		70						
	Graphes			14		35						
	Intelligence Artificielle		10		20	70-115				R	R	R
	Artificial Intelligence (UFAZ)	10			8	40			R	R		
M1	Programmation Orientée Objet avec Java *	10		10	10	30			R			
	Intelligence Artificielle & Apprentissage profond		16		14	20			R	R	R	R
	Collective and Artificial Intelligence (UFAZ)		16		14	20					R	

A : semestre d'automne
P : semestre de printemps
R : responsable de l'UE

* Cours donné en M1 BSIB (Biologie Structurale Intégrative et Bio-informatique)

Figure 7.1 : Synthèse de mes activités d'enseignement depuis 2016

2.2 Responsabilités pédagogiques

Responsable du parcours informatique de la licence Franco-Azerbaidjanaise (septembre 2017 – décembre 2020) L'université Franco-Azerbaidjanaise (UFAZ) est située à Bakou, Azerbaijan. Il s'agit d'un projet co-porté par l'université de Strasbourg et l'université d'État du pétrole et de l'industrie (ASOIU) à Bakou et démarré en septembre 2016. Les étudiants en sortent avec un double diplôme, de l'ASOIU et de l'université de Strasbourg (ou de l'université de Rennes 1 pour la filière Pétrole). La licence est un programme en 4 ans (qui comprend une année 0, dite « de fondation », pour remettre à niveau les étudiants recrutés qui, en Azerbaijan, finissent le lycée un an avant les étudiants français). À partir de la L1, la licence compte quatre parcours : génie chimique, génie géophysique, génie gazier et pétrolier (en partenariat avec l'université de Rennes 1) et informatique. 40 étudiants sont recrutés sur concours pour chacun des parcours tous les ans, pour un total de 160 étudiants par parcours sur les 4 années du diplôme.

Responsable de la L3 informatique (septembre 2020 – août 2022[1]) La promotion de L3 informatique comporte environ 130 étudiants. Les missions vont de la gestion des inscriptions pédagogiques et de l'élaboration des groupes de TD et TP, à la préparation et la tenue des jurys de semestres et à l'examen des dossiers de candidature à la formation durant le printemps.

J'ai également été membre élue du conseil d'UFR de décembre 2017 à décembre 2021.

2.3 Encadrement de Travaux d'Études et de Recherche (TER)

— Nassime Mountasir [2020-21] Encadrement 100% — *Mise en place de mesure d'incertitudes pour apprendre à un réseau de neurones profond à répondre "je ne sais pas"*

— Elias Chetouane [2020-21] Encadrement 40% avec S. Marc-Zwecker — *Sémantique pour l'explicabilité des réseaux de neurones profonds*

— Freddy Nawfal [2019-20] Encadrement 100% — *Étude comparative des méthodes d'apprentissage par transfert*

— Quentin Christoffel [2019-20] Encadrement 80% avec R. Orhand — *Expliquer les prédictions de réseaux de neurones convolutifs par applications de masques générés par évolution artificielle*

— Elisa Kalbé [2018-19] Encadrement 100% — *Détection de collisions en environnement virtuel : comparaison d'approches cognitive et géométrique*

— Gaël Mukunde [2018-19] Encadrement 50% avec O. Poch et L. Moulinier — *Apprentissage et identification de propriétés au sein de séquences de protéines*

— Matthieu Haller [2018-19] Encadrement 50% avec C. Mayer — *Apprentissage profond pour la classification de repliements de protéines*

— Amarin Hutt [2018-19] Encadrement 80% avec R. Orhand — *Apprentissage par transfert, apprentissage multi-tâches : adaptabilité des réseaux de neurones*

— Elisa Kalbé [2017-18] Encadrement 100% — Approches non représentationnelles pour la modélisation de comportements d'agents virtuels

[1] Il est prévu que je reprenne cette responsabilité à la rentrée 2023 suite à mon CRCT.

— Maxime Seyer [2017-18] Encadrement 100% — *Approches non représentationnelles pour la modélisation de comportements d'agents virtuels*

3 Activités de recherche

Je suis rattachée, du point de vue de la recherche, à l'équipe CSTB (*Complex Systems and Translational Bioinformatics*) du laboratoire ICube UMR7357 (laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie). L'équipe CSTB travaille sur deux thématiques principales : « génomique évolutive et médicale » (extraction de connaissances depuis des données omiques pour caractériser les relations génotypes-phénotypes, compréhension de l'évolution des systèmes biologiques) et « intelligence artificielle de confiance » (conception d'intelligences artificielles explicables, éthiques et autonomes). Le défi de notre équipe est de mener une recherche cohérente et complémentaire intégrant les singularités de chaque discipline dans des projets transverses.

J'ai opéré un virage radical dans mes thématiques de recherche lors de mon arrivée dans l'équipe en septembre 2016, passant de travaux sur des thématiques de biologie computationnelle et de bio-informatique à l'intelligence artificielle, en collaboration avec Pierre Collet. Les différents travaux que nous avons menés depuis 2016 ont gravité autour des questions d'autonomie et d'explicabilité des intelligences artificielles *via* un prisme épistémologique.

Je suis, depuis janvier 2022, responsable de la thématique « intelligence artificielle de confiance » de l'équipe. Notre objectif est de combiner des aspects fondamentaux et appliqués dans une approche interdisciplinaire comptant l'informatique et les mathématiques, mais aussi des sciences humaines comme la philosophie et la sociologie. Nos méthodes vont du raisonnement à l'ingénierie des connaissances en passant par la théorie des graphes et le calcul évolutionnaire. Les applications développées par les membres de la thématique touchent à la protéomique et la génomique, en lien avec la thématique « génomique évolutive et médicale », au traitement d'image, l'industrie 4.0 et la cybersécurité, l'optimisation stochastique et l'apprentissage par programmation génétique et l'éducation.

Par ailleurs, nous avons monté un petit groupe de travail au sein de la thématique « IA de confiance » pour nous interroger sur l'élaboration d'une formalisation mathématique de l'éthique. L'idée est de formaliser des règles d'éthiques en utilisant des clauses de Horn que l'on pourrait appliquer comme des métarègles sur un système.

3.1 Co-encadrements de thèses

— Quentin Christoffel [Oct. 2021 –] Encadrement 70%
Directrice : Aline Deruyver
Réduction des incertitudes de modèles d'apprentissage automatique par introduction de connaissances
Le travail de M. Quentin Christoffel traite de l'évaluation et la réduction des incertitudes de modèles d'apprentissage profond. Évaluer la certitude d'un modèle vise à permettre à celui-ci de ne pas réaliser de prédiction à tout prix lorsque des données en dehors de la distribution des données utilisées pour entraîner ce modèle lui sont passées en entrée. Si l'incertitude du modèle est forte, il est raisonnable de penser que la donnée ayant conduit à la prédiction incertaine est totalement hors distribution. Il est plus intéressant d'étudier les cas où le modèle a une incertitude relative, sans pour autant être suffisamment certain pour faire une prédiction : dans ce cas, notre hypothèse est que cette donnée partage des traits avec les traits appris par le modèle et qu'une identification de cette donnée soit possible à gros grains. Nous travaillons actuellement à l'extraction de ces traits et de ceux appris par un réseau de neurones lors de leur apprentissage.

— Hiba Khodji [Oct. 2020 –] Encadrement 60%
Directeurs : Pierre Collet et Julie Thompson
Characterization of the transferability of deep neural network latent features. Application to the detection of gene prediction error
Mme Hiba Khodji travaille sur la prédiction d'erreurs de prédictions dans des séquences de gènes dans des alignements multiples de séquences par réseaux de neurones convolutifs ainsi que sur la quantification de l'explicabilité *post-hoc* de ces modèles, comme nous l'avons présenté dans le chapitre II.5. Une autre approche développée dans le cadre de ce travail de thèse pour détecter ces erreurs est d'utiliser un réseau de neurones détecteur d'objets : cette approche permet en particulier de caractériser les types des erreurs présentes dans l'alignement. Nous avons dans ce cadre proposé le modèle MERLIN (*Msa ERror Localization and IdentificatioN*) (Khodji et al., 2022).

