



HAL
open science

Conférence Nationale d'Intelligence Artificielle Année 2023

Emmanuel Adam, Thomas Guyet, Benoit Le Blanc, Dominique Longin, Zied Bouraoui, Sandra Bringay, Romaric Gaudel, Charlotte Laclau, Christelle Launois, Maxime Morge, et al.

► **To cite this version:**

Emmanuel Adam, Thomas Guyet, Benoit Le Blanc, Dominique Longin, Zied Bouraoui, et al.. Conférence Nationale d'Intelligence Artificielle Année 2023. Association Française pour l'Intelligence Artificielle, 2023. hal-04310171v1

HAL Id: hal-04310171

<https://ut3-toulouseinp.hal.science/hal-04310171v1>

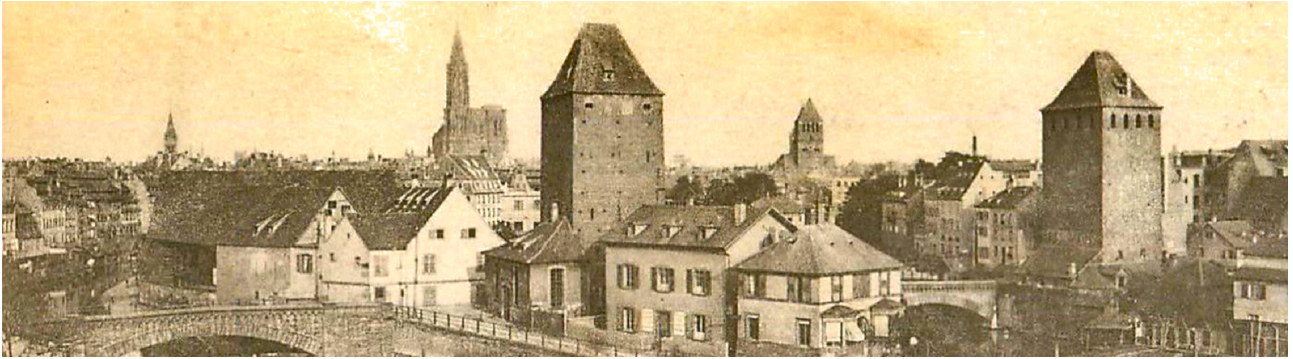
Submitted on 6 Dec 2023 (v1), last revised 22 Oct 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License



Conférence Nationale d'Intelligence Artificielle Année 2023

Sections Spéciales

Zied Bouraoui
Sandra Bringay
Romaric Gaudel
Charlotte Laclau
Christelle Launois
Maxime Morge
Catherine Roussey
François Schwarzentruher
Cassia Trojahn
Élise Vareilles
Anaëlle Wilczynski

Coordination AFIA

Emmanuel Adam
Thomas Guyet
Benoit Le Blanc
Dominique Longin



**Conférence Nationale
d'Intelligence Artificielle
Année 2023**

Activité AFIA

1^{er} août 2021 – 31 juillet 2023

Éditeurs : Conseil d'Administration de l'AFIA – Année 2023

Tables des matières

À propos de l'AFIA	III
Conseils d'administration 2022 et 2023	V
Collège Apprentissage Artificiel	VII
Collège Création d'Événements Collaboratifs, Inclusifs et Ludiques en IA	IX
Collège Industriel	XI
Collège Interaction avec l'Humain	XIII
Collège Représentation et Raisonnement	XV
Collège Science de l'Ingénierie des Connaissances	XVII
Collège Systèmes Multi-Agents et Agents Autonomes	XIX
Collège Technologies du Langage Humain	XXI
Prix de Thèse IA 2022 et 2023	XXIII
Bulletins	XXV
Journée thématique EFIA 2022	XXVII
Journée thématique PDIA 2022	XXVIII
Journée thématique FIIA 2022	XXIX
Journée thématique EFIA 2023	XXX
Journée thématique PDIA 2023	XXXI
Journée thématique Résilience & IA 2023	XXXII
Journée commune Classification & IA 2021	XXXIII
Journée commune RI & IA 2021	XXXIV
Journée commune RV & IA 2022	XXXV
Journée commune Simulation & IA 2022	XXXVI
Journée commune IoT & IA 2022	XXXVII
Journée commune Santé & IA 2022	XXXVIII
Journée commune EIAH & IA 2022	XXXIX
Journée commune Résilience & IA 2022	XL
Journée commune EGC & IA 2023	XLI
Journée commune IHM & IA 2023	XLII
Journée commune Jeux & IA 2023	XLIII
Journée commune Santé & IA 2023	XLIV
Plate-forme Intelligence Artificielle 2022	XLV
Plate-forme Intelligence Artificielle 2023	XLVI

À propos de l'AFIA

L'objet de l'AFIA, Association Loi 1901 sans but lucratif, est de promouvoir et de favoriser le développement de l'Intelligence Artificielle (IA) sous ses différentes formes, de regrouper et de faire croître la communauté française en IA et, à la hauteur des forces de ses membres, d'en assurer la visibilité.

L'AFIA anime la communauté par l'organisation de grands rendez-vous. Se tient ainsi chaque été une semaine de l'IA, la Plate-forme IA (PfIA 2022 à Saint-Étienne, PfIA 2023 à Strasbourg) au sein de laquelle se tiennent la Conférence Nationale d'Intelligence Artificielle (CNIA), les Rencontres des Jeunes Chercheurs en IA (RJCIA) et la Conférence sur les Applications Pratiques de l'IA (APIA) ainsi que des conférences/journées thématiques hébergées qui évoluent d'une année à l'autre, sans récurrence obligée.

Ainsi, PfIA 2023 héberge du 3 au 7 juillet 2023 à Strasbourg, outre la 26^e CNIA, les 21^{es} RJCIA et la 9^e APIA : les 6 conférences CAP, IC, JFPC, JFSMA, JIAF-JFPDA et SFC, 4 journées thématiques (ACAI, Jeux & IA, Résilience & IA, Santé & IA), et plusieurs tutoriels hébergés.

Forte du soutien de ses 374 adhérents à jour de leur cotisation en 2022, l'AFIA assure :

- le maintien d'un site Web dédié à l'IA reproduisant également les Brèves de l'IA ;
- une *journée industrielle* « Forum Industriel en IA » (FIIA 2022) ;
- une *journée recherche* « Perspectives et Défis en IA » (PDIA 2022) ;
- une *journée enseignement* « Enseignement et Formation en IA » (EFIA 2023) ;
- une « École Saisonnnière en IA » (ESIA2023) ;
- la remise annuelle d'un *prix de thèse* en IA ;
- le soutien à 8 collèges ayant leur propre activité :
 - collège Industriel (janvier 2016),
 - collège Apprentissage Artificiel (janvier 2020),
 - collège Interaction avec l'Humain (juillet 2020),
 - collège Représentation et Raisonnement (avril 2017),
 - collège Science de l'Ingénierie des Connaissances (avril 2016),
 - collège Systèmes Multi-Agents et Agents Autonomes (janvier 2017),
 - collège Technologies du Langage Humain (juillet 2019),
 - collège Création d'Événements Collaboratifs, Inclusifs et Ludiques en IA (octobre 2021) ;
- la parution trimestrielle des Bulletins de l'AFIA ;
- un lien entre ses membres et sympathisants sur les réseaux sociaux LinkedIn, Facebook et Twitter ;
- le parrainage scientifique, mais aussi éventuellement financier, d'événements en IA ;
- la diffusion mensuelle de Brèves sur les actualités de l'IA en France (abonnement ou envoi à la liste) ;
- la réponse aux consultations officielles ou officieuses (Ministères, Missions, Organismes) ;
- la réponse aux questions de la presse, écrite ou orale, également sur internet ;
- la divulgation d'offres de collaborations, de formations, d'emploi, de thèses et de stages.

L'AFIA organise aussi des journées communes avec d'autres associations. Pour 2022 : EIAH & IA avec l'ATIEF; IoT & IA avec l'IMT; Résilience & IA avec la région ARA; Réalité Virtuelle & IA avec le GDR IG-RV; Santé & IA avec l'AIM; Simulation & IA avec le réseau DEVS/RED.

Enfin, l'AFIA encourage la participation de ses membres aux grands événements de l'IA, dont PfIA. Ainsi, les membres de l'AFIA, pour leur inscription à PfIA, bénéficient d'une réduction équivalente à deux fois le coût de leur adhésion, leur permettant d'assister à PfIA 2023 sur 5 jours au tarif de 114€ TTC !

Rejoignez-nous vous aussi et adhérez à l'AFIA pour contribuer au développement de l'IA en France. L'adhésion peut être individuelle ou au titre de personne morale. Merci également de susciter de telles adhésions en diffusant ce document autour de vous !

**AFIA**Association française
pour l'Intelligence Artificielle

Conseils d'administration 2022 et 2023

Conseil d'Administration 2022

Président	Benoit LE BLANC	GT Bureau, Prix Thèse, International, Représentation
V.-Présidente	Domitile LOURDEAUX	GT Bureau, GT Collèges, Parrainages, Représentation
Trésorière	Isabelle SESÉ	GT Bureau, Adhésions, Partenariats
Secrétaire	Grégory BONNET	GT Rédaction, Prix Thèse
Rédacteur	Dominique LONGIN	GT Rédaction, GT Communication
Webmestre	Emmanuel ADAM	GT Web & Mail, GT Communication, Brèves, Réseaux
	Yves DEMAZEAU	GT Communication, GT Plateforme, PFIA 2022
	Gaël DIAS	GT Enseignement, École saisonnière
	Bernard GEORGES	Collège Industriel
	Thomas GUYET	GT Plateforme, PFIA 2022
	Frédéric MARIS	GT Bureau, Adhésions
	Davy MONTOCILO	GT Plateforme, PFIA 2022, PDIA 2022
	Engelbert Mephu NGUIFO	GT Journées, Dossier n°114, PDIA 2022
	Gauthier PICARD	GT Enseignement, EFIA 2022
	Valérie REINER	Collège Industriel, École saisonnière, Représentation
	Catherine ROUSSEY	GT Web & Mail
	Céline ROUVEIROL	Collège Industriel, Cartographie
	Fatiha SAIS	GT Journées, PDIA 2022
	Ahmed SAMET	GT Plateforme, Tutoriels PFIA 2022
	Charlotte TRUCHET	GT Collèges

Conseil d'Administration 2023

Président	Benoit LE BLANC	GT Bureau, Prix Thèse, International, Représentation
V.-Présidente	Domitile LOURDEAUX	GT Bureau, GT Collèges, Parrainages, Représentation
Trésorière	Isabelle SESÉ	GT Bureau, Adhésions, Partenariats
Secrétaire	Grégory BONNET	GT Rédaction, Prix Thèse
Porte-parole	Emmanuel ADAM	GT Communication, Brèves, Réseaux
Rédacteur	Dominique LONGIN	GT Rédaction, GT Communication
Webmestre	Catherine ROUSSEY	GT Web & Mail, GT Communication, Brèves, Réseaux
	Gayo DIALLO	Collège Industriel
	Gaël DIAS	GT Enseignement, École saisonnière
	Bernard GEORGES	Collège Industriel
	Thomas GUYET	GT Plateforme, PFIA 2023
	Frédéric MARIS	GT Bureau, Adhésions
	Davy MONTOCILO	GT Plateforme, PFIA 2023, PDIA 2023
	Gauthier PICARD	GT Enseignement, EFIA 2022
	Valérie REINER	Collège Industriel, École saisonnière, Représentation
	Céline ROUVEIROL	Collège Industriel, Cartographie
	Fatiha SAIS	GT Journées, PDIA 2022
	Ahmed SAMET	GT Plateforme, Tutoriels PFIA 2022

Collège Apprentissage Artificiel

Objectif du collège

L'objectif du Collège **Apprentissage Artificiel** (C2A) de l'**AFIA** est de contribuer à l'animation de la communauté de recherche française en apprentissage automatique (ou artificiel), et ce en synergie avec les structures d'animation déjà existantes. Les thématiques de recherche sont celles d'apprentissage artificiel, adossées aux principales conférences de la communauté que sont : Conférence d'Apprentissage Artificiel (CAp), Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes (JFRB), Rencontres de la Société Francophone de Classification (SFC), Reconnaissance de Formes, Image, Apprentissage et Perception (RFIAP), etc. et concerne principalement l'ensemble des travaux autour de l'étude, la conception et l'évaluation d'algorithmes d'apprentissage artificiel, sous ses différentes formes : supervisé, non supervisé ou semi-supervisé ; statistique ou symbolique ; par renforcement ; par transfert.

Plus précisément, en lien avec le CA de l'AFIA, le collège a pour buts l'animation de la communauté autour de l'apprentissage artificiel (parrainage d'événements, organisation de journées bilatérales ou thématiques), et la représentation de la communauté en apprentissage artificiel au sein de l'AFIA (communications sur le thème de l'apprentissage artificiel, participation aux comités de programme).

Programme de travail

Le Collège Apprentissage Artificiel sera impliqué dans les activités suivantes :

- organisation de journées communes, par exemple :
 - Cla&IA : Classification et IA, en lien avec la SFC,
 - Stats&IA : Statistiques et IA, en lien avec la SFdS,
 - RF&IA : Reconnaissance de formes et IA (partie Apprentissage), en lien avec l'AFRIF ;
- organisation de journées thématiques, en particulier :
 - atelier sur la plateforme de l'AFIA,
 - atelier en lien avec une conférence spécialisée (CAp, JFRB, SFC, RFIAP),
 - journée industrielle en lien avec le Collège **Industriel** de l'AFIA ;
- interaction avec des conférences (hors AFIA) pour lesquelles l'apprentissage est un thème central, parmi lesquels : CAp, JFRB, SFC, RFIAP (partie Apprentissage) ;
- interaction avec les autres associations existantes de la communauté en apprentissage artificiel :
 - Société Savante Francophone sur l'Apprentissage Machine (SSFAM),
 - Société Francophone de Classification (SFC),
 - Société Française de Statistique (SFdS).

Comité de pilotage

Le comité de pilotage est composé des personnes suivantes :

- Jérôme AZÉ, Université de Montpellier, LIRMM ;

- Isabelle BLOCH, Télécom Paris, LTCI ;
- Antoine CORNUÉJOLS, AgroParisTech, MMIP ;
- Elisa FROMONT, Université Rennes 1, IRISA ;
- Charlotte LACLAU, Université Jean Monnet à Saint-Etienne, LaHC ;
- Engelbert Mephu NGUIFO, Université Clermont Auvergne, LIMOS ;
- Amedeo NAPOLI, CNRS, LORIA ;
- Philippe PREUX, Université Lille 3, CRIStAL ;
- Céline ROUVEIROL, Université Paris 13, LIPN ;
- Christel VRAIN, Université d'Orléans, LIFO.

Le comité de pilotage peut être amené à inviter des membres de la communauté à participer aux discussions et réunions du collège.

Contacts

Coordinateur du collège : engelbert.mephu_nguifo@uca.fr.

Listes de diffusion : info-ic@inria.fr.

Deux membres du comité de pilotage sont membres du conseil d'administration de l'AFIA :

- Engelbert Mephu NGUIFO, engelbert.mephu_nguifo@uca.fr ;
- Céline ROUVEIROL, celine.rouveirol@lipn.univ-paris13.fr.

Collège Création d'Événements Collaboratifs, Inclusifs et Ludiques en IA

Objectif du collège

Le Collège **Création d'Événements Collaboratifs, Inclusifs et Ludiques en IA** (CECILIA) de l'AFIA défend l'apprentissage de l'IA grâce à la pratique coopérative et l'expérimentation. Il a deux objectifs :

- mettre à disposition des ressources pour l'Intelligence Artificielle par la pratique, en particulier pour les étudiant(e)s/lycéen(ne)s (par exemple lors de la Nuit de l'info) ;
- favoriser les rencontres dans la communauté IA au travers d'événements instructifs, ludiques et conviviaux (par exemple lors de PFIA).

Ce collège propose de reprendre les activités du Collège Compétition 2018-2020, et de les étendre à d'autres activités comme des ateliers inspirés des *game jams* pour être plus inclusives en termes de public et d'approches scientifiques.

Pour mémoire, le Collège Compétition a organisé pour l'AFIA les compétitions et animations : « IA sur Robot » lors de RFIA'16 à Clermont-Ferrand, et « BotContest » lors de PFIA'17 à Caen, « DriveToGæther » lors de PFIA'19 (PFIA'20 ayant été finalement organisé en virtuel, nous avons dû annuler notre événement cette année-là).

Programme de travail

À ce jour, les événements identifiées par le collège sont les suivants :

- **Nuit de l'Info.** Les participantes et les participants à la Nuit de l'Info répondent en une nuit sur un défi national de programmation, tout en relevant divers défis connexes. Leur programme et un document doivent être remis le lendemain matin avant 8h. Ces résultats sont évalués par un jury national et les jurys des défis connexes. Cette compétition a lieu tous les ans en décembre et l'AFIA y participe en y proposant un défi : intégrer de l'IA dans l'application développée. Le jury de ce défi se compose de membres du collège et d'autres personnes de l'AFIA.
- **IA sur Robots.** Le but de ce défi est de mettre en évidence l'IA au sein d'une ou plusieurs plateformes robotiques, dans un scénario figé avec des règles, et une mesure des performances. Ce défi est ouvert à toutes et tous : personnels travaillant dans la recherche, l'enseignement, l'ingénierie, étudiantes et étudiants ainsi que le grand public. Ce défi s'intitule « DriveToGæther » et dispose d'un scénario et d'un règlement. Les projets sélectionnés seront testés par le public (par exemple lors de PFIA). Un de nos souhaits est de formaliser le règlement et les supports de cet événement pour le rendre facilement organisable par des tierces organisations locales (lycées, clubs, etc.).
- **IA et Créativité.** Le but de cet événement est de faire expérimenter au public le potentiel créatif, l'originalité, la performance stratégique et/ou l'adaptabilité des méthodes d'IA. Il se déroule sur une ou plusieurs soirées (par exemple lors de PFIA), dans un cadre coopératif et festif.
- Un rassemblement annuel des membres du collège pour revenir sur les activités organisées (par exemple lors de PFIA).

Comité de pilotage

Le comité de pilotage se réunira mensuellement ou plus fréquemment en préparation des événements.

- Carole ADAM, Université Grenoble Alpes, LIG ;
- Florence BANNAY, Université Toulouse 3 Paul Sabatier, IRIT ;
- Anne-Gwenn BOSSER, École Nationale d’Ingénieurs de Brest, Lab-STICC ;
- Victor CHARPENAY, École des Mines de Saint-Etienne, LIMOS ;
- Liana ERMAKOVA, Université de Bretagne Occidentale, Brest, HCTI ;
- Thomas GUYET, Agrocampus Ouest, Rennes, IRISA ;
- Philippe MORIGNOT, Aspertise, Paris
- Nicolas PÉPIN-HERMANN, BA Healthcare, Rennes.

Un comité d’organisation sera déterminé en fonction des événements, et intégrera des membres du comité de pilotage et des membres de comité d’organisation de l’événement. Un appel à participation sera effectué pour constituer un comité consultatif afin de fédérer une communauté d’actrices et d’acteurs intéressés par l’IA pour tous (responsables d’enseignement ou de FabLab, etc.).

Ceci pourra nous permettre d’avoir une adhésion plus large à nos projets sur tout le territoire et dans les divers lieux d’enseignements, de pratique et de recherche en IA. Cette communauté sera réunie une fois par an, par exemple lors de PFIA, et/ou en visio selon l’évolution de la situation sanitaire.

Contacts

Coordinateur du collège : Nicolas.PepinHermann@gmail.com.

Adresse de contact : contact@afia-competitions.fr.

Un membre du comité de pilotage est membres du conseil d’administration de l’AFIA :

- Thomas GUYET, thomas.guyet@irisa.fr.

Collège Industriel

Objectif du collège

L'objet du Collège Industriel (CI) de l'AFIA est de favoriser les échanges en France dans le domaine de l'IA entre sa composante industrielle et sa composante académique ainsi que diverses actions de promotion de l'IA. Le rejoindre c'est, pour une société, en plus des bénéfices accordés à toutes les personnes morales de l'AFIA (pointeur vers site Web adhésions) :

- accroître la visibilité du CI de l'AFIA ;
- pouvoir faire état de ses relations académiques et leurs recherches de partenariat académique sur des problématiques ciblées ;
- proposer aux collèges thématiques des actions intéressant les membres du CI ;
- pouvoir discuter avec les autres sociétés et adhérents au CI de problématiques dans le domaine de l'IA, et partager des solutions en garantissant la confidentialité des échanges ;
- promouvoir l'IA auprès des décideurs et dirigeants industriels ;
- contribuer à équilibrer tous les domaines de l'IA et leurs hybridations ;
- témoigner auprès des collèges thématiques de cas d'usage qui intéressent le CI. ;
- témoigner de l'apport de l'IA dans l'industrie lors d'événements AFIA (FIIA, tutoriels). ;
- faire du lobbying au niveau français auprès des Ministères, des pôles de compétitivité, de l'ANR et tout autre organisme, également au niveau européen ;
- faire connaître aux académiques ses besoins en recrutement ;
- offrir des opportunités à la communauté académique de valoriser leurs formations.

En outre, les sociétés membres du CI à jour de leur cotisation mensuelle au printemps apparaissent comme partenaires de la Plateforme Intelligence Artificielle de la même année.

Programme de travail

En délégation du CA de l'AFIA, le programme de travail du CI consiste à :

- contribuer au pilotage d'événements annuels à forte visibilité, le Forum Industriel de l'IA (FIIA) et la Conférence sur les Applications Pratiques de l'IA (APIA) ;
- cartographier les relations académiques et industrielles (services du 1^{er} Ministre, MA, MC, MEAE, MESRI, MI, MINEF, MJ, MS, MTES, ALLISTENE, CNRS, IMT, INRAE, INRIA, ONERA + IRT) ;
- solliciter les collèges thématiques de l'AFIA pour des contacts ou des interventions ;
- solliciter les collèges thématiques de l'AFIA pour des partenariats de projet ;
- organiser des réunions régulières au sein du collège ;
- organiser des réunions avec invités externes ;
- coprogrammer le prochain *AI Summit France* ;
- diffuser des bulletins ou des dossiers du collège en français avec résumés en anglais ;
- motiver les facilités d'accès à toutes les approches d'IA et leurs hybridations dans les formations, open-sourcer.

Ce programme est complété en début d'année civile par les membres du collège. Le CI se réunit mensuellement pour coordonner les avancées sur les actions engagées par le collège et en décider d'autres, échanger sur un sujet particulier et/ou sur l'actualité en IA sur le mois écoulé. Les réunions du CI font l'objet de comptes rendus qui sont distribués à ses membres et au CA de l'AFIA.

Comité de pilotage

Le CI est composé de l'ensemble des sociétés s'étant acquittées des droits d'adhésion pour l'année en cours et d'au moins deux académiques membres du CA de l'AFIA. Son comité de pilotage est constitué de dix personnes physiques, dont au moins : le coordinateur du CI, le responsable de la feuille de route et le responsable des séminaires. Ces responsables sont désignés pour une durée d'une année par les membres du CI lors de la première réunion annuelle. En 2022, le Collège Industriel est composé de :

- Bruno CARRON et Frédéric PERLANT, AIRBUS, Elancourt ;
- Alain BERGER, ARDANS, Montigny-le-Bretonneux ;
- Mustapha DERRAS, Youssef MILOUDI et Valérie REINER, BERGER-LEVRAULT, Boulogne Billancourt ;
- Stéphane DURAND et Bruno PATIN, DASSAULT Aviation, Saint-Cloud ;
- Pierre FEILLET et Christian DE SAINTE-MARIE, IBM FRANCE, Gentilly ;
- Ghislain ATEMEZING et Christophe PRIGENT, MONDECA, Paris ;
- Jean-Pierre DESMOULINS, Jean-Baptiste FANTUN et Véronique VENTOS, NUKKAI, Paris ;
- Julien BOHNE, Bernard GEORGES et Christelle LAUNOIS, SOCIETE GENERALE, Val de Fontenay ;
- Patricia BESSON, Juliette MATTIOLI et David SADEK, THALES, Palaiseau ;
- Yves DEMAZEAU et Céline ROUVEIROL, CA AFIA, Grenoble et Paris.

Contacts

Coordinateur et Responsable de la feuille de route : yves.demazeau@afia.asso.fr.

Responsable des séminaires : valerie.reiner@berger-levrault.com.

Quatre membres du comité de pilotage sont membres du conseil d'administration de l'AFIA :

- Yves DEMAZEAU, yves.demazeau@afia.asso.fr ;
- Bernard GEORGES, bernard.georges.777@gmail.com ;
- Valérie REINER, valerie.reiner@berger-levrault.com ;
- Céline ROUVEIROL, celine.rouverirol@afia.asso.fr.



AFIA

Association française
pour l'Intelligence Artificielle

Collège Interaction avec l'Humain

Objectif du collège

Le collège [Interaction avec l'Humain](#) (IH) de l'[AFIA](#) a pour mission de contribuer aux activités menées par l'AFIA, par des actions relatives au domaine de l'Intelligence Artificielle (IA), en lien avec les domaines de l'Interaction Homme-Machine (IHM), des Environnements Informatiques pour l'Apprentissage Humain (EIAH), de la Narration Interactive (NI) et des Environnements Virtuels Interactifs (EVI, incluant la réalité virtuelle, la réalité augmentée, ou encore la réalité mixte).

Les thématiques de recherche couvertes par ce collège sont celles relevant de :

- l'ingénierie et la modélisation des connaissances :
 - connaissances des utilisateurs d'environnement numérique : les apprenants dans les EIAH (tuteurs intelligents, jeux sérieux, MOOC, etc.), les utilisateurs d'une IHM ou d'un EVI, les lecteurs de récits élaborés en NI,
 - connaissances nécessaires au bon fonctionnement de l'environnement numérique : les connaissances des domaines d'enseignements, les connaissances liées à la cognition humaine, les connaissances liées aux environnements virtuels,
 - connaissances nécessaires à l'interaction entre les humains et leur environnement numérique ;
- les algorithmes d'apprentissage automatique et semi-supervisés utilisés notamment pour la fouille des données d'interaction : par exemple, l'*Educational Data Mining* et les *Learning Analytics* dans le cadre des EIAH ;
- les systèmes de diagnostic et de prise de décisions pour adapter l'environnement aux besoins, capacités et préférences de leurs utilisateurs ;
- les modèles de collaboration au sein des environnements complexes où l'humain a une place prépondérante ;
- l'Intelligence Artificielle pour soutenir l'interaction 3D informée en Environnements Immersifs ;
- les approches génératives pour la personnalisation de scénarios en environnements virtuels et narration interactive.

Ces thématiques seront coordonnées avec celles couvertes par les autres collèges, notamment pour les parties qui se trouvent aux intersections, comme par exemple la prise en compte de l'humain dans les systèmes à base de connaissances, également couverte par le collège [Science de l'Ingénierie des Connaissances](#).

Programme de travail

Les missions du collège IH concernent le soutien à l'organisation de manifestations scientifiques (conférences, ateliers), l'animation de groupes de travail, l'édition de dossiers techniques ou de numéros spéciaux de journaux sur des thématiques d'intérêt pour la communauté, ainsi que la diffusion et la communication autour des recherches menées par les communautés françaises sur les disciplines ciblées.

Ce collège contribue aux actions initiées par l'AFIA sur la mise en place de journées bilatérales, notamment les [journées communes](#) Environnements Informatiques pour l'Apprentissage Humain et

Intelligence Artificielle (EIAH & IA), Interaction Homme-Machine et Intelligence Artificielle (IHM & IA), et Réalité Virtuelle et Intelligence Artificielle (RV & IA). Il proposera également un soutien similaire à d'autres événements, ponctuels ou récurrents, relevant de son périmètre scientifique.

Il contribue également au [Bulletin](#) de l'AFIA en proposant notamment des dossiers sur l'état des lieux des recherches combinant IHM et IA, comme cela vient d'être fait pour la thématique [EIAH & IA](#).

Les actualités du collège IH, et plus largement des communautés françaises de recherche associées aux journées, sont publiées régulièrement sur un site web dédié au collège.

Comité de pilotage

Le comité de pilotage du collège IH est constitué de chercheurs spécialisés dans les domaines des IHM, des EIAH et de la RV avec une approche IA :

- Armelle BRUN, Université de Lorraine, LORIA, Nancy ;
- Pierre CHEVAILLIER, ENI Brest, CERV, Brest ;
- Nadine COUTURE, ESTIA, LABRI, Bordeaux ;
- Catherine FARON, Université Côte d'Azur, I3S, Sophia Antipolis ;
- Benoit LE BLANC, ENSC Bordeaux INP, IMS, Bordeaux ;
- Marie LEFEVRE, Université Lyon 1, LIRIS, Lyon ;
- Domitile LOURDEAUX, Université de Technologie de Compiègne, HEUDIASYC, Compiègne ;
- Vanda LUENGO, Sorbonne Université, LIP6, Paris ;
- Nicolas SABOURET, Université Paris-Saclay, LIMSI, Saclay.

Contacts

Coordinateur du collège : benoit.leblanc@ensc.fr.

Trois membres du comité de pilotage du collège sont membres du Conseil d'Administration de l'AFIA :

- Catherine FARON, faron@i3s.unice.fr ;
- Marie LEFEVRE, marie.lefevre@liris.cnrs.fr ;
- Domitile LOURDEAUX, domitile.lourdeaux@hds.utc.fr.

Collège Représentation et Raisonnement

Objectif du collège

L'objectif du Collège **Représentation et Raisonnement** (R&R) de l'**AFIA** est d'animer les communautés de recherche françaises dans ce domaine. Les thématiques de recherche sont relatives aux méthodes et outils fondamentaux de l'Intelligence Artificielle. Elles portent sur :

- la définition de modèles de représentation des informations (croyances, connaissances, préférences, obligations et permissions, actions, incertitude, confiance, réputation) comme les langages des logiques classiques ou non classiques, les modèles possibilistes, les ontologies, les langages à base de contraintes, les représentations graphiques, etc. ;
- la définition et l'automatisation de raisonnements sur ces informations : raisonnement spatio-temporel, dynamique des informations, révision de croyances, fusion d'informations symboliques, raisonnement par argumentation, raisonnement causal, raisonnement abductif, raisonnement à partir de cas, etc. ;
- la perspective algorithmique et de représentation pour des concepts utilisés dans des théories connexes comme la théorie des jeux ou la théorie du choix social (équilibre, stratégie gagnante, manipulation, etc.) : théorie des jeux algorithmique et choix social computationnel ;
- la mise au point de méthodes de codage des informations et d'algorithmes de traitement efficaces : compilation de connaissances, SAT, ASP, etc. ;
- la modélisation formelle de l'interaction : entre utilisateurs et systèmes informatiques, entre entités informatiques autonomes (agents) ;
- et généralement le lien avec différentes techniques liées à la décision, la planification, l'ordonnancement, le diagnostic, l'apprentissage, les sciences des données, etc.

Ces thématiques couvrent de très nombreux contextes d'application, comme par exemple le Web sémantique, le Web des données, les systèmes de recommandation ou d'aide à la décision, les agents conversationnels et assistants personnels, la programmation des jeux, la robotique, etc.

Programme de travail

Le collège R&R est impliqué dans les activités suivantes :

- IAF : Journées d'Intelligence Artificielle Fondamentales. Ces journées ont lieu tous les ans, à l'initiative du comité IAF. Elles sont articulées autour de 3 ou 4 exposés de synthèse invités, ainsi que d'un programme constitué après appel à communication ;
- JFPDA : Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes. Ces journées ont lieu tous les ans, à l'initiative des anciens présidents de comités de programme JFPDA et des participants à la liste de diffusion jfpda@loria.fr. Elles sont articulées autour de 2 ou 3 exposés de synthèse invités, ainsi que d'un programme constitué après appel à communication ;
- JFPC : Journées Francophones de Programmation par Contraintes. Ces journées ont lieu tous les ans à l'initiative de l'AFPC (Association Française pour la Programmation par Contraintes). Elles sont articulées autour de 2 ou 3 exposés de synthèse invités, ainsi que d'un programme constitué après appel à communication ;

- des ateliers thématiques, qui ont lieu lors d'autres évènements et la contribution à l'organisation de journées communes.

Le collège consacre une partie de son budget de fonctionnement à l'attribution de bourses permettant à des étudiants d'assister à l'un des évènements organisés par le collège, ou d'autres manifestations pertinentes au regard des thématiques scientifiques.

Comité de pilotage

Le comité de pilotage du collège est constitué des personnes suivantes :

- Elise BONZON, Université Paris-Descartes, LIPADE, Paris ;
- Zied BOURAOUI, co-président du comité de programme IAF, Université d'Artois, CRIL, Lens ;
- Sylvie DOUTRE, co-présidente du comité de programme IAF, Université Toulouse 1 Capitole, IRIT, Toulouse ;
- Sébastien KONIECZNY, ancien directeur du GDR IA, CNRS, CRIL, Lens ;
- Frédéric MARIS, membre du CA et représentant pour l'AFIA, Université Toulouse 3 Paul Sabatier, IRIT, Toulouse ;
- Nicolas MAUDET, Université Pierre et Marie Curie, LIP6, Paris ;
- François SCHWARZENTRUBER, ancien président du comité de programme JFPDA, ENS Rennes, IRISA, Rennes ;
- Laurent SIMON, membre du CA de l'AFIA, président de l'AFPC, Université de Bordeaux, LaBRI, Bordeaux ;
- Elise VAREILLES, membre du CA de l'AFPC ISAE Supaero, Toulouse ;
- Bruno ZANUTTINI, ancien président du comité de programme JFPDA, Université Caen Normandie, GREYC, Caen.

La composition du comité est amenée à être modifiée bi-annuellement.

Contacts

Coordinateur du collège : frederic.maris@irit.fr.

Listes de diffusion : bull-i3@irit.fr, bull-ia@gdria.fr, jfpda@loria.fr.

Deux membres du comité de pilotage du collège sont membres du Conseil d'Administration de l'AFIA :

- Frédéric MARIS, frederic.maris@irit.fr ;
- Laurent SIMON, lsimon@labri.fr.



AFIA

Association française
pour l'Intelligence Artificielle

Collège Science de l'Ingénierie des Connaissances

Objectif du collège

À l'heure numérique, les données et les outils se multiplient mais assurer un accès intelligent aux données reste un défi et ce malgré les langages et les technologies qui sont maintenant à disposition des informaticiens et des experts de domaine. Partager des données et des connaissances au sein d'une communauté, d'une entreprise ou sur le web suppose leur explicitation, leur représentation, leur mise en relation, leur diffusion et leur maintenance. L'ingénierie des connaissances est au cœur de ces problématiques.

L'essor des sciences et technologies de l'information et de la communication, et notamment des technologies du web, dans l'ensemble de la société engendre des mutations dans les pratiques individuelles et collectives. L'ingénierie des connaissances accompagne cette évolution, en inventant les modèles, les méthodes, les algorithmes et les outils permettant l'intégration de connaissances et de raisonnements automatiques sur ces connaissances dans des environnements informatiques. De fait, on constate aujourd'hui une omniprésence des connaissances, en même temps qu'un décloisonnement des communautés de recherche en Intelligence Artificielle pour répondre aux problèmes qui se posent. Ainsi, la construction des systèmes à base de connaissances fait appel à l'ingénierie des connaissances, au traitement automatique des langues, à l'apprentissage automatique, à la fouille de données et à l'interaction homme-machine.

La représentation, l'intégration, l'interrogation, la découverte de connaissances ou le contrôle de la qualité de données et de connaissances issues de données textuelles, semi-structurées, graphes, ou multimédia, potentiellement multisources et évolutives, associent ingénierie des connaissances avec recherche d'information, recherche opérationnelle, raisonnement, et analyse de données ; en particulier, la définition de principes évolutifs de représentation des connaissances et de raisonnement dans le cadre unifié des graphes de connaissances devient une thématique émergente traitant par exemple du rôle des ontologies et de l'intégration de données dans ces graphes ; la modélisation des environnements informatiques, de ses acteurs humains ou logiciels, allie ingénierie des connaissances et systèmes multi-agents ; enfin l'ingénierie des connaissances touche à des thématiques classiques dans les communautés de l'interaction homme-machine et des environnements informatiques pour l'apprentissage humain. En plaçant ainsi l'utilisateur au cœur des systèmes informatiques, l'ingénierie des connaissances est sollicitée pour assister l'utilisateur dans le traitement de la masse de données disponibles, pour les visualiser, contextualiser, classer, prédire, approximer, sécuriser, fiabiliser, recommander, expliquer, valider, pallier leur imprécision voire incohérence.

Programme de travail

Le collège [Science de l'Ingénierie des Connaissances](#) (SIC) de l'[AFIA](#) a premièrement un rôle de diffusion de l'information. Il gère une liste de diffusion (comptant plus de 800 inscrits) sur l'ingénierie des connaissances et maintient une page web sur le site Web de l'[AFIA](#). Il contribue également à l'animation et au dynamisme de la communauté de recherche en ingénierie des connaissances. Il est moteur dans l'organisation de la conférence nationale Ingénierie des Connaissances (IC) et des Rencontres du

Web des Données. Il soutient l'organisation de la conférence internationale *Terminology & Ontology : Theories and applications* (Toth 2021), l'organisation d'une session spéciale *Interaction Management in Digitized Ecosystems* dans le cadre de SMC 2021 et l'organisation du workshop international *Toth : « Terminology, interoperability and Data integration : Issues and Challenges »*. Une journée thématique « Le temps dans tous ses états : détection, représentation et exploitation des aspects temporels dans les graphes de données » est prévue.

Ses membres contribuent également à la mise en place de certaines journées communes de l'AFIA avec d'autres sociétés savantes, à l'organisation d'ateliers sur la Plate-forme Intelligence Artificielle (PFIA). Enfin, ils participent activement à des événements internationaux tels que EKAW, K-CAP ou encore ISWC.

Comité de pilotage

Le comité de pilotage du collège est constitué de 10 chercheurs spécialisés dans le domaine de l'ingénierie des connaissances, tous membres des comités de programme de la conférence nationale Ingénierie des Connaissances (IC) et/ou de la conférence Journées Francophones des Ontologies (JFO), et pour la plupart anciens présidents d'IC :

- Marie-Hélène ABEL, Université de Technologie de Compiègne, HEUDIASYC ;
- Djamel BENSLIMANE, Université Claude Bernard Lyon 1, LIRIS ;
- Sandra BRINGAY, Université Paul Valéry Montpellier 3, LIRMM ;
- Jean CHARLET, Assistance Publique-Hôpitaux de Paris & INSERM, LIMICS ;
- Sylvie DESPRÉS, Université Paris 13, LIMICS ;
- Catherine FARON, Université Nice Sophia Antipolis, I3S et Inria ;
- Nathalie HERNANDEZ, Université Toulouse 2 Jean Jaurès , IRIT ;
- Nathalie PERNELLE, Université Paris 13, LIPN ;
- Raphaël TRONCY, EURECOM, Sophia Antipolis ;
- Haifa ZARGAYOUNA, Université Paris 13, LIPN.

Contacts

Coordinateur du collège : sylvie.despres@univ-paris13.fr.

Listes de diffusion : info-ic@inria.fr.

Deux membres du comité de pilotage sont membres du conseil d'administration de l'AFIA :

- Sandra BRINGAY, sandra.bringay@lirmm.fr ;
- Catherine FARON, faron@unice.fr.



AFIA

Association française
pour l'Intelligence Artificielle

Collège Systèmes Multi-Agents et Agents Autonomes

Objectif du collège

Le collège [Systèmes Multi-Agents et Agents Autonomes](#) (SMAA) de l'AFIA a pour mission d'organiser les activités du collège au sein de l'AFIA et d'assurer l'interaction entre l'AFIA et la communauté francophone SMA et Agents Autonomes, concernant leur volet IA. Il participe à l'animation scientifique au sein de l'AFIA, en coordination avec les communautés concernées, pour susciter l'implication des membres du collège dans les événements organisés par l'AFIA (p. ex. PFIA, CNIA) tant en participant aux comités scientifiques, qu'en organisant des manifestations d'intérêt pour la communauté IA en France.

Le collège SMAA évolue dans deux directions :

- Accentuation des interactions avec les communautés robotique, automatique d'une part et simulation, systèmes complexes d'autre part. Concrètement cela se traduira par l'organisation d'événements joints sur des travaux associant SMA et/ou agents conversationnels animés à ces disciplines. Lorsque d'autres champs d'IA seront ciblés, ces événements seront préparés en association avec les autres collèges de l'AFIA concernés.
- Mise en place de webinaires réguliers, issues des équipes impliquées dans le collège SMAA. Les séminaires d'équipes, soutenances de thèses, les soutenances de HDR pourront, sur base du volontariat être diffusées à l'ensemble du collège SMAA.

Programme de travail

Les missions du collège SMAA concernent l'organisation de manifestations scientifiques (conférences, journées thématiques), l'édition de dossiers techniques ou de numéros spéciaux de journaux sur des thématiques d'intérêt pour la communauté.

Le collège SMAA accompagne notamment l'organisation régulière des JFSMA et de WACAI :

- JFSMA 2021 à PFIA Bordeaux ;
- JFSMA 2022 à Saint-Etienne ;
- WACAI 2022 (lieu à définir).

Le collège SMAA accompagne également l'organisation des journées communes suivantes :

- Journée commune MACS & IA ;
- Journée commune Robotique & IA ;
- Journée commune Simulation & IA.

Il propose également un soutien similaire à d'autres événements, ponctuels ou récurrents, relevant de son périmètre scientifique.

Plus largement le collège envisage des actions d'animation à destination des doctorants, et des actions de médiation scientifique et de communication à destination du public.

Comité de pilotage

Le comité de pilotage du collège SMAA est constitué de chercheurs spécialisés dans le domaine des systèmes multi-agents et des agents autonomes, tous membres des comités de programme ou du comité consultatif des Journées Francophones en Systèmes Multi-Agents (JFSMA) et/ou du Workshop Affect, Compagnon Artificiel, Interaction (WACAI).