— Nicolas Scalzitti [dec. 2018 – sept. 2021] Encadrement 30% – Soutenue le 29 septembre 2021

Directeurs : Julie Thompson et Pierre Collet
New gene annotation and detection strategies for genome sequencing projects, using artificial intelligence
M. Nicolas Scalzitti a travaillé, durant sa thèse, à l'amélioration des méthodes d'annotation des génomes eucaryotes et plus particulièrement sur la prédiction des sites d'épissages, des sites de démarcation entre introns et exons dans des séquences de nucléotides. Trois contributions principales sont sorties de ces travaux : (a) l'élaboration d'un jeu de données de haute qualité, G3PO (Scalzitti et al., 2020), (b) un modèle de réseau de neurones profond pour la prédiction des sites d'épissage ()scalzittiSpliceatorMultispeciesSplice2021a et (c) une approche basée sur la programmation génétique pour, là aussi, caractériser les sites d'épissage d'une séquence.
Depuis septembre 2022, Nicolas est chercheur postdoctoral dans le laboratoire du professeur Wolfgang Banzhaf à l'université d'état du Michigan, États-Unis.

— Romain Orhand [oct. 2018 – nov. 2022] Encadrement 75% – Soutenue le 10 novembre 2022
Directeurs : Pierre Collet et Pierre Parrend
Towards autonomous computers by combination of artificial intelligence and artificial evolution
Comme nous l'avons présenté dans le chapitre II.4, M. Romain Orhand à travaillé à la conception d'un modèle d'intelligence artificielle autonome et explicable capable d'évoluer en environnements incertains. Basé sur un système de classeurs à anticipation, le modèle BEACS (*Behavioral and Enhanced Anticipatory Classifier System*) (Orhand et al., 2022) issu de ce travail, couple des séquences comportementales (permettant au système de réaliser plusieurs actions successives) pour gérer les situations environnementales non déterministes, ainsi que des prédictions améliorées par l'expérience lui permettant de construire des représentations complètes et exactes d'environnements, même si ces derniers sont incertains. Les mécanismes de BEACS en font un système de classeurs non seulement plus autonome, mais aussi plus explicable, car ses représentations environnementales permettent d'appréhender avec précisions des relations de cause à effet conduisant aux différentes situations de l'environnement.

— Anna Ouskova-Leonteva [août 2018 – mai 2022] Encadrement 30% – Soutenue le 24 mai 2022
Directeurs : Pierre Collet et Pierre Parrend
Evolutionary and quantum inspired algorithms for the optimization of magnetic cooling systems
Le travail de thèse de Mme Anna Ouskova-Leonteva a traité de l'optimisation, par approches évolutionnaires, de dispositifs de réfrigération magnétocalorique. Le modèle FastEMO (Leonteva et al., 2020), présenté dans le chapitre II.3 de ce document, a été conçu pour résoudre le problème de l'optimisation d'un régénérateur actif magnétique (AMR), un problème multiobjectif dont certains objectifs sont de surcroît antagonistes. Un algorithme inspiré par le quantique intégrant des stratégies d'évolution (Ouskova Leonteva et al., 2020) a été proposé pour déterminer les propriétés optimales du matériel magnétocalorique utilisé dans l'AMR.

3.2 Encadrements et co-encadrements de master 2

— Léo Wehrli [février 2022 - août 2022] Co-encadrement 30% avec S. Marc-Zwecker et D. Bernhard —*Explicabilités de réseaux de neurones profonds par approches neuro-symboliques*
M. Léo Wehrli a, durant son stage de fin d'études, étudié différentes approches d'explicabilité de réseaux de neurones par des approches symboliques et s'est focalisé en particulier sur deux approches : les *Logic Explained Networks* (LEN) (Ciravegna et al., 2021) qui permettent en théorie d'expliquer les prédictions de réseaux de neurones par la génération de règles logiques et TREPAN (Craven and Shavlik, 1995; Confalonieri et al., 2020), une approche visant à générer un arbre de décision représentatif du comportement d'un réseau de neurones. Le travail de M. Werhlir a mis en lumière la difficulté d'expliquer des réseaux de neurones par des approches *post-hoc*. Dans le cas des LEN, les règles générées restent trop triviales pour représenter le comportement du modèle ; TREPAN, en l'état, est limité dans les architectures de réseaux de neurones qu'il peut traiter et de futurs travaux sont donc nécessaires pour étendre les possibilités d'application de cette approche.

— Quentin Christoffel [15 février 2021 - 15 août 2021] Encadrement 100% — *Implementation of uncertainty measures in deep neural networks*

L'objectif du travail de fin d'études de M. Quentin Christoffel était d'étudier et mettre en place une mesure de l'incertitude des prédictions réalisées par des réseaux de neurones profonds afin de permettre à un modèle de répondre qu'il ne sait pas, plutôt que de chercher à classer à tout prix une donnée même si celle-ci est en dehors de la distribution des données utilisées pour entraîner le modèle. Les réseaux de neurones bayésiens permettent de répondre à ce problème, en apprenant non plus des paramètres, mais des distributions de paramètres. Une prédiction est acceptée si la médiane des distributions de résultats est au-delà d'un certain seuil et rejetée sinon.

— Hiba Khodji [1 mars 2020 - 15 août 2020] Encadrement 100% — *QUANTA : QUANtitative measure of latent feature TrAnsferability in deep neural networks*
À l'occasion de son stage de fin d'études, Mme Hiba Khodji a travaillé sur une problématique d'apprentissage par transfert. L'apprentissage par transfert permet de réutiliser la connaissance apprise par un modèle source pour résoudre une tâche donnée, dans un modèle cible devant résoudre une tâche proche. Dans les réseaux de neurones, un transfert se réalise le plus souvent en réutilisant les paramètres des n premières couches du modèle source. Il a été observé, empiriquement, que les premières couches des réseaux de neurones profonds sont les plus transférables, car elle apprennent des caractéristiques plus générales que les couches plus proches des sorties qui deviennent de plus en plus spécifiques à la tâche à résoudre. Nous avons proposé une approche quantitative de la transférabilité pour mieux comprendre, dans la résolution de différentes tâches, comment le transfert influence l'apprentissage du modèle cible (Orhand, Khodji, Hutt and Jeannin-Girardon, 2021).

— Nesrine Banour [1 mars 2018 - 31 juillet 2018] Co-encadrement 20% avec E. Schneider et N. Lachiche — *Résistance au bruit et à la rareté de la détection d'anomalies par arbre de décisions de systèmes physiques simulés*
Mme Nesrine Bannour a travaillé, lors de son stage de fin d'études, sur la détection d'anomalies dans des données synthétiques de séries temporelles par l'utilisation d'arbres de décision. L'idée de ce travail était de réaliser une classification binaire normal / anormal déterminée par le critère de séparation calculé par l'arbre de décision.

— Nestor Demeure [1 février 2017 - 31 juillet 2017] Encadrement 100% — *Réseaux de régulation génétique de lymphocytes B*
Le travail de M. Nestor Demeure a porté sur l'intégration d'un modèle de régulation génétique à base de fonctions de Hill (Santillán, 2008) à un modèle de lymphocyte B pour simuler les cascades de régulation de ces cellules.

3.3 Co-encadrements d'étudiants en alternance (recherche)

— Alexandre Bruyant [sept. 2016 - août 2018] Maître d'apprentissage, encadrement 50% avec P. Parrend — *Gestion de fichiers informatisés avec RADAR (Robust Anonymous DAta Record)*
Lors de son apprentissage, M. Alexandre Bruyant a travaillé sur le système RADAR (*Robust Anonymous DAta Records*), un système de stockage distribué et sécurisé fondé sur le paradigme des systèmes complexes. Un réseau RADAR est constitué d'un ensemble de machines faiblement interconnectées et dont la mise en réseau est supposée peu fiable où les données à stocker sont fragmentées et stockées de manière redondante sur n machines choisies aléatoirement dans le réseau.

— Marc Haegelin [sept. 2016 - août 2018] Maître d'apprentissage, encadrement 60% avec C. Rolando (DR CNRS Univ. Lille) — *Traitement massivement parallèle du signal d'un spectromètre de masses FT-ICR*
M. Marc Haegelin a travaillé, durant son apprentissage, sur l'analyse de données de spectrométrie de masse et plus précisément sur la mise en œuvre sur GPU de l'algorithme de débruitage urQRd (Chiron et al., 2014). Marc a par la suite été embauché par l'équipe de C. Rolando en tant qu'Ingénieur d'Études dans le laboratoire MSAP et est maintenant en thèse au MSAP sous la direction de C. Rolando.

— Anastasia Villien [sept. 2019 - août 2020] Encadrement à 40% avec P. Collet — *Elaboration d'un écosystème intelligent pour une éducation de masse personnalisée*
Mme Anastasia Villien a réalisé son apprentissage dans le cadre de la plateforme POEM (*Personalized Open Education for the Masses*), visant à offrir une personnalisation de masse de contenus pédagogiques grâce à la mise en œuvre d'hommilières (Valgiani et al., 2007) pour déterminer des trajectoires pédago-

giques optimales pour les utilisateurs et utilisant une évaluation collaborative entre pairs, COPA (*Cooperative Open Peer Assesment*) Collet et al. (2017). Anastasia est maintenant ingénieure pédagogique à l'Université de Haute Alsace à Colmar.

3.4 Implication dans des projets

ANR THIA ArtIC (intelligence artificielle pour la santé) 900 k€ / 2020–2025. Demi-financement de la thèse de Mme Hiba Khodji.

ANR PRCE CoolMagEvo 687 k€ / 2018–2021 / porté par UbiBlue (`https://ubiblue.com/`). Participation au *workpackage* 4 : "optimisation bio-inspirée du modèle multi-physique". Financement de la thèse de Mme Anna Ouskova-Leonteva.

ANR Dune Eole 2.2 M€ / 2017–2021 / porté par l'université de Strasbourg. Participation à l'action 2 de l'axe 2 ; financement de l'apprentissage de Mme Anastasia Villien.

API ICube GEM (*Germinal Center Modelling*) 9.8 k€ 2017–2018, porté par Anne Jeannin-Girardon.

API ICube *Cartes concentriques multi-valuées* 9.6 K€ 2017–2018, porté par Basile Sauvage.

API ICube ACE-game (Apprentissage de Comportement en Environnement contrôlé) 9.3 k€ 2018–2019, porté par Nicolas Lachiche.

API ICube DEEPISH (*Deep lEarning ExPlainabilIty through Symbolic approacHes*) 6 k€ 2021-2023, porté par Stella Marc-Zwecker.

3.5 Diffusion scientifique

— Juin 2022 — "Intelligence artificielle : de la technique aux enjeux sociaux," Intervention la journée doctorale de l'université de Haute Alsace à Mulhouse.

— Fév. 2020 — "Quelle intégration des systèmes intelligents dans nos sociétés ?" Intervention à la table ronde *Intelligence artificielle et démocratie : vers quelles interactions et enjeux ?* du Jardin des Sciences de l'université de Strasbourg.

— Nov. 2019 — "Transfer learning : review and recent advances," workshop de l'axe de recherche *Data Science and Artificial Intelligence* du laboratoire ICube.

— Oct. 2018 — "Défis théoriques et pratiques de l'intelligence artificielle," Présentation aux *Journées Systèmes 2018* à Strasbourg.

— 2016 et 2017 — Animation (avec Pierre Collet) du stand du Campus Numérique des Systèmes Complexes de la Fête de la Science.

3.6 Responsabilités collectives

Membre du comité d'expert de la section 27 de l'UFR de mathématique et d'informatique depuis septembre 2016.

Participation à des comités de sélection de MCF 27 (IUT Robert Schumann en 2017 et 2021 ; UFR de mathématique et d'informatique en 2020, Télécom Physique Strasbourg en 2019, sur une chaire industrielle en Sciences des Données et Intelligence Artificielle,)

Présidence d'un comité de sélection de MCF 27 pour l'UFR de mathématique et d'informatique sur un double profil réseaux et/ou sciences des données.

3.7 Autres responsabilités scientifiques

Révision d'articles

— IEEE Transactions on Artificial Intelligence (2022) ;

— IEEE Transactions on Neural Networks and Learning Systems (2020, 2021) ;

— International Conference on Engineering Applications and Advances of Artificial Intelligence (2022) ;

— AIMS Cell and Tissue Engineering (2017) ;

— International Conference on Artificial Evolution (2017, 2022) ;

— International Conference on Artificial Life and Robotics (2018).

Comité d'organisation International Conference on Artificial Evolution (2017) / Gestion de la plateforme de soumission des articles.

Expertise scientifique Dossiers de thèses CIFRE (2017, 2018) ; pôle de compétitivité BioWin (Wallonie).

3.8 Publications depuis 2016

Journaux internationaux avec comité de lecture	5
Conférences internationales avec comité de lecture	13
Chapitre d'ouvrage	1
Communications nationales	3

Ouvrages scientifiques ou participations à des ouvrages

[1] C. Zanni-Merk, and A. Jeannin-Girardon, "Towards the Joint Use of Symbolic and Connectionist Approaches for Explainable Artificial Intelligence," *Advances in Selected Artificial Intelligence Areas*, Maria Virvou, George A. Tsihrintzis, Lakhmi C. Jain (Eds.), Springer, Cham, Learning and Analytics in Intelligent Systems, 2022.

Articles dans des revues internationales avec comité de lecture

[2] N. Scalzitti, A. Kress, R. Orhand, T. Weber, L. Moulinier, A. Jeannin-Girardon, P. Collet, O. Poch, and J. D. Thompson, "Spliceator : Multi-species splice site prediction using convolutional neural networks," *BMC Bioinformatics*, 2021.

[3] C. Meyer, N. Scalzitti, A. Jeannin-Girardon, P. Collet, O. Poch, and J. Thompson, "Understanding the causes of errors in eukaryotic protein-coding gene prediction : A case study of primate proteomes," *BMC Bioinformatics*, vol. 21, no. 1, p. 513, 2020.

[4] N. Scalzitti, A. Jeannin-Girardon, P. Collet, O. Poch, and J. Thompson, "A benchmark study of ab initio gene prediction methods in diverse eukaryotic organisms," *BMC Genomics*, 2020.

[5] U. Abdulkarimova, A. Ouskova Leonteva, C. Rolando, A. Jeannin-Girardon, and P. Collet, "The PARSEC machine : A non-newtonian supra-linear supercomputer," *Azerbaijan Journal of High Performance Computing*, vol. 2, no. 2, 2019.

[6] A. Jeannin-Girardon, A. Bruyant, N. Toussaint, I. Diouf, P. Collet, and P. Parrend, "Management of digital records inspired by Complex Systems with RADAR," *Journal of Robotics Networking and Artificial Life* 5(1), June 2018.

Communications internationales avec actes

[7] H. Khodji, L. Herbay, P. Collet, J. Thompson and A. Jeannin-Girardon, "MERLIN : Identifying inaccuracies in Multiple Sequence Alignments using Object Detection," *International Conference on Artificial Intelligence Applications and Innovations*, Hersonissos, Greece, June 2022

[8] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "Accurate and Interpretable Representations of Environments with Anticipatory Learning Classifier Systems," European Conference on Genetic Programming (Part of EvoStar), Madrid, Spain, Springer, Jan. 2022.

[9] R. Orhand, H. Khodji, A. Hutt, and A. Jeannin-Girardon, "Quantification of the transferability of features between deep neural networks," in *25th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Sept. 2021.

[10] A. Ouskova Leonteva, R. Hamane, M. Risser, A. Jeannin-Girardon, P. Parrend, and P. Collet, "New evolutionary method for studying physical properties of magneto caloric materials," in *International Conference on the IEEE Congress on Evolutionary*, June 2021.