- Emmanuel ADAM, Université Polytechnique Hauts-de-France, LAMIH, Valenciennes ;
- Fabien MICHEL, Université de Montpellier, LIRMM, Montpellier ;
- Frédéric MIGEON, Université Toulouse 3 Paul Sabatier, IRIT, Toulouse ;
- Maxime MORGE, Université de Lille 1, CRIStAL, Lille ;
- Magalie OCHS, Université Aix-Marseille, LSIS, Marseille ;
- Gauthier PICARD, ONERA, Toulouse ;
- Nicolas SABOURET, Université Paris-Sud, LIMSI, Saclay ;
- Olivier SIMONIN, INSA Lyon, CITI, Lyon ;
- Mahdi ZARGAYOUNA, IFSTTAR, Paris.

Contacts

Coordinateur du collège : emmanuel.adam@uphf.fr.

Listes de diffusion : sma@loria.fr, acai@poleia.lip6.fr.

Deux membres du comité de pilotage sont membres du conseil d'administration de l'AFIA :

- Emmanuel ADAM, emmanuel.adam@uphf.fr ;
- Gauthier PICARD, gauthier.picard@emse.fr.

Collège Technologies du Langage Humain

Objectif du collège

Les Technologies du Langage Humain (TLH) proposent des méthodes permettant une communication homme-machine naturelle, pouvant s'étendre à une interaction homme-homme médiée. Ainsi, les TLH permettent d'analyser, d'interpréter et de produire des actes du langage écrit, parlé ou signé, mais aussi d'interagir avec des données langagières. Ainsi, les TLH englobent traditionnellement le Traitement Automatique des Langues (TAL), la Communication Parlée (CP) et leurs applications les plus emblématiques comme la Recherche d'Information (RI) et la Traduction Automatique.

L'étude du langage humain est une entreprise multidisciplinaire qui nécessite une expertise dans les domaines de la Linguistique, de la Psychologie, des Sciences Cognitives, des Sciences du Numérique, et notamment de l'Intelligence Artificielle (IA). Les TLH occupent une part entière en IA sous le prisme du Test de Turing. Ainsi, elles regroupent tous les axes de recherche de l'IA dans le cadre des données langagières comme la représentation (*e.g.* plongements lexicaux, analyse syntaxique), le raisonnement (*e.g.* systèmes de question-réponse), la planification (*e.g.* argumentation), l'apprentissage (*e.g.* analyse de sentiments), ou même l'intelligence collective (*e.g.* détection de communautés). Créer des modèles pouvant interagir de manière élégante et naturelle en utilisant la langue nécessite une compréhension profonde de l'acoustique, de la phonétique et de la prosodie pour le langage oral d'une part (dans le cadre de la reconnaissance et de la synthèse de la parole), et d'autre part, de la morphologie, de la syntaxe, de la sémantique et de la pragmatique pour le langage écrit ou signé (dans le cadre de l'analyse et de la génération de textes). Seulement à cette condition les applications des TLH peuvent offrir la promesse d'un accès « universel » à l'information, notamment à travers les moteurs de recherche, la traduction automatique, le résumé de textes, la veille automatique ou les systèmes de question-réponse. La compréhension globale du langage permet également de caractériser les textes suivant leurs objectifs communicationnels. Ainsi, l'analyse des sentiments et des émotions, l'identification de discours haineux, la détection de plagiat, l'identification et la vérification du locuteur proposent autant de cadres applicatifs importants pour les sociétés numériques.

Puisqu'à l'ère du numérique les données langagières prolifèrent dans des quantités toujours plus importantes et accessibles (*big data*), les champs d'applications des TLH sont nombreux. Ainsi, les archives numériques, les réseaux sociaux, les plateformes collaboratives, les entretiens clients, les textos, les courriels, les commentaires sur des plateformes de vente en ligne sont autant de matières premières pour le développement d'applications des TLH. En particulier, quelques domaines privilégiés sont la santé, l'éducation, le droit, le journalisme et le handicap, mais d'autres émergent comme la finance, l'agriculture, la sécurité, le marketing et les humanités numériques.

Parallèlement au développement de modèles des TLH, de nombreux défis connexes doivent être pris en compte dans le cadre d'une démarche globale, comme la confiance (*e.g.* reproductibilité, explicabilité, confidentialité), l'éthique (*e.g.* biais d'apprentissage, représentativité) et l'évaluation (*e.g.* métriques dédiées) des systèmes proposés.

Le collège [Technologies du Langage Humain](#) (TLH) de l'[AFIA](#) a donc pour mission de promouvoir l'animation et l'interaction scientifiques entre les communautés TAL, CP et RI, et l'ensemble des communautés en IA ayant des intérêts communs dans le but de consolider les collaborations transversales.

Programme de travail

Afin de mener à bien sa mission, le collège TLH s'engage à soutenir l'organisation de manifestations scientifiques (conférences, ateliers), animer des groupes de travail, éditer des dossiers techniques, organiser des journées thématiques et diffuser et communiquer autour des recherches des communautés françaises du TAL, de la CP, de la RI et de l'IA.

Le collège TLH s'engage à servir de canal de communication entre l'AFIA et ses collègues ainsi qu'avec l'ATALA (Association pour le Traitement Automatique des Langues), l'ARIA (Association Francophone de Recherche d'Information et Applications), et l'AFCP (Association Francophone de la Communication Parlée).

Comité de pilotage

Le comité de pilotage du collège est constitué de 10 chercheuses et chercheurs spécialisés dans le domaine du TAL, de la CP et de la RI.

- Florian BOUDIN, Université de Nantes, LS2N, Nantes ;
- Davide BUSCALDI, Université Paris 13, LIPN, Villetaneuse ;
- Gaël DIAS, Université de Caen Normandie, GREYC, Caen ;
- Emmanuelle ESPERANÇA-RODIER, Université Grenoble Alpes, LIG, Grenoble ;
- Corinne FREDOUILLE, Université d'Avignon, LIA, Avignon ;
- José MORENO, Université Toulouse 3 Paul Sabatier, IRIT, Toulouse ;
- Aurélie NÉVÉOL, CNRS, LISN, Saclay ;
- Yannick PARMENTIER, Université de Lorraine, LORIA, Nancy ;
- Mathieu ROCHE, CIRAD, TETIS, Montpellier ;
- Serena VILLATA, CNRS, I3S, Sophia Antipolis.

Contacts

Coordinateur du collège : mathieu.roche@cirad.fr.

Listes de diffusion :

- ln@cines.fr (TAL) ;
- parole@listes.afcp-parole.org (CP) ;
- info-aria@lsis.org (RI).

Un membre du comité de pilotage est membres du conseil d'administration de l'AFIA :

- Gaël DIAS, gael.dias@afia.asso.fr.



Afia

Association française
pour l'Intelligence Artificielle

Prix de Thèse IA 2022 et 2023

Prix de Thèse IA 2022

Le jury

- Présidente : Hélène FARGIER ;
- Membres : Meghyn BIENVENU, Isabelle BLOCH, Antoine CORNUEJOLS, Sébastien DESTERCKE, Jean-Gabriel GANASCIA, Jérôme LANG, Pierre MARQUIS, Marie-Christine ROUSSET, Thomas SCHIEX, Mathieu SERRURIER, Florent TEICHTEL-KOENINGSBUCH.

Les lauréats

- 1^{er} prix : Cédric COLAS, « *Towards Vygotskian autotelic agents : learning skills with goals, language and intrinsically motivated deep reinforcement learning* », dir. : Pierre-Yves OUDEYER, Olivier SIGAUD, le 30/06/2021, Université de Bordeaux, Inria ;
- accessit (ex-æquo) : Vincent LE GUEN, « *Deep learning for spatio-temporal forecasting – application to solar energy* », dir. : Nicolas THOME, 30/11/2021, Université Paris, CEDRIC, HESAM ;
- accessit (ex-æquo) : Elise PERROTIN, « *Approches légères pour le raisonnement sur les connaissances et les croyances* », dir. : Andreas HERZIG, Emiliano LORINI, 11/10/2021, Université Toulouse 3, IRIT.

Prix de Thèse IA 2023

Le jury

- Présidente : Hélène FARGIER ;
Isabelle BLOCH, Antoine CORNUEJOLS, Sébastien DESTERCKE, Jean-Gabriel GANASCIA, Jérôme LANG, Pierre MARQUIS, Edouard PAUWELS, Marie-Christine ROUSSET, Thomas SCHIEX, Mathieu SERRURIER, Bruno ZANUTTINI.

Les lauréats

- 1^{er} prix (ex-æquo) : Vincent GRARI, « *Adversarial mitigation to reduce unwanted biases in machine learning* », dir. : Marcin DETYNIĘCKI et Sylvain LAMPRIER, 22 Mai 2022, Sorbonne Université ;
- 1^{er} prix (ex-æquo) : Munyque MITTELMANN, « *Logiques pour la représentation et la conception d'enchères* », dir. : Laurent PERRUSSEL, 01/09/2022, Université Toulouse Capitole, IRIT.

**AFIA**Association française
pour l'Intelligence Artificielle

Bulletins

Le [Bulletin](#) de l'AFIA fournit un cadre de discussions et d'échanges au sein de la communauté. Toutes les contributions, pour peu qu'elles aient un intérêt général, sont les bienvenues. Le bulletin contient des rubriques régulières de comptes rendus des conférences, journées et autres événements que l'AFIA organise ou parraine, les résumés d'HDR et de Thèses de Doctorat, et un Dossier qui dresse un état de l'art sur un domaine particulier de l'IA, présente des équipes françaises de recherche en IA (académiques ou industrielles), ou PFIA. Les bulletins de l'AFIA sont accessibles librement depuis le site de l'AFIA.

Le comité de rédaction

Le comité de rédaction 2022 avait Dominique LONGIN comme rédacteur en chef, aidé de Emmanuel ADAM, Grégory BONNET (rédacteur adjoint) et Gaël LEJEUNE. En 2023, c'est la même équipe qui a souhaité continuer à s'investir dans le comité de rédaction.

Bulletin 114

Dossier « PFIA 2021 »

octobre 2021

- Le dossier est consacré à PFIA 2021 (Bordeaux, du 28 juin au 2 juillet 2021) qui a hébergé pas moins de 7 conférences (APIA, CNIA, IC, JFPDA, JFSMA, JIAF, RJCIA), 3 journées (Défense & IA, Jeux & IA, Santé & IA), des tutoriels, des conférences invitées et 3 prix AFIA de thèse. Les différents comptes rendus ont été centralisés par Engelbert Mephu NGUIFO.
- Ce bulletin comporte en outre un compte rendu de la journée PDIA sur le thème de « L'explicabilité en IA » qui s'est tenue le 8 avril 2021. Il contient également les thèses et HDR du 3^e trimestre 2021.

Bulletin 115

Dossier « Formations en IA »

janvier 2022

- Le dossier, sous la direction de Anne BOYER et Laurent VERCOUTER, recense les formations présentes dans trois des quatre instituts interdisciplinaires en IA (3IA), ainsi que celles de trois écoles universitaires de recherche (EUR), offre riche et diversifiée !
- Ce bulletin contient également les comptes rendus de FIIA 2021, en plus des thèses et HDR du dernier trimestres 2021 dont nous avons eu connaissance.

Bulletin 116

Dossier « IA & Explicabilité »

avril 2022

- Le dossier, dirigé par Nicolas MAUDET, présente pas moins de 12 contributions autour de cette thématique, alors que celle-ci est un des grands centres d'intérêt du moment en IA.
- Ce bulletin est complété par le compte rendu des journées « Réalité Virtuelle et Intelligence Artificielle » et « Perspectives et Défis de l'IA », et de la journée commune « AFIA & Réseau DEVS », ainsi que les thèses et HDR soutenues pendant le 1^{er} trimestre 2022 et dont nous avons eu connaissance.

Bulletin 117 Dossier « Équipes industrielles en IA » juillet 2022

- Ce dossier, dirigé par Yves DEMAZEAU, présente 9 équipes de recherche industrielles en IA au travers de leurs travaux du moment, équipes sollicitées auprès des membres du Collège Industriel de l'AFIA et des partenaires industriels de PFIA.
- Ce bulletin comporte également le compte rendu de la journée PDIA 2022 sur le thème « IA & créativité », ainsi que les thèses et HDR soutenues durant le 2^e trimestre 2022 et dont nous avons eu connaissance.

Bulletin 118 Dossier « PFIA 2022 » octobre 2022

- Le dossier de ce Bulletin est dirigé par Engelbert Mephu NGUIFO. Il est consacré à PFIA 2022 (Saint-Étienne, du 27 juin au 1^{er} juillet 2022) qui a hébergé 8 conférences (APIA, CNIA, IC, JFPC, JFPDA, JFSMA, JIAF et RJCIA), 4 journées (EIAH & IA, IoT & IA, Résilience & IA, Santé & IA), des tutoriels, des conférences invitées et la remise des 2 prix de thèse AFIA, en plus d'un événement consacré à l'année PROLOG.
- Un compte rendu des journées PFIA autour de l'IoT, de la santé et de la résilience est également disponible, ainsi qu'un résumé des exposés invités. Ce Bulletin recense également les thèses et HDR du 3^e trimestre de 2022 dont nous avons eu connaissance.

Bulletin 119 Dossier « Équipes académiques en IA » janvier 2023

- Ce Bulletin a été dirigé par Dominique LONGIN qui également dirigé son dossier. Ce dernier dévoile les travaux de 11 équipes de recherche en IA réparties dans toute la France.
- Ce bulletin présente également la liste des thèses et HDR soutenues durant le 4^e trimestre 2022 et dont nous avons eu connaissance.

Bulletin 120 Dossier « IA & Normes » avril 2023

- Le dossier très riche de ce Bulletin, dirigé par Nathalie NEVEJANS, était autour de la thématique « IA et normes » déclinée au travers de 16 contributions.
- Ce Bulletin contient également un compte rendu des Nuits de l'Info 2022, ainsi que la liste des thèses et HDR soutenues pendant le 1^{er} trimestre 2023 et dont nous avons eu connaissance.

Bulletin 121 Dossier « IA & Énergies » juillet 2023

- Ce second dossier thématique de 2023 a été dirigé par Nouredine HADHJ-SAID et rassemble près d'une dizaine de contribution différentes.
- Ce Bulletin contient également les thèses et HDR dont nous avons eu connaissance et soutenue durant la période du 2^e trimestre 2023.

**AFIA**Association française
pour l'Intelligence Artificielle

EFIA 2022

L'Association Française pour l'Intelligence Artificielle (AFIA) organise sa troisième journée « Enseignement et Formation en IA » sur le thème « IA pour l'enseignement ».

La journée a pour objectif d'échanger entre acteurs institutionnels, acteurs du monde industriel, chercheurs travaillant dans ce domaine, enseignants et praticiens intéressés par l'usage de l'IA en classes. L'idée étant de faire rencontrer les besoins des uns et la vision des autres sur l'utilisation de l'IA en éducation, son apport et ses perspectives à court et moyen termes.

Date et lieu

- Date : 06 janvier 2022
- Lieu : Sorbonne Université, CICSU, tour 44, 4 place Jussieu 75005 Paris

Programme

- 08h30 Accueil.
- 08h45 Ouverture de Benoit LE BLANC (Président de l'AFIA). Introduction par Gauthier PICARD, Marie LEFEVRE (membres du CA de l'AFIA) et Amel YESSAD (membre de l'AFIA).
- 09h30 « Les enjeux IA pour l'enseignant autour des projets opérationnels actuels (et futurs) » par Axel JEAN (DNE Paris).
- 10h00 « IA : former les enseignants du secondaire » par Colin DE LA HIGUERA (U. Nantes et UNESCO).
- 10h30 « GTNum Scolia » par Margarida ROMERO (U. Côte d'Azur).
- 11h00 Pause.
- 11h30 « Institutions, curriculums et pratiques : Quels usages, présents et à venir, de l'Intelligence Artificielle dans les apprentissages ? » par Agathe PAIGNEAU (L'IA pour l'École Paris).
- 12h00 « Un campus d'excellence de l'IA PV associé à une chaire IA » par Françoise STURBAUT (Lycée Paul Valéry Paris).
- 12h30 Buffet.
- 14h00 « L'IA pour l'éducation au sein de Lalilo » par Thomas SERGENT (Lalilo Paris).
- 14h30 « L'IA pour l'éducation au sein d'Évidence B » par Thierry DE VULPILLIÈRES (Evidence B Paris).
- 15h00 « L'IA pour l'éducation au sein de Nomad Education » par Julien ROTROU (Nomad Education Paris).
- 15h30 Pause.
- 16h00 Table ronde « Défis et apports de l'IA pour l'éducation » avec Armelle BRUN (U. Lorraine Nancy), Nathalie GUIN (U. Lyon 1), Vanda LUENGO (Sorbonne Université Paris) et Didier ROY (INRIA Bordeaux).
- 17h00 Clôture.

Organisation

Cet événement est organisé par Marie LEFEVRE (U. Lyon 1), Gauthier PICARD (ONERA Toulouse) et Amel YESSAD (Sorbonne Université Paris) pour le Conseil d'Administration de l'AFIA.

Inscriptions

Les inscriptions à la journée (gratuites pour les membres AFIA, 30 € sinon) sont obligatoires et à effectuer sur le site : <https://afia.asso.fr/inscription-efia/>. Le déjeuner est garanti à toutes les personnes qui se seront inscrites avant le 03/01/22, et seulement à celles-ci.

PDIA 2022

L'Association Française pour l'Intelligence Artificielle (AFIA) organise sa huitième journée annuelle « Perspectives et Défis en Intelligence Artificielle » sur le thème « IA et Créativité ».

La journée est composée d'interventions regroupées selon deux sessions : « Texte, IHM et IA » le matin, « Arts et IA » l'après-midi.

Date et lieu

- Date : 07 avril 2022
- Lieu : CNAM, Amphi Friedmann, Accès 33, 2 rue Conté, 75003 Paris

Programme

- 09h00 Accueil.
- 09h15 Ouverture de Benoit LE BLANC (Président de l'AFIA). Introduction par Fatiha SAÏS et Engelbert Mephu NGUIFO (membre du CA de l'AFIA).
- 09h30 « Repenser l'interaction avec les technologies d'apprentissage » par Baptiste CARAMIAUX (CNRS Paris).
- 10h30 « Intelligence artificielle pour assister l'idéation et la conception amont » par Alex GABRIEL (Université de Lorraine Nancy).
- 11h30 Pause.
- 11h45 « Machines à écrire : créer des programmes qui créent pour apprendre à se servir de l'IA » par Anne-Gwenn BOSSER (ENIB Brest).
- 12h45 Buffet.
- 14h00 « Angelia : une intelligence artificielle pour la musique électronique » par Jean-Claude HEUDIN (Artificial Creature Paris).
- 15h00 « Musique et IA pour instruments artificiels » par Jérôme NIKA (IRCAM Paris).
- 16h00 Pause.
- 16h15 « Créativité computationnelle liée à la musique » François PACHET (Spotify Paris).
- 17h15 Discussion.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Fatiha SAÏS, Fayçal HAMDI, Davy MONTICOLO et Engelbert Mephu NGUIFO pour le CA de l'AFIA et par les responsables des Collèges de l'AFIA.

Inscriptions

Les inscriptions à la journée (gratuites pour les membres AFIA, 30 € sinon) sont obligatoires et à effectuer sur le site : <https://afia.asso.fr/inscription-pdia/>. Le déjeuner est garanti à toutes les personnes qui se seront inscrites avant le 04/04/22, et seulement à celles-ci.

**AFIA**Association française
pour l'Intelligence Artificielle

FIIA 2022

L'Association Française pour l'Intelligence Artificielle (AFIA) organise sa septième « Forum Industriel de l'IA » sur le thème « Hybridation & IA ». Il prolonge le quatrième forum tenu en 2019 dont le thème était les « Systèmes mixtes ».

La journée est constituée de présentations pour permettre des échanges accrus entre académiques et industriels. Une dernière session est réservée à des présentations rapides par des industriels invités, dont candidats à rejoindre le Collège Industriel de l'AFIA.

Date et lieu

- Date : 6 Octobre 2022
- Lieu : **TOTEM** / Institut des Systèmes Complexes Paris IdF, 11 Place Nationale, 75013 Paris

Programme

- 09h00 Ouverture de David CHAVALARIAS (Directeur de l'ISC IdF) et Benoit LE BLANC (Président de l'AFIA). Introduction de la journée.
- 09h15 « *PINNs and other approaches to accelerate simulations* » par SINCLAIR Laboratory (Total Energy, EDF, Thales).
- 10h15 Pause.
- 10h45 « *Quantum machine learning at Pasqal* » par PASQAL.
- 11h25 « *Hybrid AI in the ANITI projects and for natural language processing* » par Nicholas ASHER (CNRS, IRIT).
- 12h30 Buffet.
- 13h30 « *Symbolic and data-driven AI for image interpretation* » par Céline HUDELLOT (Centrale Supélec).
- 14h10 « *Designing rule-based decision policy with reinforcement learning* » par Thales Research & Technology.
- 14h50 « *Rule-based-model discovery for business automation* » par IBM France.
- 15h30 Pause.
- 16h00 Présentations de sociétés concernées par l'IA en « Trois planches ».
- 16h05 « Une approche hybride pour accompagner la mémoire du REX dans l'élaboration d'avis technique » par Jean-Pierre COTTON et Alain BERGER (Ardans).
- 16h25 « De l'analyse d'images jusqu'au traitement des données numériques et textuelles ; au cœur des travaux de recherche appliquée » par Valérie REINER (Berger-Levrault).
- 17h00 Clôture.

Organisation

Cet événement est organisé par Patricia BESSON (Thales Research & Technology), Remy KUSTERS (IBM France) et Patrick FABIANI (Dassault Aviation) pour le Collège Industriel de l'AFIA.

Inscriptions

Les inscriptions à la journée (gratuites pour les membres AFIA, 30€ sinon) sont obligatoires et à effectuer sur le site : <https://afia.asso.fr/inscription-fia/>. Un lien de connexion à une salle visioconférence sera envoyé aux inscrits.

EFIA 2023

L'Association Française pour l'Intelligence Artificielle ([AFIA](#)) organise sa quatrième journée « Enseignement et Formation en IA » sur le thème « Compétences et métiers d'avenir en IA ».

La demi-journée (en distanciel) a pour but d'échanger autour des nouvelles compétences et métiers d'avenir en IA, avec les projets lauréats de l'Appel à Manifestations d'Intérêt « Compétences et Métiers d'Avenir » (AMI CMA 2022). Ce sera l'occasion d'identifier les innovations pédagogiques, les évolutions de contenus ainsi que la projection sur de nouveaux métiers d'avenir, dans le cadre de ces projets de grande envergure.

Date et lieu

- Date : 05 janvier 2023
- Lieu : Journée organisée en distanciel

Programme

- 13h30 Ouverture de Benoit LE BLANC (Président de l'AFIA). Introduction par Gauthier PICARD (membres du CA de l'AFIA et responsable du GT Enseignement).
- 13h45 « Stratégie d'accélération française sur la formation en IA » par Anne BOYER (MESR Paris).
- 14h00 « *Artificial Intelligence for All* Sorbonne Université » par Xavier FRESQUET (Sorbonne Université Paris) et Agnès DUDYCH (Sorbonne Université Paris).
- 14h30 « *SaclAI-School : Paris-Saclay AI Education Program* » par Frédéric PASCAL (Université Paris-Saclay) et Sarah COHEN-BOULAKIA (Université Paris-Saclay).
- 15h00 « Massification et accélération des Formations en IA par IP Paris et HEC Paris » par Dominique ROSSIN (Ecole Polytechnique Paris).
- 15h30 Pause.
- 15h45 « École Française de l'Intelligence Artificielle » par Chantal SOULÉ-DUPUY et Julie AL-ATRACH (Université Toulouse Capitole) et Jean-Louis ROCH (Université Grenoble Alpes).
- 16h45 Table ronde.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Laurent VERCOUTER pour l'AFIA et Gauthier PICARD pour le Conseil d'Administration de l'AFIA.

Inscriptions

Les inscriptions à la journée (<https://afia.asso.fr/inscription-efia/>) sont gratuites mais obligatoires.

**AFIA**Association française
pour l'Intelligence Artificielle

PDIA 2023

L'Association Française pour l'Intelligence Artificielle (AFIA) organise sa neuvième journée « Perspectives et Défis en Intelligence Artificielle » sur le thème « IA et écologie ».

La journée est composée d'interventions regroupées selon deux sessions : Impact de l'IA sur l'écologie en matinée ; l'IA au service de l'écologie l'après-midi.

Date et lieu

- Date : 04 Avril 2023
- Lieu : CNAM, Amphi Friedmann, Accès 33, 2 rue Conté, 75003 Paris

Programme

- 09h00 Accueil.
- 09h15 Présentation de l'AFIA . Introduction.
- 09h30 « Bénéfices/risques de l'IA pour l'environnement : que faire en contexte d'incertitude ? » par Julien LEFEVRE (Aix-Marseille Université, INT).
- 10h30 « Une dystopie de l'apprentissage sobre » par Denis TRYSTAM (Université Grenoble Alpes, Institut Universitaire de France).
- 11h30 Pause.
- 11h45 Courtes présentations de doctorants.
- 12h45 Buffet.
- 14h00 « L'IA pour l'agro-écologie » par Christine DILLMANN (Université Paris Saclay, INRAE).
- 15h00 « *Machine Learning for Climate Change and Environmental Sustainability* » par Claire MONTELEONI (University of Colorado Boulder, INRIA Paris).
- 16h00 Pause.
- 16h15 Courtes présentations de doctorants.
- 17h15 Discussion.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Anne-Laure LIGOZAT (LISN, Université Paris Saclay), Fatiha SAIS (LISN, Université Paris Saclay) pour le Conseil d'Administration de l'AFIA, et Fayçal HAMDY (CEDRIC, CNAM Paris).

Inscriptions

Les inscriptions à la journée (gratuites pour les membres AFIA, 30€ sinon) sont obligatoires et à effectuer sur le site : <https://www.linscription.com/pro/activite.php?P1=128448>. Le déjeuner est garanti à toutes les personnes qui se seront inscrites avant le 31/03/2023, et seulement à celles-ci.

Résilience & IA 2023

L'Association Française pour l'Intelligence Artificielle (**AFIA**) organise au travers de son collègue industriel sa seconde journée « Résilience et IA » au sein de la Plate-Forme Intelligence Artificielle 2023 (**PFIA 2023**).

Cette journée a pour objectif de faire un point sur les travaux menés sur la gestion de crise et la sortie de crise, communément appelé résilience, avec l'appui du traitement des données, de l'information et du Big Data. Outre le constat souvent appuyé sur des chiffres d'une crise ou d'un dysfonctionnement, les sorties de crise et de situations difficiles imposent la collecte et un traitement efficace des données pour une meilleure prise de décision traduite par des politiques de développement, de mise en place des outils, des réponses précises. Des techniques d'Intelligence Artificielle peuvent ou pourraient aider à la résilience.

Il sera aussi intéressant de voir pour des situations précises quelles sont les données massives ou non qui sont à disposition (accès libre par exemple) et des desiderata en terme des connaissances à extraire.

Date et lieu

— Date :

5 juillet 2023

— Lieu :

PFIA 23, Strasbourg

Programme

- 10h30 « *AI Research for Climate Change and Environmental Sustainability* » par Claire MONTELEONI (INRIA-Paris).
- 11h30 « Les défis du glissement de contexte géographique » par Theophile BAYET, Christophe DENIS, Alassane BAH, Jean-Daniel ZUCKER.
- 12h30 Repas.
- 14h40 « Fonder le concept de résilience sur la théorie de la viabilité dans le cas de dynamiques incertaines » par Jean-Denis MATHIAS (INRAE Clermont ARA).
- 15h40 Panel autour de la résilience et l'apport de l'IA.
- 16h30 « Escape-SG : Un jeu sérieux pour mieux préparer les évacuations de masse. Simulation à base d'agents » par Arnaud SAVAL, Mathieu BOURGAIS, Éric DAUDÉ, Pierrick TRANOUEZ, Oliviet GILLET.
- 17h00 « Gestion des connaissances partagées par des agents à ressources et connectivité limitées : étude, analyse et expérimentation » par Mohamed LIMAME, Julien HENRIET, Christophe LANG, Nicolas MARILLEAU.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Ghislain A. ATEMEZING (Mondeca, Paris) et Mihaela JUGANARU-MATHIEU (Mines de Saint Etienne, IMT).

Classification & IA 2021

L'Association Française pour l'Intelligence Artificielle (**AFIA**) et la Société Francophone de Classification (**SFC**) organisent la troisième journée « Classification et Intelligence Artificielle » sur le thème « Avancées dans le domaine non-supervisé ».

Date et lieu

- Date : 21 septembre 2021
- Lieu : en distanciel

Programme

- 09h30 Ouverture de Mohamed NADIF (Président de la SFC) et Benoit LE BLANC (Président de l'AFIA). Introduction par Mohamed NADIF (membre de la SFC) et Engelbert Mephu NGUIFO (membre du CA de l'AFIA).
- 10h00 « Classification par modèles de Markov cachés et prévision de la consommation énergétique de bâtiments » par Allou-Badara SAME (Université Gustave Eiffel Marne-la-Vallée).
- 10h45 « Forêts d'arbres aléatoires pour le clustering de données complexes » par Miguel COUCEIRO (Université de Lorraine Nancy).
- 11h30 « Quelques réflexions sur le clustering de documents textuels » par Éric GAUSSIER (Université Grenoble Alpes).
- 12h15 Pause déjeuner.
- 14h00 « Co-clustering de séries temporelles multivariées pour la validation du véhicule autonome » par Loïc GIRALDI (CEA Saint-Paul-lez-Durance).
- 14h45 « Modèles de blocs latents semi-supervisés par champs de Markov cachés » par Paul RIVERAIN (Thales Palaiseau), Simon FOSSIER (Thales Palaiseau) et Mohamed NADIF (Université de Paris).
- 15h30 Clôture.

Organisation

Cet événement est organisé par Mustapha LEBBAH représentant du Collège Apprentissage Artificiel de l'AFIA (**CAA**) et par Véronique CARIOU et Mohamed NADIF (SFC).

Inscriptions

Inscriptions gratuites mais obligatoires sur : <https://claia.sciencesconf.org/registration/>.

RI & IA 2021

Cette quatrième journée « Recherche d'Information et Intelligence Artificielle » prolonge les discussions entre les deux communautés, cette fois sur le thème « Accès interactif à l'information ».

L'objectif de la journée est de confronter les expériences de la recherche interactive dans différents domaines d'applications, tant du côté avancé de la recherche que du côté des applications industrielles.

Date et lieu

- Date : 09 décembre 2021
- Lieu : Sorbonne Université, Amphi 25, 4 place Jussieu 75005 Paris

Programme

- 08h30 Accueil.
- 09h00 Présentations de l'AFIA, par Benoit LE BLANC (Président de l'AFIA) et de l'ARIA par Benjamin PIWOWARSKI (Président de ARIA). Présentation de la journée par Gaël DIAS (Université Caen Normandie) et Jean-Pierre CHEVALET (Université Grenoble Alpes).
- 09h30 « Méthodes d'évaluation des systèmes de dialogue » par Sophie ROSSET (CNRS Saclay).
- 10h30 Pause-café.
- 11h00 Session « Agents conversationnels, Recherche interactive de documents, Systèmes de questions-réponses, Recommandation et simplification, Accès à l'information pour les non- voyants/malentendants ».
- 13h00 Buffet et session posters.
- 14h30 « Des tâches de recherche simples aux tâches de recherche complexes : tirer parti de l'interaction en recherche d'information » par Lynda TAMINE-LECHANI (Université Toulouse 3).
- 15h30 Pause-café.
- 16h00 Table ronde « Accès interactif à l'information » avec Ludovic DENOYER (Facebook Paris), Cédric LOPEZ (Emvista Montpellier), Catherine PELACHAUD (CNRS Paris), Sophie ROSSET (CNRS Saclay), Lynda TAMINE-LECHANI (Université Toulouse 3) et Charles TEISSÈDRE (Synapse Développement Toulouse).
- 17h00 Clôture.

Organisation

Cet événement est co-organisé par Gaël DIAS et Mathieu ROCHE pour le Collège Technologies du Langage Humain (TLH) de l'Association Française pour l'Intelligence Artificielle (AFIA), Jean-Pierre CHEVALLET et Benjamin PIWOWARSKI pour l'Association Francophone de Recherche d'Information et Applications (ARIA), avec le soutien du GdR TAL.

Inscriptions

Les inscriptions se font sur <https://jaii2021.sciencesconf.org/registration>. L'inscription est gratuite mais obligatoire. Elle inclut la participation au buffet et aux pause-café.

RV & IA 2022

L'AFIA et le GdR IG-RV du CNRS organisent une troisième journée commune « Réalité Virtuelle et Intelligence Artificielle » sur le thème « Environnements virtuels : adaptation du système à l'humain ou de l'humain au système ? ».

L'objectif de cette journée est de réunir actrices et acteurs du domaine, afin d'aborder des questions scientifiques, technologiques, ou des questions portant sur les facteurs humains et les usages, au travers de présentations invitées, de démonstrations et de contributions directes.

Date et lieu

- Date : 9 Mars 2022
- Lieu : Centre d'Innovation de l'UTC, 57 avenue de Landshut, 60200 Compiègne & distanciel

Programme

- 09h30 Accueil.
- 10h00 Ouverture de Dominique BECHMANN (Directrice du GdR IG-RV), Benoît LE BLANC (Président de l'AFIA) et Philippe BONNIFAIT (Directeur de [Heudiasyc](#)). Introduction par Domitile LOURDEAUX (UTC Compiègne) et Indira THOUVENIN (UTC Compiègne).
- 10h30 « Les agents virtuels dans les environnements immersifs d'apprentissage » par David PANZOLI (Université Toulouse Capitole).
- 11h15 « La réalité virtuelle en tant qu'outil de formation adapté aux pratiques de l'enseignant » par Ludovic HAMON (Le Mans Université Laval).
- 11h45 « Facteurs humains et réalité virtuelle : détecter les effets secondaires avec capteurs physiologiques et machine learning » par Alexis SOUCHET (CNRS Compiègne).
- 12h10 « Génération d'un profil dynamique du stress pour l'entraînement à la gestion de situations de crise » par Luca PELISSERO-WITOSLAWSKI (UTC Compiègne).
- 12h30 Buffet et présentation des posters.
- 14h00 Visite du *Cave Automated Virtual Environment* (CAVE) et des démonstrations.
- 15h00 « Environnements virtuels et modèles de décision pour l'interaction » par Marc MACÉ (CNRS Rennes).
- 15h30 « Toucher social pour l'interaction humain-agent incarné en environnement virtuel » par Fabien BOUCAUD (UTC Compiègne).
- 15h55 « Retours adaptatif en réalité augmentée basés sur l'état du conducteur de véhicule hautement automatisé lors de la reprise de contrôle » par Baptiste WOJTKOWSKI (UTC Compiègne).
- 16h15 Table Ronde.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Domitile LOURDEAUX pour le Collège Interaction avec l'Humain (IH) de l'Association Française pour l'Intelligence Artificielle (AFIA) et Indira THOUVENIN pour le GdR Informatique Géométrique et Graphique, Réalité Virtuelle et Visualisation (IG-RV) du CNRS.

Inscriptions

Les inscriptions à la journée sont gratuites mais obligatoires : <https://journee-rvia.hds.utc.fr/>.

Simulation & IA 2022

L'AFIA et le réseau DEVS/RED organisent une 1^{re} journée commune « Simulation et Intelligence Artificielle » sur le thème « Simulation pour l'IA & IA pour la simulation ».

La journée se tient dans le cadre des 4^{es} Journées Francophones de la Modélisation et de la Simulation ([JFMS 2022](#)).

Date et lieu

- Date : 28 mars 2022
- Lieu : Institut d'Etudes Scientifiques 20130 Cargèse

Programme

- 09h30 Accueil.
- 09h45 Ouverture de Paul-Antoine BISGAMBIGLIA (Coordinateur DEVS/RED) et Benoît LE BLANC (Président de l'AFIA). Introduction par Paul-Antoine BISGAMBIGLIA (U. Corse Corte) et Fabien MICHEL (U. Montpellier).
- 10h00 « Simulation des systèmes multi-robots : outils et enjeux » par Olivier SIMONIN (INSA Lyon).
- 11h00 « Apprentissage automatique pour l'amélioration de la qualité des modèles » par Grégory BEURIER (CIRAD Montpellier).
- 11h45 Pause.
- 12h00 « Vers des équipes humains-IA : Ecosystèmes d'intelligence pour cas d'utilisation à fort enjeu » par Clodéric MARS (AI Redefined Montréal, CDN-QC).
- 12h30 Pause déjeuner.
- 14h30 « Simulations and risk, from simulations in simulation to agent-based model and reinforcement learning » par Arthur CHARPENTIER (U. Québec à Montréal, CDN-QC).
- 15h30 « SCAMP : a stigmergic approach to modeling intelligent behavior » par H. VAN DYKE PARUNAK (Parallax Advanced Research, Beavercreek, USA-OH).
- 16h30 Pause.
- 17h00 « Facing complexity with self-organization » par Carlos GERSHENSON (U. Nacional Autonoma, Mexico, MX).
- 18h00 Clôture.

Organisation

Cet événement est organisé par Fabien MICHEL pour le Collège Systèmes Multi-Agents et Agents Autonomes ([SMAA](#)) de l'Association Française pour l'Intelligence Artificielle ([AFIA](#)) et Paul-Antoine BISGAMBIGLIA pour le réseau [DEVS/RED](#).

Inscriptions

Les inscriptions en présentiel sont à prendre depuis le site des 4^{es} Journées Francophones de la Modélisation et de la Simulation ([JFMS 2022](#)). Les inscriptions en distanciel sont gratuites mais obligatoires, et sont à effectuer sur [cette page](#).

**AFIA**Association française
pour l'Intelligence Artificielle

IoT & IA 2022

L'AFIA et l'IMT organisent une 2^e journée « Internet des Objets et Intelligence Artificielle » sur le thème « Intégration des modèles et technologies ».

Cette journée [Internet des Objets et IA](#) est inscrite au sein de la Plate-forme IA 2022 ([PFIA 2022](#)).

Date et lieu

- Date : 27 juin 2022
- Lieu : Espace Fauriel, Le Corbusier 5, 23 rue P. et D. Ponchardier, 42100 Saint-Étienne

Programme

- 09h00 « Visualisation de l'information adaptée au monitoring d'un processus industriel » par Élodie TOUFAILI (Berger-Levrault Lyon), Y. MILOUDI et Ch. BORTOLASO.
- 09h30 « Visualisation Node-RED des données sémantiques des capteurs de bâtiments » par Ghislain ATEMEZING (Mondeca Paris), Y. GRIPAY, S. SERVIGNE et F. LAFOREST.
- 10h00 Pause-café.
- 10h30 « WildCount » par Didier DONSEZ (U. Grenoble).
- 10h50 « Approches légères pour une solution de maison intelligente pour personnes âgées » par Sara KEBIR (U. Artois), C. CARRA et K. TABIA.
- 11h10 « Gestion prédictive embarquée et frugale pour le bâtiment intelligent » par Louis CLOSSON (U. Grenoble).
- 11h30 « Jumeau Numérique de l'Espace Fauriel » par Maxime LEFRANÇOIS (EMSE Saint-Étienne).
- 12h00 Déjeuner.
- 13h30 « Détecter des situations médicales d'importance en utilisant le Stream Reasoning sur des données issues de l'IoT » par Mathieu BOURGAIS (INSA Rouen), F. GIUSTOZZI et L. VERCOUTER.
- 13h50 « Développement de nouvelles stratégies de monitoring grâce à une approche Massive IoT » par Gwen MAUDET (IMT Nantes), M. BATTON-HUBERT, P. MAILLE et L. TOUTAIN.
- 14h30 Pause.
- 14h45 « Un langage dédié au domaine du Web Sémantique des objets contraints » par Fatma-Zohra HANNOU (EMSE Saint-Etienne).
- 15h05 « LiRoT » par Alexandre BENTO (INSA Lyon), L. MÉDINI, K. SINGH et F. LAFOREST.
- 15h25 « Une plateforme d'apprentissage fédéré pour IoT » par Hamza SAFRI (Berger-Levrault Toulouse), M. KANDI, Y. MILOUDI, Ch. BORTOLASO, D. TRYSTRAM, et F. DESPREZ.
- 15h45 « Détection d'intrusions fédérée et semi-supervisée pour l'IoT » par Ons AOUEDI (U. Nantes), K. PIAMRAT, G. MULLER et K. SINGH.
- 16h15 Pause.
- 16h45 « The Data Uncertainty Principle » par Leonardo LINGUAGLOSSA (Telecom Paris).
- 17h05 Clôture.

Organisation

Cet événement est organisé par Patrick ALBERS pour l'Association Française pour l'Intelligence Artificielle ([AFIA](#)) et Olivier BOISSIER pour l'Institut Mines-Télécom de Saint-Étienne ([IMT](#)).

Inscriptions

Inscriptions : <https://ci.mines-stetienne.fr/pfia2022/infos-locales/inscriptions/>.



Santé & IA 2022

La journée a pour objectif de faire un point sur les travaux menés actuellement en ingénierie des connaissances dans le domaine de la santé. Un intérêt particulier est porté aux approches permettant de prendre en compte l'incertitude et la complexité des données de santé en alliant des algorithmes d'apprentissage et des méthodes fondées sur les connaissances.

Cette journée **Santé et IA** est inscrite au sein de la Plate-forme IA 2022 (**PFIA 2022**).

Date et lieu

- Date : 28 juin 2022
- Lieu : Espace Fauriel, Le Corbusier 5, 23 rue P. et D. Ponchardier, 42100 Saint-Étienne

Programme

- 09h00 Conférence invitée de Gaël DIAS (U. Caen-Normandie).
- 10h00 Pause-café.
- 10h30 « Conceptions de phénotypes computationnels pour la recherche en santé publique » par Pegdwendé SAWADOGO (AP-HP Paris), Thomas GUYET et Étienne AUDUREAU.
- 10h50 « Combiner plongements de graphes et *clustering* pour l'alignement de connaissances pharmacogénomiques » par Pierre MONNIN (Mines Nancy), Cheddy RAÏSSI, Amedeo NAPOLI et Adrien COULET.
- 11h00 Pause.
- 11h15 « Intégration continue et architecture à base de containers pour exposer un modèle d'apprentissage profond comme un service » par Théophile TIFFET (CHU Saint-Etienne), Guillaume GUERDOUX et Cédric BOUSQUET.
- 11h35 « Facial movement assessment based on anomaly detection » par Eder Alejandro RODRIGUEZ MARTINEZ (U. Picardie), Olga POLEZHAeva, Félix MARCELLIN, Émilien COLIN, Lisa BOYAVAl, François-Régis SARHAN et Stéphanie DAKPE.
- 12h00 Déjeuner.
- 13h30 Conférence invitée de Gilles DEQUEN (U. Picardie).
- 14h30 Pause.
- 16h15 Pause-café.
- 16h45 [Départ en bus depuis le cours Fauriel.] Visite de la plateforme #FutureMedicine du Centre ingénierie et Santé des Mines de Saint-Étienne.

Organisation

Cet événement est organisé par Fleur MOUGIN pour le Collège Science de l'Ingénierie des Connaissances (**SIC**) de l'Association Française pour l'Intelligence Artificielle (**AFIA**) et Lina SOUALMIA pour l'Association française d'Informatique Médicale (**AIM**).

Inscriptions

Inscriptions : <https://ci.mines-stetienne.fr/pfia2022/infos-locales/inscriptions/>.

EIAH & IA 2022

L'AFIA et l'ATIEF organisent une quatrième journée « Environnements Informatiques pour l'Apprentissage Humain & Intelligence Artificielle » sur le thème « Systèmes de recommandation pour l'éducation ».

L'objectif de cette journée est de rassembler les chercheurs issus des communautés de l'Intelligence Artificielle et des Environnements Informatiques pour l'Apprentissage Humain pour échanger autour de problématiques, concepts et techniques communs. La journée est composée d'une session de présentation de travaux sur le thème de la recommandation pour l'éducation, et d'une session plus générale sur les interactions entre IA et éducation. Ces sessions seront agrémentées d'une conférence invitée sur les liens et les questions de recherche qui existent entre ces deux communautés.

Cette journée [EIAH et IA](#) est inscrite au sein de la Plate-forme IA 2022 ([PFIA 2022](#)).

Date et lieu

- Date : 29 juin 2022
- Lieu : Espace Fauriel, Le Corbusier 5, 23 rue P. et D. Ponchardier, 42100 Saint-Étienne

Programme

- 10h30 « Un survol de la modélisation de l'expertise et du défi de l'évaluer dans un contexte ouvert » par Michel DESMARAIS (École Polytechnique de Montréal).
- 12h00 Déjeuner.
- 13h30 « Fairness et confidentialité en IA pour l'éducation : risques et opportunités » par Jill Jênn VIE (Inria Palaiseau).
- 14h30 Pause.
- 14h45 « Essai de prédiction de la réussite à un exercice de programmation » par Maxime BOUTON (Université de Lille), Marielle LÉONARD, Yvan PETER et Nour EL MAWAS.
- 15h15 « Vers une Approche éthique pour la prédiction automatique du risque d'échec scolaire » par Fahima DJELIL (IMT Brest) et Jean-Marie GILLIOT.
- 16h15 Pause-café.
- 16h45 « Comparaison de l'utilisation d'un référentiel standard et d'un référentiel interne pour la recommandation d'objets d'apprentissage » par Molka TOUNSI DHOUB (U. Côte d'Azur) et Catherine FARON.
- 17h30 Discussion avec les participants.
- 18h00 Clôture.

Organisation

Cet événement est organisé par Marie LEFEVRE pour le Collège Interaction avec l'Humain (IH) de l'Association Française pour l'Intelligence Artificielle (AFIA) et Jean-Marie GILLIOT pour l'Association des Technologies de l'Information pour l'Education et la Formation (ATIEF).

Inscriptions

Inscriptions : <https://ci.mines-stetienne.fr/pfia2022/infos-locales/inscriptions/>.

Résilience & IA 2022

L'AFIA et l'IMT organisent une 1^{re} journée commune « Résilience et Intelligence Artificielle ».

Cette journée a pour objectif de faire un point sur les travaux menés sur la gestion de crise et la sortie de crise, communément appelé résilience, avec l'appui du traitement des données, de l'information et du Big Data. Outre le constat souvent appuyé sur des chiffres d'une crise ou d'un dysfonctionnement, les sorties de crise et de situations difficiles imposent la collecte et un traitement efficace des données pour une meilleure prise de décision traduite par des politiques de développement, de mise en place des outils, des réponses précises. Des techniques d'Intelligence Artificielle peuvent ou pourraient aider à la résilience. Il serait aussi intéressant de voir pour des situations précises quelles sont données massives ou non qui sont à disposition et des desideratas en termes des connaissances à extraire.

Cette journée **Résilience et IA** est inscrite au sein de la Plate-Forme IA 2022 (**PFIA 2022**).

Date et lieu

- Date : 30 juin 2022
- Lieu : Espace Fauriel, Le Corbusier 5, 23 rue P. et D. Ponchardier, 42100 Saint-Étienne

Programme

- 09h00 « La résilience selon la théorie de la viabilité » par Guillaume DEFFUANT (IRSTEA Aubière).
- 10h00 Pause-café.
- 10h30 « Annotation sémantique pour la géolocalisation d'entités spatiales dans des tweets » par Gaëtan CAILLAUT (BRGM Orléans), Cécile GRACIANNE, Samuel AUCLAIR, Nathalie ABADIE et Guillaume TOUYA.
- 11h00 « Méthodologies pour la résilience de l'intelligence artificielle » par Mathilde JAY (U. Grenoble Alpes), Laurent LEFEVRE et Denis TRYSTRAM.
- 11h30 « *Transport Network Design Based on Origin/Destination Clustering During the COVID-19 Pandemic Use Case* » par Matthieu GUILLOT (IFSTTAR Bron), Angelo FURNO, El-Houssaine AGHEZZAF et Nour-Eddin EL.
- 12h00 Déjeuner.
- 14h45 Panel avec les présentateurs et les inscrits autour des apports pratiques de l'IA dans la résilience.

Organisation

Cet événement est organisé par Ghislain ATEMEZING pour le Collège Industriel (CI) de l'Association Française pour l'Intelligence Artificielle (AFIA) et Mihaela JUGANARU pour pour l'Institut Mines-Télécom de Saint-Étienne (IMT).

Inscriptions

Inscriptions : <https://ci.mines-stetienne.fr/pfia2022/infos-locales/inscriptions/>.

EGC & IA 2023

L'Association Française pour l'Intelligence Artificielle (AFIA) et l'association internationale francophone d'Extraction et de Gestion des Connaissances (Association EGC) organisent, avec l'aide locale du groupe de travail Gestion et l'Analyse de données Spatiales et Temporelles (GAST), une journée « Gestion et Analyse des données Maritimes » (GAM' 23) sur le thème de la représentation, de la gestion, de l'analyse et du stockage des données maritimes. Cette journée réunira les acteurs de la recherche académique ou industrielle autour de cette thématique.

Avec la multiplication des capteurs, des satellites et des systèmes d'émission, une grande variété de données liées au monde de la mer est désormais disponible et nécessite d'être analysée. Cette analyse est cruciale pour répondre à différentes problématiques, par exemple celles liées à l'activité des navires en mer (sécurité maritime, routage, détection d'activités illégales) ou aux enjeux environnementaux (réchauffement climatique, préservation de la biodiversité, pollution en mer). Le but de cette journée thématique est donc de réunir les gens intéressés par le traitement de données maritimes, celles-ci peuvent être par exemple des positions de navires, des données météo, des données d'images ou satellitaires, des données de qualité de l'eau ou de pollution qu'il s'agit d'assimiler et de traiter afin d'extraire de l'information.

Date et lieu

- Date : 11 mai 2023
- Lieu : EPITA, 14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre

Programme

- 08h30 Accueil.
- 09h30 Mot d'accueil par Aurélie LEBORGNE (membre du comité de programme GAM) et par Thierry GERAUD (directeur du LRE-EPITA).
- 09h45 « Traitement et analyse de données de positions de navires pour le suivi du trafic maritime à différentes échelles » par Loïc SALMON (ISEA—Université de la Nouvelle-Calédonie).
- 10h30 « De l' AIS à la prédiction de temps d'arrivée : détection de routes maritimes » par Jacques EVERWYN (SINAY – Caen).
- 11h15 « Exploitation et croisement de données complexes d'aquaculture » par Nazha SELMAOUI (ISEA—Université de la Nouvelle-Calédonie).
- 12h00 Buffet.
- 13h30 « Les navires à l'escale : une approche des dynamiques maritimes caribéennes au regard des fréquentations portuaires » par Clément IPHAR (LETG – Université de Bretagne Occidentale).
- 14h00 « Maritime 4.0 : Challenges et opportunités dans la marine marchande » par Pedro MERINO LASO (ENSM — Nantes).
- 14h45 « Segmentation d'objets mobiles pour les images sous-marines en utilisant des réseaux de neurones de graphes » par Wieke PRUMMEL (MIA Lab — Université de La Rochelle).
- 15h30 « Surveillance en mer » par Olivier RICOU (LRE — Le Kremlin-Bicêtre).
- 16h15 Mot de la fin par Aurélie LEBORGNE, Nida MEDDOURI et Loïc ALMON (membre du comité de programme GAM).

Organisation

Cet événement est organisé par Nida MEDDOURI (LRE/EPITA Kremlin-Bicêtre), Aurélie LEBORGNE (ICube/UNISTRA Strasbourg) et Loïc ALMON (ISEA/UNC (Nouméa – Nouvelle-Calédonie).

IHM & IA 2023

L'Association Française pour l'Intelligence Artificielle (AFIA) et l'Association Francophone d'Interaction Homme-Machine (AFIHM) organise une sixième journée commune « IHM et IA » sur les thèmes « Concevoir des interactions avec et pour l'IA » (en matinée) et « Construire, évaluer, et déployer des IA de confiance » (l'après-midi).

Cette journée s'articulera autour de présentations, débats, posters et démos mettant en évidence les liens entre les deux disciplines.

Date et lieu

- Date : 7 juin 2023
- Lieu : Sorbonne Center for Artificial Intelligence (SCAI), 4 Pl. Jussieu, 75005 Paris

Programme

- 09h30 Café d'accueil.
- 10h00 Présentation de l'AFIA et de l'AFIHM.
- 10h15 « Composer les interactions avec les systèmes d'apprentissage automatique » par Jules FRANÇOISE (CNRS, Université Paris-Saclay).
- 10h45 « Évaluation et utilisations expertes des systèmes d'apprentissage automatique interactifs » par Nadia BOUKHELIFA (INRAE, Université Paris-Saclay).
- 11h15 « Concevoir par des voies détournées : comment les designers parviennent à façonner les systèmes algorithmiques » par Nolwenn MAUDET (ACCRA, Université de Strasbourg).
- 11h45 Débat « Quelle est la place du design et des designers dans la fabrique des systèmes intelligents ? » : échange entre Jules FRANÇOISE, Nadia BOUKHELIFA et Nolwenn MAUDET, modéré par Téo SANCHEZ.
- 12h30 Buffet.
- 14h00 « Les traces d'interaction comme support d'explications » par Béatrice FUCHS (LIRIS, Université Jean Moulin Lyon III).
- 14h30 « Confiance humain-IA pour la prise de décision : définitions, facteurs et évaluation au travers de prisme académique et industriel » par Oleksandra VERESCHAK (Sorbonne Université).
- 15h00 « Analyse et coordination de l'automatisation de décisions pour les professionnels » par Thomas BAUDEL (IBM, Université Paris-Saclay).
- 15h30 Débat « L'explicabilité : condition nécessaire ou suffisante pour mériter la confiance des utilisateurs ? » : échange entre Béatrice FUCHS, Oleksandra VERESCHAK et Thomas BAUDEL, modéré par Benoît LE BLANC.
- 16h00 Posters et démos.
- 17h30 Clôture.

Organisation

Cet événement est organisé par Benoît LE BLANC et Eya BEN CHAABEN pour l'AFIA, et Téo SANCHEZ pour l'AFIHM.

Inscriptions

Les inscriptions à la journée (gratuites pour les membres AFIA ou AFIHM, 30 € sinon) sont obligatoires et à effectuer sur le site : <https://www.linscription.com/pro/activite.php?P1=135557>. Le déjeuner est garanti à toutes les personnes qui se seront inscrites avant le 2 juin 2023, et seulement à celles-ci.

Jeux & IA 2023

L'Association Française pour l'Intelligence Artificielle (**AFIA**) et le groupe de travail Jeux et Planification Multi-Agents, Flexible, Temporelle, Épistémique et Contingente (**MAFTEC**) du groupement de recherche Raisonnement, Apprentissage, et Décision en Intelligence Artificielle (**GDR RADIA**) organisent conjointement une quatrième journée commune « **Jeux et IA** » au sein de la Plate-Forme Intelligence Artificielle 2023 (**PFIA 2023**).

Date et lieu

— Date :

3 juillet 2023

— Lieu :

PFIA 2023, Strasbourg

Programme

- 10h30 Présentation des Groupes de Travail.
- 10h45 « Vers plus de raisonnement dans EL-O : l'exemple de Hanabi » par Elise PERROTIN (CRIL).
- 11h10 « Évaluation de méthodes d'XAI diverses sur une tâche de pronostic d'e-sport » par Corentin BOIDOT (ENIB, Lab-STICC).
- 11h35 « Prolog et ontologies, une autre approche pour les comportements des PNJ » par Sylvain LAPEYRADE (Université Clermont-Auvergne, LIMOS).
- 12h00 Repas.
- 14h40 « Perspectives sur l'automatisation de l'évaluation de l'expérience de jeu » par Thomas CONSTANT (CNAM, CEDRIC).
- 15h05 « Modélisation récursive d'opposants dans les jeux à information incomplète » par Junkang LI (NukkAI et GREYC, Université de Caen Normandie).
- 15h30 Discussion.
- 16h10 Clôture.

Organisation

Cet événement est organisé par Anne-Gwenn BOSSER (Lab-STICC, Ecole Nationale d'Ingénieur de Brest) et Tristan CAZENAVE (LAMSADE, Université Paris-Dauphine) pour le Conseil d'Administration de l'AFIA, et Tiago DE LIMA (CRIL, Université d'Artois) et Bruno ZANUTTINI (GREYC, Université Caen Normandie) pour le GT MAFTEC.

Santé & IA 2023

L'Association Française pour l'Intelligence Artificielle (**AFIA**) et l'Association française d'Informatique Médicale (**AIM**) organisent conjointement une sixième journée commune « Santé et IA » au sein de **PFIA 2023** afin de faire un point sur les travaux actuels en ingénierie des connaissances pour la santé.

Date et lieu

- Date : 6 Juillet 2023
- Lieu : PFIA 23, Strasbourg

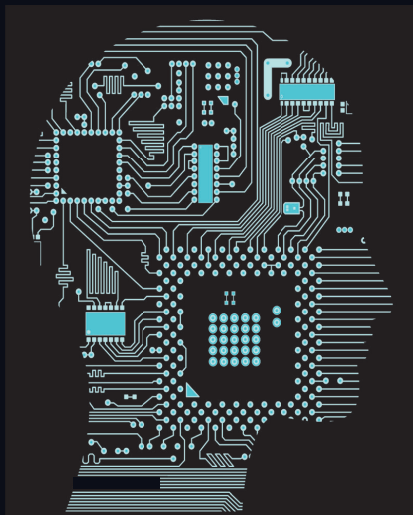
Programme

- 10h40 « Conception et visualisation d'un graphe de connaissances à partir de données en hématologie : Prise en charge de l'anémie chez l'adulte » par Soulaymane HODROJ, Sylvie DESPRES, Jérôme NOBECOURT, Florence CYMBALISTA.
- 11h10 « Graphe de connaissance et ontologie pour la représentation des données de la LLC » par Chiraz PIRIOU, Sylvie DESPRES, Jérôme NOBECOURT, Claudine IRLLES, Christine LE ROY.
- 11h25 « *Enhancing a Biomedical Ontology with Knowledge from Discharge Summaries* » par Sylvie DESPRES, Catherine DUCLOS, Chan LE DUC, Pascal VAILLANT.
- 11h40 « *Weak Controllability of multi-agent plans with uncertainty : towards temporal flexibility negotiation* » par Ajdin SUMIC, Thierry VIDAL, Mohamed HEDI KARRAY.
- 11h55 « *Towards Trustworthy-AI-by-Design Methodology for Intelligent Radiology Systems* » par Clotilde BRAYE, Jérémy CLECH, Arnaud GOTLIEB, Nadjib LAZAAR, Patrick MALLEA.
- 12h10 Repas.
- 14h40 « Détection automatique des pics d'un signal de pression intracrânienne : comparaison d'algorithmes combinant apprentissage profond et fonction de courbure » par Donatien LEGÉ, Marion PRUD'HOMME, Julien HENRIET.
- 15h10 « Analyse automatique de négations pour la radiologie et autres textes cliniques en français par modèles de langage » par Salim SADOUNE, Antoine FRABOULET, Antoine RICHARD, François TALBOT, Loïc BOUSSEL, Hugues BERRY.
- 15h40 « Anonymisation de documents médicaux en textes libres et en français via réseaux de neurones » par Antoine RICHARD, François TALBOT, David GIMBERT.
- 16h10 Pause.
- 16h30 « Un système d'aide au dialogue en santé intime des femmes » par Xingyu LIU, François PORTET, Didier SCHWAB.
- 16h45 « *Automatic detection of schwa in French hypersomniac patients using Automatic Speech Recognition* » par Colleen BEAUMARD, Vincent MARTIN, Yaru WU, Jean-Luc ROUAS, Pierre PHILIP.
- 17h15 « Explorer des mentions d'interventions non médicamenteuses dans des données issues des médias sociaux » par Alexis DELAFORGE, Jérôme AZÉ, Sandra BRINGAY, Caroline MOLLEVI, Arnaud SALLABERRY, Maximilien SERVAJEAN.
- 17h45 « *Heterogeneous incomplete multi-view data for Neurotoxicity biomarkers Identification* » par Quentin RUIN, David BALAYSSAC, Issam FALIH, Engelbert MEPHU NGUIFO.
- 18h00 Clôture.

Organisation

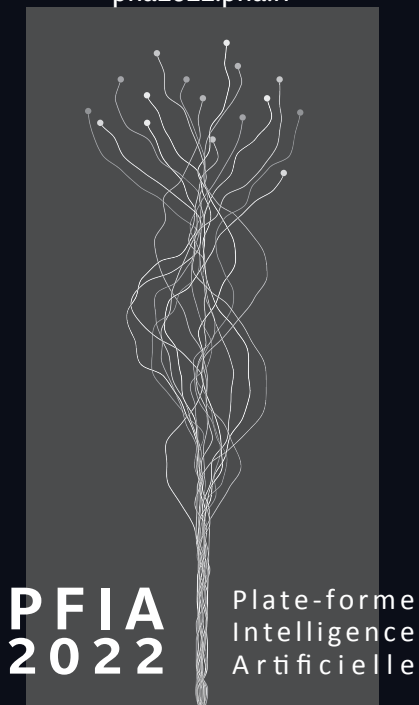
Cet événement est organisé par Fleur MOUGIN (AHeaD, Université de Bordeaux & Inserm) et Lina SOUALMIA (LITIS & LIMICS, Normandie Universités & Inserm).

La rencontre annuelle
de la communauté Intelligence Artificielle
autour de conférences et ateliers thématiques



SAINT-ÉTIENNE
27 juin - 1^{er} juillet 2022

pfia2022.pfia.fr



PFIA
2022 Plate-forme
Intelligence
Artificielle



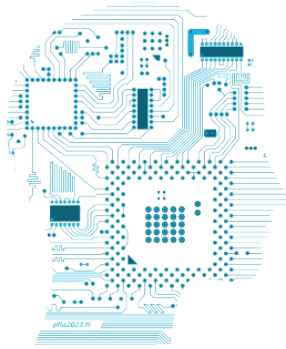
PROGRAMME

- Conférence Applications Pratiques de l'Intelligence Artificielle
- Conférence Nationale en Intelligence Artificielle
- Journées francophones d'Ingénierie des Connaissances
- Journées Francophone de Programmation par Contraintes
- Journée Francophone sur la Planification, la Décision et l'Apprentissage
- Journées Francophones sur les Systèmes Multi-Agents
- Journées d'Intelligence Artificielle Fondamentale
- Rencontre des Jeunes Chercheurs en Intelligence Artificielle
- Tutoriels
- Journée thématique EIAH et IA
- Journée thématique IoT et IA
- Journée thématique Résilience et IA
- Journée thématique Santé et IA
- IA dans le tissu industriel local



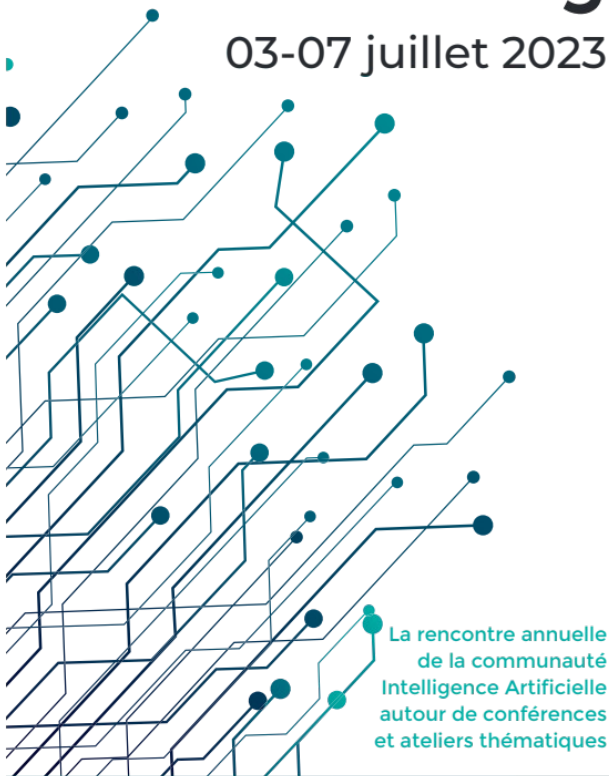
PFIA 2023

Plate-forme
Intelligence
Artificielle



Strasbourg

03-07 juillet 2023



La rencontre annuelle
de la communauté
Intelligence Artificielle
autour de conférences
et ateliers thématiques



pfia2023.pfia.fr

PROGRAMME

- APIA** : Applications Pratiques de l'Intelligence Artificielle
- CAp** : Conférence sur l'Apprentissage automatique
- CNIA** : Conférence Nationale en Intelligence Artificielle
- IC** : Journées francophones d'Ingénierie des Connaissances
- JFPC** : Journées Francophone de Programmation par Contraintes
- JFSMA** : Journées Francophones sur les Systèmes Multi-Agents
- JIAF** : Journées d'Intelligence Artificielle Fondamentale
- RJCIA** : Rencontres des Jeunes Chercheurs en Intelligence Artificielle
- SFC** : Rencontres de la Société Francophone de Classification

Tutoriels

- Journée thématique **ACAI** : Affects, Compagnons Artificiels et Interactions
- Journée thématique **Jeux & IA**
- Journée thématique **Santé & IA**
- Journée thématique **Résilience & IA**

Soirée IA en Alsace



Actes CNIA 2023
Conférence Nationale d'Intelligence Artificielle

Éditeurs : Président(e)s des Comités de Programme de PFIA 2023

Tables des matières

Strasbourg, au cœur de l'Intelligence Artificielle de demain	3
Comité de programmation	5
Comités de programme	5
Comité d'organisation	12
Logo partenaires PFIA 2023	13
E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier (JFSMA)	
Consommation adaptative par négociation continue	15
G. Bied, E. Perennes, S. Nathan, V. Naya, P. Caillou, B. Crepon, C. Gaillac, M. Sebag (APIA)	
RECTO : REcommandation diminuant la Congestion par Transport Optimal	25
L. Cervoni, J. Brasseur, M. Roudesli (APIA)	
Prévention et auto-rééducation de pathologies ostéoarticulaires assistées par intelligence artificielle	35
M.C. Cooper, A. Lequen, F. Maris (JIAF-JFPDA)	
Analysis of planning instances without search	41
S. Doutre, T. Duchatelle, M.-C. Lagasquie-Schiex (JIAF-JFPDA)	
Classes of Explanations for the Verification Problem in Abstract Argumentation	51
G. Dubuisson Duplessis, M. Richard, A.-L. Guénet (APIA)	
Segmentation de phases de dialogue dans des retranscriptions de conversations de centres d'appels	61
A. Ferdjaoui, S. Affeldt, M. Nadif (SFC)	
Modèles graphiques causaux interactifs pour les données textuelles	65
C. Fèvre, H. Zgaya-Biau, P. Mathieu, S. Hammadi (JFSMA)	
L'optimisation du covoiturage dynamique multi-saut	71
S. Forest, J.-C. Quinton, M. Lefort (CNIA)	
Champ neuronal et apprentissage profond de topologies pour la fusion multimodale	81
A. Godinot, E. Le Merrer, C. Penzo, F. Taïani, G. Tredan (RJCIA)	
Change-Relaxed Active Fairness Auditing	91
K. Le Gall, L. Bellanger, A. Stamm, D.A. Laplaud (SFC)	
Génération de données synthétiques de marche : application au cas de patients atteints de sclérose en plaques	97
A. Malapert, M. Schmied, D. Fissore, M. Pelleau, A. Picard Marchetto (JFPC)	
Jouer avec des cryptarithmes en programmation par contraintes	101
J. Rouzot, C. Artigues, P. Garnier, E. Hebrard, P. Lopez (JFPC)	
Une contrainte globale pour l'ordonnancement des transferts de données dans les missions spatiales	111
T. Soulard, F. Saïs, J. Raad, G. Quercini (IC)	
Étude de transférabilité des clés pour le liage de données entre graphes de connaissances	115

Strasbourg, au cœur de l'Intelligence Artificielle de demain

Nicolas Lachiche

Professeur, ICube, Université de Strasbourg

Président du Comité d'Organisation de la Plate-Forme IA 2023

Thomas Guyet, Davy Monticolo, Ahmed Samet

Chercheur INRIA Lyon, Professeur Université de Lorraine, Maître de Conférence INSA Strasbourg,

Membres du Conseil d'Administration de l'AFIA,

Co-Présidents du Comité de Programmation de la Plate-Forme IA 2023

L'AFIA et le laboratoire ICube s'associent pour organiser du 3 au 7 juillet 2023 la seizième Plate-Forme Intelligence Artificielle – [PFIA 2023](#). Les précédentes éditions se sont tenues à Palaiseau (1999), Grenoble (2001), Laval (2003), Nice (2005), Grenoble (2007), Hammamet (2009), Chambéry (2011), Lille (2013), Rennes (2015), Caen (2017), Nancy (2018), Toulouse (2019), Angers (2020), Bordeaux (2021) et Bordeaux (2022).

La plate-forme IA constitue un point de rencontre unique pour la communauté IA permettant de rapprocher les différentes disciplines qui la composent et d'établir des passerelles entre elles. À cette intention, la Plate-forme IA s'adresse à l'ensemble de la communauté francophone en IA pour aborder des problématiques communes.

L'Alsace compte des acteurs de premier plan dans le domaine de l'Intelligence Artificielle, avec de réelles compétences tant dans son cœur scientifique et technique que dans son intégration à d'autres technologies ou dans ses applications dans différents domaines. Une soirée organisée le mercredi 5 juillet 2023 autour de posters et de spécialités culinaires alsaciennes permettra aux industriels de la région d'échanger avec les chercheurs présents.

Pour son édition 2023, la Plate-Forme IA héberge les neuf conférences suivantes...

APIA	9 ^e Conférence Nationale sur les Applications Pratiques de l'IA	C. Launois et C. Roussey
CAp	25 ^e Conférence sur l'Apprentissage automatique	C. Laclau et R. Gaudel
CNIA	26 ^e Conférence Nationale d'Intelligence Artificielle	S. Bringay
IC	34 ^{es} Journées Francophones d'Ingénierie des Connaissances	C. Trojhan
JFPC	18 ^{es} Journées Francophones de Programmation par Contraintes	E. Vareilles
JFSMA	31 ^{es} Journées Francophones sur les Systèmes Multi-Agents	M. Morges
JIAF	17 ^{es} Journées d'Intelligence Artificielle Fondamentale	Z. Bouraoui et A. Wilczynski
RJCIA	21 ^{es} Rencontres des Jeunes Chercheurs en Intelligence Artificielle	B. Ravenet
SCF	28 ^{es} rencontres de la Société Francophone de Classification	P. Préa

les 4 journées thématiques et 5 ateliers suivantes...

ACAI	Atelier Affects, Compagnons Artificiels et Interactions en partenariat avec le GT CNRS ACAI
CECILIA	JAM « DriveToGæther »
Decade	DÉcouverte de Connaissances et Apprentissage dans les Données graphEs
Jeux et IA	en partenariat avec les groupes de travail « IA et Jeux » et MAFTEC du GDR RADIA
IN-OVIVE	INtégration de sources/masses de données hétérogènes et Ontologies, dans le domaine des sciences du VIVant et de l'Environnement
RADIA	Lancement du Groupement de Recherche (GDR) RADIA (Raisonnement, Apprentissage et Décision en IA)
Résilience et IA	2 ^e Journée Résilience et IA, en partenariat avec IMT
Santé et IA	7 ^e Journée Informatique Médicale et IA, en partenariat avec AIM
SOSEM	Science Ouverte et Sémantique
TLH	Réunion du collège TLH

... ainsi que dix [tutoriels](#) sur les thèmes :

- théorie des fonctions de croyance dans le cadre de l'apprentissage automatique ;
- agents : Comportements et protocoles de communications ;
- programmation orientée multiagent ;
- intelligence artificielle consciente, holistique et inclusive ;
- DLinker ;
- résolution parallèle de problèmes sous contraintes ;
- apprentissage automatique pour les séries temporelles ;
- approches par ensembles ;

- initiation à l'IA explicable ;
- introduction à la planification.

Neuf collègues ont également été invités pour des conférences en début de chaque demi-journée :

Leman Akoglu	Carnegie Mellon University	États-Unis
Anne-Laure Boulesteix	Ludwig-Maximilians-Universitat	Allemagne
Ulle Endriss	University of Amsterdam	Pays-Bas
Siham Essodaigui	Renault	France
Sarit Kraus	Université Bar-Ilan	Israël
Heiko Paulheim	University of Mannheim	Allemagne
Clément Raïevsky	Université de Grenoble	France
Bernd Stahl	University of Nottingham	Royaume-Uni
Devis Tuia	École Polytechnique Fédérale de Lausanne	Suisse

plus les 2 lauréats du prix de thèse de l'AFIA :

Vincent Grari	AXA	France
Munyeque Mittelmann	IRIT	France

Nous remercions les présidents et membres de comités de programme des conférences et journées hébergées, les orateurs, les membres des comités de programmation et d'organisation, nos partenaires institutionnels et industriels, ainsi que tous les participants, pour leurs contributions précieuses à la réussite de cette plate-forme.

L'ensemble des informations sur ces événements est disponible à partir du site de la Plate-Forme IA 2023 (<http://pfia2023.pfia.fr/>) dont le contenu sera maintenu jusqu'à l'été 2027.

Comité de programmation

Conseil d'Administration de l'Association Française pour l'Intelligence Artificielle

Président

- Thomas GUYET, AIstroSight, Inria, Centre de Lyon

Membres

- Davy Monticolo, Université de Lorraine, ERPI;
- Ahmed Samet, INSA Strasbourg, ICube.

Comités de programme

Conférence nationale sur les Applications Pratiques de l'IA (APIA)

Présidence

- Christelle Launois (Société Générale, Paris);
- Catherine Roussey (INRAE MISTEA, Montpellier).

Membres

- Florence Amardeilh (Elzeard, Bordeaux);
- Fabien Amarger (Logilab, Toulouse);
- Ghislain Ateazing (Mondeca, Paris);
- Nicolas Audebert (CNAM CEDRIC, Paris);
- Romain Benassi (Publicis Sapient, Paris);
- Alain Berger (Ardans, Montigny-le-Bretonneux);
- Sandra Bringay (LIRMM, Montpellier);
- Stephan Brunessaux (Sensei Consult, Louviers);
- Patrice Buche (INRAE IATE, Montpellier);
- Davide Buscaldi (LIPN, Université Sorbonne Paris Nord);
- Bruno Carron (Airbus defense and Space);
- Caroline Chopinaud (Hub France IA, Paris);
- Gaël de Chalendar (CEA LIST, Saclay);
- Yves Demazeau (LIG, Grenoble);
- Sylvie Despres (LIMICS, Bobigny);
- Valentina Dragos (Onera, Palaiseau);
- Catherine Faron (I3S, Sophia Antipolis);
- Bernard Georges (Société Générale, Paris);
- Céline Hudelot (CentraleSupélec MICS, Gif-sur-Yvette);
- Dino Ienco (INRAE TETIS, Montpellier);
- Hedi Karray (INP-ENIT, Tarbes);
- Christine Largouët (IRISA, Rennes);
- Arnaud Lallouet (Huawei Technologies Ltd, Boulogne-Billancourt);
- Mustapha Lebbah (DAVID, Université Paris-Saclay, Versailles);
- Dominique Lenne (Université de Technologie de Compiègne, Compiègne);
- Bertrand Le Saux (European Space Agency ESRIN, Rome);
- Domitile Lourdeaux (Université de Technologie de Compiègne, Compiègne);

- Sylvain Mahé (EDF Recherche et Développement, Chatou) ;
- Denis Marraud (Airbus Defence and Space) ;
- Juliette Mattioli (Thales) ;
- Marie-Christine Rousset (LIG, Grenoble) ;
- Céline Rouveirol (LIPN, Paris) ;
- Frederique Segond (INRIA, Montpellier) ;
- Françoise Soulié-Fogelman (Hub France IA, Paris) ;
- Elodie Thiéblin (Logilab, Toulouse).

Conférence sur l'Apprentissage automatique (CAp)

Présidence

- Charlotte Laclau ;
- Romaric Gaudel.

Membres

- Marta Avalos (Université de Bordeaux) ;
- Stephane Ayache (LIS, Aix-Marseille Université) ;
- Nicolas Baskiotis (MLIA, UPMC) ;
- Aurélien Bellet (INRIA Lille) ;
- Younès Bennani (LIPN, Université Sorbonne Paris Nord) ;
- Tristan Cazenave (LAMSADE, Université Paris Dauphine) ;
- Laetitia Chapel (IRISA, Université de Bretagne Sud) ;
- Julien Chiquet (MIA Paris-Saclay, INRAE) ;
- Marianne Clausel (Institut Elie Cartan, Université de Lorraine) ;
- Nicolas Courty (IRISA, Université de Bretagne Sud) ;
- Rémi Emonet (Laboratoire Hubert Curien, Université Jean Monnet) ;
- Florence D'Alché-Buc (Télécom Paris) ;
- Aurélien Garivier (UMPA, ENS de Lyon) ;
- Gilles Gasso (LITIS, INSA de Rouen) ;
- Pascal Germain (GRAAL, Université Laval) ;
- Remi Gilleron (Cristal, Université de Lille) ;
- Antoine Gourru (Laboratoire Hubert Curien, Université Jean Monnet) ;
- Yves Grandvalet (Heudiasyc, Université Technologie de Compiègne) ;
- Yann Guermeur (LORIA, Université de Lorraine) ;
- Vincent Guigue (LIP6, UPMC) ;
- Amaury Habrard (Laboratoire Hubert Curien, Université Jean Monnet) ;
- Hachem Kadri (LIS, Aix-Marseille Université) ;
- Mikaela Keller (Cristal, Université de Lille) ;
- Batiste Le Bars (INRIA Lille) ;
- Engelbert Mephu Nguifo (LIMOS, Université Blaise Pascal Clermont-Ferrand) ;
- Guillaume Metzler (ERIC, Université Lyon 2) ;
- Emilie Morvant (Laboratoire Hubert Curien, Université Jean Monnet) ;
- Michaël Perrot (INRIA Lille) ;
- Liva Ralaivola (Criteo) ;
- Marc Sebban (Laboratoire Hubert Curien, Université Jean Monnet) ;
- Marta Soare (LIFO, Université d'Orléans) ;
- Romain Tavenard (Université Rennes 2) ;
- Marc Tommasi (INRIA Lille) ;
- Titouan Vayer (INRIA Lyon) ;
- Christel Vrain (LIFO, Université d'Orléans).

Conférence Nationale en Intelligence Artificielle (CNIA)

Présidence

- Sandra Bringay, Université Paul-Valéry Montpellier.

Membres

- Jérôme Azé, Université de Montpellier ;
- Isabelle Bloch, Sorbonne Université ;
- Olivier Boissier, Mines Saint-Etienne, LIMOS ;
- Grégory Bonnet, Université de Caen Normandie ;
- Robert Bossy, INRAE Centre de Jouy en Josas, MaIAGE ;
- Armelle Brun, Université de Lorraine ;
- Sylvie Coste-Marquis, Université d'Artois ;
- Benjamin Dalmas, Centre de Recherche Informatique de Montréal ;
- Yves Demazeau, CNRS, LIG ;
- Arnaud Doniec, IMT Lille Douai ;
- Jean-Gabriel Ganascia, Sorbonne Université ;
- Eric Gaussier, Université Grenoble Alpes ;
- Guillaume Gravier, CNRS, IRISA ;
- Andreas Herzig, Université de Toulouse, IRIT ;
- Nathalie Hernandez, Université de Toulouse, IRIT ;
- Camille Kurtz, Université Paris Cité ;
- Nicolas Lachiche, Université de Strasbourg ;
- Frederique Laforest, INSA Lyon ;
- Florence Le Ber, École Nationale du Génie de l'Eau et de l'Environnement de Strasbourg ;
- Philippe Lenca, IMT Atlantique ;
- Marie-Jeanne Lesot, Sorbonne Université ;
- Pascal Poncelet, Université de Montpellier ;
- Catherine Roussey, INRAE Centre Occitanie-Montpellier, MISTEA ;
- Ana Roxin, Université Bourgogne Franche-Comté ;
- Nicolas Sabouret, Université Paris-Saclay ;
- Pascale Sébillot, INSA Rennes ;
- Nazha Selmaoui, Université de la Nouvelle Calédonie ;
- Catherine Tessier, ONERA ;
- Laurent Vercoüter, INSA Rouen Normandie ;
- Bruno Zanuttini, Université de Caen Normandie.

Journées francophones d'Ingénierie des Connaissances (IC)

Présidence

- Cassia Trojahn - IRIT, Université de Toulouse, IRIT.

Membres

- Nathalie Abadie, IGN/COGIT ;
- Marie-Helene Abel, Université de Technologie de Compiègne ;
- Mehwish Alam, Télécom Paris ;
- Xavier Aimé, Cogsonomy ;
- Yamine Ait Ameer, Université de Toulouse, IRIT ;
- Nathalie Aussenac-Gilles, Université de Toulouse, IRIT ;

- Bruno Bachimont, University de technologie de Compiègne ;
- Nacéra Bennacer, Centrale Supélec ;
- Nathalie Bricon-Souf, Université de Toulouse, IRIT ;
- Sandra Bringay, LIRMM - Université Paul Valéry ;
- Patrice Buche, INRAE ;
- Davide Buscaldi, École Polytechnique ;
- Sylvie Calabretto, INSA de Lyon ;
- Pierre-Antoine Champin, ERCIM ;
- Jean Charlet, AP-HP & INSERM UMRS 1142 ;
- Victor Charpenay, Mines Saint-Etienne ;
- Jérôme David, INRIA & Université Grenoble Alpes ;
- Sylvie Despres, LIMICS - Université Sorbonne Paris Nord ;
- Gayo Diallo, Université de Bordeaux ;
- Gilles Falquet, University of Geneva ;
- Catherine Faron, Université Côte d’Azur ;
- Béatrice Fuchs, LIRIS - université de Lyon ;
- Frédéric Fürst, MIS - Université de Picardie ;
- Alban Gaignard, CNRS ;
- Jean-Gabriel Ganascia, Pierre and Marie Curie University - LIP6 ;
- Ollivier Haemmerlé, Université de Toulouse, IRIT ;
- Mounira Harzallah, LS2N - University of Nantes ;
- Nathalie Hernandez, Université de Toulouse, IRIT ;
- Dominique Lenne Heudiasyc, Université de Technologie de Compiègne ;
- Liliana Ibanescu, AgroParisTech ;
- Sébastien Iksal, LIUM - Le Mans Université ;
- Antoine Isaac, Europeana & VU University Amsterdam ;
- Khadija Jradeh, Université de Toulouse, IRIT ;
- Clement Jonquet, MISTEA (INRAE) and LIRMM (U. Montpellier) ;
- Mouna Kamel, Université de Toulouse, IRIT ;
- Gilles Kassel, University of Picardie Jules Verne ;
- Michel Leclère, University of Montpellier (LIRMM/INRIA) ;
- Maxime Lefrançois, MINES Saint-Etienne ;
- Pascal Molli, University of Nantes ;
- Jérôme Nobécourt, LIPN - Université Sorbonne Paris Nord ;
- Nathalie Pernelle, LIPN - Université Sorbonne Paris Nord ;
- Yannick Prié, LINA - University of Nantes ;
- Cédric Pruski, Luxembourg Institute of Science and Technology ;
- Joe Raad, University of Paris-Saclay ;
- Sylvie Ranwez, LGI2P - Ecole des Mines d’Alès ;
- Catherine Roussey, INRAE ;
- Pascal Salembier, UTT ;
- Fatiha Saïs, LISN, CNRS & Université Paris Saclay ;
- Karim Sehaba, LIRIS CNRS ;
- Danai Symeonidou, INRAE ;
- Konstantin Todorov, LIRMM ;
- Rallou Thomopoulos, INRAE ;
- Raphaël Tronec, EURECOM ;
- Haifa Zargayouna, Université Sorbonne Paris Nord.

Journées Francophones de Programmation par Contraintes (JFPC)

Présidence

- Élise Vareilles, ISAE-SUPAERO, Toulouse, France.