[11] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "Explainability and performance of anticipatory learning classifier systems in non-deterministic environments," in *Genetic and Evolutionary Computation Conference (GECCO)*, July 2021.

[12] A. Ouskova Leonteva, R. Hamane, M. Risser, A. Jeannin-Girardon, P. Parrend, and P. Collet, "Evolutionary optimization of Hamiltonian model for the study of magnetocaloric materials," in *THERMAG IX. 9th IIR International Conference on Caloric Cooling and Applications of Caloric Materials.*, June 2021.

[13] A. Ouskova Leonteva, U. Abdulkarimova, A. Jeannin-Girardon, P. Parrend, and P. Collet, "Quantum-inspired algorithm with evolution strategy," in *TPNC 2020 : 9th International Conference on the Theory and Practice of Natural Computing*, p. 12, Aug. 2020.

[14] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "BACS : A thorough study of using behavioral sequences in ACS2," in *PPSN*, Aug. 2020.

[15] A. Ouskova Leonteva, U. Abdulkarimova, T. Wintermantel, A. Jeannin-Girardon, P. Parrend, and P. Collet, "A quantum simulation algorithm for continuous optimization (poster)," in *GECCO 2020 : The Genetic and Evolutionary Computation Conference*, p. 2, Mar. 2020.

[16] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "PEPACS : Integrating probability-enhanced predictions to ACS2," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Jan. 2020.

[17] A. Ouskova Leonteva, P. Parrend, A. Jeannin-Girardon, and P. Collet, "Fast evolutionary algorithm for solving large-scale multi-objective problems," in *EA2019 Artificial Evolution*, Oct. 2019.

Conférences internationales avec comité de lecture, sans acte

[18] P. Parrend, A. Kress, E.-A. Sauleau, A. Jeannin-Girardon, F. Colin, J.-L. Mandel, and P. Collet, "Global data and local action for wellbeing science : When 4P-medicine meets health self-assessment," in *Conference on Complex Systems (CCS2018), Digital Epidemiology and Surveillance Satellite* (C. S. Society, ed.), Sept. 2018.

[19] A. Kress, E.-A. Sauleau, A. Jeannin-Girardon, F. Colin, J.-L. Mandel, P. Collet, and P. Parrend, "HOPE : Emergent solutions for analysis and management of patient cohort data," in *Conference on Complex Systems (CCS2018)* (C. S. Society, ed.), Jan. 2018.

Communications nationales ou francophones avec actes

[20] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "Les réseaux de neurones peuvent-ils être créatifs ?," in *Les Journées de Rochebrune 2020*, Jan. 2020.

[21] R. Orhand, A. Jeannin-Girardon, P. Parrend, and P. Collet, "DeepExpert : Vers une Intelligence Artificielle autonome et explicable," in *RJCIA*, July 2019.

[22] N. Bannour, A. Jeannin-Girardon, N. Lachiche, and E. Schneider, "Résistance au bruit et à la rareté de la détection d'anomalies par arbre de décision de systèmes physiques simulés," in *19ième Conférence Francophone Sur l'Extraction et La Gestion Des Connaissances (EGC)*, pp. 273–278, Jan. 2019.

Bibliographie

Alfares, I., Javaid, M. S., Chen, Z., Anderson, A., Antonic-Baker, A. and Kwan, P. (2021), 'Sex Differences in the Risk of Cutaneous Adverse Drug Reactions Induced by Antiseizure Medications : A Systematic Review and Meta-analysis', *CNS Drugs* **35**(2), 161–176.

Alves, G., Amblard, M., Bernier, F., Couceiro, M. and Napoli, A. (2021), Reducing Unintended Bias of ML Models on Tabular and Textual Data, *in* '2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)', pp. 1–10.

Assael, Y. M., Shillingford, B., Whiteson, S. and de Freitas, N. (2016), 'LipNet : End-to-End Sentence-level Lipreading'.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S. and Salomon, L. (2021), 'Performance indicators in multiobjective optimization', *European Journal of Operational Research* **292**(2), 397–422.

Badreddine, S., d'Avila Garcez, A., Serafini, L. and Spranger, M. (2022), 'Logic tensor networks', *Artificial Intelligence* **303**, 103649.

Bagnall, A. and Zatuchna, Z. (2005), On the Classification of Maze Problems, *in* L. Bull and T. Kovacs, eds, 'Foundations of Learning Classifier Systems', Studies in Fuzziness and Soft Computing, Springer, Berlin, Heidelberg, pp. 305–316.

Bordt, S., Finck, M., Raidl, E. and von Luxburg, U. (2022), Post-Hoc Explanations Fail to Achieve their Purpose in Adversarial Contexts, *in* '2022 ACM Conference on Fairness, Accountability, and Transparency', FAccT '22, Association for Computing Machinery, New York, NY, USA, pp. 891–905.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A.,

Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D. (2020), Language Models are Few-Shot Learners, *in* 'Advances in Neural Information Processing Systems', Vol. 33, Curran Associates, Inc., pp. 1877–1901.

Butz, M. V. (2002*a*), *Anticipatory Learning Classifier Systems*, Vol. 4 of *Genetic Algorithms and Evolutionary Computation*, Springer US, Boston, MA.

Butz, M. V. (2002*b*), Biasing Exploration in an Anticipatory Learning Classifier System, *in* P. L. Lanzi, W. Stolzmann and S. W. Wilson, eds, 'Advances in Learning Classifier Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 3–22.

Butz, M. V., Goldberg, D. E. and Stolzmann, W. (2001), Probability-Enhanced Predictions in the Anticipatory Classifier System, *in* P. Luca Lanzi, W. Stolzmann and S. W. Wilson, eds, 'Advances in Learning Classifier Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 37–51.

Butz, M. V. and Pelikan, M. (2001), Analyzing the evolutionary pressures in XCS, *in* 'Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation', GECCO'01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 935–942.

Butz, M. V. and Stolzmann, W. (2002), An Algorithmic Description of ACS2, *in* P. L. Lanzi, W. Stolzmann and S. W. Wilson, eds, 'Advances in Learning Classifier Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 211–229.

Celis, L. E., Kapoor, S., Salehi, F. and Vishnoi, N. K. (2019), 'Controlling Polarization in Personalization : An Algorithmic Framework', *Proceedings of the Conference on Fairness, Accountability, and Transparency* .

CEST (n.d.), 'Quelle est la différence entre éthique et morale ? - Commission de l'éthique en science et technologie'.

Chalmers, A. (2013), *What Is This Thing Called Science ?*, Univ. of Queensland Press.

Cheung, M., Shi, J., Wright, O., Jiang, L. Y., Liu, X. and Moura, J. M. F. (2020), 'Graph Signal Processing and Deep Learning : Convolution, Pooling, and Topology', *IEEE Signal Processing Magazine* **37**(6), 139–149.

Chiner-Oms, A. and González-Candelas, F. (2016), 'EvalMSA : A Program to Evaluate Multiple Sequence Alignments and Detect Outliers', *Evol Bioinform Online* **12**, EBO.S40583.

Chiron, L., van Agthoven, M. A., Kieffer, B., Rolando, C. and Delsuc, M.-A. (2014), 'Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry', *Proceedings of the National Academy of Sciences* **111**(4), 1385–1390.

Chowdhury, B. and Garai, G. (2017), 'A review on multiple sequence alignment from the perspective of genetic algorithm', *Genomics* **109**(5), 419–431.

Chui, M., Harryson, M., Manyika, J., Roberts, R., Chung, R., van Heteren, A. and Nel, P. (2018), 'Notes from the AI frontier : Applying AI for social good', *McKinsey Global Institute* .

Chung, H., Park, C., Kang, W. S. and Lee, J. (2021), 'Gender Bias in Artificial Intelligence : Severity Prediction at an Early Stage of COVID-19', *Frontiers in Physiology* **12**.

Chung, J., Senior, A., Vinyals, O. and Zisserman, A. (2017), Lip reading sentences in the wild, CVPR 2017, Institute of Electrical and Electronics Engineers.

Ciravegna, G., Barbiero, P., Giannini, F., Gori, M., Lió, P., Maggini, M. and Melacci, S. (2021), 'Logic Explained Networks', *arXiv :2108.05149 [cs]* .