Membres

- Quentin Cappart, École Polytechnique de Montréal, Canada ;
- Clément Carbonnel, LIRM Montpellier, France ;
- Mohamed Sami Cherif, Université Aix-Marseille, France ;
- Thi-Bich-Hanh Dao, Université d'Orléans, France ;
- Jean-Guillaume Fages, Cosling, France ;
- Hao Jin-Kao, Université d'Angers, France ;
- Jean-Marie Lagniez, CRIL Lens, France ;
- Arnaud Lallouet, Huawei Technologies, France ;
- Olivier Lhomme, IBM, France ;
- Chu-Min Li, Université de Picardie Jules Verne, France ;
- Giovani Lo Bianco, Université de Toronton, Canada ;
- Xavier Lorca, IMT Mines Albi, France ;
- Ndiaye Samba Ndjogh, Liris, France ;
- Bertrand Neveu, LIGM Imagine École des Ponts ParisTech, France ;
- Abdelkader Ouali, Université de Caen, France ;
- Eric Piette, Maastricht Université ;
- Cédric Pralet, ONERA Toulouse, France ;
- Charles Prud'Homme, IMT Atlantique, France ;
- Nicolas Procvic, Université Aix-Marseille, France ;
- Matthieu Py, Université Aix-Marseille, France ;
- Abdourahim Sylla, G-SCOP Grenoble, France ;
- Julien Vion, LAMIH, France ;
- Romain Wallon, CRIL Lens, France.

Journées Francophones sur les Systèmes Multi-Agents (JFSMA)

Présidence

- Maxime Morge (CRISAL, Université de Lille, France).

Comité consultatif

- Valérie Camps (IRIT, Université de Toulouse, France) ;
- Jean-Paul Jamont (LCIS, Université Grenoble Alpes, France) ;
- Gauthier Picard (DTIS - SYD, ONERA, France) ;
- Nicolas Sabouret (LIMSI, Université Paris Sud, France) ;
- Olivier Simonin (CITI lab, INSA Lyon, France).

Membres

- Emmanuel Adam (LAMIH, Université Polytechnique Hauts-De-France, France) ;
- Aurélie Beynier (LIP6, Sorbonne Université, France) ;
- Olivier Boissier (LIMOS, École des Mines de Saint-Etienne, France) ;
- Grégory Bonnet (GREYC, Université Caen Normandie, France) ;
- Jonathan Bonnet (Continental CSV, Toulouse) ;
- Florent Carlier (CREN, Le Mans Université, France) ;
- Anne-Cécile Caron (CRISAL, Université de Lille, France) ;

- François Charpillet (LORIA, INRIA, Nancy, France) ;
- Vincent Chevrier (LORIA, Université de Lorraine, France) ;
- Rémy Courdier (LIM, Université de la Réunion, France) ;
- Stéphane Galland (CIAD, Université de Technologie Belfort-Montbéliard, France) ;
- Sylvain Giroux (DOMUS, Université de Sherbrooke, Canada) ;
- Zahia Guessoum (LIP6 & CReSTIC, Université de Reims, France) ;
- Cédric Herpson (LIP6, Sorbonne Université, France) ;
- Guillaume Hutzler (IBISC, Université Evry-Paris Saclay, France) ;
- Elsy Kaddoum (IRIT, Université de Toulouse, France) ;
- Christophe Lang (FEMTO-ST, Université de Franche-Comté, France) ;
- Domitile Lourdeaux (UTC, Université de Technologie de Compiègne) ;
- René Mandiau (LAMIH, Université Polytechnique Hauts-De-France, France) ;
- Philippe Mathieu (CRIStAL, Université de Lille, France) ;
- Bruno Mermet (GREYC, Université Caen Normandie, France) ;
- Fabien Michel (LIRMM, Université de Montpellier, France) ;
- Frédéric Migeon (IRIT, Université de Toulouse, France) ;
- Gildas Morvan (LGI2A, Université d'Artois, France) ;
- Guillaume Muller (Institut Henri Fayol, École des Mines de Saint-Etienne, France) ;
- Jean-Pierre Müller (SENS, CIRAD, Université de Montpellier, France) ;
- Antoine Nongaillard (CRIStAL, Université de Lille, France) ;
- Michel Occello (LCIS, Université Grenoble Alpes, France) ;
- Alexandre Pauchet (LITIS, INSA Rouen Normandie, France) ;
- Sébastien Picault (INRAE, UMR BIOEPAR, France) ;
- Suzanne Pinson (LAMSADE, Université Paris Dauphine, France) ;
- Manon Prédhumeau (School of Geography, University of Leeds, UK) ;
- Clément Raievsky (LCIS, Université Grenoble Alpes, France) ;
- Lilia Rejeb (SMART Lab, Institut Supérieur de Gestion, Tunisie) ;
- Valérie Renault (CREN, Le Mans Université, France) ;
- Julien Saunier (LITIS, INSA Rouen Normandie, France) ;
- Patrick Taillandier (COGIT, IGN, France) ;
- Laurent Vercouter (LITIS, INSA Rouen Normandie, France).

Journées d'Intelligence Artificielle Fondamentale & Journées Francophones sur la Planification, la Décision et l'Apprentissage (JIAF-JFPDA)

Présidence

- Zied Bouraoui (CRIL, Univ Artois & CNRS) ;
- François Schwarzentruher (IRISA, ENS Rennes) ;
- Anaëlle Wilczynski (MICS, CentraleSupélec, Université Paris-Saclay).

Membres

- Francesco Belardinelli (IBISC, Université d'Évry) ;
- Nawal Benabbou (LIP6, Sorbonne Université) ;
- Elise Bonzon (LIPADE, Université Paris Descartes) ;
- Nadia Creignou (LIS, Aix-Marseille Université) ;
- Aurélie Beynier (LIP6, Sorbonne Université) ;
- Olivier Buffet (INRIA / LORIA) ;
- Martin Cooper (IRIT, Université Paul Sabatier) ;

- Tiago de Lima (CRIL, Univ Artois & CNRS);
- Sylvie Doutre (IRIT, Université de Toulouse);
- Alain Dutech (Loria - Inria);
- Jérôme Euzenat (LIG, INRIA);
- Hugo Gilbert (LAMSADE, Université Paris-Dauphine);
- Sébastien Konieczny (CRIL, CNRS);
- Jean Lieber (LORIA, INRIA);
- Jérôme Lang (CNRS, LAMSADE, Université Paris-Dauphine);
- Frédéric Maris (IRIT, Université de Toulouse);
- Pierre Marquis (CRIL, IUF, Univ Artois & CNRS);
- Amedeo Napoli (LORIA Nancy, CNRS - Inria - Université de Lorraine);
- Célia da Costa Pereira (I3S, Université Nice Sophia Antipolis);
- Damien Pellier (Laboratoire d'Informatique de Grenoble);
- Laurent Perrussel (IRIT, Université de Toulouse);
- Sophie Pinchinat (IRISA, INRIA);
- Philippe Preux (INRIA, LIFL, Université de Lille);
- Emmanuel Rachelson (ISAE-SUPAERO);
- Stéphanie Roussel (ONERA);
- Julien Rossit (LIPADE, Université Paris Decartes);
- Régis Sabbadin (INRAE);
- Vincent Thomas (LORIA);
- Paul Weng (UM-SJTU Joint Institute);
- Bruno Zanuttini (GREYC, UNICAEN).

Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA)

Présidence

- Brian Ravenet (LISN-CNRS, Université Paris-Saclay, France).

Membres

- Mathieu Chollet (IMT Atlantique, University of Glasgow, Ecosse);
- Maxime Devanne (IRIMAS, Université Haute-Alsace, France);
- Madeleine El-Zaher (LINEACT, CESI, France);
- Mireille Fares (ISIR, Sorbonne Université, France);
- Maxime Folschette (CRIStAL, Centrale Lille, France);
- Jules François (LISN-CNRS, Université Paris-Saclay, France);
- Sahar Ghannay (LISN-CNRS, Université Paris-Saclay, France);
- Maxime Guériau (LITIS, INSA Rouen Normandie, France);
- Camille Guinaudeau (LISN-CNRS, Université Paris-Saclay, France);
- Marie-Jeanne Lesot (LIP6, Sorbonne Université, France);
- Guillaume Lozenguez (Center Digital Systems, IMT Lille Douai, France);
- Jean-Guy Mailly (LIPADE, Université de Paris, France);
- Florian Pecune (SANPSY, Université de Bordeaux, France);
- Charlotte Pelletier (IRISA, Université Bretagne Sud, France);
- Mohamed-Lamine Messai (ERIC, Université Lumière Lyon 2, France);
- Arianna Novaro (Université Paris 1 Panthéon-Sorbonne, France);
- Nicolas Verstaevael (IRIT, Université de Toulouse, France);
- Genane Youness (LINEACT, CESI, France).

Rencontres de la Société Francophone de Classification (SFC)

Présidence

- Pascal Préa, École Centrale Marseille.

Membres

- Rafik Abdesselam, Université Lumière Lyon ;
- Séverine Affeldt, Université de Paris ;
- Alexandre Bazin, LIRMM, Montpellier ;
- Patrice Bertrand, Université Paris Dauphine ;
- Paula Brito, Université de Porto, Portugal ;
- François Brucker, École Centrale Marseille ;
- Véronique Cariou, ONIRIS Nantes ;
- Christian Derquenne, EDF R&D ;
- Dominique Desbois, INRAE-Paris-Saclay ;
- Jean Diatta, Université de La Réunion ;
- Nadia Ghazalli, Université du Québec à Trois-Rivières ;
- Pascale Kuntz, Université de Nantes ;
- Lazhar Labiod, Université Paris Descartes ;
- Mustapha Lebbah, Université Paris 13 ;
- Ahmed Moussa, ENSA Tanger, Maroc ;
- Mohamed Nadif, Université Paris Descartes ;
- Amedeo Napoli, LORIA, Nancy ;
- Ndèye Niang, CNAM Paris ;
- Allou Samé, Université Gustave Eiffel ;
- Rosanna Verde, Université della Campania, Caserta, Italie.

Comité d'organisation

ICube, Université de Strasbourg

Présidence

- LACHICHE Nicolas, ICube, Université de Strasbourg.

Membres

- BRAUD Agnès, ICube, Université de Strasbourg ;
- DOLQUES Xavier, ICube, Université de Strasbourg ;
- FREY Gabriel, ICube, Université de Strasbourg ;
- GANÇARSKI Pierre, ICube, Université de Strasbourg ;
- GIUSTOZZI Franco, ICube, INSA de Strasbourg ;
- LAFABREGUE Baptiste, ICube, Université de Strasbourg ;
- LAMPERT Thomas, ICube, Université de Strasbourg ;
- LEBORGNE Aurélie, ICube, Université de Strasbourg ;
- LE BER Florence, ICube, ENGEES ;
- MARC-ZWECKER Stella, ICube, Université de Strasbourg ;
- NAEGEL Benoît, ICube, Université de Strasbourg ;
- SAGET Antoine, ICube, Université de Strasbourg ;
- SAMET Ahmed, ICube, INSA de Strasbourg ;
- WEMMERT Cédric, ICube, Université de Strasbourg.



Sciences et technologies
de l'information pour la santé |
HealthTech
Les Instituts thématiques interdisciplinaires de l'Université de Strasbourg & CNRS & Inserm
dans le cadre de l'Initiative d'excellence



Consommation adaptative par négociation continue

E. Beauprez, A-C. Caron, M. Morge, J-C. Routier

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISAL

Ellie.Beauprez@univ-lille.fr, Anne-Cecile.Caron@univ-lille.fr, Maxime.Morge@univ-lille.fr,
Jean-Christophe.Routier@univ-lille.fr

Résumé

Nous étudions ici le problème de l'allocation de jobs concurrents, composés de tâches situées, sous-jacent au déploiement distribué du patron de conception MapReduce sur une grappe de serveurs. Afin de mettre en œuvre notre stratégie multi-agents qui vise à minimiser le délai moyen de réalisation des jobs (flowtime), nous proposons une architecture composite d'agent qui permet la concurrence des négociations et des consommations. Nos expérimentations montrent que, lorsqu'elle est exécutée en continue lors du processus de consommation, notre stratégie de réallocation : (1) améliore le flowtime; (2) ne pénalise pas la consommation ; (3) est robuste aux aléas d'exécution.

Mots-clés

Résolution collective de problèmes, négociation multi-agents, architecture d'agent.

Abstract

In this paper, we study the problem of allocating concurrent jobs, composed of situated tasks, underlying the distributed deployment of the MapReduce design pattern on a cluster. In order to implement our multi-agent strategy that aims at minimizing the mean flowtime of jobs, we propose a composite agent architecture that allows negotiation and consumption concurrency. Our experiments show that, when executed continuously during the consumption process, our reallocation strategy: (1) improves the flowtime; (2) does not penalise the consumption; (3) is robust to execution hazards.

Keywords

Distributed problem solving, agent-based negotiation, agent architecture.

1 Introduction

Les sciences des données exploitent de larges volumes de données sur lesquelles des calculs sont effectués en parallèle par différents nœuds. Ces applications mettent à l'épreuve l'informatique distribuée en ce qui concerne l'allocation de tâches et l'équilibrage de charge. C'est le cas de l'application pratique que nous considérons dans cet article : le modèle de traitement le plus répandu pour traiter des données massives sur une grappe de serveurs,

c'est-à-dire le patron de conception MapReduce [10]. Les jobs, qui doivent être exécutés le plus tôt possible, sont composés d'un ensemble de tâches, exécutées par les différents nœuds. L'exécution d'une tâche consiste à traiter des ressources situées sur les nœuds. Comme plusieurs ressources sont requises pour réaliser une tâche sur un nœud, son exécution peut nécessiter de récupérer des ressources disponibles sur d'autres nœuds, ce qui induit un surcoût.

De nombreux travaux adoptent le paradigme multi-agents pour aborder le problème de la réallocation de tâches et de l'équilibrage de charge parmi des exécutants multiples [2]. L'approche centrée individus permet la distribution d'heuristiques pour des problèmes impraticables à cause de la combinatoire des ordonnancements, autorisant ainsi le passage à l'échelle. De plus, intrinsèquement réactives, les méthodes multi-agents de réaffectation s'adaptent aux estimations inexactes des temps d'exécution et aux perturbations (consommation/libération de tâches, ralentissement des exécutants, etc.).

Dans [4], nous avons proposé une stratégie multi-agents de réallocation de tâches pour un ensemble de jobs devant être exécutés le plus tôt possible. Afin de minimiser le délai moyen de réalisation des jobs (*flowtime*), les agents, qui sont coopératifs, négocient pour déterminer les prochaines tâches à déléguer, voire à échanger. Cette stratégie nécessite le déploiement distribué d'agents autonomes qui consomment les tâches et échangent de manière continue certaines d'entre elles pour équilibrer l'allocation courante. Dans cet article, nous formalisons les opérations de consommation de tâches et celles de réallocation et nous proposons une architecture composite d'agent qui permet la concurrence des négociations et des consommations. Selon le principe de la séparation des préoccupations, un premier agent composant est dédié à la consommation (i.e. l'exécution) des tâches, un second aux négociations des réallocations et un troisième à la coordination locale de ces opérations à travers la gestion du lot de tâches. La difficulté réside dans la conception des comportements des agents composants qui ne partagent pas un état global du système (e.g. l'allocation) mais disposent de connaissances locales et partielles. Nos expérimentations montrent que, lorsqu'elle est exécutée en continue lors du processus de consommation, notre stratégie de réallocation ne pénalise pas la consommation et peut améliorer le *flowtime* jusqu'à

37 %, même lorsque les agents ont une connaissance imparfaite de l'environnement d'exécution comme lors d'aléas (i.e. le ralentissement de nœuds).

Après un aperçu des travaux connexes dans la section 2, nous rappelons dans la section 3 la formalisation du problème d'allocation de jobs composés de tâches situées. La section 4 formalise les opérations de consommation/réallocation. Nous décrivons ensuite, dans la section 5, comment le processus de consommation et celui de réallocation sont entrelacés, puis nous détaillons notre architecture d'agents dans la section 6. La section 7 présente nos résultats expérimentaux. La section 8 résume notre contribution et présente nos perspectives.

2 Travaux connexes

De nombreux travaux ont abordé le problème de la réallocation de tâches parmi de multiples exécutants. L'approche centrée individus permet de surmonter les limites des solutions centralisées : l'impossibilité de résoudre les problèmes à grande échelle et la faible réactivité aux changements [2]. Les problèmes d'allocation dynamique des tâches nécessitent notamment de proposer des processus dynamiques qui s'ajustent en permanence aux changements de l'environnement d'exécution ou de performance des exécutants [11]. La plupart de ces travaux s'appuie sur l'algorithme à base de consensus (CBBA – *Consensus Based Bundle Algorithm*) [7] qui est une méthode multi-agents d'assignation en deux phases qui consiste à : (a) sélectionner les tâches à négocier à travers un processus d'enchère ; (b) déterminer les offres qui remportent ces enchères en résolvant les conflits potentiels. En particulier, notre architecture d'agent modulaire s'inspire largement de [1]. Toutefois, nos agents ne visent pas à minimiser le *makespan* (i.e. le temps d'exécution global) mais le *flowtime*. De plus, nous avons préféré ici un protocole de négociation bilatérale qui permet, en sélectionnant l'interlocuteur, de faire des propositions ciblées et donc de réduire les coûts computationnel et communicationnel liés à la négociation. Finalement, la simulation de l'environnement d'exécution nous permet d'en contrôler les perturbations.

Chen et al. envisagent des problèmes d'allocation dynamique de tâches où les tâches sont libérées à des moments incertains [6]. Ils proposent d'ajuster l'allocation des tâches de façon continue en combinant le réordonnement local des agents avec la réallocation des tâches entre agents. De manière similaire, notre stratégie multi-agents s'appuie sur une stratégie de consommation pour définir l'ordonnement local des tâches et sur une stratégie de négociation des tâches à réallouer. Contrairement à [6], nous faisons ici l'hypothèse que l'ensemble des jobs sont initialement connus, mais nos agents sont susceptibles d'avoir une connaissance imparfaite de l'environnement d'exécution.

La plupart des travaux qui considèrent que les perturbations de l'environnement d'exécution font varier le coût des tâches s'appuient sur des techniques de recherche

opérationnelle comme l'analyse de sensibilité pour évaluer la robustesse des optima aux perturbations [15], des méthodes incrémentales pour réparer l'allocation optimale initiale lorsque les coûts changent [14] ou l'optimisation combinatoire pour exploiter les mesures de dégradation [13]. De manière similaire, notre stratégie mesure notamment l'écart entre les progrès attendus et ceux observés en vue de modifier l'allocation. Toutefois, notre approche centrée individus permet de résoudre des problèmes à grande échelle.

Creech et al. aborde le problème de l'allocation des ressources et de la hiérarchisation des tâches dans les systèmes multi-agents distribués pour des environnements dynamiques [8]. Ils proposent un algorithme d'optimisation de l'allocation des ressources multi-groupes (MG-RAO) qui combine des algorithmes de mise à jour et de priorisation et qui utilise des techniques d'apprentissage par renforcement. À l'inverse des techniques d'apprentissage, notre solution ne nécessite aucun modèle préalable, ni des données, ni de l'environnement, et aucune phase d'exploration car cela ne serait pas pertinent pour l'application pratique qui nous concerne. En effet, le volume de données rend les pré-traitements et l'exploration trop coûteux. De plus, la variabilité des données les rend rapidement obsolète.

Nos précédentes expérimentations ont montré que la durée moyenne de réalisation atteinte par notre stratégie est meilleure que celle obtenue avec les techniques d'optimisation sous contraintes distribuée (DCOP) et reste proche de celle obtenue avec une heuristique classique, avec dans tous les cas un temps de réordonnement significativement réduit [3]. Nous montrons dans cet article comment déployer cette stratégie de manière continue au cours du processus de consommation.

3 Problème

Cette section présente la formalisation, introduite dans [4], du problème d'allocation des jobs concurrents composés de tâches situées.

Un système distribué est composé d'un ensemble de nœuds de calcul capables d'exécuter des tâches. Ces tâches requièrent des ressources, transférables et non consommables, réparties parmi différents nœuds de ressources.

Définition 1 (Système distribué). *Un système distribué est un quadruplet $\mathcal{D} = \langle \mathcal{P}, \mathcal{N}_r, \mathcal{E}, \mathcal{R} \rangle$ où :*

- \mathcal{P} est un ensemble de p nœuds de calcul;
- \mathcal{N}_r est un ensemble de r nœuds de ressource;
- $\mathcal{E} : \mathcal{P} \times \mathcal{N}_r \rightarrow \{\top, \perp\}$ est une propriété de voisinage qui évalue si un nœud de calcul de l'ensemble \mathcal{P} est local à un nœud de ressource dans \mathcal{N}_r ;
- $\mathcal{R} = \{\rho_1, \dots, \rho_k\}$ est un ensemble de ressources ayant des tailles $|\rho_i|$. La localisation des ressources, qui

sont éventuellement répliquées, est déterminée par la fonction :

$$l : \mathcal{R} \rightarrow 2^{\mathcal{N}_r} \quad (1)$$

Une ressource peut être locale ou distante d'un nœud de calcul, selon sa présence ou non sur un nœud de ressource dans le voisinage du nœud de calcul. À partir des fonctions \mathcal{E} et l , nous définissons le prédicat de localité :

$$\begin{aligned} \forall v_c \in \mathcal{P}, \forall \rho \in \mathcal{R}, \text{local}(\rho, v_c) \text{ ssi} \\ \exists v_r \in l(\rho) \text{ t.q. } \mathcal{E}(v_c, v_r) \end{aligned} \quad (2)$$

Les ressources sont accessibles pour tous les nœuds de calcul, même celles sur les nœuds de ressource distants.

Un job est un ensemble de tâches indépendantes, non divisibles et non-préemptives. L'exécution de chaque tâche nécessite l'accès à des ressources distribuées sur les nœuds du système. L'exécution d'un job (sans date butoir) consiste à exécuter l'ensemble de ses tâches pour produire un résultat.

Définition 2 (Job/Tâche). Soit \mathcal{D} un système distribué. On considère un ensemble de ℓ jobs $\mathcal{J} = \{J_1, \dots, J_\ell\}$. Chaque job J_i , associé à la date de libération $t_{j_i}^0$, est un ensemble non vide de k_i tâches $J_i = \{\tau_1, \dots, \tau_{k_i}\}$.

On note $\mathcal{T} = \cup_{1 \leq i \leq \ell} J_i$ l'ensemble des n tâches sous-jacentes à \mathcal{J} et $\mathcal{R}_\tau \subseteq \mathcal{R}$ l'ensemble des ressources requises pour la tâche τ . Par souci de concision, on note $\text{job}(\tau)$ le job contenant la tâche τ . Nous faisons l'hypothèse que le nombre de jobs est négligeable par rapport au nombre de tâches, $|\mathcal{J}| \ll |\mathcal{T}|$.

Le coût d'une tâche pour un nœud v_i est une estimation *a priori* de son temps d'exécution.

Définition 3 (Coût d'une tâche pour un nœud). Soient \mathcal{D} un système distribué et \mathcal{T} un ensemble de tâches. La fonction de coût $c : \mathcal{T} \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est telle que :

$$\begin{aligned} c(\tau, v_j) &= \sum_{\rho_j \in \mathcal{R}_\tau} c(\rho_j, v_j) \\ \text{avec } c(\rho_j, v_i) &= \begin{cases} |\rho_j| & \text{si local}(\rho_j, v_i) \\ \kappa \times |\rho_j| & \text{avec } \kappa > 1 \text{ sinon.} \end{cases} \end{aligned} \quad (3)$$

Comme la collecte de ressources distantes représente un surcoût, une tâche est plus coûteuse si les ressources nécessaires sont « moins locales ». La fonction de coût peut être étendue à un ensemble de tâches :

$$\forall T \subseteq \mathcal{T}, c(T, v_i) = \sum_{\tau \in T} c(\tau, v_i) \quad (4)$$

En substance, nous considérons le problème d'allocation de jobs composés de tâches situées.

Définition 4 (STAP). Un problème d'allocation de tâches situées est un quadruplet $STAP = (\mathcal{D}, \mathcal{T}, \mathcal{J}, c)$ où :

- \mathcal{D} est un système distribué de m nœuds ;
- $\mathcal{T} = \{\tau_1, \dots, \tau_n\}$ est un ensemble de n tâches ;

- $\mathcal{J} = \{J_1, \dots, J_\ell\}$ est un partitionnement des tâches en ℓ jobs ;

- $c : \mathcal{T} \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est la fonction de coût.

Une allocation de tâches est une répartition des tâches dans des lots ordonnés.

Définition 5 (Allocation). Une allocation pour un problème STAP à l'instant t est un vecteur de m lots de tâches ordonnées

$\vec{A}_t = ((B_{1,t}, \prec_1), \dots, (B_{m,t}, \prec_m))$ où chaque lot $(B_{i,t}, \prec_i)$ est l'ensemble des tâches $(B_{i,t} \subseteq \mathcal{T})$ affectées au nœud v_i à l'instant t , associé à un ordre total strict $(\prec_i \subseteq \mathcal{T} \times \mathcal{T})$. $\tau_j \prec_i \tau_k$ signifie que si $\tau_j, \tau_k \in B_{i,t}$ alors τ_j est exécutée avant τ_k par v_i . L'allocation \vec{A}_t vérifie pour l'instant t :

$$\forall \tau \in \mathcal{T}, \exists v_i \in \mathcal{N}, \tau \in B_{i,t} \quad (5)$$

$$\forall v_i \in \mathcal{N}, \forall v_j \in \mathcal{N} \setminus \{v_i\}, B_{i,t} \cap B_{j,t} = \emptyset \quad (6)$$

Toutes les tâches sont allouées (Eq. 5) et chacune n'est allouée qu'à un seul nœud (Eq. 6). Par souci de concision, on note :

- $\vec{B}_{i,t} = (B_{i,t}, \prec_i)$, le lot trié de v_i ;
- $\min_{\prec_i} B_{i,t}$, la prochaine tâche à exécuter par v_i .

Pour évaluer la qualité d'une allocation de tâches, nous considérons le délai moyen de réalisation (*flowtime*), qui mesure le temps moyen écoulé entre la date de libération des jobs et leur date d'achèvement.

Définition 6 (Flowtime). Soient STAP un problème et \vec{A}_t une allocation à l'instant t . On définit :

- le délai d'attente de la tâche τ dans le lot $\vec{B}_{i,t}$,

$$\Delta(\tau, v_i) = \sum_{\tau' \in B_{i,t} | \tau' \prec_i \tau} c(\tau', v_i) \quad (7)$$

- la durée de réalisation de la tâche $\tau \in \mathcal{T}$ pour l'allocation \vec{A}_t ,

$$\begin{aligned} C_\tau(\vec{A}_t) &= \Delta(\tau, v(\tau, \vec{A}_t)) + t - t_{\text{job}(\tau)}^0 \\ &+ c(\tau, v(\tau, \vec{A}_t)) \end{aligned} \quad (8)$$

- la durée de réalisation de $J \in \mathcal{J}$ pour \vec{A}_t ,

$$C_J(\vec{A}_t) = \max_{\tau \in J} \{C_\tau(\vec{A}_t)\} \quad (9)$$

- le délai moyen de réalisation de \mathcal{J} pour \vec{A}_t ,

$$\begin{aligned} C_{\text{mean}}(\vec{A}_t) &= \frac{1}{\ell} C(\vec{A}_t) \\ \text{avec } C(\vec{A}_t) &= \sum_{J \in \mathcal{J}} C_J(\vec{A}_t) \end{aligned} \quad (10)$$

Plus particulièrement le délai d'attente (équation 7) correspond au délai d'attente à partir de l'instant courant t avant que la tâche τ ne soit traitée. Il est à noter que les temps de réalisation, et par conséquent le *flowtime*, dépendent de l'ordre d'exécution des tâches sur chacun des nœuds.

4 Opérations

Nous formalisons ici les opérations de consommation et de réallocation de tâches.

La **consommation** d'une tâche par un nœud consiste pour ce dernier à retirer cette tâche de son lot pour l'exécuter. Cette opération mène à une nouvelle instance de problème où cette tâche a été retirée. En d'autres termes, l'accomplissement d'une tâche est un évènement disruptif qui modifie non seulement l'allocation des tâches mais également le problème sous-jacent.

Définition 7 (Consommation de tâche). Soient $STAP = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ un problème d'allocation de tâches et \vec{A}_t une allocation. La consommation par un nœud consommateur v_i à un instant t , dont le lot n'est pas vide ($B_{i,t} \neq \emptyset$), aboutit à l'allocation $\vec{A}'_t = \lambda(v_i, \vec{B}_{i,t})$ pour le problème $STAP' = \langle \mathcal{D}, \mathcal{T}', \mathcal{J}', c \rangle$ où :

$$\mathcal{T}' = \mathcal{T} \setminus \{\min_{<i>_i B_{i,t}\}$$

$$\mathcal{J}' = \begin{cases} \mathcal{J} \setminus \{job(\min_{<i>_i B_{i,t})\} & \text{si } job(\min_{<i>_i B_{i,t}) = \\ & \{\min_{<i>_i B_{i,t}\} \\ \mathcal{J} & \text{sinon} \end{cases}$$

Dans ce dernier cas :

$$\forall J_j \in \mathcal{J} \exists J'_j \in \mathcal{J}' \text{ t.q. } J'_j = \begin{cases} J_j \setminus \{\min_{<i>_i B_{i,t}\} \\ \text{si } job(\min_{<i>_i B_{i,t}) = J_j \\ J_j \text{ sinon} \end{cases}$$

et $\vec{A}'_t = (\vec{B}'_{1,t}, \dots, \vec{B}'_{m,t})$ avec

$$\vec{B}'_{j,t} = \begin{cases} \overline{B_{i,t} \ominus \min_{<i>_i B_{i,t}} & \text{si } j = i \\ \vec{B}_{j,t} & \text{sinon} \end{cases}$$

Lorsqu'une tâche est consommée, elle est retirée du problème résultant non seulement dans l'ensemble des tâches mais également du job correspondant. Ce dernier peut également être retiré si la tâche était la seule (la dernière) du job. L'allocation résultante est également modifiée. La tâche est retirée du lot où elle se trouvait. Les tâches sont destinées à être consommées une à une jusqu'à atteindre l'allocation vide.

À l'évidence, la consommation d'une tâche ne peut augmenter le *flowtime* à un instant t . En effet, la consommation d'une tâche fait décroître localement le *flowtime*, à l'instant t ,

$$\sum_{J \in \mathcal{J}} C_J(\lambda(v_i, \vec{B}_{i,t})) < \sum_{J \in \mathcal{J}} C_J(\vec{B}_{i,t}) \quad (11)$$

Cela n'est cependant pas toujours vrai au cours du temps car les coûts effectifs des tâches peuvent être différents des coûts estimés. Si une tâche s'avère plus coûteuse que prévu lors de son exécution, le *flowtime* peut augmenter après sa consommation, comme dans l'exemple 1.

Exemple 1. Soit le problème $STAP = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ avec :

- $\mathcal{D} = \langle \mathcal{P}, \mathcal{N}_r, \mathcal{E}, \mathcal{R} \rangle$, un système distribué avec un unique nœud de calcul $\mathcal{P} = \{v_1\}$ associé au seul nœud de ressource $\mathcal{N}_r = \{v_1^r\}$, tel que $\mathcal{E}(v_1, v_1^r) = \top$ et une unique ressource $\mathcal{R} = \{\rho_1\}$ sur le nœud de ressource v_1^r ;
- deux tâches $\mathcal{T} = \{\tau_1, \tau_2\}$;
- un unique job $\mathcal{J} = \{J_1\}$ libéré à $t_{J_1}^0 = 0$ composé des deux tâches $J_1 = \{\tau_1, \tau_2\}$;
- la fonction de coût c telle que $c(\tau_1, v_1) = 2$ et $c(\tau_2, v_1) = 4$.

L'allocation $\vec{A}_0 = (\vec{B}_{1,t})$ avec $\vec{B}_{1,t} = (\tau_1, \tau_2)$. Selon l'équation 10, le *flowtime* est $C_{mean}(\vec{A}_0) = C_{J_1}(\vec{A}_0) = C_{\tau_2}(\vec{A}_0) = \Delta(\tau_2, v_1) + t + t_{J_1}^0 + c(\tau_2, v_1) = c(\tau_1, \vec{A}_t) + 0 + 0 + c(\tau_2, v_1) = 2 + 4 = 6$.

Si la consommation de τ_1 se termine à l'instant $t_1 = 3$, cela signifie que cette tâche s'avère plus coûteuse que prévu lors de son exécution. Par conséquent, le *flowtime* de $\vec{A}'_t = (\vec{B}_{v_1, t_1})$ avec $B_{v_1, t_1} = (\tau_2)$ est $C_{mean}(\vec{A}'_t) = C_{J_1}(\vec{A}'_t) = t_1 + t_{J_1}^0 + c(\tau_2, v_1) = 3 + 0 + 4 = 7 > C_{mean}(\vec{A}_0)$.

Une **réallocation bilatérale** est une opération lors de laquelle une ou plusieurs tâches sont déplacées d'un lot à un autre.

Définition 8 (Réallocation bilatérale). Soit $\vec{A}_t = (\vec{B}_{1,t}, \dots, \vec{B}_{m,t})$ une allocation pour le problème $STAP = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ à l'instant t . La réallocation bilatérale de la liste non vide de tâches T_1 allouées au proposant v_i en échange de la liste de tâches T_2 allouées au répondant v_j dans \vec{A}_t ($T_1 \subseteq B_{i,t}$ et $T_2 \subseteq B_{j,t}$) aboutit à l'allocation $\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)$ avec les m lots $\gamma(T_1, T_2, v_i, v_j, \vec{B}_{k,t})$ définis tels que :

$$\gamma(T_1, T_2, v_i, v_j, \vec{B}_{k,t}) = \begin{cases} \overline{B_{i,t} \ominus T_1 \oplus T_2} & \text{si } k = i, \\ \overline{B_{j,t} \ominus T_2 \oplus T_1} & \text{si } k = j, \\ \vec{B}_{k,t} & \text{sinon} \end{cases} \quad (12)$$

Si T_2 est vide, on parle de *délégation*. Sinon, une réallocation bilatérale est un échange de tâches.

Nous nous restreignons ici aux échanges mais les réallocations multilatérales mériteraient d'être explorées. Contrairement à la plupart des autres travaux (e.g. [9]), nos agents ne sont pas individuellement rationnels mais ils ont un but commun qui prime sur leur intérêt individuel : réduire le *flowtime*.

Définition 9 (Réallocation bilatérale socialement rationnelle). Soit \vec{A}_t une allocation à l'instant t pour le problème $STAP = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$. La réallocation bilatérale $\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)$ est socialement rationnelle ssi le *flowtime* décroît,

$$C(\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)) < C(\vec{A}_t) \quad (13)$$

Une allocation est dite **stable** s'il n'existe aucune réallocation bilatérale socialement rationnelle. Dans [3], nous avons démontré la terminaison du processus qui itère ce type de réallocations.

5 Processus

Afin d'exécuter de manière concurrente le processus de consommation et celui de réallocation, nous considérons deux types d'agents : (a) les agents de nœud, chacun d'entre eux représentant un nœud de calcul en gérant son lot de tâches (cf. section 6) ; (b) le superviseur qui synchronise les phases du processus de négociation.

Le processus de consommation se résume à l'exécution concurrente ou séquentielle des différentes tâches par les nœuds de calcul sous la supervision de leur agent. Le processus de réallocation est constitué de multiples réallocations locales qui sont le résultat de négociations bilatérales entre agents de nœud, réalisées de manière séquentielle ou concurrente. Ces processus sont complémentaires. Tandis que les consommations se déroulent en continu, les agents négocient leurs lots de tâches jusqu'à atteindre une allocation stable. La consommation d'une tâche peut rendre instable une allocation et ainsi déclencher de nouvelles négociations. Le processus de consommation se termine quand toutes les tâches sont exécutées. L'allocation finale, qui est vide, met un terme au processus.

Il est important de noter que ce système multi-agents est intrinsèquement adaptatif. En effet, si le coût d'une tâche s'avère plus élevé que prévu, parce que le temps d'exécution a été sous-estimé ou parce que le nœud l'exécutant est ralenti, alors le processus de réallocation qui se déroule en continu permet de corriger l'allocation en prenant en compte le temps effectivement mesuré après la réalisation des tâches.

La **stratégie de consommation** des agents, détaillée dans [4], spécifie l'ordonnancement des tâches, au sein du lot de tâches, pour leur exécution par le nœud dont ils ont la charge. Afin de réduire la durée de réalisation des jobs, cette stratégie exécute les tâches des jobs les moins coûteux avant celles des jobs les plus coûteux.

La **stratégie de négociation** des agents de nœud, également détaillée dans [4], s'appuie sur un modèle des pairs, notamment une base de croyances, construit à partir des messages échangés et grâce auquel elle détermine si une réallocation est socialement rationnelle, selon les croyances de l'agent. Les agents disposent : (a) d'une stratégie d'offre qui propose des réallocations bilatérales ; (b) d'une règle d'acceptabilité qui évalue si une proposition reçue est ou non socialement rationnelle avant de l'accepter ou de la refuser ; et (c) d'une stratégie de contre-offre qui sélectionne une contre-partie à une délégation afin de proposer un échange de tâches.

Le processus de négociation se décompose en deux phases successives : (1) les agents proposent des délégations qu'ils croient socialement rationnelles et qui sont acceptées ou refusées par leurs pairs ; (2) les agents proposent des délégations qui ne sont pas nécessairement socialement rationnelles mais qui sont susceptibles de déclencher des contre-offres et ainsi des échanges socialement rationnels. Les phases de négociations s'alternent successivement, de façon concurrente à la consommation.

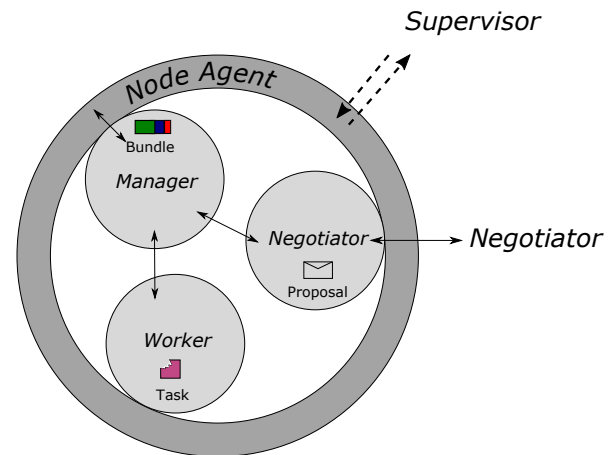


Figure 1: Architecture d'un agent de nœud

6 Architecture

Pour concevoir un agent de nœud, nous avons adopté une architecture modulaire qui permet la concurrence des négociations et des consommations.

Un agent de nœud est un agent composite constitué de trois agents composants (cf. figure 1), chacun ayant un rôle limité :

- le **worker** exécute (consomme) les tâches ;
- le **negotiator** maintient une base de croyances pour négocier des tâches avec ses pairs ;
- le **manager** gère le lot de tâches du nœud de calcul pour ordonner leur exécution par le **worker** en y ajoutant ou supprimant les tâches selon les réallocations bilatérales marchandées par le **negotiator**.

Afin de prioriser la consommation des tâches, le **manager**, dès qu'il est informé que le **worker** est libre, fournit à ce dernier la prochaine tâche à exécuter conformément à la stratégie de consommation, quitte à annuler la réallocation de cette tâche en cours négociation. Cette tâche n'est alors plus éligible pour une potentielle réallocation.

Nous représentons ici les interactions entre les agents composants sous la forme de diagrammes d'interaction où les flèches pleines représentent des appels synchrones (comme dans la figure 2b), les flèches ouvertes des messages asynchrones (comme dans la figure 2a) et les lignes pointillées des messages de réponse.

Après que le **manager** a confié au **worker** une tâche, ce dernier signale au **manager** quand elle est accomplie (cf. figure 2a). Pour prioriser la consommation plutôt que la négociation, la demande de la prochaine tâche à accomplir par le **worker** au **manager** est prioritaire et préempte les interactions de ce dernier avec le **negotiator**. Pour raffiner son estimation du délai d'attente des tâches dans son lot, le **manager** peut demander au **worker** une estimation du temps d'exécution restant pour la tâche en cours (cf. figure 2b).

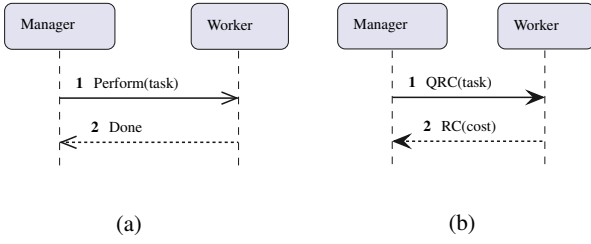


Figure 2: Interactions *manager/worker*

Lors d’une première phase de négociation, les agents marchandent des délégations (cf. figure 3). Afin de confirmer une telle réallocation bilatérale, le *negotiator* de l’agent proposant demande de manière synchrone au *manager* de mettre à jour le lot de tâches afin que lui-même puisse mettre à jour sa base de croyances avant de s’engager dans de nouvelles négociations. Après cette confirmation, le *negotiator* du répondant en fait de même. Les étiquettes QRC indiquent que le *manager* interagit avec le *worker* selon le protocole de la figure 2b pour prendre en compte le temps d’exécution restant pour la tâche en cours. Lors d’une seconde phase de négociation, les agents marchandent des échanges de tâches et les interactions sont similaires.

Malgré notre architecture modulaire, la principale difficulté subsiste dans la conception des comportements des agents, qui sont spécifiés dans [5] par des automates¹, et dont la complexité est mesurée dans la table 1 en nombre d’états, de transitions et de lignes de code.

Agent	États	Transitions	Lignes
<i>worker</i>	2	7	173
<i>manager</i>	5	23	465
<i>negotiator</i>	9	74	1306
superviseur	9	69	626

Table 1: Complexité des comportements

Le *worker* est : soit disponible ; soit en train d’exécuter une tâche et il peut donc estimer le temps d’exécution restant pour la tâche en cours.

Le *manager* gère le lot de tâches et coordonne les opérations de consommation des tâches réalisées par le *worker* avec celles de réallocations marchandées par le *negotiator*. Quand ce dernier lui signale qu’il n’a plus de délégation socialement rationnelle à proposer et qu’il attend les propositions des autres agents de nœud, le *manager* en informe le superviseur. Il continue également de distribuer les tâches à exécuter au *worker* jusqu’à ce que son lot soit vide. Informé qu’aucun agent de nœud ne détecte d’opportunité de réallocation, le superviseur enclenche le changement de phase de négociation. Enfin, le superviseur clôt le processus quand il apprend par les *managers* que toutes les tâches ont été consommées.