Collet, P., Seereekissoon, R., Abotsi, I., Michaud-Maret, M., Scius-Bertrand, A., Tillich, E. and Parrend, P. (2017), POEM-COPA Collaborative Open Peer Assessment, *in* P. Bourgine, P. Collet and P. Parrend, eds, 'First Complex Systems Digital Campus World E-Conference 2015', Springer Proceedings in Complexity, Springer International Publishing, Cham, pp. 15–27.

Commision, E. (2020), 'White Paper on Artificial Intelligence : Public consultation towards a European approach for excellence and trust'.

Confalonieri, R., Coba, L., Wagner, B. and Besold, T. R. (2021), 'A historical perspective of explainable Artificial Intelligence', *WIREs Data Mining and Knowledge Discovery* **11**(1), e1391.

Confalonieri, R., Weyde, T., Besold, T. R. and Moscoso del Prado Martín, F. (2020), Trepan Reloaded : A Knowledge-driven Approach to Explaining Artificial Neural Networks, *in* '24th European Conference on Artificial Intelligence (ECAI 2020)', Vol. 325, IOS Press, Santiago de Compostela, Spain, pp. 2457–2464.

Cottraux, J. (2020), *Les Psychothérapies Cognitives et Comportementales*, Collection Médecine et Psychothérapie, seventh edn, Elsevier Masson.

Craven, M. and Shavlik, J. (1995), 'Extracting tree-structured representations of trained networks', *Advances in neural information processing systems* **8**.

Ćurković, M., Košec, A., Roje Bedeković, M. and Bedeković, V. (2021), 'Epistemic responsibilities in the COVID-19 pandemic : Is a digital infosphere a friend or a foe ?', *Journal of Biomedical Informatics* **115**, 103709.

Damasio, A. R. (2000), A neurobiology for consciousness, *in* 'Neural Correlates of Consciousness : Empirical and Conceptual Questions', The MIT Press, Cambridge, MA, US, pp. 111–120.

Dastin, J. (2018), 'Amazon scraps secret AI recruiting tool that showed bias against women', *Reuters* .

De Regt, H. W. and Dieks, D. (2005), 'A Contextual Approach to Scientific Understanding', *Synthese* **144**(1), 137–170.

Deb, K. and Jain, H. (2014), 'An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I : Solving Problems With Box Constraints', *IEEE Transactions on Evolutionary Computation* **18**(4), 577–601.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002*a*), 'A fast and elitist multiobjective genetic algorithm : NSGA-II', *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002*b*), 'A fast and elitist multiobjective genetic algorithm : NSGA-II', *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2002), Scalable multi-objective optimization test problems, *in* 'Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)', Vol. 1, pp. 825–830 vol.1.

Deb, K. and Tiwari, S. (2008), 'Omni-optimizer : A generic evolutionary algorithm for single and multi-objective optimization', *European Journal of Operational Research* **185**(3), 1062–1087.

Dennis, L., Fisher, M., Slavkovik, M. and Webster, M. (2016), 'Formal verification of ethical choices in autonomous systems', *Robotics and Autonomous Systems* **77**, 1–14.

Doerr, B., Le, H. P., Makhmara, R. and Nguyen, T. D. (2017), Fast genetic algorithms, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference', GECCO '17, Association for Computing Machinery, New York, NY, USA, pp. 777–784.

Doshi, T. (2018), 'Introducing the Inclusive Images Competition'.

Drake, J. (2018), *Introduction to Logic*, Scientific e-Resources.

Dworkin, G. (2015), 'The nature of autonomy', *Nordic Journal of Studies in Educational Policy* **2015**(2), 28479.

Ebrahimi, M., Eberhart, A., Bianchi, F. and Hitzler, P. (2021), 'Towards bridging the neuro-symbolic gap : Deep deductive reasoners', *Appl Intell* **51**(9), 6326–6348.

El Achkar, G., Bichat, B., Colasson, S., Dianoux, A., Mazet, T., Maillet, D., Feidt, M. and Kheiri, A. (2015), Etude expérimentale de l'effet magnétocalorique d'un régénérateur magnétique actif fonctionnant à température ambiante, *in* S. F. Thermique, ed., 'Congrès Français de Thermique SFT 2015', Société Française Thermique, La Rochelle, France.

equivant (2018), 'Response to ProPublica : Demonstrating accuracy equity and predictive parity'.

Etzioni, A. and Etzioni, O. (2017), 'Incorporating Ethics into Artificial Intelligence', *J Ethics* **21**(4), 403–418.

Européenne, U. (n.d.), 'Une approche européenne de l'intelligence artificielle | Bâtir l'avenir numérique de l'Europe'.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. and Song, D. (2018), Robust Physical-World Attacks on Deep Learning Visual Classification, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'.

Faraut, M. (2015), Apprendre à apprendre dans un environnement incertain, et dynamique des réseaux corticaux pour la flexibilité comportementale, PhD thesis, Université Claude Bernard - Lyon I.

Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P. and Vayena, E. (2018), 'AI4People—An Ethical Framework for a Good AI Society : Opportunities, Risks, Principles, and Recommendations', *Minds & Machines* **28**(4), 689–707.

Fürnkranz, J., Kliegr, T. and Paulheim, H. (2020), 'On cognitive preferences and the plausibility of rule-based models', *Mach Learn* **109**(4), 853–898.

Gardner, H. (1987), 'The theory of multiple intelligences', *Annals of Dyslexia* **37**(1), 19–35.

Gaur, M., Kursuncu, U., Sheth, A., Wickramarachchi, R. and Yadav, S. (2020), Knowledge-infused Deep Learning, *in* 'Proceedings of the 31st ACM Conference on Hypertext and Social Media', Association for Computing Machinery, New York, NY, USA, pp. 309–310.

Georgopoulos, M., Oldfield, J., Nicolaou, M. A., Panagakis, Y. and Pantic, M. (2021), 'Mitigating Demographic Bias in Facial Datasets with Style-Based Multi-attribute Transfer', *Int. J. Comput. Vision* **129**(7), 2288–2307.

Giancola, M., Bringsjord, S., Govindarajulu, N. S. and Varela, C. (2020), 'Ethical reasoning for autonomous agents under uncertainty', *Smart Living and Quality Health with Robots, Proceedings of ICRES* .

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., USA.

Gong, W., Duan, Q., Li, J., Wang, C., Di, Z., Ye, A., Miao, C. and Dai, Y. (2016), 'Multiobjective adaptive surrogate modeling-based optimization for parameter estimation of large, complex geophysical models', *Water Resources Research* **52**(3), 1984–2008.

Goodfellow, I., Shlens, J. and Szegedy, C. (2015), Explaining and Harnessing Adversarial Examples, *in* 'International Conference on Learning Representations'.

Google Cloud AI platform (n.d.), 'Introduction to AI Explanations for AI Platform | AI Platform Prediction'.

Graves, M. (2022), 'Theological Foundations for Moral Artificial Intelligence', *Journal of Moral Theology* **11**(Special Issue 1), 182–211.

Guidotti, R. (2022), 'Counterfactual explanations and how to find them : Literature review and benchmarking', *Data Min Knowl Disc* .

Gunning, D. and Aha, D. (2019), 'DARPA's Explainable Artificial Intelligence (XAI) Program', *AI Magazine* **40**(2), 44–58.

Hansen, N., Gemperle, F., Auger, A. and Koumoutsakos, P. (2006), When Do Heavy-Tail Distributions Help ?, *in* T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley and X. Yao, eds, 'Parallel Problem Solving from Nature - PPSN IX', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 62–71.

Hasselt, H. (2010), 'Double Q-learning', *Advances in neural information processing systems* **23**.

Heaven, W. D. (2021), 'Hundreds of AI tools have been built to catch covid. None of them helped.'.

Hempel, C. G. and Oppenheim, P. (1948), 'Studies in the logic of explanation', *Philosophy of science* **15**(2), 135–175.