Le *negotiator* répond aux propositions de ses pairs et met à jour sa base de croyances, ce qui lui permet de

¹<https://gitlab.univ-lille.fr/maxime.morge/smastaplus/-/tree/worker/doc/specification>

détecter des opportunités de réallocation. Après avoir proposé une délégation, il attend, avant une date butoir, une acceptation, un refus ou une contre-proposition. Lorsque le *negotiator* a accepté une proposition ou fait une contre-proposition, il attend la confirmation ou l’abandon de son interlocuteur (si la tâche a été consommée depuis). Lorsque l’agent a confirmé son acceptation d’une contre-offre, il attend également la double-confirmation de son homologue. Quand la stratégie d’offre ne suggère aucune délégation, la base de croyances est mise à jour jusqu’à ce qu’une nouvelle opportunité soit trouvée.

7 Expérimentations

Nos expériences visent à valider que, lorsqu’elle est exécutée en continue lors du processus de consommation, la stratégie de réallocation : (1) améliore le *flowtime* ; (2) ne pénalise pas la consommation ; (3) est robuste aux aléas d’exécution (i.e. le ralentissement de nœuds). Nous présentons ici nos métriques, notre protocole expérimental et nos résultats².

Plutôt que le temps estimé d’exécution des tâches par les nœuds (cf. équation 3), nous considérons le **coût simulé** $c^S(\tau, v)$ comme le coût effectif de la réalisation de la tâche τ par le nœud v :

- avec une connaissance parfaite de l’environnement d’exécution,

$$c^{SE}(\tau, v_i) = c(\tau, v_i) \tag{14}$$

- avec le ralentissement de la moitié des nœuds,

$$c^{SH}(\tau, v_i) = \begin{cases} 2 \times c(\tau, v_i) & \text{si } i \bmod 2 = 1 \\ c(\tau, v_i) & \text{sinon.} \end{cases} \tag{15}$$

C’est pourquoi nous distinguons :

- le **flowtime simulé** $C_{mean}^S(\vec{A}_t)$ calculé à partir d’une allocation \vec{A}_t selon les coûts simulés ;
- le **flowtime réalisé** $C_{mean}^R(\vec{A}_t)$ calculé à partir des temps de réalisation des tâches effectivement mesurés.

On définit le **taux d’amélioration de performance** :

$$\Gamma = \frac{C_{mean}^R(\vec{A}_0) - C_{mean}^R(\vec{A}_e)}{C_{mean}^R(\vec{A}_0)} \tag{16}$$

où \vec{A}_e est l’allocation des tâches au moment de leur exécution et \vec{A}_0 est l’allocation initiale. Il est à noter que si aucune réallocation n’a lieu pendant le processus, $\vec{A}_e = \vec{A}_0$. Le taux d’amélioration est positif si le *flowtime* réalisé de l’allocation atteinte par le processus de réallocation est meilleur (c’est-à-dire plus faible) que celui de l’allocation initiale.

²Ces expérimentations sont reproductibles à partir des instructions suivantes : <https://gitlab.univ-lille.fr/maxime.morge/smastaplus/-/tree/master/doc/experiments>

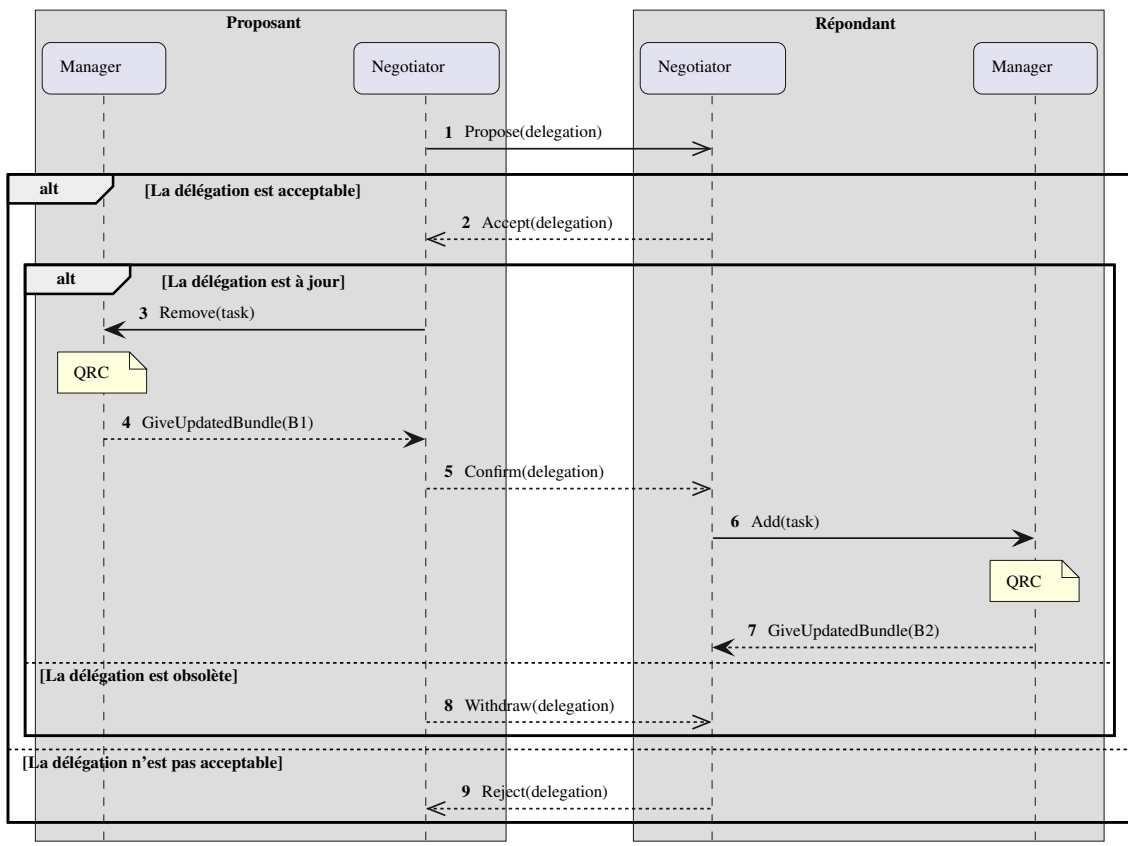


Figure 3: Interactions entre le *manager* et le *negotiator* lors de la première phase de négociation

Notre prototype [5] est implémenté avec le langage de programmation Scala et la bibliothèque Akka [12] adaptée aux applications orientées messages, fortement concurrentes, distribuées et robustes. Les expériences ont été réalisées sur une lame munie de 20 CPUs avec 512Go de RAM. Le fait que, dans nos expériences, la différence entre le *flowtime* réalisé et le *flowtime* simulé de l'allocation initiale ($C_{mean}^R(\vec{A}_0) - C_{mean}^S(\vec{A}_0)$), qui mesure le coût de l'infrastructure, est négligeable, conforte ce choix technologique. Le protocole expérimental consiste, pour les différentes expériences, à générer aléatoirement 25 allocations initiales pour des problèmes STAP distincts. Nous avons fixé empiriquement $\kappa = 2$ comme une valeur réaliste pour capturer le surcoût induit par la récupération des ressources non locales dans un réseau homogène. Nous considérons $m = 8$ nœuds, $l = 4$ jobs, $n \in [40; 320]$ tâches avec 10 ressources par tâche. Chaque ressource ρ_i est répliquée 3 fois et $|\rho_i| \in [0; 500]$. Afin de ne pas déclencher des négociations inutiles dues à l'asynchronisme des opérations de consommations, nous considérons dans nos expérimentations qu'une réallocation bilatérale est socialement rationnelle si elle fait décroître d'au moins une seconde le *flowtime*.

Hypothèse 1 : la stratégie de réallocation améliore le *flowtime*. Nous considérons ici que les allocations initiales sont aléatoires et que les agents ont une connaissance parfaite de l'environnement d'exécution (c^{SE}). La figure 4 montre les médianes et les écarts types de nos métriques en fonction du nombre de tâches. Nous observons que le *flowtime* réalisé de la réallocation est meilleur que le *flowtime* réalisé de l'allocation initiale et borné par le *flowtime* simulé de la réallocation (si un oracle calcule la réallocation en temps constant). Notre stratégie améliore le *flowtime* en réallouant en continu lors du processus de consommation les tâches non locales dont la délégation réduit le coût. Le taux d'amélioration de performance (Γ) se situe entre 20 % et 37 %.

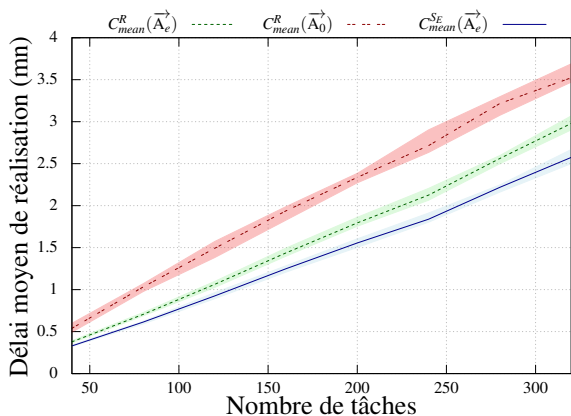


Figure 4: Depuis une allocation aléatoire

Hypothèse 2 : la stratégie de réallocation ne pénalise pas la consommation. Nous considérons ici que les allocations initiales sont stables. Dans la figure 5 le *flowtime* réalisé de la réallocation est similaire au *flowtime*

réalisé de l'allocation initiale. Le surcoût de la négociation est négligeable car aucune négociation n'est déclenchée lorsque les agents estiment que l'allocation est stable.

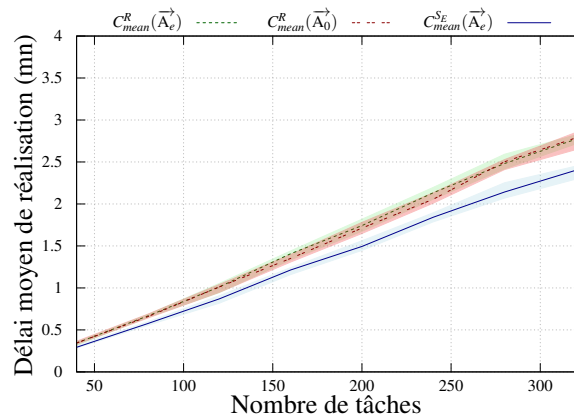


Figure 5: Depuis une allocation stable

Hypothèse 3 : la stratégie de réallocation s'adapte aux aléas d'exécution. Nous considérons ici le coût effectif des tâches qui simule le ralentissement de la moitié des nœuds, c^{SH} . Nous observons dans la figure 6 que les délais moyens de réalisation ont doublé à cause des aléas d'exécution. De plus, le *flowtime* réalisé de la réallocation reste meilleur que le *flowtime* réalisé de l'allocation initiale malgré une connaissance imparfaite de l'environnement d'exécution des agents. Prendre en compte les temps d'exécution effectifs des tâches déjà réalisées permet au taux d'amélioration de performance (Γ) de se situer entre 30 % et 37 %.

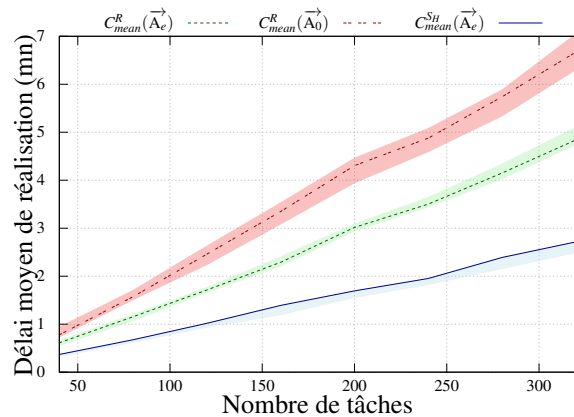


Figure 6: Avec aléas d'exécution

8 Discussion

Afin de concevoir des agents autonomes qui réalisent de manière concurrente des opérations de consommation et de réallocation, nous avons proposé une architecture modulaire d'agent composé de trois agents composants : le *worker* qui exécute les tâches ; le *negotiator* qui marchandise des réallocations avec ses pairs ; et le *manager*

qui coordonne localement ces opérations en gérant le lot de tâches. Sans pour autant connaître l'état global du système, i.e. l'allocation, ces agents disposent de connaissances locales (e.g. la tâche courante, le lot de tâches) et de croyances qui guident leur comportement dans les interactions.

Nos expérimentations montrent que le taux d'amélioration de performance dû à notre stratégie de réallocation, lorsqu'elle est exécutée en continue lors du processus de consommation, peut atteindre 37 %. De plus, le surcoût lié aux négociations est négligeable car, lorsque l'allocation est stable, elles sont suspendues. De plus, même si un ou plusieurs nœuds sont ralentis, notre stratégie de réallocation s'adapte au contexte d'exécution en distribuant plus de tâches aux nœuds qui ne sont pas ralentis, car elle tient compte du temps d'exécution effectif des tâches déjà réalisées, sans pour autant nécessiter de phase d'apprentissage.

Une analyse de sensibilité pour étudier l'influence du facteur de réplication, du surcoût induit par la récupération des ressources non locales (κ) ou du délai d'annulation de la négociation (*timeout*) va au-delà de la portée de cet article, mais mériterait une étude approfondie. Nous souhaitons également évaluer la réactivité de notre stratégie face à la libération de *jobs* au fil de l'eau.

Plus généralement, nos travaux futurs porteront sur l'intégration de la réallocation des tâches dans un processus d'approvisionnement qui ajoute ou supprime des nœuds de calcul au cours de l'exécution en fonction des besoins des utilisateurs afin de proposer une stratégie multi-agents élastique de passage à l'échelle.

Remerciements

Nous adressons nos remerciements aux relecteurs pour leur travail minutieux et leurs précieux conseils.

Références

- [1] Q. Baert, A.-C. Caron, M. Morge, J.-C. Routier, K. Stathis. Un système multi-agent adaptatif pour la réallocation de tâches au sein d'un job Map-Reduce. *Revue Ouverte d'Intelligence Artificielle* (2022), pp. 557–585.
- [2] E. Beauprez, L. Bigand, A.-C. Caron, M. Morge, J.-C. Routier. Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience. *Actes des JFSMA*. Cépaduès, 2021, pp. 51–60.
- [3] E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier. Délégation de lots de tâches pour la réduction de la durée moyenne de réalisation. *ROIA* (2023). À paraître, pp. 1–29.
- [4] E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier. Échange de tâches pour la réduction de la durée moyenne de réalisation. *Actes des JFSMA*s. Cépaduès, 2022, pp. 19–28.
- [5] E. Beauprez, M. Morge. *Scala implementation of the Extended Multi-agents Situated Task Allocation*. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus>. 2020.
- [6] Y. Chen, X. Mao, F. Hou, Q. Wang, S. Yang. Combining re-allocating and re-scheduling for dynamic multi-robot task allocation. *Proc. of SMC*. 2016, pp. 395–400.
- [7] H.-L. Choi, L. Brunet, J. P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics* 25.4 (2009), pp. 912–926.
- [8] N. Creech, N. C. Pacheco, S. Miles. Resource allocation in dynamic multiagent systems. *CoRR* abs/2102.08317 (2021).
- [9] A. Damamme, A. Beynier, Y. Chevaleyre, N. Maudet. The Power of Swap Deals in Distributed Resource Allocation. *Proc. of AAMAS*. 2015, pp. 625–633.
- [10] J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Proc. of OSDI*. 2004, pp. 137–150.
- [11] K. Lerman, C. Jones, A. Galstyan, M. J. Matarić. Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotics Research* 25.3 (2006), pp. 225–241.
- [12] Lightbend. *Akka is the implementation of the Actor Model on the JVM*. <http://akka.io>. 2020.
- [13] S. Mayya, D. S. D'antonio, D. Saldaña, V. Kumar. Resilient task allocation in heterogeneous multi-robot systems. *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1327–1334.
- [14] G. A. Mills-Tettey, A. Stentz, M. B. Dias. The dynamic hungarian algorithm for the assignment problem with changing costs. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27* (2007).
- [15] C. Nam, D. A. Shell. When to do your own thing: Analysis of cost uncertainties in multi-robot task allocation at run-time. *Proc. of ICRA*. 2015, pp. 1249–1254.

RECTO : REcommandation diminuant la Congestion par Transport Optimal

G. Bied^{1,2}, E. Perennes¹, S. Nathan², V. Naya², P. Caillou², B. Crepon¹, C. Gaillac³, M. Sebag²

¹ CREST, ENSAE

² LISN, Université Paris-Saclay

³ Oxford University

bied@lri.fr

Résumé

La recommandation de biens rivaux (sur le marché du travail ou des rencontres) fait face au danger majeur de la congestion. Pour concevoir un système de recommandation évitant la congestion, une approche possible repose sur le transport optimal, cherchant un appariement global entre l'ensemble des utilisateurs et l'ensemble des items selon un coût de transport à définir. L'originalité de l'approche RECTO (REcommandation diminuant la Congestion par Transport Optimal) est de construire le coût de transport en fonction du score de recommandation, et de définir la politique de recommandation en fonction du plan de transport entre utilisateurs et items. Une validation expérimentale comparative, menée sur une base de données publique relative au marché matrimonial et une base de données propriétaire relative au marché du travail, illustre le compromis entre précision et congestion. La discussion porte sur l'intégration de critères individuels et collectifs.

Mots-clés

Systèmes de recommandation, congestion, transport optimal, appariement.

Abstract

The matching setting, a particular case of recommendation problem, focuses on applications where a so-called item can be attributed to at most one user, with the job market and the matrimonial market as chief examples. In such cases, recommending the items preferred by each user might contribute to a congestion issue as users aiming at the same item cannot be all satisfied. While some state of art approaches proceed by repairing the recommendation policy to account for the congestion issue, other approaches take inspiration from the optimal transport (OT) framework, and aim to map the user population onto the item population in order to minimize some global transportation cost. In OT-based recommendation state-of-art approaches, the collaborative matrix (reporting the user-item matches) is interpreted as if it were the result of an OT plan; the underlying transport cost is inferred and used e.g. to propose new congestion-avoiding recommendation policies. In this paper, another OT-based recommendation strategy is

defined, noting that the collaborative matrix reflecting the individual decisions can hardly be considered as the result of a (centralized) OT plan. Accordingly, the proposed algorithm first learns a recommendation score from the data and then defines a cost matrix, with the transportation cost decreasing depending on the matching relevance. The optimal transport plan is thereafter used for the recommendation. The experimental validation of the approach presents comparative results on benchmark data on the matrimonial market, and proprietary real-world data on the job market, illustrating the trade-off between the recall and the congestion indicators. The discussion focuses on the integration of individual and collective criteria.

Keywords

Recommender systems, Congestion, Optimal Transport, Matching.

1 Introduction

La motivation principale de cet article consiste à s'appuyer sur l'état de l'art des systèmes de recommandation [24] pour assister les services publics de l'emploi. Si les systèmes de recommandation usuels visent à recommander à chaque utilisateur l'offre d'emploi qui leur est la plus désirable, il peut sembler peu approprié de recommander la même offre à de nombreux demandeurs d'emploi : cela induirait un phénomène de congestion au niveau de la population et une satisfaction individuelle finale médiocre. Plus généralement, dans des domaines tels que les marchés du travail ou matrimoniaux, correspondant à une configuration de *recommandation réciproque* [18], une politique de recommandation adéquate devrait prendre en compte les populations de demandeurs d'emploi et d'offres d'emploi dans leur globalité, et connecter les deux populations de manière à éviter la congestion.

S'inspirant de travaux connexes sur les systèmes de recommandation [16, 17, 5] et en économétrie [7, 10], cet article étudie le couplage du transport optimal [9, 21] avec les systèmes de recommandation. L'approche proposée, appelée *REcommandation diminuant la Congestion par Transport Optimal* (RECTO), apprend des correspondances entre les populations d'utilisateurs (demandeurs d'emploi) et

d’items (offres d’emploi), visant à trouver un compromis entre l’intérêt des articles recommandés et une diversité suffisante des recommandations au niveau de la population (par opposition à la sérendipité en recommandation [15], qui vise la diversité des items recommandés au niveau individuel). Les questions scientifiques abordées dans l’article concernent donc : i) la définition d’un indicateur mesurant la qualité de recommandations en termes de congestion ; ii) la mise au point d’un algorithme visant à limiter cette congestion ; iii) l’évaluation du compromis entre l’indicateur de performance de recommandation usuel qu’est le recall et la congestion.

La contribution de l’article est triple. Tout d’abord, l’évitement de la congestion est formalisé dans le cadre du transport optimal (section 3). Deuxièmement, l’algorithme RECTO proposé pour résoudre ce problème est agnostique quant à la distribution des données (par opposition aux conditions requises dans [16, 17, 10, 7]) et moins exigeant computationnellement que les approches d’optimisation combinatoire, par exemple [26]. Troisièmement, les mérites de RECTO sont démontrés expérimentalement sur i) un benchmark public dans le domaine du marché matrimonial, utilisé pour l’évaluation comparative avec [16] ; ii) un vaste jeu de données propriétaire relatif au marché du travail¹, et fournissent des leçons inattendues sur les interactions des indicateurs de recall et de congestion.

2 Notations et État de l’art

Cette section présente le problème de recommandation réciproque, renvoyant le lecteur à [18] pour un état de l’art plus complet, et discute quelques travaux connexes. Le cadre du transport optimal [9, 21] est ensuite brièvement introduit.

Notations. Soit n (respectivement m) le nombre d’utilisateurs (resp. d’items), avec x_i (resp. y_j) la description du i -ème utilisateur (resp. du j -ème item). La matrice collaborative booléenne $M_{i,j}$ est telle que $M_{i,j} = 1$ si et seulement si l’utilisateur i a sélectionné l’item j .

Position du problème. Un système de recommandation apprend généralement une fonction de score ϕ telle que la matrice définie à partir de $\phi_{i,j} = \phi(x_i, y_j)$ soit proche de la matrice collaborative M (en termes d’erreur quadratique moyenne ou de divergence de Kullback-Leibler), éventuellement pénalisée par un terme de régularisation [1]. Sans perte de généralité, on suppose par la suite que les items recommandés au i -ème utilisateur sont ordonnés par ordre croissant de $\phi_{i,j}$.

Dans les problèmes de recommandation réciproque [18], l’item j est soumis à une contrainte de capacité n_j : seuls les n_j meilleurs utilisateurs sélectionnant cet item peuvent être servis. De nouveaux objectifs d’optimisation et des algorithmes spécifiques doivent être définis pour tenir compte de ces contraintes.

Travaux connexes. Une première approche de recommandation réciproque, [12], apprend une fonction de score ϕ comme solution d’un problème d’optimisation contraint.

[26] considère la recommandation réciproque comme un problème d’optimisation multi-objectif (NP-difficile), introduisant la satisfaction des contraintes de capacité comme un objectif supplémentaire ; l’optimisation est réalisée en utilisant une approche gloutonne. Dans [2], un modèle auxiliaire prédit la popularité attendue d’un item ; la recommandation est réparée au niveau individuel, en décalant vers le haut ou vers le bas les items recommandés à un utilisateur donné en fonction de leur popularité globale prévue. Dans le contexte de sites de rencontre, [6], s’inspirant de modèles économiques décentralisés, construisent une fonction d’utilité reflétant l’intérêt de x_i pour y_j et réciproquement ; ils utilisent une approche de transport optimal (voir ci-dessous) pour calculer les recommandations finale à partir du score d’utilité.

Transport optimal computationnel. Le transport optimal (OT) vise à faire correspondre une certaine distribution (continue ou discrète) μ à une autre distribution ν . Dans la suite de l’article, μ (respectivement ν) désigne la distribution discrète uniforme sur l’ensemble des n utilisateurs (resp. sur l’ensemble des m items). En notant $\Gamma(\mu, \nu)$ l’ensemble des mesures telles que leurs marginales par rapport au premier et deuxième argument sont respectivement μ et ν , et en définissant $C_{i,j}$ comme le coût de la correspondance de i à j , le problème du transport optimal est de trouver une distribution jointe $\gamma^* \in \Gamma(\mu, \nu)$ telle que :

$$\gamma^*(C) \in \arg \min_{\gamma \in \Gamma(\mu, \nu)} \sum_{i=1}^n \sum_{j=1}^m \gamma_{i,j} C_{i,j} \quad (1)$$

Une relaxation de ce problème d’optimisation, proposée par [9], y ajoute un terme entropique :

$$\gamma^*(C, \varepsilon) \in \arg \min_{\gamma \in \Gamma(\mu, \nu)} \sum_{i=1}^n \sum_{j=1}^m \gamma_{i,j} (C_{i,j} + \varepsilon \log(\gamma_{i,j})) \quad (2)$$

avec ε le poids donné à la régularisation. La solution optimale de l’équation (2) prend la forme $\gamma_{i,j} = \alpha_i \exp(-C_{i,j}/\varepsilon) \beta_j$, où α et β reflètent les contraintes sur les marginales de γ .

Discussion Les principales approches de recommandation basées sur l’OT supposent que la matrice collaborative observée M est générée comme optimum d’un plan d’OT basé sur un coût d’appariement C à estimer [7, 11, 16, 17] : elles apprennent C à partir des données d’entraînement et utilisent le modèle de coût estimé pour construire des appariements sur de nouvelles données.

Dans le contexte du marché du travail, il est cependant discutable de considérer que les appariements réels (*i.e.* la matrice collaborative observée M) doivent être vus comme la solution d’un plan de transport optimal : par construction, M est le résultat d’un processus décentralisé tandis que la solution d’OT résulte d’un processus centralisé (les deux ne coïncidant que sous de fortes hypothèses). En conséquence, l’approche proposée est structurée en deux phases : l’apprentissage de la fonction de coût d’appariement C à partir de M (sans supposer que M soit une solution d’un problème d’OT), et l’utilisation de C dans un problème d’OT.

1. Fourni par Pôle emploi.

3 Présentation de RECTO

Soit $\phi_{i,j} \in \mathbb{R}$ le score de recommandation de l’item j pour l’utilisateur i , et définissons l’indicateur booléen $\mathbb{1}_{i \rightarrow j, k, \phi}$ comme valant 1 si j figure parmi les k meilleures recommandations pour i d’après ϕ . L’indice ϕ sera omis par la suite lorsque le contexte ne présente pas d’ambiguïté.

Critères de performance. En plus de l’indicateur standard qu’est le Recall@ k , qui mesure la fraction d’utilisateurs pour lesquels l’item préféré est classé parmi les k meilleures recommandations,

$$\text{Recall}@k(\phi) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m M_{i,j} \cdot \mathbb{1}_{i \rightarrow j, k} \quad (3)$$

nous définissons la notion de part de marché de l’item $MS_\ell(j)$, correspondant à la fraction d’utilisateurs i pour lesquels j figure parmi les ℓ meilleurs articles recommandés à i (avec $\ell < m$) :

$$MS_\ell(j) = \frac{1}{n \times \ell} \sum_{i=1}^n \mathbb{1}_{i \rightarrow j, \ell}$$

Nous définissons la congestion comme moins l’entropie des parts de marché : plus les recommandations sont réparties uniformément sur tous items, plus leurs parts de marché sont similaires, moins la congestion est importante.

$$\text{Congestion}@l(\phi) := \sum_{j=1}^m MS_\ell(j) \log(MS_\ell(j)) \quad (4)$$

Cet indicateur de congestion est normalisé sur $[-1, 0]$ en le divisant par $\log(m)$; l’évitement parfait de la congestion est obtenu pour des parts de marché égales des items, avec -1 comme valeur correspondante de l’indicateur.

Utilisation du Transport Optimal. Comme indiqué précédemment, le transport optimal s’applique sur une matrice de coûts $C_{i,j}$ dépendant du score de recommandation appris $\phi_{i,j}$. L’apprentissage de bout en bout de $C_{i,j}$ sera considéré dans des travaux ultérieurs. Nous considérons ici $C_{i,j} = g(\phi_{i,j})$, où g est une fonction scalaire monotone, considérée comme un hyperparamètre de l’approche, telle que le coût $C_{i,j}$ de transporter i vers j augmente avec $\phi_{i,j}$ (c’est-à-dire que la pertinence de la recommandation de j à i diminue). $\phi_{i,j}$ est plafonné au score du millièmème élément recommandé à chaque i par ϕ , noté $\phi_i^{(1000)}$ ($\phi_{i,j} \leftarrow \max(\phi_{i,j}, \phi_i^{(1000)})$ dans la suite).

Quatre fonctions g ont été considérées, respectivement linéaires, exponentielles de $\phi_{i,j}$, basées sur le rang, ou de type NDCG [3].

Pour une comparaison équitable des résultats obtenus avec le même poids de régularisation entropique ε , les $C_{i,j}$ sont normalisés de sorte que $\sum_{i,j} C_{i,j} = 1$.

L’algorithme RECTO. RECTO est un processus en 2 étapes : i) apprentissage d’une fonction de score ϕ ; ii) résolution du problème de transport optimal défini à partir de $C_{ij} = g(\phi_{i,j})$.

RECTO : 1. Apprentissage de ϕ . Les deux approches d’apprentissage de ϕ considérées sont XGBOOST et des réseaux de neurones (NN). XGBOOST est un système de recommandation basé sur des arbres boostés utilisé par de nombreux algorithmes de l’état de l’art [24], qui peut être efficacement entraîné en sous-échantillonnant agressivement les paires négatives (i, j) , au détriment d’un passage à l’échelle plus limité dû à l’utilisation de variables croisées telles que la distance. NN utilise un réseau de neurones profond, dont l’architecture est adaptée aux spécificités du domaine (par exemple, en considérant des sous-modules dédiés aux informations géographiques ou aux compétences). Les descriptions de l’utilisateur x_i et de l’item y_j sont respectivement projetés dans des espaces latents, notés $z_{x,i}$ et $z_{y,j}$, et leur adéquation $\phi_{i,j}$ est paramétrée comme $z_{x,i}^T A z_{y,j}$ avec A une matrice [4]. Les projections et la matrice A sont apprises de manière end-to-end en utilisant une perte “triplet” [25]. Formellement, en supposant que chaque utilisateur de l’entraînement i est associé à un seul item j , on peut réordonner les colonnes de la matrice de telle sorte que l’utilisateur i soit associé à l’item i , et l’objectif d’apprentissage est alors :

$$\text{Minimiser} \sum_{i=1}^n \sum_{j=1, j \neq i}^m (\phi_{i,i} - \phi_{i,j} + \eta)_+ \quad (5)$$

avec $A_+ = \max(0, A)$ et $\eta > 0$. En pratique, un échantillonnage négatif est utilisé pour faire face au nombre de paires négatives. Les hyperparamètres de XGBOOST et NN sont détaillés en annexe B.

RECTO : 2. Transport optimal. En fonction du poids de régularisation ε et de la fonction g (avec $C_{ij} = g(\phi_{i,j})$), la distribution discrète γ est entraînée en optimisant l’équation (2). Remarquons que l’extension de l’approche au cas général de recommandation réciproque (e.g. où plusieurs postes peuvent être ouverts pour l’offre d’emploi j) est immédiate en rendant ν_j proportionnelle à la contrainte de capacité de l’item j .

Enfin, pour déterminer la recommandation finale, RECTO procède de manière déterministe, en classant les j articles recommandés à l’utilisateur i par ordre décroissant par rapport à $\gamma_{i,j}$.

4 Validation expérimentale

Cette section présente la validation expérimentale de l’approche, réalisée sur deux jeux de données : des données publiques sur le marché matrimonial (notées MAR), utilisées par [16], pour une comparaison avec l’état de l’art ; et un ensemble de données propriétaires (notées JOB) fournies par *Pôle emploi*.

Le premier objectif des expériences est d’évaluer l’efficacité de l’approche proposée en termes de compromis entre le recall et la congestion. Le second objectif est d’étudier comment les résultats dépendent des hyperparamètres de l’approche - ϕ appris avec XGBOOST ou NN ; poids de régularisation entropique ε variant entre $10^{-2}, \dots, 10^2$; coût de transport C_{ij} défini comme $g(\phi_{ij})$ avec g variant entre {Id, Exp, Ndcg, Rank}.

Sept indicateurs de performance sont associés à chaque hyperparamètre : $\text{recall}@k$ avec $k = 1, 10, 100$, $\text{congestion}@k$ avec $k = 1, 10$, et $\text{coverage}@k$ avec $k = 1, 10$, indiquant la fraction d'éléments impliqués dans la recommandation top- k d'au moins un utilisateur. D'autres indicateurs de performance sont présentés et discutés dans l'annexe C.

4.1 Données MAR

Description des données Les données comprennent 2 475 hommes et 2 475 femmes, répartis en 50 groupes. Chaque individu est décrit par 11 caractéristiques principalement ordinales. L'appariement 1 à 1 est décrit au niveau individuel et les données comprennent également la matrice collaborative $M_{c,c'}$, qui indique la fraction d'appariements entre les hommes du groupe c et les femmes du groupe c' .

Benchmarks. Les résultats de référence sur MAR sont ceux de RIOT [16], utilisant une factorisation basée sur SVD et itemKNN [8]. Les indicateurs de performance comprennent le RMSE et le MAE entre la matrice collaborative M au niveau des groupes et la matrice de recommandation estimée, mesurée à l'aide d'une validation croisée à 5 plis. RECTO est également évalué au niveau individuel, à l'aide des indicateurs de performance définis en section 4.

Resultats. Les tableaux 1 et 2 affichent respectivement les résultats obtenus sur MAR au niveau du groupe² et au niveau individuel.

Au niveau des groupes, γ^{XGB} obtient de légèrement meilleurs résultats (de manière statistiquement significative) par rapport à RIOT en termes de RMSE ($8,89 \pm 0,11$ contre $8,98 \pm 0,17$) et de MAE ($5,80 \pm 0,13$ contre $5,79 \pm 0,12$). γ^{NN} est également légèrement meilleur que RIOT. Les autres algorithmes de référence (aléatoire, PMF, SVD et itemKNN) sont aussi dominés.

Au niveau individuel, XGBOOST est significativement plus performant que NN en termes de recall et de congestion pour toutes les valeurs de k considérées.

γ^{XGB} améliore la congestion au détriment du recall : l'amélioration de la congestion (de $-.84$ à $-.98$) est obtenue en diminuant le $\text{recall}@10$ (de 28.4% à 23.7% pour la combinaison d'hyperparamètres $g = Id, \varepsilon = 10^{-2}$).

Pour γ^{NN} , la congestion peut être considérablement améliorée (de $-.84$ à $-.98$, pour $g = Id, \varepsilon = 10^{-2}$) tout en préservant le $\text{recall}@10$ (environ 15.4%).

Les indicateurs de recall et de congestion ne sont toutefois pas nécessairement antagonistes dans le problème de MAR : par construction, la matrice collaborative recherchée est une permutation. La principale difficulté de ce problème de recommandation semble donc provenir de la petite taille du jeu de données et de la description peu détaillée des individus.

2. La différence avec [16] s'explique par le fait qu'un bug a été trouvé (et corrigé) dans le code public pour RIOT et les autres algorithmes utilisés par [16], au niveau de la division de l'erreur par le nombre de plis dans chaque itération. Le classement final des algorithmes n'est pas affecté par la correction du bug.

4.2 Données JOB

Description des données L'ensemble d'entraînement comprend environ 1 650 000 demandeurs d'emploi, 477 000 offres d'emploi et 43 000 appariements (contrats signés) en Ile de France au cours de la période février-octobre 2018. La description x_i (respectivement y_j) d'un demandeur d'emploi (resp. d'une offre d'emploi) se trouve dans \mathbb{R}^{448} (resp. \mathbb{R}^{582}). La fonction ϕ est apprise sur l'ensemble d'apprentissage; le plan de transport optimal γ est calculé sur l'ensemble de test restreint au secteur d'emploi de la logistique pour des raisons de passage à l'échelle : il comprend 110 000 demandeurs d'emploi, 14 200 annonces d'emploi et 450 embauches en nov. 2018.

Résultats. Le tableau 3 résume les résultats, avec trois enseignements principaux. Premièrement, NN est dominé par XGBOOST sur les trois indicateurs de performance : recall, coverage et congestion; en outre, le moindre recall de NN (4% de perte de $\text{recall}@100$) s'accompagne d'un coverage beaucoup plus faible (7% de perte de $\text{coverage}@1$). Cette contre-performance est imputée à l'architecture du réseau neuronal (des expériences en cours suggèrent que l'ajout d'un second réseau n'utilisant que le Top 1000 de NN en entrée permet à NN de rattraper son retard). Deuxièmement, la coverage ($@1$ et $@10$) augmente de façon monotone et le recall ($@1, @10, @100$) diminue de façon monotone lorsque ε passe de 1 à 0,01, ce qui laisse peu d'espoir de pouvoir combiner un bon coverage avec un recall décent. Il est plus encourageant de constater que la congestion@1 peut être améliorée de manière significative (de $-0,62$ à $-0,78$) au prix d'une perte de recall modérée (le $\text{recall}@10$ passe de 62% à 56%) pour $g = Id, \varepsilon = 0, 1$.

L'option $g = ndcg$ a peu d'effets (légèrement positifs) pour $\varepsilon = 1$ et des effets fortement négatifs pour $\varepsilon = .1$ ou $.01$.

Il est surprenant de constater qu'en diminuant ε , on obtient une meilleure congestion (plus faible) au prix d'un moins bon recall. Cet effet est inattendu car plus ε est élevé, plus le plan de transport γ est uniforme (toutes choses étant égales par ailleurs) et plus la congestion semblerait devoir être faible. Une interprétation possible (étayée par des expériences complémentaires présentées dans l'annexe A) est que tous les indicateurs de performance dépendent de l'ordre induit par γ , par opposition aux valeurs réelles de $\gamma_{i,j}$. Bien que la variance de $\gamma_{i,j}$ diminue à mesure que ε augmente, le phénomène du "winner takes all" persiste, c'est-à-dire que les premières recommandations de tous les utilisateurs ne couvrent qu'un maigre 13% à 20% des offres.

Les temps de calcul (tableau 4) pour XGBOOST (environ 2 heures) et NN (environ 30 mn) restent relativement limités. Le temps associé au calcul du transport optimal augmente à mesure que ε diminue, jusqu'à environ 10 mn pour $\varepsilon = .01$ (voir aussi [22]). La majorité du temps de calcul est consacrée au calcul des recommandations avec XGBOOST et γ^{XGB} , du fait de la nécessité de calculer les ϕ_{ij} pour toutes les paires $i - j$.

TABLE 1 – Résultats sur MAR au niveau des groupes ; moyenne et écart-type de la RMSE et de la MAE par rapport à la matrice collaborative M . Les résultats pour RECTO correspondent à $g = Id+, \varepsilon = 1$.

	Random	PMF	SVD	itemKNN	RIOT	γ^{NN}	γ^{XGB}
RMSE	10.71± 0.13	446.6± 9.86	441.4± 11.2	9.36± 0.12	9.12± 0.12	8.98± 0.17	8.89± 0.11
MAE	7.22± 0.06	251.3± 6.00	249.2± 5.71	6.30± 0.03	5.98± 0.10	5.80± 0.13	5.79± 0.12

TABLE 2 – Résultats sur MAR au niveau individuel : Recall, Coverage et Congestion.

		Algorithm	Recall (%)		Coverage (%)		Congestion	
			@1	@10	@1	@10	@1	@10
		ϕ Random	0.16	2.27	63.32	100	-0.90	-0.98
RECTO-XGB		ϕ XGBOOST	7.93	27.88	48.55	98.69	-0.84	-0.94
		$\gamma^{XGB}, g = Id+, \varepsilon = 1.0$	8.05	28.41	49.77	99.18	-0.85	-0.95
		$\gamma^{XGB}, g = Id+, \varepsilon = 0.1$	8.01	27.02	72.73	100	-0.93	-0.95
		$\gamma^{XGB}, g = Id+, \varepsilon = 0.01$	6.47	23.77	96.05	100	-0.98	-0.84
		$\gamma^{XGB}, g = ndcg, \varepsilon = 1.0$	7.93	28.2	48.55	99.02	-0.84	-0.95
		$\gamma^{XGB}, g = ndcg, \varepsilon = 0.1$	8.10	25.72	59.42	100	-0.89	-0.93
		$\gamma^{XGB}, g = ndcg, \varepsilon = 0.01$	6.06	19.49	94.26	100	-0.98	-0.73
RECTO-NN		ϕ NN	3.82	15.50	46.27	98.00	-0.83	-0.93
		$\gamma^{NN}, g = Id+, \varepsilon = 1.0$	2.84	14.32	38.86	92.47	-0.80	-0.90
		$\gamma^{NN}, g = Id+, \varepsilon = 0.1$	3.94	15.46	70.12	100	-0.92	-0.98
		$\gamma^{NN}, g = Id+, \varepsilon = 0.01$	3.78	15.46	93.48	100	-0.98	-0.95
		$\gamma^{NN}, g = ndcg, \varepsilon = 1.0$	3.82	15.63	46.27	98.73	-0.83	-0.94
		$\gamma^{NN}, g = ndcg, \varepsilon = 0.1$	4.23	13.87	57.99	99.91	-0.88	-0.93
		$\gamma^{NN}, g = ndcg, \varepsilon = 0.01$	2.89	11.60	93.44	100	-0.98	-0.72

TABLE 3 – Résultats sur JOB : Recall, Coverage et Congestion.

		Algorithm	Recall (%)			Coverage (%)		Congestion	
			@1	@10	@100	@1	@10	@1	@10
		ϕ Random	0	0.21	0.65	99.95	100	-0.99	-0.99
RECTO-XGB		ϕ XGB	9.62	31.40	61.59	12.94	25.16	-0.62	-0.64
		$\gamma^{XGB}, g = Id+, \varepsilon = 1.0$	4.81	21.99	57.87	21.61	31.76	-0.74	-0.75
		$\gamma^{XGB}, g = Id+, \varepsilon = 0.1$	2.18	15.31	56.01	27.54	41.24	-0.78	-0.81
		$\gamma^{XGB}, g = Id+, \varepsilon = 0.01$	4.37	20.45	43.21	46.75	57.61	-0.85	-0.79
		$\gamma^{XGB}, g = ndcg, \varepsilon = 1.0$	9.62	31.61	62.36	12.96	26.14	-0.62	-0.67
		$\gamma^{XGB}, g = ndcg, \varepsilon = 0.1$	8.97	25.38	46.06	14.69	30.84	-0.67	-0.74
		$\gamma^{XGB}, g = ndcg, \varepsilon = 0.01$	5.03	14.00	18.81	36.81	57.52	-0.82	-0.81
RECTO-NN		ϕ NN	5.68	28.66	57.98	6.02	17.78	-0.46	-0.49
		$\gamma^{NN}, g = Id+, \varepsilon = 1.0$	6.78	26.14	60.39	11.99	26.30	-0.62	-0.65
		$\gamma^{NN}, g = Id+, \varepsilon = 0.1$	2.40	19.03	50.43	28.23	40.16	-0.80	-0.79
		$\gamma^{NN}, g = Id+, \varepsilon = 0.01$	3.93	16.30	27.89	53.38	62.35	-0.83	-0.70
		$\gamma^{NN}, g = ndcg, \varepsilon = 1.0$	5.68	27.46	59.08	6.02	19.75	-0.46	-0.55
		$\gamma^{NN}, g = ndcg, \varepsilon = 0.1$	5.25	23.3	49.01	8.85	26.40	-0.53	-0.65
		$\gamma^{NN}, g = ndcg, \varepsilon = 0.01$	1.53	12.36	24.28	35.41	51.56	-0.81	-0.81

TABLE 4 – Temps de calcul en secondes sur JOB (moyenne sur toutes les options g). NN a été entraîné sur un serveur équipé de 2 processeurs Intel Xeon Silver 4214 2.2 GHz, de 192Go de RAM et d’un GPU Tesla T4. XGBOOST est entraîné sur un serveur DELL PowerEdge R640 avec 2X Intel Xeon Gold 6130 2.10GHz CPUs (2×16 cores) et 384Go RAM. Le plan de transport optimal est calculé sur le DELL avec les mêmes ressources que pour XGBOOST.

Temps de calcul	ϕ		γ^{XGB}			γ^{NN}		
	XGBoost	NN	$\varepsilon = 0.01$	$\varepsilon = 0.1$	$\varepsilon = 1$	$\varepsilon = 0.01$	$\varepsilon = 0.1$	$\varepsilon = 1$
Total	104,340	4,104	104,778	104,394	104,385	4,611	4,156	4,148
(incl. Appr./OT)	(7,454/–)	(2,039/–)	(7,454/438)	(7,454/54)	(7,454/45)	(2,039/507)	(2,039/52)	(2,039/44)

5 Conclusion et Perspectives

Dans la lignée de l’IA éthique [14], cet article vise à prévenir les effets indésirables des systèmes de recommandation de bien rivaux, notamment dans le domaine du marché du travail. En effet, si certaines offres d’emploi sont recommandées à de nombreux demandeurs d’emploi, un phénomène de congestion est observé au niveau global, entraînant un gaspillage de temps et d’autres effets préjudiciables pour les demandeurs d’emploi comme pour les recruteurs.

L’approche proposée s’inspire du transport optimal, avec l’idée de "transporter" globalement la population des demandeurs d’emploi sur la population des offres d’emploi, en visant un recall décent avec une faible congestion. La question clé devient alors la définition du coût de transport. Dans cet article, le coût de transport est basé sur un score de recommandation appris de façon classique (sans prendre en compte la congestion). Les leçons surprenantes tirées de l’application de l’approche sur un vaste jeu de données issu du monde réel sont que le coût de transport et la régularisation du transport (utilisée pour garantir le passage à l’échelle de l’OT) interagissent de manière subtile. En particulier, une forte régularisation (qui devrait normalement donner un plan de transport uniforme) dégrade considérablement le recall, *tout en n’améliorant pas la congestion*.

Ce travail ouvre deux perspectives principales. Du côté algorithmique, des travaux futurs se pencheront sur l’apprentissage de bout en bout du plan de recommandation, en tenant compte à la fois du recall et du coverage. Un objectif intermédiaire est d’apprendre la fonction g utilisée pour définir les coûts de transport à partir de la fonction de score.

Une autre perspective, dans le cadre du marché du travail ou des rencontres matrimoniales, consisterait à estimer les caractéristiques structurelles du comportement individuel, telles que l’aversion au risque ou à la concurrence. Ces caractéristiques pourraient à leur tour être incorporées aux recommandations, permettant de prendre en compte le niveau de congestion acceptable pour différents types d’individus dans la définition d’une stratégie globale évitant la congestion.

Plus fondamentalement, la construction d’un "système de recommandation d’emploi équitable" devrait être considérée comme l’apprentissage d’un modèle prescriptif (*suivez cette politique pour atteindre les objectifs souhaités*), plutôt que d’un modèle prédictif (*suivez cette politique car elle estime avec précision les préférences des utilisateurs*) [13, 20, 23]. Les mérites et les limites d’un tel modèle prescriptif nécessiteront des expériences fondées sur la métho-

dologie des essais contrôlés randomisés [19].

Remerciements

Cette recherche a été soutenue par l’institut convergences DATAIA dans le cadre du "Programme d’Investissement d’Avenir" (ANR-17-CONV-0003) géré par l’Institut Polytechnique de Paris et l’Université Paris-Saclay.

Références

- [1] Charu C Aggarwal. *Recommender systems*, volume 1. Springer, 2016.
- [2] Fedor Borisyuk, Liang Zhang, and Krishnaram Kenthapadi. Lijar : A system for job application redistribution towards efficient career marketplace. In *Proceedings of the 23rd ACM SIGKDD*, pages 1397–1406. ACM, 2017.
- [3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.
- [4] Gal Chechik, Uri Shalit, Varun Sharma, and Samy Bengio. An online algorithm for large scale image similarity learning. *Advances in Neural Information Processing Systems*, 22 :306–314, 2009.
- [5] Kuan-Ming Chen, Yu-Wei Hsieh, and Ming-Jen Lin. Prediction and congestion in two-sided markets : Economist versus machine matchmakers. *SSRN Electronic Journal*, 2019.
- [6] Kuan-Ming Chen, Yu-Wei Hsieh, and Ming-Jen Lin. Reducing recommendation inequality via two-sided matching : A field experiment of online dating. *SSRN Electronic Journal*, 01 2020.
- [7] Pierre-André Chiappori and Bernard Salanié. The econometrics of matching models. *Journal of Economic Literature*, 54(3) :832–61, 2016.
- [8] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys ’10*, page 3946, New York, NY, USA, 2010. Association for Computing Machinery.
- [9] Marco Cuturi. Sinkhorn distances : Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, page 22922300, 2013.

- [10] Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2018.
- [11] Alfred Galichon and Bernard Salanié. Cupids invisible hand : Social surplus and identification in matching models. *Available at SSRN 1804623*, 2020.
- [12] Stanislao Gualdi, Matús Medo, and Yi-Cheng Zhang. Crowd avoidance and diversity in socio-economic systems and recommendation. *CoRR*, abs/1301.1887, 2013.
- [13] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.
- [14] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27, 2018.
- [15] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems—a survey. *Knowledge-Based Systems*, 123 :154–162, 2017.
- [16] Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. *J. Mach. Learn. Res.*, 20(80) :1–37, 2019.
- [17] Ruishan Liu, Akshay Balsubramani, and James Zou. Learning transport cost from subset correspondence. *arXiv preprint arXiv :1909.13203*, 2019.
- [18] Ivan Palomares, Carlos Porcel, Luiz Pizzato, Ido Guy, and Enrique Herrera-Viedma. Reciprocal recommender systems : Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *arXiv preprint arXiv :2007.16120*, 2020.
- [19] W. Pearce and S. Raman. The new randomised controlled trials (RCT) movement in public policy : challenges of epistemic governance. *Policy Sci*, 47 :387402, 2014.
- [20] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference : foundations and learning algorithms*. MIT press, 2017.
- [21] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6) :355–607, 2019.
- [22] Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *arXiv preprint arXiv :1610.06519*, 2019.
- [23] Flavian Vasile and Stephen Bonner. Causal embeddings for recommendation : An extended abstract. In *IJCAI-19*, pages 6236–6240. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [24] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. Content-based neighbor models for cold start in recommender systems. In *Proceedings of the Recommender Systems Challenge 2017 - RecSys Challenge 17*. ACM Press, 2017.
- [25] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10(2), 2009.
- [26] Bin Xia, Junjie Yin, Jian Xu, and Yun Li. We-rec : A fairness-aware reciprocal recommendation based on walrasian equilibrium. *Knowledge-Based Systems*, 182, 08 2019.

Appendices

Annexe A : Une régularisation entropique plus élevée peut ne pas réduire la congestion

D’après [21] (prop. 4.1), lorsque le poids ε du terme de régularisation entropique tend vers ∞ , la solution γ du problème de transport optimal régularisé tend vers une distribution uniforme. Lorsque $\varepsilon \rightarrow 0$, en revanche, la solution converge vers le plan de transport optimal avec entropie maximale. De manière informelle, augmenter ε conduit à des solutions γ moins parcimonieuse.

Cependant, de manière inattendue, l’utilisation de γ pour un processus de recommandation par classement n’implique pas nécessairement qu’un γ plus uniforme entraîne moins de congestion.

Ce phénomène peut être analysé en simulation. 1 000 matrices de coût C de taille $n = 30$, $m = 10$ sont générées indépendamment, avec $C_{ij} \sim \mathcal{U}(\frac{j}{m}, \frac{j}{m} + 1)$ (les items étant ordonnés par attractivité croissante). Les plans de transport γ avec marges uniformes par rapport aux utilisateurs et aux items sont ensuite calculés à l’aide de l’algorithme de Sinkhorn et un poids de régularisation entropique $\varepsilon = 100$ et $\varepsilon = 0,01$. La moyenne et l’écart type sur les 1 000 exécutions de la congestion obtenue après les classements issus ces plans indiquent que la congestion est significativement plus élevée pour la valeur de ε la plus élevée :

ε	moyenne de congestion@1	écart-type
100	-0.940521	0.029445
0.01	-0.996059	0.003586

Les figures 1, 2, 3 et 4 illustrent ce phénomène sur une seule simulation représentative. Les valeurs de γ_{ij} sont plus uniformes lorsque $\varepsilon = 100$ que lorsque $\varepsilon = 0,01$, mais le classement par ligne conduit à une répartition plus inégale des recommandations vers les différentes offres.

Dans l’ensemble, une régularisation entropique plus élevée a un impact indéterminé sur la congestion et peut même l’augmenter en pratique. Le choix de ε doit donc être basé sur un ensemble de validation, ainsi que sur des critères numériques pour la convergence de l’algorithme de Sinkhorn. On peut noter que l’utilisation de valeurs extrêmement faibles de ε avec une implémentation naïve de l’algorithme de Sinkhorn peut avoir des conséquences néfastes sur la stabilité numérique ainsi que sur la vitesse de convergence, bien que des alternatives aient été développées, par exemple dans [22].

FIGURE 1 – Coûts bruts

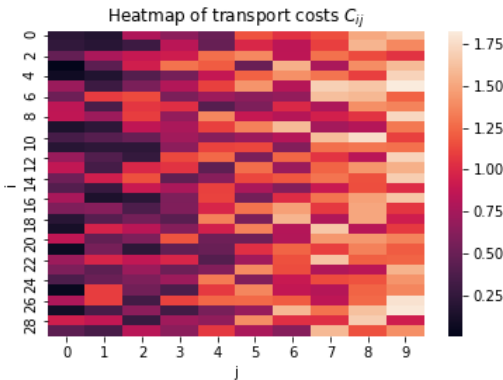


TABLE 7 – NN : Hyperparamètres pour MAR

Couche 1	<i>tanh</i> , dim. 300
Embedding	<i>tanh</i> , dim. 300
Algorithme d'apprentissage	Adam
Taux d'apprentissage	0.001
Epochs	300
Taille de <i>batch</i>	64
Ratio d'échantillonnage négatif (par <i>epoch</i>)	10

Annexe B : Hyperparamètres

Cette annexe détaille les hyperparamètres utilisés pour entraîner XGBOOST et NN sur les deux jeux de données.

XGBOOST

Sur MAR, XGBOOST est utilisé avec ses paramètres par défaut, à l'exception du nombre d'arbres, fixé à 200. Une perte logistique est utilisée et le taux d'échantillonnage négatif est fixé à 50 (Tableau 5).

TABLE 5 – XGBOOST Hyperparamètres pour MAR

Nombre d'arbres	200
Fonction de perte	Logistique
Ratio d'échantillonnage négatif	50

Sur JOB, XGBOOST est utilisé avec les hyperparamètres indiqués dans le tableau 6. Les autres hyperparamètres sont fixés à leur valeur par défaut.

TABLE 6 – XGBOOST Hyperparamètres pour JOB

<i>col_sample_bytree</i>	0.6
<i>eta</i>	0.075
<i>gamma</i>	0.85
<i>max_depth</i>	12
<i>min_child_weight</i>	1
<i>subsample</i>	0.9
Nombre d'arbres	400
Fonction de perte	Logistique
Ratio pour l'échantillonnage négatif	50

NN

Le paramètre de marge η dans la fonction de perte est fixé à 1 dans toutes les expériences.

Sur MAR, NN est utilisé avec les hyperparamètres du Tableau 7. Dans chaque batch, 10 paires négatives sont sélectionnées uniformément pour chaque positive.

Sur JOB, le modèle NN utilise une architecture définie pour le marché de l'emploi, avec trois modules neuronaux. Le

premier module "géographique" prend les coordonnées latitude et longitude standardisées d'une commune, et produit une représentation de dimension 50 en utilisant deux couches cachées de taille 100 et des fonctions d'activation *tanh*. Le module est entraîné pendant 100 *epochs* avec l'algorithme d'optimisation Adam et un taux d'apprentissage initial de 10^{-4} . Les exemples négatifs sont choisis en prenant des items plus éloignés (géographiquement) que l'item positif. Le deuxième module "compétences" prend les compétences standardisées et les réduit à une représentation de dimension 100 avec une couche cachée de taille 200 et une fonction d'activation RELU. La matrice de similarité est diagonale, et le module est entraîné pendant 100 *epochs* avec l'algorithme d'optimisation Adam et un taux d'apprentissage initial 10^{-4} . Le troisième module "autre" prend les autres variables, avec une couche cachée de taille 500 et produit une représentation de dimension 200. Le module est entraîné pendant 100 *epochs* avec l'algorithme d'optimisation Adam et un taux d'apprentissage initial 10^{-4} . L'architecture d'ensemble est initialisée en utilisant les poids pré-entraînés des trois modules précédents, et la matrice de similarité A est contrainte d'être diagonale par blocs. Le module est ensuite entraîné pendant 35 *epochs* avec l'algorithme d'optimisation Adam et un taux d'apprentissage initial 10^{-4} , en utilisant des exemples négatifs choisis uniformément. Les autres hyperparamètres sont détaillés dans la Table 6.

Annexe C : Résultats complets

La figure 5 présente les résultats de toutes les méthodes dans le plan 2D $\text{recall}@10$, $\text{congestion}@10$. La figure 6 présente des courbes de Lorenz pour illustrer la qualité de recommandation. Enfin, le tableau 4 présente les temps de calcul pour chaque méthode. Les temps d'entraînement pour XGBOOST et NN sont limités à respectivement 2 heures et 30 minutes. Le temps du transport optimal augmente avec la diminution de ϵ jusqu'à environ 10 minutes pour $\epsilon = 0,01$. Le coût le plus important vient du calcul des recommandations avec XGBOOST et γ^{XGB} en raison de la nécessité de calculer des variables d'adéquation jointe pour toutes les paires (utilisateur, item).

FIGURE 2 – $\varepsilon = 100$

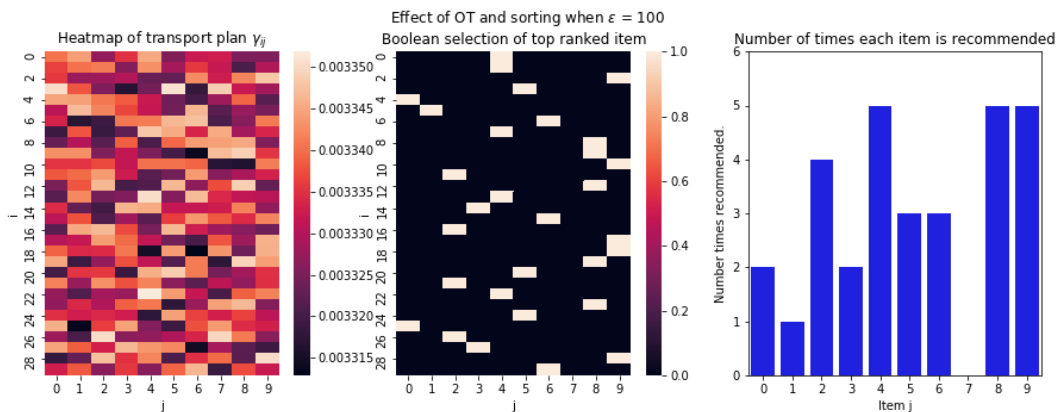


FIGURE 3 – $\varepsilon = 0.1$

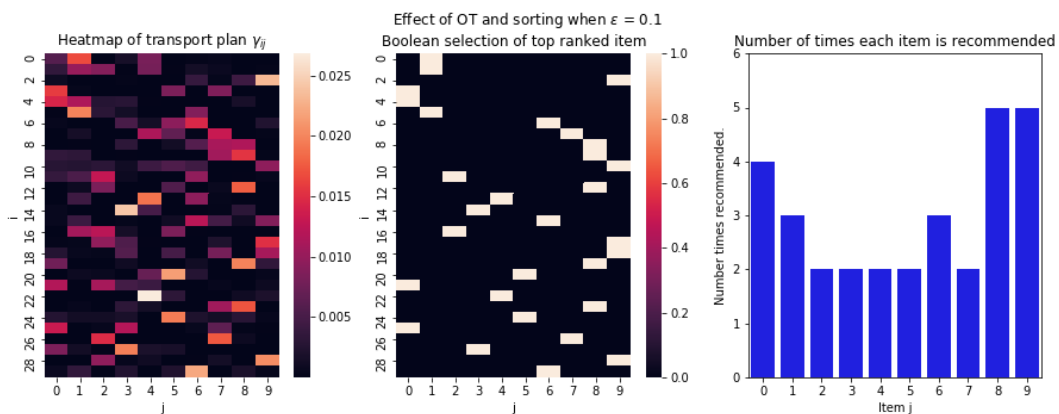


FIGURE 4 – $\varepsilon = 0.01$

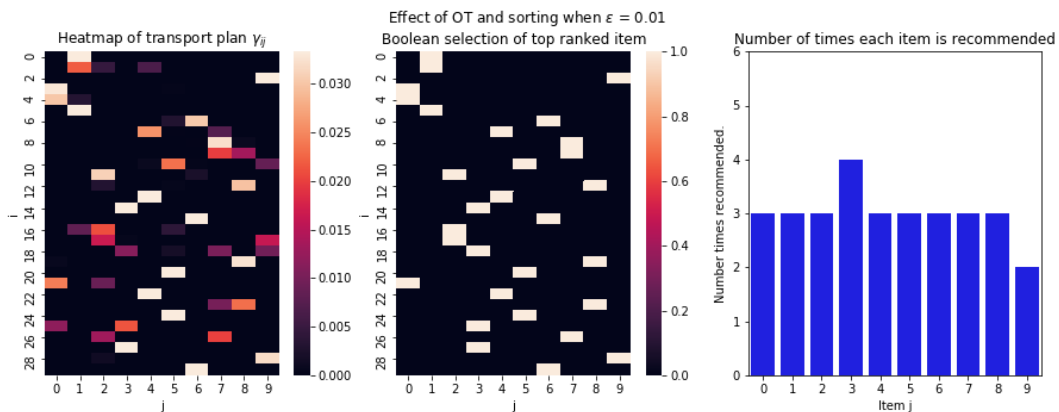


FIGURE 5 – Front de Pareto congestion (-Congestion@10) - précision de la recommandation (Recall@10), ensemble de données JOB

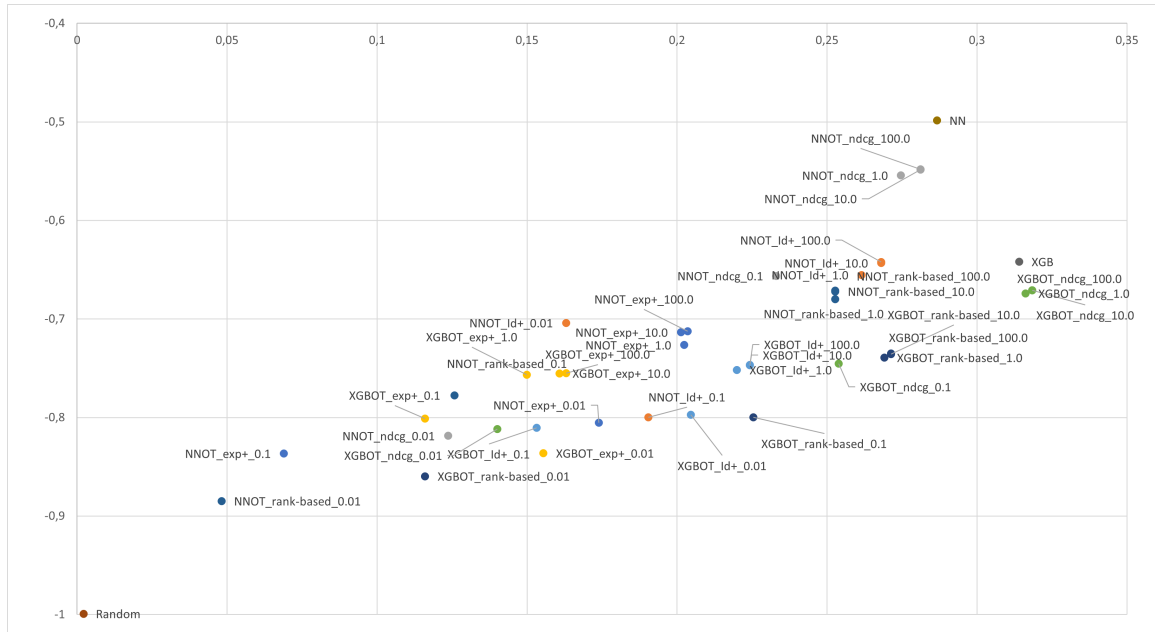
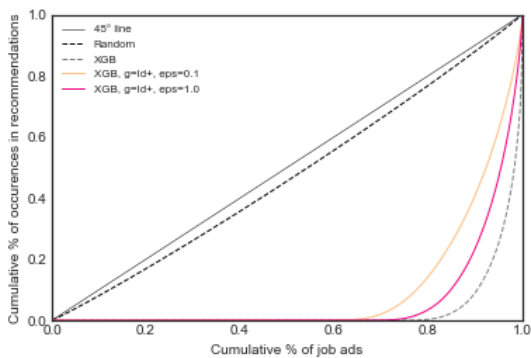
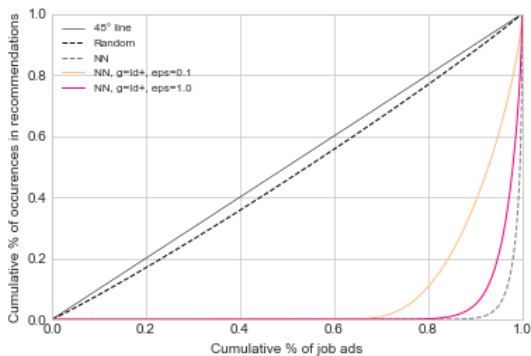


FIGURE 6 – Courbes de Lorenz calculées sur les recommandations du Top10



(a) XGB sur le jeu de données JOB



(b) NN sur le jeu de données JOB

TABLE 8 – Hyperparamètres - NN (JOB)

Sous-module géographique	
Couche 1	<i>tanh</i> , dim. 100
Couche 2	<i>tanh</i> , dim. 100
Embedding	<i>tanh</i> , dim. 50
Algorithme d'apprentissage	Adam
Taux d'apprentissage	0.0001
Epochs	100
Taille de batch	32
Sous-module compétences	
Couche 1	ReLU, dim. 200
Embedding	<i>tanh</i> , size = 100
Algorithme d'apprentissage	Adam
Taux d'apprentissage	0.0001
Epochs	100
Taille de batch	32
Module "général"	
Couche 1	ReLU, dim. 500
Embedding	<i>tanh</i> , dim. 200
Algorithme d'apprentissage	Adam
Taux d'apprentissage	0.0001
Epochs	100
Taille de batch	256
Apprentissage à chaud	
Structure bloc-diagonale	Oui
Epochs	10 / 25
Algorithme d'apprentissage	Adam
Taux d'apprentissage	0.0001 / 0.00001
Taille de batch	256

Prévention et auto-rééducation de pathologies ostéoarticulaires assistées par intelligence artificielle

L. Cervoni¹, J. Brasseur¹, M. Roudesli²

¹ Centre de Recherche et d'Innovation de Talan, Paris, France

² Centre de l'appareil locomoteur de l'estuaire, Le Havre

laurent.cervoni@talan.com, julien.brasseur@talan.com, m.roudesli@skeewai.com

Résumé

Les pathologies ostéo-articulaires et tout particulièrement les lombalgies ou les entorses de cheville sont un enjeu de santé publique. Les données décrivant l'impact des traitements sur l'évolution des pathologies sont insuffisantes pour imaginer un processus d'auto-apprentissage. Nous avons donc conçu une application basée, notamment, sur des règles de consensus médicales capable de générer des séances d'exercices adaptées à la pathologie du patient et à son évolution. Nous avons retenu une approche qui, partant de l'entorse de cheville, a pu être généralisée à 12 pathologies et a donné lieu à une application opérationnelle. Recov'up, issue de ces travaux, est disponible sous forme de WebApp (<https://recovup.com>).

Mots-clés

Troubles musculo-squelettiques, Intelligence Artificielle, Santé, Programmation Logique, Règles de consensus.

Abstract

Osteoarticular pathologies, particularly low back pain and ankle sprains, are a public health issue. The data describing the impact of treatments on the evolution of pathologies are insufficient to imagine a self-learning process. We have therefore designed an application based, in particular, on medical consensus rules capable of generating exercise sessions adapted to the patient's pathology and its evolution. We selected an approach that, starting from ankle sprains, could be generalized to 12 pathologies and resulted in an operational Web application. Recov'up, resulting from this work, is available as a WebApp (<https://recovup.com>).

Keywords

Musculoskeletal disorders, Artificial Intelligence, Health, Logical Programming, Consensus rules.

1 Introduction

Les pathologies ostéoarticulaires des membres inférieurs représentent 25 % de l'ensemble des pathologies ostéoarticulaires. La cheville, avec près de 6000 personnes atteintes [8] par jour en France, représente, à elle seule, 7 à 10 % des urgences hospitalières et plus d'un quart des pathologies des membres inférieurs. Il s'agit d'un problème de Santé

publique dont le coût est estimé à 1,2 millions d'euros par jour. Le coût total annuel, pour la société, des entorses de cheville a même été estimé approximativement à 40 millions d'euros par million d'habitants.

Le chiffre des récurrences est éloquent : 30 % des gens présentent un problème persistant au-delà de 1 an. ([11] et [10]). D'autres études pointent des séquelles encore plus importantes. Ceci souligne l'importance de la prévention et de traitements rapide, notamment par la mise en œuvre d'exercices de rééducation ([12]).

La prise en charge se fait, généralement, sur la base de « règles de consensus » médicales qui permettent d'évaluer la gravité de la pathologie. Les approches par proprioception, par exemple, ont entraîné une réduction de 50 % des récurrences [5]. Il est aussi mis en avant la place de l'*auto-rééducation dans la stratégie de rééducation*.

La comparaison d'une rééducation supervisée par un praticien et une auto-rééducation à domicile tend à montrer des résultats fonctionnels similaires pour la cheville. On observe une meilleure observance, notamment pour le respect des rendez-vous médicaux et un meilleur résultat de suivi du programme de soin.

Les lombalgies constituent aussi un enjeu de santé publique similaire, 80 % de la population étant susceptible de souffrir de lombalgie au cours de sa vie, et plus de la moitié de la population française déclare un épisode de mal de dos sur une année¹. En première intention, une autogestion et des exercices ciblés sont recommandés par la HAS. Pour cette pathologie aussi plusieurs études ont cherché à démontrer l'intérêt des soins en autonomie éventuellement assistés par une application ([2] et [6]).

Il paraît donc pertinent de pouvoir autonomiser le patient dans la réalisation de tests fonctionnels seuls ou assistés par informatique, afin que les praticiens aient accès à davantage d'informations pour personnaliser le parcours de soins.

L'intelligence artificielle peut donc jouer un rôle significatif dans l'accompagnement des rééducations des pathologies ostéoarticulaires et cette stratégie d'autonomisation. Toutefois, si l'IA connaît un nombre croissant d'applications médicales, la plupart d'entre-elles portent sur l'aide au diagnostic ou l'imagerie et beaucoup plus rarement sur la thé-

1. <https://www.ameli.fr/medecin/santeprevention/pathologies/lombalgies/enjeu-sante-publique>

rapeutique. Une des raisons majeures à cela réside dans le caractère sensible des données médicales. Aussi, dans bien des cas, il n'existe aucune base de données disponible, ce qui est le cas des pathologies ostéoarticulaires.

Dans cet article, nous présentons un cas pratique (Recov'Up) conçu dans le cadre d'une collaboration entre la start-up Skeewai et le Centre de Recherche et d'Innovation du groupe Talan. Celle-ci repose principalement sur des règles de consensus, les recommandations de la Haute Autorité de Santé² et la mise en pratique de l'expertise de praticiens (médecins du sport et kinésithérapeutes) exprimée sous forme de règles « logiques ».

2 Genèse et contexte des travaux

Au coût total pour la société, et pour les entreprises privées, de la prise en charge des pathologies telles que l'entorse de cheville et des arrêts de travail qui en découlent, s'ajoute le risque de récurrence. Une re-mobilisation rapide dans le cadre de l'entorse de cheville semble être favorable à la réduction des risques de récurrences [1] et [4].

Pourtant, seuls 50 % des cas font l'objet d'un suivi médical. Une large proportion des patients risque donc de développer une instabilité chronique de la cheville. ([9] et [3]).

Il n'existe pourtant pas de méthode médicale universelle « déterministe » pour traiter l'entorse de cheville. Cette situation n'est pas surprenante au regard du nombre de facteurs qui entrent en jeu dans la pathologie et l'essence même de la pratique médicale qui doit être la plus personnalisée.

Le site Ameli précise, concernant la rééducation, que « *Le nombre de séances préconisées est de 10 pour une entorse de cheville récente. La rééducation n'est pas systématique, et se propose en fonction des besoins estimés par le médecin.* ».

Ainsi, il existe des recommandations sur avis d'experts dont la réalisation de tests d'évaluation des déficiences sensorimotrices. Notamment, dans les travaux les plus récents et qui constituent la référence internationale en matière d'analyse de l'entorse de cheville, on observe que « *Le comité exécutif de l'International Ankle Consortium est parvenu à un consensus sur les recommandations pour une évaluation clinique structurée des lésions latérales aiguës de l'entorse de la cheville. Les recommandations sont fournies sur la base d'une évaluation clinique diagnostique initiale. L'International Ankle Consortium ROAST [...] met l'accent sur l'évaluation des déficiences mécaniques et sensorimotrices connues pour être associées à une instabilité chronique de la cheville.* » (Source : *Clinical assessment of acute lateral ankle sprain injuries (ROAST) : 2019 consensus statement and recommendations of the International Ankle Consortium*).

L'absence de démarche « algorithmique » ou simplement formelle conduit à envisager des approches expérimentales qui ne peuvent s'appuyer sur une programmation informa-

tique traditionnelle.

L'absence de base de données de patients avec les suivis thérapeutiques associés évoquée précédemment vient s'ajouter aux motivations de retenir un système à base de règles pour proposer un accompagnement thérapeutique.

Le projet RecovUp a donc été initié sur la base de ces constats, début 2020. L'équipe médicale (1 kiné, 1 médecin du sport) a collecté l'ensemble des exercices utilisés par les praticiens et disponibles dans la littérature, utilisant ou non du matériel simple. La liste des exercices varie d'une vingtaine à plus de cinquante selon les pathologies prises en compte par l'application.

La mise en œuvre des exercices dans le parcours de soin a été définie par l'équipe médicale au regard du bénéfice apporté selon la date initiale de la pathologie, de sa nature, de la capacité du patient à effectuer certains tests fonctionnels simples et de réponses sur l'évolution de la douleur, notamment. Le développement du moteur modélisant les exercices et l'interface associé s'est étalé sur un an environ avant la mise en situation réelle.

3 Complexité de la génération d'exercices d'auto-rééducation

Dans le traitement de la cheville, nous avons retenu un panel de 30 exercices environ, classés dans différentes familles (sur la base notamment des recommandations de la Haute Autorité de Santé ([7])).

L'objectif est de proposer des séances de 3, 4 ou 5 exercices (le nombre étant fixé par les règles proposées par l'équipe médicales en fonction de l'évolution de la pathologie et en évitant la monotonie des séances afin de ne pas lasser le patient), choisis dans cette liste de 30 exercices.

L'espace complet pour 3 à 5 exercices est théoriquement $C_{30}^3 + C_{30}^4 + C_{30}^5 = 4060 + 27405 + 142506 = 173971$. Dans la pratique, l'espace réel est un peu plus complexe à calculer, certains exercices devant, selon les situations, être pris dans une famille ou une autre et le nombre d'exercices ayant été étendu. Certaines familles sont exclusives des autres ou au contraire être associées.

Si on introduit les notions de durées, l'espace est nettement plus important. La gestion des durées se traduit par l'ajout de répétitions afin de rentrer dans un intervalle de temps pour qu'une séance soit acceptable pour le patient avec une durée minimale efficace.

Si on souhaite « simuler », dans un domaine discret, la gestion des exercices, cela reviendrait à ajouter à la liste des exercices tous les cas de répétitions possibles. Comme les exercices peuvent être réalisés sur des séries de 8 à 15 répétitions, 3 à 5 fois chacun, il y aurait donc un espace de 40 à 75 représentants de chaque exercice.

Dans ce cas, l'espace total dépasse plusieurs millions de combinaisons possibles (dans le pire des cas) puisqu'il faudrait calculer toutes les associations de répétitions.

Définition des règles

1. des règles générales :

2. https://www.has-sante.fr/upload/docs/application/pdf/2019-04/fm_lombalgie_v2_2.pdf et <https://www.has-sante.fr/upload/docs/application/pdf/recomentors.pdf>.

- une séance doit comporter entre 3 et 5 exercices (selon l'évolution du patient et la pathologie);
 - les règles de progressivité qui précisent l'évolution du nombre d'exercices et de séries d'un exercice doivent toujours être respectées;
 - une séance doit débiter par une mesure de la douleur au de repos, Echelle Visuelle Analogique (EVA), et terminer par une EVA post-séance;
 - la rééducation se déroule sur une durée minimale (celle-ci pourra varier selon d'autres pathologies que la cheville) et ne s'achève que si un certain nombre de critères est atteint;
2. des règles de progressivité dont un échantillon est :
 - les exercices proposés doivent être issus d'au moins 2 familles différentes (celles-ci étant, par exemple, proprioception, renforcement, excentrique, etc.);
 - avant J5 (ie dans les 5 jours qui suivent la blessure), appliquer le protocole « Glaçage, Repos, élévation Compression » (GREC);
 - aucun exercice des familles médicalement considérées comme peu efficaces, après J15;
 - aucun exercice de sur-sollicitation avant J15;
 - ne jamais proposer moins de 2 exercices de certaines familles à partir de J21;
 - etc.;
 3. des règles d'irritabilité comme, par exemple :
 - si l'EVA post-séance est supérieure à l'EVA de repos de plus de 20 % alors ne pas suivre la règle de progressivité pour la séance suivante;
 4. des règles « de cause à effet » :
 - des familles d'exercices différentes sont proposées en fonction de tests fonctionnels réalisés et auto-évalués par le patient;
 - selon le résultat de scores (comme le Limb Symmetry Index), les exercices sont adaptés en intégrant aussi le fait que le patient dispose ou non du matériel nécessaire (élastique, coussin instable).

Étude de complexité

Considérons, à titre d'exemple, la règle « avant le cinquième jour, ne faire que des exercices de type A (dans la modélisation les familles d'exercices sont numérotées par des lettres), et jamais après ». Cette dernière peut s'écrire sous la forme :

$$L_{prog}^{(1)} := \left(\bigwedge_{k \leq 5} NbEx(A, 1, k) \wedge \bigwedge_{j \notin A} \overline{Exer(j, k)} \right) \wedge \left(\bigwedge_{k > 5} \overline{NbEx(A, 1, k)} \right)$$

Ici, $Exer(j, k)$ est un booléen qui est actif lorsqu'un exercice de type j est réalisé au jour k , et $NbEx(A, 1, k)$ est un booléen construit à l'aide de $Exer(j, k)$ et dont la définition

formelle est :

$$NbEx(A, 1, k) := \bigvee_{\substack{\Lambda \subset \Omega \\ \#(\Lambda \cap A) \geq 1}} \bigwedge_{j \in \Lambda} Exer(j, k)$$

où A correspond à l'ensemble des exercices de type A et Ω à l'ensemble de tous les exercices. Autrement dit, $NbEx(A, 1, k)$ est un booléen qui est actif lorsque l'on réalise au moins 1 exercice de type A au jour k .

L'ensemble des règles peuvent s'exprimer sous Forme Normale Conjonctive de manière analogue. Une fois ceci réalisé, l'on peut alors déduire la complexité mathématique de notre modèle.

En effet, le problème est exprimable sous forme booléenne, *via* :

$$L_{cheville} := L_{règles} \wedge L_{progressivité} \wedge L_{Questions-tests}$$

où $L_{règles}$, $L_{progressivité}$ et $L_{Questions-tests}$ sont, respectivement, des clauses logiques regroupant l'ensemble des règles régissant : 1) les règles générales et d'irritabilité (regroupés par commodité), 2) les règles de progressivité et, enfin, 3) les règles de cause à effet des questions fonctionnelles.

Notons que, par construction de $L_{règles}$, $L_{cheville}$ se réécrit : $L_{cheville} = C \wedge L'$ où $C := \bigvee_{k=28}^{70} Fin(k)$ pour une certaine formule logique L' , et avec $Fin(k)$ où $k \in \llbracket 28, 70 \rrbracket$ (intervalle de durée des soins) vrai si l'utilisateur peut sortir de rééducation au jour k (dans la version en ligne la durée de soins peut-être prolongée mais cela ne modifie pas le raisonnement).

Comme L' peut toujours se réécrire sous forme normale conjonctive, on peut supposer sans perte de généralité qu'elle est sous cette forme.

Comme la clause C comporte 42 termes, il vient que $L_{cheville}$ est équivalent à un λ -SAT avec $\lambda \geq 42$.

4 Structure générale de Recov'Up

Le moteur de génération d'exercices correspond à la modélisation des règles exprimées par les médecins et kiné impliqués dans le projet et à la recherche d'un n -uplet satisfaisant ces règles.

L'objectif est de disposer d'un outil « intelligent » capable de construire des séances identiques à celles qu'un praticien pourrait les proposer en disposant d'un mécanisme d'explicitabilité du raisonnement et en étant plus « imagitatif » qu'un humain par l'agrégation de différentes catégories d'exercices.

Prolog a été retenu pour sa facilité à représenter les contraintes descriptives de la situation, sa capacité à inférer sur celles-ci pour déterminer si une séquence d'exercices pouvait être proposée et sa lisibilité pour les médecins. Le déroulement du programme et les solutions sont explicables ce qui permet d'ajouter ou de faire évoluer des règles pour être au plus près des situations réelles des patients.

En outre, l'application est parfaitement conforme à la gestion des données privées, aucune donnée personnelle n'étant nécessaire pour générer des séances (en effet, le genre, l'âge ou le poids, par exemple, n'apporte en réalité

aucune information médicalement utile quand on dispose des réponses aux questions posées). Cette approche permet donc de construire cette base de données inédites à partir de laquelle l’algorithme pourra devenir encore plus pertinent dans ses propositions d’exercices.

La structure des termes Prolog a été imaginée en anticipant la prise en compte d’une évolution vers d’autres pathologies ultérieures. L’interface utilisateur est conçue en PHP et la communication entre PHP et Prolog s’effectue via un JSON.

Les exercices sont classés comme des faits Prolog standards et organisés par famille et leur éligibilité potentielle dépend du jour courant, de la présence ou non de matériel spécifique ou de la capacité du patient à effectuer certains tests physiologiques :

```
exoeligible (Patient, cheville, J, il, _) :-
    J >= 15,
    exercicesCumules (Patient, cheville, _, ListeExo),
    comptemembre (h, ListeExo, Num), Num >= 3,
    questiontest (Patient, cheville, J, coussin, _, o, _).

exoeligible (Patient, cheville, J, ExoFamilleK, _) :-
    J >= 21,
    questiontest (Patient, cheville, _, sport, _, o, _),
    questiontest (Patient, cheville, _, qf11, _, Repqf11, _),
    Repqf11 \= a,
    questiontest (Patient, cheville, _, qf12, _, Repqf12, _),
    Repqf12 \= a,
    questiontest (Patient, cheville, _, qf13, _, Repqf13, _),
    Repqf13 \= a,
    exercice (cheville, k, _, ExoFamilleK, _, _).
```

Le générateur d’exercices proprement dit exploite ensuite une « capacité de la logique d’ordre 2 » de Prolog en recueillant tous les exercices potentiellement éligibles avec une commande bagof :

```
bagof (Exo, exoeligible (Patient, Pathologie, Jour, Exo, _),
      ListeExo)
```

Le prédicat exoeligible détermine les situations d’applicabilité d’un exercice et est ainsi déclenché par bagof qui remplit la ListeExo de tous les Exos correspondant à un Patient donné pour le Jour courant.