Hermann, E. (2022), 'Artificial intelligence and mass personalization of communication content—An ethical and literacy perspective', *New Media & Society* **24**(5), 1258–1277.

Hern, A. (2022), 'Meta's BlenderBot 3 wants to chat – but can you trust it ?', *The Guardian* .

Hill, K. (2022), 'Accused of Cheating by an Algorithm, and a Professor She Had Never Met', *The New York Times* .

Hoffmann, J. (2003), Anticipatory Behavioral Control, *in* M. V. Butz, O. Sigaud and P. Gérard, eds, 'Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 44–65.

Hoffmann, J. (2009), ABC : A Psychological Theory of Anticipative Behavioral Control, *in* G. Pezzulo, M. V. Butz, O. Sigaud and G. Baldassarre, eds, 'Anticipatory Behavior in Adaptive Learning Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 10–30.

Holland, J. H. (1976), Adaptation, *in* R. Rosen and F. M. Snell, eds, 'Progress in Theoretical Biology', Academic Press, pp. 263–293.

Huband, S., Barone, L., While, L. and Hingston, P. (2005), A Scalable Multi-objective Test Problem Toolkit, *in* C. A. Coello Coello, A. Hernández Aguirre and E. Zitzler, eds, 'Evolutionary Multi-Criterion Optimization', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 280–295.

Huszár, F., Ktena, S. I., O'Brien, C., Belli, L., Schlaikjer, A. and Hardt, M. (2022), 'Algorithmic amplification of politics on Twitter', *Proceedings of the National Academy of Sciences* **119**(1), e2025334119.

Islam, M. R., Ahmed, M. U., Barua, S. and Begum, S. (2022), 'A Systematic Review of Explainable Artificial Intelligence in Terms of Different Application Domains and Tasks', *Applied Sciences* **12**(3), 1353.

Jehl, P., Sievers, F. and Higgins, D. G. (2015), 'OD-seq : Outlier detection in multiple sequence alignments', *BMC Bioinformatics* **16**(1), 269.

Jiang, L., Hwang, J. D., Bhagavatula, C., Bras, R. L., Liang, J., Dodge, J., Sakaguchi, K., Forbes, M., Borchardt, J., Gabriel, S., Tsvetkov, Y., Etzioni, O., Sap, M., Rini, R. and Choi, Y. (2022), 'Can Machines Learn Morality ? The Delphi Experiment'.

Jobin, A., Ienca, M. and Vayena, E. (2019), 'The global landscape of AI ethics guidelines', *Nat Mach Intell* **1**(9), 389–399.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P. and Hassabis, D. (2021), 'Highly accurate protein structure prediction with AlphaFold', *Nature* **596**(7873), 583–589.

Khenoussi, W., Vanhoutrève, R., Poch, O. and Thompson, J. D. (2014), 'SIBIS : A Bayesian model for inconsistent protein sequence estimation', *Bioinformatics* **30**(17), 2432–2439.

Khodji, H., Herbay, L., Collet, P., Thompson, J. and Jeannin-Girardon, A. (2022), MERLIN : Identifying inaccuracies in Multiple Sequence Alignments using Object Detection, *in* 'International Conference on Artificial Intelligence Applications and Innovations'.

Kim, W. and Lee, K. (2020), Building Ethical AI from News Articles, *in* '2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)', pp. 210–217.

Klaus, S., HannaAlex and DentonEmily (2021), 'Do Datasets Have Politics ? Disciplinary Values in Computer Vision Dataset Development', *Proceedings of the ACM on Human-Computer Interaction* .

Konak, A., Coit, D. W. and Smith, A. E. (2006), 'Multi-objective optimization using genetic algorithms : A tutorial', *Reliability Engineering & System Safety* **91**(9), 992–1007.

Krizhevsky, A. (2009), 'CIFAR-10 and CIFAR-100 datasets'.

Lanzi, P. L. (1999), 'An Analysis of Generalization in the XCS Classifier System', *Evolutionary Computation* **7**(2), 125–149.

Larson, J., Angwin, J., Kirchner, L. and Mattu, S. (2016), 'How We Analyzed the COMPAS Recidivism Algorithm'.

Laugel, T., Lesot, M.-J., Marsala, C., Renard, X. and Detyniecki, M. (2019), The dangers of post-hoc interpretability : Unjustified counterfactual explanations, *in*

'Proceedings of the 28th International Joint Conference on Artificial Intelligence', IJCAI'19, AAAI Press, Macao, China, pp. 2801–2807.

Le Coadic, R. (2006), 'L'autonomie, illusion ou projet de société ?', *Cahiers internationaux de sociologie* **no 121**(2), 317–340.

Lebouc, A., Allab, F., Fournier, J.-M. and Yonnet, J.-P. (2005), 'Réfrigération magnétique', *Techniques de l'Ingénieur* **28**, 16.

LeCun, Y. (1998), 'MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges'.

Leonteva, A. O., Parrend, P., Jeannin-Girardon, A. and Collet, P. (2020), Fast Evolutionary Algorithm for Solving Large-Scale Multi-objective Problems, *in* L. Idoumghar, P. Legrand, A. Liefooghe, E. Lutton, N. Monmarché and M. Schoenauer, eds, 'Artificial Evolution', Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 82–95.

Leonteva, A. O., Risser, M., Hamane, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2022), A hybrid optimization tool for active magnetic regenerator, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference Companion', GECCO '22, Association for Computing Machinery, New York, NY, USA, pp. 739–742.

Li, O., Liu, H., Chen, C. and Rudin, C. (2018), Deep learning for case-based reasoning through prototypes : A neural network that explains its predictions, *in* 'Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence', AAAI'18/IAAI'18/EAAI'18, AAAI Press, New Orleans, Louisiana, USA, pp. 3530–3537.

Liao, S. (2018), 'Chinese facial recognition system mistakes a face on a bus for a jaywalker'.

Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S. (2021), 'Explainable AI : A Review of Machine Learning Interpretability Methods', *Entropy* **23**(1), 18.

Lipshitz, R. and Strauss, O. (1997), 'Coping with Uncertainty : A Naturalistic Decision-Making Analysis', *Organizational Behavior and Human Decision Processes* **69**(2), 149–163.

Liu, Z., Janowicz, K., Cai, L., Zhu, R., Mai, G. and Shi, M. (2022), 'Geoparsing : Solved or Biased ? An Evaluation of Geographic Biases in Geoparsing', *AGILE : GIScience Series* **3**, 1–13.

Lobo, D. and Levin, M. (2015), 'Inferring Regulatory Networks from Experimental Morphological Phenotypes : A Computational Method Reverse-Engineers Planarian Regeneration', *PLOS Computational Biology* **11**(6), e1004295.

Lourie, N., Bras, R. L., Bhagavatula, C. and Choi, Y. (2021), UNICORN on RAINBOW : A universal commonsense reasoning model on a new multitask benchmark, *in* 'AAAI'.

Lubin, G. (2016), ''Facial-profiling' could be dangerously inaccurate and biased, experts warn'.

Mandal, A., Leavy, S. and Little, S. (2021), Dataset Diversity : Measuring and Mitigating Geographical Bias in Image Search and Retrieval, *in* 'Proceedings of the 1st International Workshop on Trustworthy AI for Multimedia Computing', Trustworthy AI'21, Association for Computing Machinery, New York, NY, USA, pp. 19–25.

Marra, G., Giannini, F., Diligenti, M. and Gori, M. (2019), Integrating learning and reasoning with deep logic models, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 517–532.

Mason, P. (2016), 'The racist hijacking of Microsoft's chatbot shows how the internet teems with hate', *The Guardian* .

Métivier, M. and Lattaud, C. (2003), Anticipatory Classifier System Using Behavioral Sequences in Non-Markov Environments, *in* P. L. Lanzi, W. Stolzmann and S. W. Wilson, eds, 'Learning Classifier Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 143–162.

Meyer, C., Scalzitti, N., Jeannin-Girardon, A., Collet, P., Poch, O. and Thompson, J. D. (2020), 'Understanding the causes of errors in eukaryotic protein-coding gene prediction : A case study of primate proteomes', *BMC Bioinformatics* **21**(1), 513.