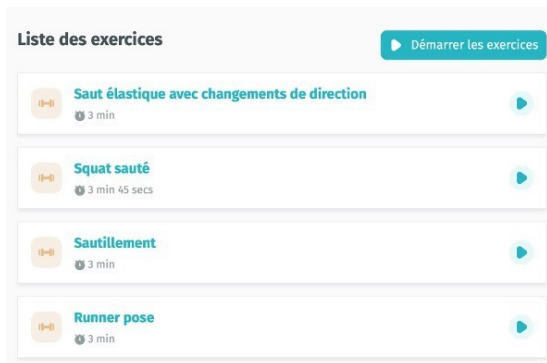


FIGURE 1 – Exemples d’exercices proposés chez un patient peu douloureux avec évolution favorable de la douleur

La série d’exercices considérés comme éligibles est ensuite filtrée par des règles supplémentaires (comme, par exemple, la non-monotonie pour éviter une succession d’exercices identiques d’un jour à l’autre) afin de générer une séquence qui est transmise via JSON au module d’interface qui affiche les exercices de la journée (voir Figure 1).

Dans l’extrait ci-dessus, on voit que la pathologie est déjà anticipée comme pouvant avoir une autre valeur que « cheville », le système ayant été étendu à 12 pathologies ostéoarticulaires.

Pour chaque pathologie, des jeux de données synthétiques ont été générées puis confrontés aux règles du moteur Prolog, les séquences d’exercices produites ayant été contrôlées par l’équipe médicale.

L’application mise à disposition des patients reste sous contrôle médical, le praticien prescripteur pouvant suivre via une interface dédiée l’évolution de la pathologie et le respect du protocole proposé. Dans les cas où la pathologie évoluerait défavorablement, l’application suggère une visite médicale.

5 Évolution vers d’autres pathologies et perspectives

Confrontés à des pathologies pour lesquelles il n’existe que peu de jeux de données permettant de construire des modèles précis d’évolution de celles-ci, nous avons adopté une approche consistant à implémenter des règles reconnues par les praticiens. Le moteur de génération d’exercices a été conçu tel que d’autres pathologies ont pu être aisément intégrées.

Prolog nous a permis d’avoir une démarche incrémentale en complétant ainsi progressivement le moteur de règles spécifiques à chaque pathologie en ayant la certitude ne de pas interférer avec les autres et en garantissant la stricte application de ces règles.

Le modèle mathématique sous-jacent confirme la complexité de la génération d’une séquence d’exercices, l’espace étant fortement combinatoire.

Recov’up propose donc une solution innovante pour la prise en charge de problèmes de santé publique. En exploitant des publications scientifiques et des règles médicales opérationnelles, nous avons conçu une application d’Intelligence Artificielle originale, frugale sur le plan de la donnée, dont l’explicabilité facilite l’adoption par les professionnels de santé.

L’expérience utilisateur a été pensée pour favoriser un usage régulier de l’application permettant ainsi le suivi de l’évolution de la pathologie. Cette collecte de données directement auprès du patient offre alors l’opportunité d’envisager des évolutions basées sur d’autres modèles d’Intelligence Artificielle.

Références

- [1] E. T. Chen, J. Borg-Stein, and K. C. McInnis. Ankle sprains : Evaluation, rehabilitation, and prevention. *Current sports medicine reports*, 18(6) :217–223, 2019.
- [2] Francis Fatoye, Tadesse Gebrye, Clara Fatoye, Chidozie Mbada, Mistura Olaoye, Adesola Odole, and Olumide Dada. The clinical and cost-effectiveness of telerehabilitation for people with nonspecific chronic low back pain : Randomized controlled trial. *Journal*

- of *Medical Internet Research mHealth uHealth*, 8(6), 2020.
- [3] Phillip Gribble, Chris Bleakley, Brian Caulfield, Carrie Docherty, Francois Fourchet, Daniel Fong, Jay Hertel, Claire Hiller, Thomas Kaminski, Patrick McKeon, Kathryn Refshauge, Evert Verhagen, Bill Vicenzino, Erik Wikstrom, and Eamonn Delahunt. 2016 consensus statement of the international ankle consortium : Prevalence, impact and long-term consequences of lateral ankle sprains. *British Journal of Sports Medicine*, 50(24) :1493–1495, June 2016.
- [4] Matthew Hoch, Jay Hertel, Phillip Gribble, Nicholas Heebner, Johanna Hoch, Kyle Kosik, Doug Long, Pinata Sessoms, Amy Silder, Danielle Torp, Katherine Thompson, and John Fraser. Effects of foot intensive rehabilitation (fire) on clinical outcomes for patients with chronic ankle instability : a randomized controlled trial protocol. *BMC Sports Science, Medicine and Rehabilitation*, 15 :1–13, 04 2023.
- [5] Maarten D. W. Hupperets, Evert A. L. M. Verhagen, and Willem van Mechelen. Effect of unsupervised home based proprioceptive training on recurrences of ankle sprain : randomised controlled trial. *BMJ*, 339, 2009.
- [6] N. Itoh, H. Mishima, Y. Yoshida, M. Yoshida, H. Oka, and K. Matsudaira. Evaluation of the effect of patient education and strengthening exercise therapy using a mobile messaging app on work productivity in japanese patients with chronic low back pain : Open-label, randomized, parallel-group trial. *JMIR mHealth and uHealth*, 10(5) :e35867, 2022.
- [7] A. Leuret, J.P. Sommereisen, J.M. Philippe, C. Le Gall, C. Busseuil, F. Carpentier, S. Le Goff, P. Atain-Kouadio, O. Fougeras, S. Stempfle, and L. Picault. Actualisation 2004 de la conférence de consensus : l’entorse de cheville au service d’urgence. https://www.sfm.u.org/upload/consensus/actualisation_entorse.pdf, 2004. 5ème conférence de consensus.
- [8] MEDiMTS. Avis. [https://www.has-sante.fr/upload/docs/evamed/CEPP-5487_A2T_15_mai_2018_\(5487\)_avis.pdf](https://www.has-sante.fr/upload/docs/evamed/CEPP-5487_A2T_15_mai_2018_(5487)_avis.pdf), mai 2018.
- [9] A. C. Pijnenburg, C. N. van Dijk, P. M. Bossuyt, and R. K. Marti. Treatment of ruptures of the lateral ankle ligaments : A meta-analysis*. *The Journal of bone and joint surgery. American volume*, 82 :761–73, 07 2000.
- [10] Hans Polzer, Karl-Georg Kanz, Wolf Prall, Florian Haasters, Ben Ockert, W. Mutschler, and Stefan Grote. Diagnosis and treatment of acute ankle injuries : Development of an evidence-based algorithm. *Orthopedic reviews*, 4(1) :e5, 2012.
- [11] Rogier M. van Rijn, Anton G. van Os, Roos M.D. Bernsen, Pim A. Luijsterburg, Bart W. Koes, and Sita M.A. Bierma-Zeinstra. What is the clinical course of acute ankle sprains ? a systematic literature review. *The American Journal of Medicine*, 121(4) :324–331.e7, 2008.
- [12] Jente Wagemans, Chris Bleakley, Jan Taeymans, Kevin Kuppens, Alexander Philipp Schurz, Heiner Baur, and Dirk Vissers. Rehabilitation strategies for lateral ankle sprain do not reflect established mechanisms of re-injury : A systematic review. *Physical Therapy in Sport*, 60 :75–83, 2023.

Analysis of planning instances without search

Martin C. Cooper¹, Arnaud Lequen¹, Frédéric Maris¹

¹ IRIT, Université Toulouse 3, France

Martin.Cooper@irit.fr, Arnaud.Lequen@irit.fr, Frederic.Maris@irit.fr

Résumé

Les planificateurs classiques actuels détectent les instances de planification insolubles au travers d'une recherche dans l'espace d'états sous-jacent. Dans cet article, nous montrons cependant qu'il est quelquefois suffisant d'utiliser un critère incomplet, mais efficace d'un point de vue calculatoire. Nous proposons une méthode permettant de tirer parti de ce critère, basé sur des techniques de programmation linéaire et en nombres entiers, dans le cas où il ne permet pas de conclure. Ce critère est central aux méthodes que nous proposons pour préciser et enrichir le modèle STRIPS, dans l'optique de collecter de nouvelles informations à son propos. Dans le cas où les informations supplémentaires ne permettent pas de s'assurer de l'insolubilité de l'instance, elles peuvent être réinvesties dans un algorithme complet intervenant ensuite, afin de l'accélérer.

Mots-clés

Planification Classique, Programmation Linéaire, Insatisfiabilité, STRIPS.

Abstract

In order to prove classical planning instances unsolvable, state-of-the-art planners resort to a state-space search. However, we show here that an incomplete, yet computationally efficient criterion is sometimes sufficient to immediately identify as unsolvable a wide range of planning instances. Based on linear and integer programming, we show in this paper how it can be leveraged, were it to fail at first. This criterion is the keystone of various techniques we propose to rewrite and enhance the STRIPS model, so as to gather new information about it. In case the newly-found bits of information are not sufficient to identify the instance as unsolvable, they can be reinvested later to speed up a complete algorithm.

Keywords

Classical Planning, Linear Programming, Unsatisfiability, STRIPS.

1 Introduction

Current classical planners resort to a search, with the goal of finding a solution-plan. They often start with the assumption that such a plan exists, and for the past few decades, significant work has been done on designing more and more efficient techniques to find solution-plans. Ho-

wever, various reasons may lead an instance not admitting any solution. Search-based planners will then explore the state-space in its entirety, potentially cutting branches of the search tree, until they realize no plan can be found. The detection of states that can not lead to any solution is often a byproduct of the heuristics used during search : an infinite heuristic value for an admissible heuristic is synonym of a dead-end state.

This is why in the recent years, there has been a renewed interest in detecting unsolvable planning instances, as illustrated by the 2016 Unsat IPC (International Planning Competition). Various techniques have been developed in the last couple of decades, such as dead-end formulas [4], traps [10, 14], and so on. However, all of these methods are based on the exploration of the state-space.

In this article, we propose to leverage a linear programming- and integer programming-based criterion to iteratively refine a planning model, to show its unsolvability. The criterion we use is fast to compute, and allows us to quickly recognize a wide range of unsolvable planning instances. However, it is not complete, in the sense that it may not recognize some unsolvable instances as such. Nevertheless, we show how to use it to iteratively refine the planning model, and keep gathering additional information about the instance with the aim that our procedure can detect that it is unsolvable.

Most of our techniques to gather information are based on a simple schema : after testing the solvability of planning instances Π' that are derived from the initial planning problem Π given as input, we deduce additional information about the problem Π if Π' is unsolvable. For instance, if the instance Π' , which is Π where operator a was removed, is proven to be unsolvable, then it means that a appears in all solution-plans of Π . In the case where one can efficiently detect some unsolvable planning instances, then lots of such derived instances Π' can be tested successfully. As the criterion we use is incomplete but fast, even though it often fails to detect unsolvable instances, it still manages to help gather new information, as lots of tests can be made in reasonable time. As more and more information is known about the planning instance, the mathematical program on which the criterion is based can also be enriched with the new knowledge, so that it can detect additional unsolvable instances.

More generally, being able to detect planning instances that have no solution can have various applications in itself. For

instance, consider the case where an instance models the attacks of a malicious user may perform on a system, with the goal of accessing restricted data. Finding that no sequence of actions may achieve this shows that the system is secure. The paper is organized as follows. In Section 2, we introduce our formalism and notations for classical planning. In Section 3, we present the mathematical-programming-based criterion we use throughout this paper. In Section 4, we show how to design tests to gather new information about a planning model. In Section 5, we report our experimental trials on standard sets of benchmarks. Section 7 is devoted to a discussion and perspectives on our findings.

2 Background

STRIPS planning instance A STRIPS planning instance is a tuple $\Pi = \langle F, I, O, G \rangle$ such that F is a set of propositional variables called *fluents*, and I is a set of fluents of F , called the *initial state*. G is a set of *literals* of F , such that no literal appears at the same time as its negation, and is called the *goal*. We will denote G^+ the set of positive literals of G , and G^- the set of negative literals. Finally, O is a set of *operators*: operators $a \in O$ are of the form $a = \langle \text{pre}(a), \text{eff}(a) \rangle$. $\text{pre}(a)$ is the *precondition* of a and $\text{eff}(a)$ is the *effect* of a , which are both sets of literals of F . We will denote $\text{eff}^+(a) = \{f \in F \mid f \in \text{eff}(a)\}$ the set of *positive effects* of a , and $\text{eff}^-(a) = \{f \in F \mid \neg f \in \text{eff}(a)\}$ its *negative effects*. We will use similar notations to define $\text{pre}^+(a)$ and $\text{pre}^-(a)$.

Note that we define a version of STRIPS with negative preconditions. However, we are not any more general than the original formulation of STRIPS. Indeed, any STRIPS instance with negative preconditions can be translated into an equivalent instance without negative preconditions in linear time, and the converse is immediate [6]. The same goes for negative goals: the original STRIPS formulation only specified positive goals. We nonetheless allow negative goals in our formulation of STRIPS, and we progressively take them into account. But one should keep in mind that most planning instances (and in particular, the ones used in our set of benchmarks) come with positive goals only: this is why we assume G^- is empty unless otherwise specified.

Without loss of generality, we assume that for all operators a , $\text{pre}^+(a) \cap \text{pre}^-(a) = \emptyset$. We also assume that $\text{eff}^+(a) \cap \text{eff}^-(a) = \emptyset$, otherwise we can remove from $\text{eff}^-(a)$ any fluent also in $\text{eff}^+(a)$. In addition, we will also suppose that $\text{eff}^+(a) \cap \text{pre}^+(a) = \emptyset$, and $\text{eff}^-(a) \cap \text{pre}^-(a) = \emptyset$, otherwise the redundant fluents from the effects can be removed. Any planning instance which does not satisfy these criteria can be transformed, in polynomial time, into an equivalent instance that complies with them.

States and plans A state s is an assignment of truth values to all fluents in F . For notational convenience, we associate s with the set of fluents of F which are true in s . An operator a can be applied to states of Π that verify its preconditions. More formally, for any state s , if $\text{pre}^+(a) \subseteq s$ and $\text{pre}^-(a) \cap s = \emptyset$, then we define the result of the application of a to s as $s[a] = (s \setminus \text{eff}^-(a)) \cup \text{eff}^+(a)$.

Given an instance $P = \langle F, I, O, G \rangle$, a *plan* is a sequence of operators $\pi = a_1, \dots, a_k$ from O such that there exists a sequence of states s_0, \dots, s_k , such that, for all $i \in 1, \dots, k$, the operator a_i is applicable in s_{i-1} , so that $s_i = s_{i-1}[a_i]$. A plan is a *solution-plan* if we have, in addition, $s_0 = I$ and $G \subseteq s_k$. We denote \mathcal{S}_Π the set of all solution-plans to Π . We say that a fluent f is established (resp. deleted) by some occurrence of an operator $a \in O$ in π if f is false (resp. true) in some state s_i , but true (resp. false) after the application of a , in state $s_{i+1} = s_i[a]$. In the rest of this paper, we will refer to solution-plans as simply plans.

3 Detecting unsolvable instances by LP

This section introduces two equivalent criteria that we use, and extend, to detect a planning instance's unsolvability. These criteria are incomplete, in the sense that they can not detect all unsolvable planning instances by themselves. However, they require very limited computational resources, and are fast to run, as they are based on linear programming, or mathematical programming in general. We will show later how to leverage those properties in order to make the most of these criteria when they are not able to detect an instance's unsolvability by themselves.

3.1 Potential-based argument

The first linear programming formulation that we worked with is based on the following argument. Suppose that we have a numerical function $\Phi : F \rightarrow \mathbb{R}^+$, that associates a *potential* to each fluent. We can then naturally define the potential of a state $s \subseteq 2^F$ as $\Phi(s) = \sum_{f \in s} \Phi(f)$. If one can prove that all goal states have a higher potential than the initial state, but the application of any operator a to any state s leads to a state s' of lesser (or equal) potential, then the planning instance has no solution-plan.

Such a function Φ can be found thanks to the following observation. In any plan, the potential of a state s' solely depends on the previous state s , and on the operator a that was applied such that $s[a] = s'$. In this case, we will say that a induced an increase in potential of $\Delta\Phi_a(s) = \Phi(s') - \Phi(s)$. One can remark that there exists an upper bound for $\Delta\Phi_a(s)$, which does not depend on s but only on a . Indeed, in the limit case, all fluents $f \in \text{eff}^+(a)$ are effectively established by a , but no fluent $f' \in \text{eff}^-(a)$ is destroyed, except when $f' \in \text{eff}^-(a) \cap \text{pre}^+(a)$. Recall that we assume, without loss of generality, that $\text{eff}^+(a) \cap \text{pre}^+(a) = \text{eff}^-(a) \cap \text{pre}^-(a) = \emptyset$.

More formally, let us consider four¹ sets of operators, with regard to some fluent f : on the one hand, the operators that will *surely add* and *surely delete* f when applied, that we denote respectively SA_f and SD_f ; on the other hand, the operators that could *possibly add* and *possibly delete* f when applied, respectively PA_f and PD_f . The latter are operators that may establish (resp. delete) f in the resulting state s' depending on whether f is false (resp. true) in the

1. Even though only three sets out of the four are needed here, we introduce all four sets as they will be useful later in the paper.

previous state s or not. More formally, the sets are defined as follows :

- $SA_f = \{a \mid f \in \text{eff}^+(a) \cap \text{pre}^-(a)\}$
- $SD_f = \{a \mid f \in \text{eff}^-(a) \cap \text{pre}^+(a)\}$
- $PA_f = \{a \mid f \in \text{eff}^+(a) \setminus \text{pre}^-(a)\}$
- $PD_f = \{a \mid f \in \text{eff}^-(a) \setminus \text{pre}^+(a)\}$

This leads to the following inequality, which models the limit case previously presented. This effectively gives us an upper bound on the change of potential induced by a from any state s , which we denote $\Delta\Phi_a(s)$. Remark that the right-hand side is independent of s .

$$\Delta\Phi_a(s) \leq \sum_{f \text{ s.t. } a \in PA_f} \Phi(f) + \sum_{f \text{ s.t. } a \in SA_f} \Phi(f) - \sum_{f \text{ s.t. } a \in SD_f} \Phi(f)$$

Now suppose that, for all operators a , the right-hand side of the previous inequation is negative. It means that applying any operator makes the potential of the state decrease. As a consequence, states that have a higher potential than the initial state cannot be reached. Note that, as the potential of a state is only determined by the potential of the fluents that are true in this state, and all potentials are positive, $\Phi(G)$ is a lower bound for the potential of any goal-state. Thus, if we also have that $\Phi(G) > \Phi(I)$, then the planning instance has no solution.

The only remaining issue is to check whether such a potential function Φ exists. As Φ is only determined by its values on the various fluents, this can be done with the following set of equations, with the set of variables $V = \{x_f \mid f \in F\}$. Intuitively, x_f corresponds to the potential $\Phi(f)$ of f .

Linear Program 1.

Variables : $V = \{x_f \mid f \in F\}$

Constraints :

$$\sum_{f \in G} x_f - \sum_{f \in I} x_f > 0 \quad (1)$$

$$\sum_{f \in \text{eff}^+(a)} x_f - \sum_{f \in \text{eff}^-(a) \cap \text{pre}^+(a)} x_f \leq 0 \quad (a \in O) \quad (2)$$

$$x_f \geq 0 \quad (f \in F) \quad (3)$$

The following proposition follows from the discussion above.

Proposition 1. *Let Π be a STRIPS instance. Suppose that there exists a solution for the Linear Program 1. Then Π has no solution.*

Note that the converse is not true : not all unsolvable planning instances are detected by the criterion we propose.

3.2 Dual linear program

The linear program presented in the previous section is hard to interpret, as the concept of potential we introduced has no reality outside of the criterion. However, we show in this section how to transform it into another program that can

equivalently allow us to detect some unsolvable instances, but whose result is easier to interpret.

To this effect, we resort to Farkas's lemma. Farkas's lemma is related to the well-known fact that in linear programming, the primal problem is feasible iff the dual problem is feasible. One version of this lemma states that exactly one of the following sets of equations has a solution : either (1) $Ay \geq d$ where $y \geq 0$, or (2) $A^t x \leq 0$ and $d^t x > 0$ where $x \geq 0$, where A is a matrix and x, y and d vectors of the appropriate sizes. Let us consider the set of equations previously mentioned. Applying Farkas's lemma, it has a solution iff the following system has no solution :

Linear Program 2.

Let $\Pi = \langle F, I, O, G \rangle$ a planning instance. We define $\mathcal{L}_{\Pi}^{op}(V, C)$ as follows :

Variables : $V = \{y_a \mid a \in O\}$

Constraints C :

$$\sum_{a \in SA_f} y_a + \sum_{a \in PA_f} y_a - \sum_{a \in SD_f} y_a \geq \delta_f^- \quad (f \in F) \quad (4)$$

$$y_a \geq 0 \quad (a \in O) \quad (5)$$

where $\delta_f^- = \mathbb{1}_G(f) - \mathbb{1}_I(f)$ ($\mathbb{1}_S(x)$ being the indicator function of set S : $\mathbb{1}_S(x) = 1$ if $x \in S$, and 0 otherwise). In this context, the variable y_a corresponds to the number of times operator a is executed in some sequence of actions. Note that y_a is positive, but not necessarily integral : this allows us to obtain a polynomial-time relaxation of the STRIPS instance. Inequality (4) states that the number of (possible) establishments of f minus the number of sure destructions of f must be greater than or equal to δ_f^- . For instance, any fluent that appears positively in the goal but not in the initial state must be established as least once. This dual version of our original linear program provides an alternative insight into the meaning of Proposition 1.

Lemma 1. *Let $\Pi = \langle F, I, O, G \rangle$ be a planning instance, $\mathcal{L}_{\Pi}^{op}(V, C)$ as defined in Linear Program 2, and π a solution-plan for Π . Let us define $c_{\pi} : O \rightarrow \mathbb{N}$ the number of occurrences of operators of O in π . Then the assignment $Y : V \rightarrow \mathbb{N}$ such that, for all $a \in O$, $Y(y_a) = c_{\pi}(a)$, is a solution for \mathcal{L}_{Π}^{op} .*

Démonstration. Let Y be as defined above. We will show that Y is a solution for \mathcal{L}_{Π}^{op} . For each fluent f , let us denote e_f the number of times a fluent is established during the execution of π , and d_f the number of times it is destroyed. Recall that a fluent f is established (resp. deleted) by some occurrence of an operator $a \in O$ in π if f is false (resp. true) before the application of the operator, but true (resp. false) after. As π is a solution plan, we have that :

$$\mathbb{1}_{G^+}(f) - \mathbb{1}_I(f) \leq e_f - d_f \leq 1 - \mathbb{1}_I(f) - \mathbb{1}_{G^-}(f)$$

which can be shown by case disjunction on whether f is in I, G^+ or G^- . We denote the inequations above in a more concise way :

$$\delta_f^- \leq e_f - d_f \leq \delta_f^+$$

2. Even though we suppose G^- empty now, we introduce the notation and argument here, for later use.

In addition, in the extreme case, f is established in π at most as many times as there are occurrences of operators a with $f \in \text{eff}^+(a)$. Remark that SA_f and PA_f form a partition of the set $\{a \mid f \in \text{eff}^+(a)\}$ ³. Hence

$$e_f \leq \sum_{a \in SA_f} Y(y_a) + \sum_{a \in PA_f} Y(y_a)$$

Similarly, the only operators $a \in O$ whose applications are guaranteed to destroy f are such that $f \in \text{pre}^+(a) \cap \text{eff}^-(a)$. Thus,

$$d_f \geq \sum_{a \in SD_f} Y(y_a)$$

By combining both inequations above, we have

$$\begin{aligned} \delta_f^- &\leq e_f - d_f \\ &\leq \sum_{a \in SA_f} Y(y_a) + \sum_{a \in PA_f} Y(y_a) - \sum_{a \in SD_f} Y(y_a) \end{aligned} \quad (6)$$

which means that Y satisfies the constraints of the form of inequation (4) of \mathcal{L}_{Π}^{op} . As a consequence, as Y is also positive, Y is a solution to \mathcal{L}_{Π}^{op} . \square

The contrapositive of Lemma 1 is an alternative proof that, if \mathcal{L}_{Π}^{op} has no solution, then neither has Π . But it allows us to show more than that, as we have the following corollaries, that we use later on :

Corollary 1. *If \mathcal{L}_{Π}^{op} has no integral solution, then the associated planning instance Π has no solution.*

Démonstration. The proof is immediate, as each operator appears an integral number of times in any solution-plan π . \square

Corollary 2. *Optimising the value of y_a within \mathcal{L}_{Π}^{op} leads to a bound on the number of times $a \in O$ must occur in a plan.*

Linear Program 2 is, in fact, a linear programming formulation of the state equation heuristic [2], as previously shown in [12]. Its efficiency for detecting unsolvable planning instances has been shown before, as it is part of the Aidos planner, which won the Unsat IPC in 2016 [13]. The planner uses the LP formulation of the operator counting heuristic to detect dead-ends during search, working on a finite domain representation (FDR) of the instance. We, however, do not resort to search, but show how to rewrite the model directly, potentially changing the linear program when doing so.

Even though we introduced \mathcal{L}_{Π}^{op} as a linear program, we showed with Lemma 1 that one can also see it as an integer program. Solving an integer program is notoriously harder and slower than solving a linear program. As the integral solutions of the set of equations form a subset of its set of rational solutions, testing the solvability of the program

³. Recall that we suppose that, for all $a \in O$, $\text{pre}^+(a) \cap \text{pre}^-(a) = \text{eff}^+(a) \cap \text{pre}^+(a) = \text{eff}^-(a) \cap \text{pre}^-(a) = \emptyset$

over integral solutions is more likely to prove that the associated planning instance has no solution. Note that Farkas's lemma does not apply in the integral case, hence the need for Lemma 1.

In the next section, we show that, in the case where the criterion introduced here fails, it can still be leveraged to gather additional information about Π .

4 Enhancing the planning problem

This section is dedicated to extending and adding information to the initial planning instance, mainly with the goal of proving it unsolvable. Through various methods, we either add or remove elements from the input model Π , or add information about Π that is not directly encodable into the model, but that can nevertheless still be included in the linear program or to make deductions. In order to do so, we will resort to two kinds of methods. In the first ones, we build variations of Π so that, if one of these variations can be deemed unsolvable through the previous linear program, then some additional information about Π can be deduced. In the second method, we do not consider *per se* a variation Π' of Π , but we directly modify the linear program \mathcal{L}_{Π}^{op} associated to Π , so that if it is unsolvable, we can deduce new specific information about Π .

In the following, we call *operation* any such method. In the specific case where the operation answers a boolean question (e.g. Is an action removable?), we call it a *test*.

In the rest of this section, we illustrate the previous general principles through various operations, that allow us to find new information about the planning instance given as input. As our goal is to detect unsolvable instances, in the following, we assume that the criterion could not detect, at first, that the instance is unsolvable and that we have to gather additional information in order to do so.

4.1 Operator counts and landmarks

Landmark detection An operator $a \in O$ is a landmark for Π if a occurs at least once in every solution-plan. We maintain through our procedure a set $L \subseteq O$ of landmarks. With regard to our framework, we can test if an operator is a landmark by removing it from the model and testing if the instance can be deemed unsolvable. More formally,

Lemma 2. *Let $\Pi = \langle F, I, O, G \rangle$ and $a \in O$. If $\Pi|_a = \langle F, I, O \setminus \{a\}, G \rangle$ is unsolvable, then a is a landmark.*

This leads us to defining the landmark detection test, as introduced below, where $\Pi|_a$ is defined in the lemma above.

LMDet

If $\Pi|_a$ is unsolvable
then add a to the set of landmarks L

Operator count One can generalize the notion of landmark, by counting the least number of times an operator appears in any solution-plan. This is why we maintain a function $n^- : O \rightarrow \mathbb{N}$, such that $n^-(a)$ is (a lower bound on) the least number of occurrences of action a in any plan. Likewise, we define $n^+(a)$ as (an upper bound on) the maximum number of times a appears in any plan. With these notations, $a \in O$ is a landmark iff $n^-(a) \geq 1$.

Reasoning on the number of occurrences of some operator $a \in O$ can be done through Linear Program 2. Indeed, as the variables are associated to the number of occurrences of each operator in some sequence of actions, one only has to find lower and upper bounds for each variable y_a in a solution of LP 2. This is why one can compute approximate values for $n^+(a)$ and $n^-(a)$ through an integral variation of our linear program, that we present below :

Integer Program 1.

Let $\Pi = \langle F, I, O, G \rangle$ a planning instance, with $O = \{a_1, \dots, a_m\}$, and $\mathcal{L}_{\Pi}^{op}(V, C)$ the associated Linear Program 2. For $a \in O$, let us define $\mathcal{L}_{\Pi}^{opt}(V, C)(a)$ such that :

Variables $V = \{y_a \mid a \in O\}$

Constraints C : Same as \mathcal{L}_{Π}^{op}

Objective function $g : \mathbb{N}^m \rightarrow \mathbb{N}$:

$$g : y_{a_1}, \dots, y_{a_m} \mapsto y_a$$

Lemma 3. Let Π a planning instance, $a \in O$ an operator and consider integer program $\mathcal{L}_{\Pi}^{opt}(V, C)(a)$ with objective function g . Then minimizing (resp. maximizing) g yields a lower (resp. an upper) bound on the value of $n^-(a)$ (resp. $n^+(a)$).

Démonstration. The proof is a consequence of Lemma 1. Let us show the case where g is minimized, as the proof for the other case is mostly identical. We denote $n_{\mathcal{L}}^-$ the value obtained by minimizing g in $\mathcal{L}_{\Pi}^{opt}(V, C)(a)$, where $a \in O$ is fixed. Suppose for a contradiction that $n^-(a) < n_{\mathcal{L}}^-$. Then there exists a plan π_a where a occurs exactly $n^-(a)$ times, by definition. By Lemma 1, there exists a solution Y_{π_a} for \mathcal{L}_{Π}^{op} where $Y_{\pi_a}(a) = n^-(a) < n_{\mathcal{L}}^-$, which contradicts the optimality of $n_{\mathcal{L}}^-$. Consequently, we have $n_{\mathcal{L}}^- \leq n^-(a)$. \square

OpCount⁺(a)

If the value $n_{\mathcal{L}}^+$ obtained by maximizing g over \mathbb{N} in $\mathcal{L}_{\Pi}^{opt}(V, C)(a)$ is bounded
 then set the current value of $n^+(a)$ to $n_{\mathcal{L}}^+$

OpCount⁻(a)

If the value $n_{\mathcal{L}}^-$ obtained by minimizing g over \mathbb{N} in $\mathcal{L}_{\Pi}^{opt}(V, C)(a)$ is non-zero
 then set the current value of $n^-(a)$ to $n_{\mathcal{L}}^-$

In the rest of this paper, we will often use the notation $\text{OpCount}(a)$ to refer to the successive application of $\text{OpCount}^-(a)$ and $\text{OpCount}^+(a)$. As experimental trials show that OpCount^- does not find all landmarks found by the test LMDet , OpCount^- does not make it redundant.

Using operator counts Once non-trivial values for some $n^+(a)$ or some $n^-(a)$ has been found (i.e. a finite or non-zero value, respectively), one can reintroduce it into the linear program in the form of additional constraints. These constraints can be introduced in either \mathcal{L}_{Π}^{op} or \mathcal{L}_{Π}^{opt} , as both programs use the same sets of variables and constraints. As the variables of the linear programs correspond to the number of occurrences of operators in some plan, adding these

constraints is straightforward for every $a \in O$:

$$y_a \leq n^+(a)$$

$$y_a \geq n^-(a)$$

4.2 Detection of removable actions

This section is concerned with finding operators $a \in O$ that never appear in any solution-plan. Even though some such operators can be detected statically by the parser of Fast Downward, some others require additional computation. We present various techniques that allow us to detect if an operator can be immediately removed from the planning instance, without altering its set of solutions.

Through a modification of the linear program We start by extending \mathcal{L}_{Π}^{op} into $\mathcal{L}_{\Pi}^{ro}(a)$ through the addition of the constraint $y_a \geq 1$. If $\mathcal{L}_{\Pi}^{ro}(a)$ has no solution, then Π has no solution where a occurs at least once, and a can thus be removed from the model.

We do not elaborate on this argument further, as it is a special case of the technique seen in Section 4.1. Indeed, it is equivalent to show that $n^+(a) = 0$, as it ensures that a does not occur in any solution-plan. However, this argument allows us to find removable operators that are not detected by a test proposed later in this subsection.

Unreachable preconditions A simple way to prove that some operator a will never be part of any plan, is to prove that no reachable state satisfies its precondition. This can be done by testing that the planning instance $\Pi_a^{\text{pre}} = \langle F, I, O, \text{pre}(a) \rangle$ is unsolvable.

Removing some operators relaxes the linear program \mathcal{L}_{Π}^{op} , by the deletion of some of the associated variables and constraints. As a consequence, it can help prove some instances unsolvable. We introduce below the notation for the associated test :

PreImp

If Π_a^{pre} is unsolvable
 then remove a from the set of operators O

Dead-end operators As it is possible to test whether or not there exists a reachable state where a can be applied, it is natural to ask the opposite : does a always lead to a dead-end, where no goal state can be reached ?

This paragraph is dedicated to finding such operators, called *dead-end operators*. In order to do so, we need to restrict ourselves to the few fluents that appear in all states resulting from the application of a , that is to say, the fluents that are true after a is applied either because of the effects of a , or by inertia. Indeed, these fluents are the only ones for which we have enough information about their truth value to reason about. Let $F_a = \text{fluents}(\text{pre}(a)) \cup \text{fluents}(\text{eff}(a))$. For any set S of literals of F , and $E \subseteq F$, we note $S|_E$ the projection of S over the fluents E . Likewise, we denote $a|_E = \langle \text{pre}(a)|_E, \text{eff}(a)|_E \rangle$ the projection of operator a over E . For any $O' \subseteq O$, we also note $O'|_E = \{a|_E \mid a \in O'\}$. This leads us to the following lemma, for which the proof is skipped due to space limitations :

Lemma 4. Let $\Pi = \langle F, I, O, G \rangle$ be a planning instance and $\Pi^{\text{post}} = \langle F_a, I_a^{\text{post}}, O|_{F_a}, G|_{F_a} \rangle$, where $I_a^{\text{post}} =$

$((\text{pre}^+(a) \setminus \text{eff}^-(a)) \cup \text{eff}^+(a)) \cap F_a$. If Π_a^{post} is unsolvable, then a is a dead-end operator in Π .

ActDLock

If Π_a^{post} is unsolvable
 then remove a from the set of operators O

4.3 Extended preconditions and goals

In this section, we propose various methods to find more precise preconditions for operators. More precisely, we try to add new fluents to operators' positive or negative preconditions. Suppose for instance that some fluent f can only be true if some other fluent f' is true. Then any operator a such that $f \in \text{pre}^+(a)$ can be extended by adding also f' to $\text{pre}^+(a)$. These more precise preconditions make the program richer and hence more likely to detect unsolvable instances. Similarly, the negative preconditions of operators can be extended, and by the same reasoning, so can the goal. In addition to that, we introduce negative goals : fluents that have to be false in any goal state.

In the rest of this section, we propose several ways to extend preconditions and goals.

Extending the goal The previous argument can also be applied to the goal, and help us add new fluents to the goal. Indeed, let $f \in F$, and $\Pi_{+f}^G = \langle F, I, O, G \cup \{f\} \rangle$. If Π_{+f}^G is unsolvable, then f can be added to the negative goals of Π . Indeed, no goal state s_G such that $s_G \models f$ is reachable : necessarily, in any goal state s_G , we have $s_G \models \neg f$. Conversely, let $\Pi_{-f}^G = \langle F, I, O, G \cup \{\neg f\} \rangle$. If Π_{-f}^G is unsolvable, then f can be safely added to the goals of Π without changing the set of solutions.

We define below the test that allows us to detect if a fluent can be added to the negative goals.

FNegGoal

If Π_{+f}^G is unsolvable
 then add f to the negative goals of Π

Taking negative goals into account The linear programs we presented earlier do not make use of the negative goals of the planning instance. Indeed, they usually do not appear in the STRIPS model, as they can be avoided by rewriting the instance during parsing time. However, the previous argument allows us to find such negative goals, and it would be costly to rewrite the whole instance to convert them into positive goals. As such, we show how to take these negative goals directly into account in our linear program.

The key elements have already been introduced in the proof of Lemma 1, where we defined for each $f \in F$ the value $\delta_f^+ = 1 - \mathbb{1}_I(f) - \mathbb{1}_{G^-}(f)$. δ_f^+ serves as an upper bound on the difference on the number of times f is established and the number of times it is destroyed, in any plan.

With a proof that is very similar to the one that leads to Equation 6 in the proof of Lemma 1, one can show that the following equation holds, for any fluent f :

$$\sum_{a \in SA_f} y_a - \sum_{a \in PD_f} y_a - \sum_{a \in SD_f} y_a \leq \delta_f^+ \quad (7)$$

Note that the above equation is symmetrically equivalent to Equation 4, found in the original Linear Program 2, that we recall below. In the initial formulation, the significant number of positive preconditions allows us to have non-empty sets of the form SD_f , thus adding negative variables in the left-hand side of the inequation. These negative variables penalize the whole sum, and make it harder to reach the threshold of δ_f^- given in the right-hand side. As our goal is to make the linear program unsatisfiable, the more positive preconditions we have, the better.

$$\sum_{a \in SA_f} y_a + \sum_{a \in PA_f} y_a - \sum_{a \in SD_f} y_a \geq \delta_f^-$$

The same case can be made for negative preconditions and Equation 7 : negative preconditions contribute to populating sets of the form SA_f , which in turn further constraint the inequation. In addition, note that having negative goals also contributes to making the inequation harder to satisfy, by lowering the bound δ_f^+ on the right-hand side. As negative goals only appear in variables δ_f^+ , without negative preconditions, there would be little interest in seeking to detect them. In addition, negative precondition do not affect the final expression of Equation 4, but only affect Equation 7. As such, negative goals and negative preconditions are closely intertwined.

4.4 Fluent mutexes and unreachable fluents

A fluent mutex is a set of fluents $M \subseteq F$ for which all states s accessible from the initial state I are such that $s \not\models M$. Some tests presented previously can be seen as testing whether some subset $M \subseteq F$ is a fluent mutex. Let us consider for instance the **PreImp** test presented in Section 4.2 : for some operator $a \in O$, checking that $\Pi_a^{\text{pre}} = \langle F, I, O, \text{pre}(a) \rangle$ is unsolvable (and thus that operator a can be removed from the instance) is equivalent to checking that $\text{pre}(a)$ is a mutex. However, our criterion allows us to check if any set of fluents $F' \subseteq F$ is a mutex, by testing the unsolvability of $\Pi_{F'}^{\text{mut}} = \langle F, I, O, F' \rangle$.

FMut

If $\Pi_{F'}^{\text{mut}}$ is unsolvable
 then F' is a fluent mutex

The criterion does not detect all fluent mutexes, and each candidate set of fluents has to be tested individually. Thus, not all fluent mutexes can be detected in reasonable time, as there exists an exponential number of candidates. Finding which sets are interesting to test is a problem in itself ; even more so since one has to know how to make use of the newly-found information that some $M \subseteq F$ is a mutex.

In the general case, we could not find a way to reinvest into the linear program the knowledge that a set of fluents is a mutex. Indeed, Linear Program 2 reasons over the number of times operators (have to) occur in a plan. As a consequence, we do not have any obvious way to reason about properties concerning states, which is precisely what fluent mutexes are. For that reason, we do not include in our routine a computation of mutexes through our linear program, even though we can detect a range of fluent mutexes.

However, some fluents are always false, in the sense that no plan will ever establish them. We call the fluents *unreachable fluents*, and they can be detected with the same argument as above :

FReach(f)
 If $\Pi_{\{f\}}^{mut}$ is unsolvable
 then f is an impossible fluent

Even though these fluents appear very rarely, as will be shown in the experimental trials, it remains linear to test for all fluents whether they are unreachable or not : thus, the computational burden is significantly lower than for other fluent “mutexes”. When an unreachable fluent is detected, one can project the whole instance on fluents $F \setminus \{f\}$. Theoretically, one could also remove operators that have f in their positive preconditions : however, any such operator a would also be detected by test $\text{PreImp}(a)$, which is more likely to succeed.

5 Experimental evaluation

Our implementation was done in Python 3.10, basing ourselves on the Fast Downward parser [8]. For linear programs, we resorted to the GLOP solver [11], while integer programs were solved with Gurobi [7]. We also used Google ORTools [11] to interface between our program and the solvers. We ran our experiments on a machine running Rocky Linux 8.5, powered by an Intel Xeon E5-2667 v3 processor, with a 30-minutes cutoff and using at most 16GB of memory per instance. Our code is available online⁴.

In addition to the evaluation of the linear program, we also implemented a procedure based on the observations of Section 4 . The main loop of this procedure consists in executing sequentially a predetermined list of operations and tests, until the instance is detected as unsolvable or the list is depleted. We elaborate further on this in Section 5.2.

We wished to evaluate our program on two different aspects : first, its ability to detect unsolvable instances, and second, its ability to find additional information when it could not conclude.

Our set of benchmarks consists of the unsolvable instances from the unplannability track of the International Planning Competition 2016 (Unsat IPC), which consists of unsolvable instances. The Unsat IPC also included solvable instances, which we tested our program on, as a sanity check, with success.

5.1 LP-based criteria

In this section, we show that our LP-based criterion suffices to detect a wide range of unsolvable planning instances. Our results are reported in detail in Table 1.

In essence, about 30% of all instances of the Unsat IPC are almost immediately found to be unsolvable by the sole use of the criterion. These results however vary greatly from one domain to the other, in a very dichotomous fashion : either the domain is (almost) entirely solved through the criterion, either few to no instances can be deemed unsol-

Set	Unsat	Total
bag-transport	19	29
bottleneck	25	25
cave-diving	1	25
chessboard-pebbling	23	23
over-tpp	2	30
pegsol-row5	14	15
tetris	20	20
Remaining	0	180
Total	104	347

TABLE 1 – Summary of the results returned by the LP-based criterion, run on the Unsat planning competition benchmark set. Each line corresponds to a domain : a set of instances modelling similar problems. The first column reports instances on which our criterion succeeds, while the second column reports the total number of instances in the benchmark set. Domains for which no instance could be solved are summed up in the last line labeled *Remaining*.

vable. In the case of domain bag-transport, which seems to be in-between, all instances the criterion has been tried on are actually found to be unsolvable : however, as the last 10 instances are too big to be parsed, we could not run the test on them. We can also note that both linear- and integer-programming-based criteria yield the same results, and that solving the IP-based program did not allow us to improve our results.