Miller, T. (2019), 'Explanation in artificial intelligence : Insights from the social sciences', *Artificial Intelligence* **267**, 1–38.

Ministère de l'Agriculture et de la Souveraineté alimentaire (n.d.), 'L'économie mondiale du froid'.

Minot, J. R., Cheney, N., Maier, M., Elbers, D. C., Danforth, C. M. and Dodds, P. S. (2022), 'Interpretable bias mitigation for textual data : Reducing genderization in patient notes while maintaining classification performance', *ACM Trans. Comput. Healthcare* .

Misra, I., Shrivastava, A., Gupta, A. and Hebert, M. (2016), Cross-stitch networks for multi-task learning, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 3994–4003.

Moradi, M. and Samwald, M. (2021), 'Post-hoc explanation of black-box classifiers using confident itemsets', *Expert Systems with Applications* **165**, 113941.

Nakamura, Y., Horiuchi, M. and Nakata, M. (2021), Convergence analysis of rule-generality on the XCS classifier system, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference', GECCO '21, Association for Computing Machinery, New York, NY, USA, pp. 332–339.

Nozza, D., Volpetti, C. and Fersini, E. (2019), Unintended Bias in Misogyny Detection, *in* 'IEEE/WIC/ACM International Conference on Web Intelligence', WI '19, Association for Computing Machinery, New York, NY, USA, pp. 149–155.

Obermeyer, Z., Powers, B., Vogeli, C. and Mullainathan, S. (2019), 'Dissecting racial bias in an algorithm used to manage the health of populations', *Science* **366**(6464), 447–453.

ÓhÉigeartaigh, S. S., Whittlestone, J., Liu, Y., Zeng, Y. and Liu, Z. (2020), 'Overcoming Barriers to Cross-cultural Cooperation in AI Ethics and Governance', *Philos. Technol.* **33**(4), 571–593.

Oltramari, A., Francis, J., Henson, C., Ma, K. and Wickramarachchi, R. (2020), 'Neuro-symbolic architectures for context understanding', *Knowledge Graphs for eXplainable Artificial Intelligence : Foundations, Applications and Challenges* **47**, 143.

Orhand, R. (2022), Vers Une Intelligence Artificielle Autonome et Explicable Pour Des Environnements Incertains, PhD thesis, Université de Strasbourg.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2019), DeepExpert : Vers une Intelligence Artificielle autonome et explicable, *in* 'Rencontres Des Jeunes Chercheurs En Intelligence Artificielle (RJCIA)'.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2020*a*), BACS : A Thorough Study of Using Behavioral Sequences in ACS2, *in* T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich and H. Trautmann, eds, 'Parallel Problem Solving from Nature – PPSN XVI', Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 524–538.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2020*b*), BACS : Integrating behavioral sequences to ACS2, *in* 'Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion', pp. 147–148.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2020*c*), PEPACS : Integrating probability-enhanced predictions to ACS2, *in* 'Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion', GECCO '20, Association for Computing Machinery, New York, NY, USA, pp. 1774–1781.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2021), Explainability and performance of anticipatory learning classifier systems in non-deterministic environments, *in* 'Genetic and Evolutionary Computation Conference (GECCO)'.

Orhand, R., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2022), Accurate and Interpretable Representations of Environments with Anticipatory Learning Classifier Systems, *in* E. Medvet, G. Pappa and B. Xue, eds, 'Genetic Programming', Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 245–261.

Orhand, R., Khodji, H., Hutt, A. and Jeannin-Girardon, A. (2021), 'Quantification of the transferability of features between deep neural networks', *Procedia Computer Science* **192**, 138–147.

Ouskova Leonteva, A. (2022), Evolutionary and Quantum-inspired Algorithms for the Optimization of Magnetic Cooling Systems, PhD thesis, Université de Strasbourg.

Ouskova Leonteva, A., Abdulkarimova, U., Jeannin-Girardon, A., Parrend, P. and Collet, P. (2020), Quantum-inspired algorithm with evolution strategy, *in* 'TPNC 2020 : 9th International Conference on the Theory and Practice of Natural Computing', p. 12.

Parlement européen (2016), 'EUR-Lex - 32016R0679 - EN - EUR-Lex'.

Pearson, W. (2003), 'Finding Protein and Nucleotide Similarities with FASTA', *Current Protocols in Bioinformatics* **4**(1), 3.9.1–3.9.23.

Prosdocimi, F., Linard, B., Pontarotti, P., Poch, O. and Thompson, J. D. (2012), 'Controversies in modern evolutionary biology : The imperative for error detection and quality control', *BMC Genomics* **13**(1), 5.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J. (2020), 'Exploring the limits of transfer learning with a unified text-to-text transformer', *Journal of Machine Learning Research* **21**(140), 1–67.

Randall, D. (1980), 'Meta-rules : Reasoning about control - ScienceDirect', *Artificial Intelligence* **15**.

Rawls, J. (1951), 'Outline of a Decision Procedure for Ethics', *The Philosophical Review* **60**(2), 177–197.

Reviglio, U. and Agosti, C. (2020), 'Thinking Outside the Black-Box : The Case for "Algorithmic Sovereignty" in Social Media', *Social Media + Society* **6**(2), 2056305120915613.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016), "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier, *in* 'Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '16, ACM, New York, NY, USA, pp. 1135–1144.

Risser, M., Vasile, C., Muller, C. and Noume, A. (2013), 'Improvement and application of a numerical model for optimizing the design of magnetic refrigerators', *International Journal of Refrigeration* **36**(3), 950–957.

Rudin, C. (2019), 'Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead', *Nat Mach Intell* **1**(5), 206–215.

Rudin, C. and Radin, J. (2019), 'Why Are We Using Black Box Models in AI When We Don't Need To ? A Lesson From an Explainable AI Competition', *Harvard Data Science Review* **1**(2).

Rudin, C., Wang, C. and Coker, B. (2020), 'The Age of Secrecy and Unfairness in Recidivism Prediction', *Harvard Data Science Review* **2**(1).

Salmon, W. (1971), *Statistical Explanation & Statistical Relevance*, Pittsburgh, PA, USA : University of Pittsburgh Press.

Santillán, M. (2008), 'On the Use of the Hill Functions in Mathematical Models of Gene Regulatory Networks', *Math. Model. Nat. Phenom.* **3**(2), 85–97.

Santos, F. P., Lelkes, Y. and Levin, S. A. (2021), 'Link recommendation algorithms and dynamics of polarization in online social networks', *Proceedings of the National Academy of Sciences* **118**(50), e2102141118.

Sarker, M. K., Zhou, L., Eberhart, A. and Hitzler, P. (2021), 'Neuro-symbolic artificial intelligence', *AI Communications* **34**(3), 197–209.

Sato, H., Aguirre, H. E. and Tanaka, K. (2007), Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs, *in* S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu and T. Murata, eds, 'Evolutionary Multi-Criterion Optimization', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 5–20.

Sattarzadeh, S., Sudhakar, M., Lem, A., Mehryar, S., Plataniotis, K. N., Jang, J., Kim, H., Jeong, Y., Lee, S. and Bae, K. (2021), Explaining convolutional neural networks through attribution-based input sampling and block-wise feature aggregation, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 35, pp. 11639–11647.

Scalzitti, N., Jeannin-Girardon, A., Collet, P., Poch, O. and Thompson, J. D. (2020), 'A benchmark study of ab initio gene prediction methods in diverse eukaryotic organisms', *BMC Genomics* **21**(1), 293.

Schramowski, P., Turan, C., Jentzsch, S., Rothkopf, C. and Kersting, K. (2020), 'The Moral Choice Machine', *Frontiers in Artificial Intelligence* **3**.

Seward, J. P. (1949), 'An experimental analysis of latent learning', *Journal of Experimental Psychology* **39**(2), 177–186.

Shane, J. (2018), 'Do neural nets dream of electric sheep ?'.

Shankar, S., Halpern, Y., Breck, E., Atwood, J., Wilson, J. and Sculley, D. (2017), No classification without representation : Assessing geodiversity issues in open data sets for the developing world, *in* 'NIPS 2017 Workshop : Machine Learning for the Developing World'.

Sharma, D. and Collet, P. (2010*a*), An archived-based stochastic ranking evolutionary algorithm (asrea) for multi-objective optimization, *in* 'Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation', GECCO '10, Association for Computing Machinery, New York, NY, USA, pp. 479–486.

Sharma, D. and Collet, P. (2010*b*), GPGPU-Compatible Archive Based Stochastic Ranking Evolutionary Algorithm (G-ASREA) for Multi-Objective Optimization, *in* R. Schaefer, C. Cotta, J. Kołodziej and G. Rudolph, eds, 'Parallel Problem Solving from Nature, PPSN XI', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 111–120.

Stolzmann, W. (2000), An Introduction to Anticipatory Classifier Systems, *in* 'Learning Classifier Systems, From Foundations to Applications', Springer-Verlag, Berlin, Heidelberg, pp. 175–194.

Stolzmann, W. and Butz, M. (2000), Latent Learning and Action Planning in Robots with Anticipatory Classifier Systems, *in* P. L. Lanzi, W. Stolzmann and S. W. Wilson, eds, 'Learning Classifier Systems', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 301–317.

Stroud, N. J. (2008), 'Media Use and Political Predispositions : Revisiting the Concept of Selective Exposure', *Polit Behav* **30**(3), 341–366.

Svegliato, J., Nashed, S. B. and Zilberstein, S. (2021), Ethically compliant sequential decision making, *in* 'AAAI'.

Thompson, J. (2016), *Statistics for Bioinformatics : Methods for Multiple Sequence Alignment*, 1st edition edn, ISTE Press - Elsevier.

Tolman, E. C. (1932), *Purposive Behavior in Animals and Men*, Purposive Behavior in Animals and Men, Century/Random House UK, London, England.

Tschechne, S. and Neumann, H. (2014), 'Hierarchical representation of shapes in visual Cortex—from localized features to figural shape segregation', *Frontiers in Computational Neuroscience* **8**.

Valgiani, G., Lutton, E., fonlupt, C. and Collet, P. (2007), 'Optimisation par « hommilière » de chemins pédagogiques pour un logiciel d'e-Learning', *Techniques et sciences informatiques* **26**(10), 1245–1267.

van der Waa, J., Nieuwburg, E., Cremers, A. and Neerincx, M. (2021), 'Evaluating XAI : A comparison of rule-based and example-based explanations', *Artificial Intelligence* **291**, 103404.

Vilone, G. and Longo, L. (2021), 'Notions of explainability and evaluation approaches for explainable artificial intelligence', *Information Fusion* **76**, 89–106.

Vincent, J. (2016), 'Twitter taught Microsoft's friendly AI chatbot to be a racist asshole in less than a day'.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C. and Silver, D. (2019), 'Grandmaster level in StarCraft II using multi-agent reinforcement learning', *Nature* **575**(7782), 350–354.

Wang, A., Narayanan, A. and Russakovsky, O. (2020), REVISE : A Tool for Measuring and Mitigating Bias in Visual Datasets, *in* 'Computer Vision – ECCV 2020 : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III', Springer-Verlag, Berlin, Heidelberg, pp. 733–751.

Wang, H. and et al. (2020), 'Score-CAM : Score-Weighted Visual Explanations for Convolutional Neural Networks', *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* .

Warnow, T. (2021), 'Revisiting Evaluation of Multiple Sequence Alignment Methods', *Methods Mol Biol* **2231**, 299–317.

Watkins, C. J. C. H. and Dayan, P. (1992), 'Q-learning', *Mach Learn* **8**(3), 279–292.

Whitehead, S. D. and Ballard, D. H. (1991), 'Learning to perceive and act by trial and error', *Mach Learn* **7**(1), 45–83.

Wilson, S. W. (1995), 'Classifier fitness based on accuracy', *Evol. Comput.* **3**(2), 149–175.

Wilson, S. W. and Goldberg, D. E. (1989), A Critical Review of Classifier Systems, *in* 'Proceedings of the 3rd International Conference on Genetic Algorithms', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 244–255.

Wong, P.-H. (2020), 'Cultural Differences as Excuses ? Human Rights and Cultural Values in Global Ethics and Governance of AI', *Philos. Technol.* **33**(4), 705–715.

Woodward, J. (2005), *Making Things Happen : A Theory of Causal Explanation*, Oxford university press.

Wright, A. T. (2022), 8 Rightful Machines, *in* H. Kim and D. Schönecker, eds, 'Kant and Artificial Intelligence', De Gruyter, pp. 223–238.

Zaal, D. and Nota, B. (2016), 'ADOMA : A Command Line Tool to Modify ClustalW Multiple Alignment Output', *Molecular Informatics* **35**(1), 42–44.

Zeiler, M. D. and Fergus, R. (2014), Visualizing and Understanding Convolutional Networks, *in* D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, eds, 'Computer Vision – ECCV 2014', Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 818–833.

Zhang, Q. and Li, H. (2007), 'MOEA/D : A Multiobjective Evolutionary Algorithm Based on Decomposition', *IEEE Transactions on Evolutionary Computation* **11**(6), 712–731.

Zhang, Y., Zhang, Y. and Yang, Q. (2019), Parameter Transfer Unit for Deep Neural Networks, *in* Q. Yang, Z.-H. Zhou, Z. Gong, M.-L. Zhang and S.-J. Huang, eds, 'Advances in Knowledge Discovery and Data Mining', Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 82–95.

Zhong, Q., Fan, X., Luo, X. and Toni, F. (2019), 'An explainable multi-attribute decision model based on argumentation', *Expert Systems with Applications* **117**, 42–61.

Zitzler, E. and Künzli, S. (2004), Indicator-Based Selection in Multiobjective Search, *in* X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán and H.-P. Schwefel, eds, 'Parallel Problem Solving from Nature - PPSN VIII', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 832–842.

Zoshak, J. and Dew, K. (2021), Beyond Kant and Bentham : How Ethical Theories are being used in Artificial Moral Agents, *in* 'Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems', CHI '21, Association for Computing Machinery, New York, NY, USA, pp. 1–15.

— **Résumé** —

L'objectif de notre travail de recherche est de nous interroger sur les différentes manières possibles de construire artificiellement des connaissances ; de nous intéresser, donc, à une épistémologie de l'intelligence artificielle (IA). On parle bien ici d'une « construction de savoirs » par des systèmes intelligents artificiels et non pas d'une interrogation plus profonde sur la nature même de ces systèmes artificiels. Ainsi, de la même manière que l'approche épistémologique d'un énoncé comme « la terre tourne autour du soleil » interroge sur les raisons de penser que la terre tourne effectivement autour du soleil et pourquoi ces raisons sont admissibles, nous souhaitons nous interroger sur les énoncés produits par des intelligences artificielles et, un peu plus largement, sur les impacts qu'ont de tels énoncés d'un point de vue éthique sur celles et ceux qui les utilisent. Pour aborder ces questions, nous proposons tout d'abord de cadrer plus spécifiquement les aspects épistémologiques qui vont nous intéresser ici : la caractérisation des énoncés produits par des IA et l'explication de ces énoncés et leurs conséquences. Dans une seconde partie, nous ferons la synthèse de travaux que nous avons menés ces dernières années concernant l'utilisation de différentes approches d'intelligence artificielle pour résoudre des tâches diverses : nous nous sommes intéressés en particulier à l'optimisation, par le calcul évolutionnaire, de systèmes de réfrigération magnétocalorique, au développement d'une intelligence artificielle autonome et explicable capable d'évoluer en environnements incertains et à la caractérisation d'erreurs de prédiction de gènes dans des alignements multiples de séquences par réseaux de neurones profonds. Enfin, nous présentons la direction de recherche que nous souhaitons explorer et développer à la suite des différents travaux que nous avons menés et synthétisés ici, toujours dans cette optique de contribuer à développer une approche épistémologique de l'intelligence artificielle.