Both programs are however very lightweight : in every case, building and solving the program required less than a few seconds. In most cases, the criteria required little more than a few tenths of a second to complete. This further justifies our use of the program in the iterative procedure that we present in the next section.

Our program fails entirely on some domains, where no instance can be solved. While this is often because our criterion simply fails to detect the instance’s unsolvability, this can also be due to the size of the model. This is the case of bag-gripper, where the first instance has 5681 fluents and 60604 operators, which prevents us from building the associated linear program. In our assessments of the performances of the criteria, the limitation always came from memory.

5.2 Iterative refinement of the model

In the case where the criterion did not immediately detect that an instance Π is unsolvable, one can resort to the several operations previously introduced. In addition, the order in which operations are executed is also critical. Consider for instance an operator a that is both recognized as a landmark and as a removable operator by our operations. In the case where the operator is first removed, then it can not be detected as a landmark, and we thus missed an opportunity to return that the instance is unsolvable. In the case where a is first detected as a landmark, then our routine terminates successfully by detecting that the instance is unsolvable.

4. <https://github.com/arnaudlequen/MPRefinement>

5.2.1 Sequences of operations

We present below the different lists of operations that we chose. Note that all sequences start and end with a simple test of solvability with the criterion : initially with only the information contained in the STRIPS model, and then with all information that could be gathered after all operations.

Linear This sequence comprises all tests and operations that are linear in the size of the instance, i.e. that only require one argument. We tried to put first the tests that were the most likely to succeed, so that the followings tests and operations that come after have more information to work with. We successively apply the following tests on all relevant elements, in that order : LMDet, Prelmp, OpCount, FReach, and FNegGoal. By that, we mean that we run LMDet(a) for all $a \in O$, then Prelmp(a) for all $a \in O$, etc.

OperatorPreImpossible As will be reported later, the Prelmp tests that check an operator’s reachability are our most successful ones. We wished to gauge the time it requires and its possible impact on the model by itself.

OperatorDeadLocks Even though we choose this name to contrast with the OperatorPreImpossible sequence, this sequence tests both the reachability (through Prelmp) and co-reachability (through ActDLock) of an operator. In our trials, no operator could be shown to be a deadlock, even when we tested after the Linear sequence : as a consequence, we only include this sequence for the sake of completeness.

OperatorCount This sequence consists in finding lower, then upper bounds on the number of times each operator has to appear in any plan. It aims to show that a linear number of integer programs to optimize can be done in reasonable time, while also providing interesting information.

5.2.2 Results

We present our results below. As we prune out instances that can be immediately identified as unsolvable, domains that are immediately found unsolvable by the criterion are not reported.

Linear sequence Table 2 shows statistics for the Linear sequence. The main goal of our routine is to extract additional information from the model, so that another procedure that comes after can more easily show it unsolvable. However, we could notice that our algorithm was sometimes enough to detect unsolvable instances that are otherwise not detected as such by the criterion. There are few examples of such instances (about 9.5% of the entire benchmark set), and they are grouped in only two domains (cave-diving and pegsol). Nonetheless, they suffice to show that a well-chosen sequence of operations can sometimes replace a search, and that our work paves the way for further research in that regard.

In the cases where our procedure could not conclude, it still manages to gather valuable information about the planning instance. For example, on some domains, almost a third of all operators are pruned on average, among instances on which our procedure terminates.

The termination of our procedure is, however, the main issue of this sequence of operations, which is too computationally costly, and often stops early because of the time and memory limits imposed. In some domains, very few instances could be run through the entire sequence of operations : such domains include over-nomystery, where this sequence terminated on only 2 instances out of the 24 that could be parsed.

Individual tests Table 3 summarizes the statistics for the other sequences, that mostly consists of series of one or two of the same operations. However, it does not report comprehensive results for all remaining sequences : indeed, in the case of the OperatorDeadLock sequence, no test answered positively. Thus, no dead-end operator could be found.

Nonetheless, the results for the other sequences of operations are encouraging. Be it for the sequence centered on Prelmp or the one focused on OpCount operations, a significant proportion of operators could be removed. In some cases, it suffices to show that the instance was not solvable, as is the case for the cave-diving or pegsol domains. However, the time required for the computation is significant, which is discussed in the next section.

Note that these sequences of tests are not as powerful as the Linear sequence, when it comes to detecting unsolvable instances. This seems to indicate that the combination of different kinds of operations is crucial to draw conclusions, and studying their interactions is crucial in designing more powerful sequences.

6 Related work

The surge in interest for unsolvability detection, in the last decade, has been embodied by the first Unsolvability Planning Competition in 2016. The competition saw various adaptations of techniques that have shown themselves efficient for finding plans, in a state-space search. Such methods include heuristics specifically tailored for unsolvability detection, such as a Merge & Shrink-based heuristic [9] (which precedes the competition). Such heuristics rely on abstractions that do not preserve distance, but merely solvability.

Another heuristic that was successfully adapted was the operator-counting heuristic [2, 12, 18]. The heuristic is based on a relaxation of the orderings of the operators. Previous works showed that it admits a linear programming formulation, similar to the Linear Program 1 that we propose. However, while we only optimize the variable associated to the count of a single operator, the objective function that they minimize is the total cost of the plan. The adaptation of the linear program to the case of unsolvability detection, was carried out by the Fast Downward-based unsolvability planner Aidos [13]. It consists in checking the existence of a solution, in the same way as for Linear Program 2. However, Aidos uses this component in a state-space search, in order to detect dead-ends.

More generally, be it in unsolvable or in solvable planning tasks, the early detection of states that can not lead to a goal makes can help prune out whole branches of the search

Set	Diff.	Operators			Others		
		PreImp	OpCount	Removed	LMDet	FReach	FNegGoal
cave-diving (14)	+9	10.0%	14.1%	10.4%	1.1%	4.8%	3.0%
diagnosis (19)	0	0%	57.0%	11.6%	18.3%	4.6%	17.6%
doc-transfer (5)	0	13.0%	26.4%	27.9%	1.7%	0.0%	39.8%
over-nomystery (2)	0	33.4%	25.7%	34.8%	2.1%	0%	7.4%
over-rovers (8)	0	27.9%	17.2%	29.3%	0%	<0.1%	0%
over-tpp (8)	0	7.4%	54.8%	24.7%	0.3%	0.3%	0%
pegsol (24)	+24	13.6%	N/A%	13.6%	0.8%	N/A%	N/A%
sliding-tiles (20)	0	0%	0%	0%	0%	0%	69.2%

TABLE 2 – Statistics for the Linear sequence. The first column with the name of the domain also reports the total number of instances for which the procedure terminated entirely within the time and memory limits. The “Difference” (Diff.) column shows the number of instances that could be found unsolvable during the execution of the procedure, compared to the single use of the criterion reported in Table 1. The next set of columns shows stats for operations related to the deletion of operators. The first pair of columns show the percentage of success of each test, while the last column of the set shows the average total percentage of operators pruned at the end of the sequence of tests. The last three columns show the percentage of success of three other tests. N/A values indicate that no such test was performed as the program terminated before.

Set	OperatorPreImpossible				OperatorCount					
	Cpt	Rem.	Diff.	Time	Cpt	OpCount ⁻	OpCount ⁺	Rem.	Diff.	Time
bag-barman	4	77.2%	0	1177.8	0
cave-diving	17	6.5%	+4	147.8	17	0.9%	28.0%	7.0%	0	329.3
diagnosis	20	0%	0	6.4	20	16.9%	96.3%	19.5%	0	91.6
document-transfer	13	0%	0	475.7	8	1.7%	50.7%	29.8%	0	643.2
over-nomystery	10	18.8%	0	587.8	3	1.4%	87.2%	3.9%	0	746.2
over-rovers	11	21.9%	0	370.2	9	0%	62.4%	5.2%	0	455.1
over-tpp	14	<0.1%	0	268.1	9	0.3%	65.4%	20.2%	0	428.8
pegsol	24	16.4%	+6	0.6	24	0%	8.2%	3.0%	+22	0.51
sliding-tiles	20	0%	0	5.6	20	0%	0%	0%	0	19.4

TABLE 3 – Performances of the individually run operations. The *Completed* (Cpt) columns show the number of instances the sequence terminated on, the *Diff.* columns show the number of instances solved thanks to the iterative refinement, and the *Time* columns show, in seconds, the average time per instance. The *Rem.* columns show the average percentage of operators that could be removed thanks to the operation. OpCount⁺ and OpCount⁻ columns report the average percentage of success of their respective operations.

space. In the case of dead-end detection [4], various works have focused on the elaboration of formulas that can be efficiently evaluated, and whose only models are states that can not lead to a goal state. The notion of dead-end formula has been generalized with the notion of traps [10]: a formula ϕ such that, once it’s verified in a state s , all states reachable from s will satisfy it too. A formula ϕ that is inconsistent with the goal then shows that the current branch is not worth exploring.

In the case where our algorithm does not manage to find that the task is unsolvable, it still manages to remove unnecessary elements from the planning model, to make the task easier for the next algorithm. Various other methods prune the model in a preprocessing step: in [1], the authors show that invariants in the form of mutexes can be leveraged to remove operators that will never be part of a plan. In [5], it is shown how to combine symmetries of the planning task and operator mutexes to find operators that are redundant, in the sense that removing them preserves at least one solution-plan.

Our algorithm also learns information that is not explicitly expressible in a STRIPS planning instance. In [16], the authors draw inspiration from a well-known technique in SAT solving, to learn clauses that recognize dead-ends, through a conflict-driven approach during search. They also show how to learn traps online [15]. Learning is ubiquitous in generalized planning, which is a domain concerned with the synthesis of generalized plans, which are procedures that solve multiple instances. For instance, previous work [17] proposed to learn heuristics in the form of logical formulas, out of a set of small examples instances, so as to recognize unsolvable planning instances.

In [3], another polynomial criterion is proposed to immediately detect a class of unsolvable instances without resorting to search. The authors synthesize a function that separates the initial state from all goal states, through a linear combination of features valued in a finite field. Akin to our criterion, their technique is incomplete, but it is very efficient at detecting parity arguments.

7 Discussion and conclusion

Section 5 showed that, when our criterion failed to show an instance unsolvable, it was still possible to extract additional information from the model by leveraging the criterion. Even more so, in some cases, otherwise undetected unsolvable instances could be identified as such by this means. Yet, there is still a lot of room for improvement : a more in-depth study of our operations, as well as their interactions, could help us fine-tune the algorithm. Indeed, not all sequences of tests are equal in all aspects, and finding a sequence that avoid unnecessary computations is a way to optimize our algorithm, and to boost its detection power.

In our tests, we choose to simply run pre-determined sequences of operations and tests. This means that, regardless of how tests succeed or fail, the algorithm will linearly go through the same sequence of operations, except if it can show preemptively that an instance is unsolvable. However, the outcome of some test may help in finding which step to take next. For instance, after finding that an operator is a landmark, it might be interesting to check right away if it can be removed.

One of the main weaknesses of our iterative refinement algorithm is its computational cost. Even the most lightweight sequences, such as the OperatorPreImpossible sequence, takes significant time to complete. Our program builds each linear program from scratch each time a test is performed. However, very few constraints differ from one linear program to the other ; thus, one could modify only these constraints from one test to the next, in order to save significant time.

As a conclusion, we showed that a simple criterion was sometimes enough to prove that a planning instance is unsolvable. Even though our program is non-optimised, we have still managed to show that resorting to a search is not always necessary, as reasoning on the model directly can suffice. Even when our procedure fails, it still gathers valuable information about the instance, that can help a complete procedure terminate faster.

Références

- [1] Vidal Alcázar and Alvaro Torralba. A reminder about the importance of computing and exploiting invariants in planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, pages 2–6, 2015.
- [2] Blai Bonet. An admissible heuristic for SAS+ planning obtained from the state equation. In *IJCAI*, 2013.
- [3] Remo Christen, Salomé Eriksson, Florian Pommerening, and Malte Helmert. Detecting unsolvability based on separating functions. In Akshat Kumar, Sylvie Thiébaux, Pradeep Varakantham, and William Yeoh, editors, *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling, ICAPS 2022*, pages 44–52, 2022.
- [4] Bence Cserna, William Doyle, Jordan Ramsdell, and Wheeler Ruml. Avoiding dead ends in real-time heuristic search. In *AAAI*, 2018.
- [5] Daniel Fišer, Alvaro Torralba, and Alexander Shleyfman. Operator mutexes and symmetries for simplifying planning tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7586–7593, 2019.
- [6] Hector Geffner and Blai Bonet. *A concise introduction to models and methods for automated planning*. Morgan & Claypool Publishers, 2013.
- [7] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.
- [8] Malte Helmert. The fast downward planning system. *JAIR*, 26 :191–246, 2006.
- [9] Malte Helmert, Patrik Haslum, Jörg Hoffmann, and Raz Nissim. Merge-and-shrink abstraction : A method for generating lower bounds in factored state spaces. *J. ACM*, 61(3) :16 :1–16 :63, 2014.
- [10] Nir Lipovetzky, Christian Muise, and Hector Geffner. Traps, invariants, and dead-ends. In *ICAPS*, pages 211–215, 2016.
- [11] Laurent Perron and Vincent Furnon. Or-tools.
- [12] Florian Pommerening, Gabriele Röger, Malte Helmert, and Blai Bonet. LP-based heuristics for cost-optimal planning. In *ICAPS*, pages 226–234, 2014.
- [13] Jendrik Seipp, Florian Pommerening, Silvan Sievers, Martin Wehrle, Chris Fawcett, and Yusra Alkhazraji. Fast Downward Aidos. *Unsolvability International Planning Competition : planner abstracts*, pages 28–38, 2016.
- [14] Marcel Steinmetz, Jörg Hoffmann, Alisa Kovtunova, and Stefan Borgwardt. Classical planning with avoid conditions. In *AAAI*, pages 9944–9952, 2022.
- [15] Marcel Steinmetz and Jörg Hoffmann. Search and learn : On dead-end detectors, the traps they set, and trap learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4398–4404, 2017.
- [16] Marcel Steinmetz and Jörg Hoffmann. State space search nogood learning : Online refinement of critical-path dead-end detectors in planning. *Artificial Intelligence*, 245 :1–37, 2017.
- [17] Simon Ståhlberg, Guillem Francès, and Jendrik Seipp. Learning generalized unsolvability heuristics for classical planning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4175–4181, 2021.
- [18] Menkes Van Den Briel, J Benton, Subbarao Kambhampati, and Thomas Vossen. An LP-based heuristic for optimal planning. In *Principles and Practice of Constraint Programming–CP 2007 : 13th International Conference*, pages 651–665. Springer, 2007.

Classes of Explanations for the Verification Problem in Abstract Argumentation

S. Doutre¹, T. Duchatelle², M.-C. Lagasquie-Schiex²

¹ IRIT, Université Toulouse Capitole, France

² IRIT, Université Paul Sabatier, France

Sylvie.Doutre@irit.fr, Theo.Duchatelle@irit.fr, Marie-Christine.Lagasquie@irit.fr

Résumé

Le problème de vérification en argumentation abstraite consiste à déterminer si un ensemble est acceptable sous une sémantique donnée dans un graphe d'argumentation donné. Cet article s'attache à expliquer la réponse retournée. Des explications visuelles en termes de sous-graphes du cadre d'argumentation initial sont définies. Ces explications sont regroupées en classes, ce qui permet de sélectionner l'explication qui convient le mieux dans un contexte donné parmi l'ensemble des possibilités offertes. Des résultats montrent comment utiliser les aspects visuels de ces explications pour soutenir l'acceptabilité d'un ensemble d'arguments sous une sémantique. Les aspects computationnels d'explications spécifiques sont également étudiés.

Mots-clés

Systèmes multi-agents, XAI, explications visuelles, argumentation formelle.

Abstract

The Verification Problem in abstract argumentation consists in checking whether a set is acceptable under a given semantics in a given argumentation graph. Explaining why the answer is so is the challenge tackled by this paper. Visual explanations in the form of subgraphs of the initial argumentation framework are defined. These explanations are grouped into classes, allowing one to select the explanation that suits them best among the several offered possibilities. Results are provided on how to use the visual aspects of these explanations to support the acceptability of a set of arguments under a semantics. Computational aspects of specific explanations are also investigated.

Keywords

Multi-Agents Systems, XAI, Visual Explanations, Formal Argumentation.

1 Introduction

Abstract Argumentation is increasingly studied as a formal tool to provide explanations in the context of eXplainable Artificial Intelligence (XAI). The term argumentative XAI has emerged, with a number of application domains, ranging from machine learning, to decision, medicine or secu-

rity (see [19] for an overview). [7] presents the current approaches of argumentative XAI and their open challenges, and underlines that explanations for the argumentative process itself are necessary too.

The basic argumentation process relies on an abstract structure which takes the form of a directed graph, whose nodes are arguments and edges represent attacks between arguments [10]. Characterising the acceptability of arguments can take the form of extension-based semantics : they define sets (extensions) of arguments which are collectively acceptable according to the semantics. The main questions which have been addressed so far in this context concern the global acceptability status of an argument or of a set of arguments, that is, why, under a given semantics, they belong to at least one extension (credulous acceptance) or to every extension (skeptical acceptance). The most common explanation approach consists in identifying set(s) of arguments which act as explanation(s), as in [12, 4, 5, 18, 13, 1]. However, since the argumentative process of Abstract Argumentation already provides ways for selecting arguments, explaining this process by more selection of arguments (although different ones) may not be fully helpful. Moreover, this set approach does not highlight the attacks which are involved in the explanations.

Another question regarding the argumentation process concerns the *Verification Problem* Ver, defined as follows : given an Argumentation Framework \mathcal{A} , a set of arguments S and an extension-based semantics σ , “Is S an extension under σ in \mathcal{A} ?”. The answer to this problem is “yes” or “no”. In order to explain why the answer is so, the *eXplanation Verification Problem* XVer can be defined using the question Q_σ : “Why is S (not) an extension under σ in \mathcal{A} ?”. [2] is one of the only approaches which has addressed this problem and which has provided answers for some acceptability semantics of [10] in the form of relevant subgraphs, as in [17, 15, 16]. Such a visual approach is particularly of interest for human agents, graphs having been shown to be helpful for humans to comply with argumentation reasoning principles [20]. This graph-based approach not only highlights arguments, but also attacks. In [2], properties that these answers satisfy have been established, depending on whether the answer to the corresponding verification problem is “yes” or “no”. This methodology follows the line

of [6] in that an explanation for a set S satisfying a semantic σ is a (set of) subgraph(s) G of \mathcal{A} such that G satisfies a given graph property C . Another interesting point in [2] is that the considered semantics are based on a modular definition, which allows the explanations to be decomposed.

A limitation of [2] is however that, for each semantic principle, a *single* explanation subgraph is defined. It could be more realistic to consider classes (sets) of explanations. Indeed such classes would be particularly meaningful and useful when several agents, human or artificial, are involved around the explanation of a same problem, in that they offer a variety of answers, which all follow a same schema, but which may differ on their exact content. Any agent can choose or can be presented an explanation that suits them best, and any agent can understand an explanation given by another agent, different from theirs. Classes of explanations adapt to a wide set of agents.

As in [2], the approach that will be presented in this paper goes further, by considering the possibility that the answer to the Verification Problem is not known before an explanation be asked and given. In this case, the explanation graph and its interpretation offer at a same time the answer to the problem and a justification to this answer.

Only few related works can be found concerning this notion of classes of explanation. Such classes have already been proposed in [1] for the problem of credulous acceptance of an argument, where the authors consider explanation schemes made of several elements, one of them being fixed, the other ones varying from one explanation to another. Another related work is [4] in which the authors define a parametric computation of explanations. As such, it is more the computation processes that are grouped in classes, rather than the explanations (i.e. results of the processes) themselves.

Thus, our aim in the current paper is to define classes of explanations following a generic methodology, applied to classical semantics (conflict-free, admissible, stable, complete), by building up on the approach of [2]. Additional properties (emptiness, uniqueness, maximality, minimality, computation) of explanations on these new classes will be defined and investigated.

Sec. 2 recalls background notions relative to abstract argumentation, graph theory, and presents the explanation approach defined in [2]. Classes of explanations are defined in Sec. 3, Sec. 4 studies their properties; Sec. 5 shows how to compute their maximal and minimal explanations and illustrates the whole approach on an example. Sec. 6 concludes and presents some future works. Proofs of all the results can be found in [8]

2 Background notions

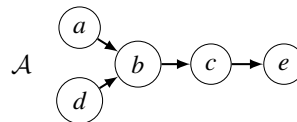
2.1 Argumentation and Graph Theory

We begin by recalling basic notions on Abstract Argumentation.

Definition 1 ([10]) A Dung’s argumentation framework (AF) is an ordered pair (A, R) such that $R \subseteq A \times A$.

Each element $a \in A$ is called an *argument* and aRb means that a attacks b . For $S \subseteq A$, we say that S attacks $a \in A$ iff bRa for some $b \in S$. Any argumentation framework can be represented as a directed graph (the nodes are the arguments and the edges correspond to the attack relation).

Example 1 Let consider $\mathcal{A} = (A = \{a, b, c, d, e\}, R = \{(a, b), (d, b), (b, c), (c, e)\})$. \mathcal{A} is depicted by the following figure :



The main asset of Dung’s approach is the definition of semantics using some basic properties in order to define sets of acceptable arguments, as follows.

Definition 2 ([10]) Let $\mathcal{A} = (A, R)$. An argument $a \in A$ is acceptable wrt $S \subseteq A$ iff for all $b \in A$, if bRa then $\exists c \in S$ st cRb .

Definition 3 ([10]) Given $\mathcal{A} = (A, R)$, a subset S of A is :

- a conflict-free set iff there are no a and b in S such that a attacks b ,
- an admissible set iff S is conflict-free and for any $a \in S$, a is acceptable wrt S ,
- a complete extension iff S is admissible and for any $a \in A$, if a is acceptable wrt S then $a \in S$,
- a stable extension iff S is conflict-free and S attacks any $a \in A \setminus S$.

Example 2 Let consider again \mathcal{A} given in Ex. 1. Here there is a unique complete and stable extension : $\{a, d, c\}$ whereas there are 6 admissible sets : $\{\}, \{a\}, \{d\}, \{a, c\}, \{d, c\}, \{a, d, c\}$.

The Verification Problem for the four semantics given in Def. 3 can be solved in polynomial time, as indicated by [11].

Example 3 Considering \mathcal{A} given in Ex. 1, an instance of the Verification problem could be : “Is $\{\}$ a stable extension?”; in this case the answer will “no”. Another instance would be : “Is $\{a, d, c\}$ a complete extension?”; in this case the answer will “yes”.

Since an AF can be represented using directed graphs, we also need to recall some basic notions of Graph Theory.

Definition 4 Let $G = (V, E)$ and $G' = (V', E')$ be two graphs.

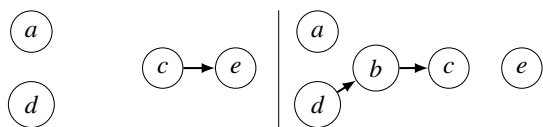
- G' is a subgraph of G iff $V' \subseteq V$ and $E' \subseteq E$.¹
- G' is a strict subgraph of G iff it is a subgraph of G and either $V' \subset V$ or $E' \subset E$.²

1. G is then a supergraph of G'
 2. G is then a strict supergraph of G'

- G' is an induced subgraph of G by V' if G' is a subgraph of G and for all $a, b \in V'$, $(a, b) \in E'$ iff $(a, b) \in E$. G' is denoted as $G[V']_V$.
- G' is a spanning subgraph of G by E' if G' is a subgraph of G and $V' = V$. G' is denoted as $G[E']_E$.

A subgraph G' of G is included in G . In an induced subgraph G' of G by a set of vertices S , some vertices of G can be missing but all the edges concerning the kept vertices are present. In a spanning subgraph G' of G by a set of edges S , all the vertices of G are present but some edges of G can be missing.

Example 4 Let consider \mathcal{A} given in Ex. 1. An example of an induced (resp. spanning) subgraph of \mathcal{A} is given in the left (resp. right) following figure :



Induced and spanning subgraphs are examples of ways to compute a graph from another single graph. Another operation producing a new graph from other ones is the union that represents the aggregation of the information contained in the two graphs :

Definition 5 (Graph union) Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The union of G_1 and G_2 is defined by $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.

Let us consider also a particular kind of graphs, bipartite graphs, whose set of vertices can be split in two disjoint sets and in which every arc connects a vertex of one part to a vertex of the other part :

Definition 6 (Bipartite Graph) Let $G = (V, E)$ be a graph. G is bipartite (with parts T and U) iff there exist $T, U \subseteq V$ such that $T \cup U = V$ and $T \cap U = \emptyset$ (T and U are a partition of V) and for every $(a, b) \in E$, either $a \in T$ and $b \in U$, or $a \in U$ and $b \in T$. G will be denoted with (T, U, E) and U is the complement part of T (and vice-versa).

Some important functions can be defined over graphs.

Definition 7 (Successor and Predecessor functions)

Let $G = (V, E)$ be a graph. The successor function of G is the function $E^+ : V \mapsto 2^V$ such that $E^+(v) = \{u \mid (v, u) \in E\}$ and the predecessor function of G is the function $E^- : V \mapsto 2^V$ such that $E^-(v) = \{u \mid (u, v) \in E\}$. Let S be a set of vertices, $E^+(S) = \bigcup_{v \in S} E^+(v)$ and $E^-(S) = \bigcup_{v \in S} E^-(v)$.

Let $n \geq 0$. The n -step successor (resp. predecessor) func-

tion of G is $E^{+n}(v) = \overbrace{E^+ \circ \dots \circ E^+}^{n \text{ times}}(v)$ (resp. $E^{-n}(v) = \overbrace{E^- \circ \dots \circ E^-}^{n \text{ times}}(v)$). By convention, we have $E^{+0}(v) = E^{-0}(v) = \{v\}$.³

3. Note that $E^{+1}(v) = E^+(v)$ and $E^{-1}(v) = E^-(v)$

Considering an argumentation framework, the successor (resp. predecessor) function represents the arguments that are attacked by (resp. are the attackers of) some argument(s). An AF being usually denoted by (A, R) , the successor and predecessor functions are thus denoted R^+ and R^- in this context.

We then recall some notions on vertices having a particular status in a graph.

Definition 8 (Source, Sink, Isolated vertex) Let $G = (V, E)$ be a graph and v be a vertex of G . v is said to be a source iff $E^-(v) = \emptyset$ and it is said to be a sink iff $E^+(v) = \emptyset$. v is said to be isolated iff it is both a source and a sink.

Thus, sources (resp. sinks) are vertices that may only be origins (resp. endpoints) of arcs. Isolated vertices are those that are connected to no other vertices.

Example 5 Let consider \mathcal{A} given in Ex. 1. Argument a is a 3-step predecessor of e , whereas c is a predecessor of e (and obviously e is a 3-step successor of a , whereas e is a successor of c). Moreover, a and d are the sources of \mathcal{A} and e is the sink of \mathcal{A} .

2.2 Explanations in Argumentation

We recall the main definitions of what explanations are in [2] but only for those answering the questions about semantics results in abstract argumentation. These questions are defined as follows : let σ represent a semantics among conflict-freeness, admissibility, completeness and stability, and given an argumentation framework $\mathcal{A} = (A, R)$ and some set $S \subseteq A$,

Q_σ : Why is S (not) an extension under σ in \mathcal{A} ?

In order to answer these questions, and hence to provide explanations, [2] uses the decomposition of semantics into principles. The idea is to identify some properties that can be used to provide a modular characterization of semantics. We refer the reader to [9] for further details. Given a set S , the following principles are considered :

Conflict-freeness (CF) : No internal conflicts in S

Defence (Def) : $\forall x \in S, x$ is acceptable wrt S

Reinstatement (Re) : $\forall x$ acceptable wrt $S, x \in S$

Complement Attack (CA) : S attacks all arguments not in S

Note that the reinstatement principle has been split into two sub-principles. Indeed, to decide whether a set S of arguments contains all the arguments acceptable wrt S , one must consider on the one hand the arguments that are unattacked and thus acceptable by lack of attackers (sub-principle denoted by Re_1), and on the other hand the arguments for which S defeats all the attackers (sub-principle denoted by Re_2).

The following has been proven in [9].

Proposition 1 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. S is :

- Conflict-free* iff S respects $\{CF\}$
- Admissible* iff S respects $\{CF, Def\}$
- Complete* iff S respects $\{CF, Def, Re_1, Re_2\}$
- Stable* iff S respects $\{CF, CA\}$

With this result, a straightforward answer arises for Q_σ : a set S is an extension under semantics σ because it respects all the principles listed for σ in Prop. 1. This moves the burden of explanation from semantics to principles. So, in order to answer Q_σ , we are going to answer intermediate questions on principles. Let $\pi \in \{CF, Def, Re_1, Re_2, CA\}$ represent a principle. Given an argumentation framework $\mathcal{A} = (A, R)$ and some set $S \subseteq A$, the questions we will define answers for are :

Q_π : Why does (not) S respect principle π ?

[2] defines visual answers to these questions. These answers take the form of a graph. This allows for the answers to be drawn, as well as to study their visual (i.e. structural) properties. More precisely, as argumentation frameworks are graphs themselves, the answers given are subgraphs of an argumentation framework.

Definition 9 ([2]) Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. $G_\pi(S)$ is defined as :

$$\begin{aligned} G_{CF}(S) &= \mathcal{A}[S]_V \\ G_{Def}(S) &= (\mathcal{A}[S \cup R^{-1}(S)]_V) \\ &\quad [\{(a, b) \in R \mid (a \in R^{-1}(S) \text{ and } b \in S) \\ &\quad \text{or } (a \in S \text{ and } b \in R^{-1}(S))\}]_E \\ G_{Re_1}(S) &= \mathcal{A}[\{a \in A \mid R^-(a) = \emptyset\}]_V \\ G_{Re_2}(S) &= (\mathcal{A}[S \cup R^2(S) \cup R^{-1}(R^2(S))]_V) \\ &\quad [\{(a, b) \in R \mid (a \in R^{-1}(R^2(S)), b \in R^2(S)) \\ &\quad \text{or } (a \in S, b \in R^{-1}(R^2(S))\}]_E \\ G_{CA}(S) &= \mathcal{A}[\{(a, b) \in R \mid a \in S \text{ and } b \notin S\}]_E \end{aligned}$$

Moreover the interpretation of these subgraphs can be done using a ‘‘checking procedure’’ in order to explicitly identify if the given subset satisfies or not the concerned principle :

Definition 10 ([2]) Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. Let G be a subgraph of \mathcal{A} . The checking procedure $C_\pi(G)$ is defined as :

$$\begin{aligned} C_{CF}(G) &= \text{no attacks in } G \\ C_{Def}(G) &= \text{no source vertices in } R^{-1}(S) \text{ in } G \\ C_{Re_1}(G) &= \text{all vertices in } G \text{ are in } S \\ C_{Re_2}(G) &= \text{all vertices in } R^2(S) \setminus S \text{ are endpoint of an} \\ &\quad \text{arc whose origin is a source vertex in } G \\ C'_{Re_2}(G) &= \text{all vertices in } R^2(S) \setminus S \text{ are endpoint of an} \\ &\quad \text{arc whose origin is a source vertex or is in} \\ &\quad \text{in } R^2(S), \text{ in } G \\ C_{CA}(G) &= \text{no isolated vertices in the complement part} \\ &\quad \text{of } S \text{ in } G \end{aligned}$$

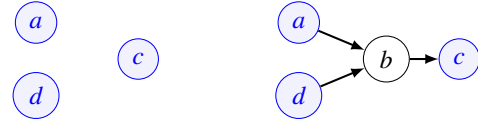
For each principle π , [2] has proven that the subgraph G_π associated with the corresponding checking procedure C_π provides an explanation that answers question Q_π .⁴ More

4. This result is slightly more complex in the case of reinstatement. See [2] and Sec. 3.3.

precisely, if a set S respects a principle π , then G_π verifies C_π , otherwise it does not. When the principles are combined into a semantics σ , the answer to Q_σ is the corresponding set of subgraphs along with their corresponding checking procedures.

Example 6 Let consider \mathcal{A} given in Ex. 1 and $S = \{a, d, c\}$. The question we are interested in is : ‘‘Why is S an extension under admissibility in \mathcal{A} ?’’. This question comes down to wondering : ‘‘Why S satisfies conflict-freeness CF and defense Def ?’’. So, an explanation of why S is admissible is a set which contains the explanation for CF and the explanation for Def .

The $G_{CF}(S)$ and $G_{Def}(S)$ explanations are given in the following figure :



There is no attack in G_{CF} , hence C_{CF} is satisfied. And so we can conclude that S is conflict-free.

Concerning G_{Def} , note that neither e nor (c, e) belong to this explanation since they have no impact on the defence of S . Then applying C_{Def} on G_{Def} , we can see that each attacker of S (here only b) is not a source vertex ; so S also satisfies the defence principle.

This allows this form of explanation to be used for two purposes as indicated in the introduction : when the answer to the corresponding verification problem is known, that is, when we know that a set is (resp. is not) acceptable under a given semantics or principle, G_π on which C_π is (resp. is not) verified, offers a visual explanation of the situation, answering XVer. When the answer to the verification problem is not known, G_π and the verification of whether C_π holds or not offers at the same time an answer to Ver and an explanation of this answer.

3 Classes of explanations

In this paper, we are interested in refining the notion of explanation proposed in [2] and recalled in Sec. 2.2. Indeed, considering Ex. 6 leads to the following remark : for explaining the respect of the defence principle it seems useless to consider the two defenders of c in G_{Def} (only one is enough for proving that c is defended). So, in order to propose a more flexible notion of explanation, another approach based on the notion of *classes of explanations* is presented in this section. Of course the definition of these classes allows to recover the explanations described in [2] but also it results in the *possibility of producing several explanations for the same question*.

Hence, for each principle π , we define our explanations so that they contain at least enough information to be able to decide whether or not S respects π . We then prove that our explanations can be used in conjunction with the checking procedures recalled in Def. 10.

3.1 Explanation about Conflict-freeness

To decide whether a set S of arguments is conflict-free, one must know whether or not there are attacks among its arguments. Thus, we firstly require our explanation to contain only arguments of S , and secondly to contain only attacks between these arguments. However, with only these two constraints, it may happen that no attacks are displayed on the explanation when there are some in the original framework, leading at best to an impossibility to decide or at worst, an incorrect decision. Hence, we add a third constraint, which is that if conflicts exist between arguments of S , then at least one must be present in the explanation.

Definition 11 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{(a, b) \in R \mid a, b \in S\}$. The subgraph (A', R') of \mathcal{A} is an explanation to Q_{CF} iff

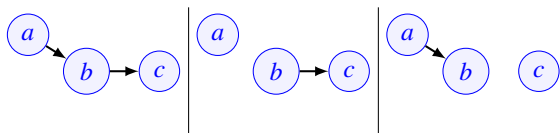
- $A' = S$
- $R' \subseteq X$
- If $X \neq \emptyset$, then $R' \neq \emptyset$

Note that the subgraph G_{CF} recalled in Def. 9 obviously belongs to the class of explanations for conflict-freeness. Moreover, in [2], a result concerning the structural property of explanations for conflict-freeness has been given : a set of arguments is conflict-free iff there is no attack in the subgraph corresponding to its explanation (checking procedure C_{CF} recalled in Def. 10). This result can be extended to all the subgraphs captured by our class of explanations.

Theorem 1 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{CF} . S is conflict-free iff $C_{CF}(A', R')$ is satisfied by S .

This provides a way of deciding whether a set is conflict-free based on an explanation to Q_{CF} . Note that this also provides a way of deciding whether a set is *not* conflict-free, hence the possibility of handling the negative version of Q_{CF} . The same goes for all the other equivalence results concerning the other principles.

Example 7 Let consider \mathcal{A} given in Ex. 1 and $S = \{a, b, c\}$. There are 3 explanations for Q_{CF} , each of them proving that S is not conflict-free :



3.2 Explanation about Defence

To decide whether a set S of arguments contains only arguments that are acceptable wrt S , one must know whether or not this set defeats all its attackers. Thus, we firstly require our explanation to contain only arguments of S and its attackers, and secondly to contain only attacks from S to its attackers and vice versa. To make sure the attackers are spotted as such, we further require that all the attacks

of the second type are contained in the explanation. However, with only these two constraints, it may happen that no attacks targeting a specific attacker are displayed on the explanation when there are some in the original framework. As we wish the explanation to show how S defends itself, this situation is certainly undesirable. Hence, we add a third constraint, which is that if an attacker is attacked by S , then at least one attack from S to this attacker must be present in the explanation.

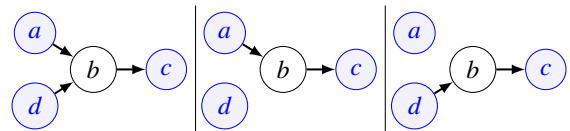
Definition 12 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. Consider $X = \{(b, a) \in R \mid b \in R^{-1}(S), a \in S\}$ and $Y = \{(a, b) \in R \mid a \in S, b \in R^{-1}(S)\}$. The subgraph (A', R') of \mathcal{A} is an explanation to Q_{Def} iff

- $A' = S \cup R^{-1}(S)$
- $X \subseteq R' \subseteq X \cup Y$
- $\forall b \in R^{-1}(S)$, if $b \in R^+(S)$, then $\exists (a, b) \in R'$ with $a \in S$

Note that the subgraph G_{Def} recalled in Def. 9 obviously belongs to the class of explanations for defence. Moreover it has been shown in [2] that a conflict-free set of arguments defends all its arguments iff there is no source vertex among its attackers in $G_{Def}(S)$ (checking procedure C_{Def} recalled in Def. 10). This result can be extended to all the subgraphs captured by our class of explanations.

Theorem 2 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ be a conflict-free set of arguments and (A', R') be an explanation to Q_{Def} . $S \subseteq F_{\mathcal{A}}(S)$ iff $C_{Def}(A', R')$ is satisfied by S .

Example 8 Let consider \mathcal{A} given in Ex. 1 and $S = \{a, c, d\}$. There are 3 explanations for proving that S satisfies the defence principle :



Additionally, the next result extends a similar result given in [2] providing more insight on the behavior of an explanation for defence : when computed using a conflict-free set, the explanation for defence takes the form of a bipartite graph.

Proposition 2 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{Def} . If S is conflict-free, (A', R') is a bipartite graph and S can always be one of its parts.

The two previous results can thus be used to decide whether a set of arguments effectively defends all its arguments or if it is not conflict-free.

3.3 Explanation about Reinstatement

The first part of the reinstatement principle concerns unattacked arguments. All these arguments are acceptable wrt S and should thus belong to S . Thus, we firstly require our

explanation to contain only unattacked arguments, and secondly to contain no attacks (which results from the only arguments displayed being unattacked). However, with only these two constraints, it may happen that an unattacked argument not belonging to S is not displayed on the explanation. Hence, we add a third constraint, which is that if there exists unattacked arguments that are not in S , then at least one must be present in the explanation.

Definition 13 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{a \in A \mid R^{-1}(a) = \emptyset\}$. The subgraph (A', R') of \mathcal{A} is an explanation to Q_{Re_1} iff

- $S \cap X \subseteq A' \subseteq X$
- $R' = \emptyset$
- If $(A \setminus S) \cap X \neq \emptyset$, then $\exists a \in (A \setminus S) \cap X$ with $a \in A'$

The second part concerns arguments for which S defeats the attackers. These arguments must belong to S if S defeats all of their attackers. Thus, we firstly require our explanation to contain the arguments of S , the arguments that S defends (two steps of the attack relation from S), and the attackers of these arguments. Secondly, we require it contains only the attacks from S to the attackers and from the attackers to the arguments S defends. In addition, we require that all the attacks of the second type are displayed on the explanation, so that none is missed. However, with only these two constraints, it may happen that no attacks targeting a specific attacker are displayed on the explanation when there are some in the original framework. Hence, we add a third constraint, which is that if an attacker is attacked by S , then at least one attack from S to this attacker must be present in the explanation.

Definition 14 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. Consider $X = \{(b, c) \in R \mid b \in R^{-1}(R^{+2}(S)), c \in R^{+2}(S)\}$ and $Y = \{(a, b) \in R \mid a \in S, b \in R^{-1}(R^{+2}(S))\}$. The subgraph (A', R') of \mathcal{A} is an explanation to Q_{Re_2} iff

- $A' = S \cup R^{+2}(S) \cup R^{-1}(R^{+2}(S))$
- $X \subseteq R' \subseteq X \cup Y$
- For every $b \in R^{-1}(R^{+2}(S))$, if $b \in R^{+1}(S)$, then $\exists (a, b) \in R'$ with $a \in S$

Note that the subgraph G_{Re_1} (resp. G_{Re_2}) recalled in Def. 9 obviously belongs to the class of explanations for the first (resp. second) part of the principle of reinstatement. Moreover in the case of reinstatement, two results have been proven in [2] and can be extended to all the subgraphs captured by our class of explanations.

The first one shows how to conclude that a set contains all the arguments that it effectively defends from both parts of the explanation on reinstatement.

Theorem 3 Let $\mathcal{A} = (A, R)$, $S \subseteq A$, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2} . If $C_{Re_1}(A', R')$ and $C_{Re_2}(A'', R'')$ are satisfied by S then $F_{\mathcal{A}}(S) \subseteq S$.

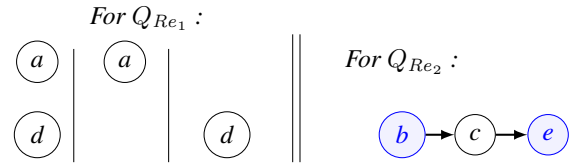
The second results shows the behavior of both parts of the explanation on reinstatement if computed on a set that contains all the arguments it effectively defends.

Theorem 4 Let $\mathcal{A} = (A, R)$, $S \subseteq A$, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2} . If $F_{\mathcal{A}}(S) \subseteq S$ then $C_{Re_1}(A', R')$ and $C'_{Re_2}(A'', R'')$ are satisfied by S .

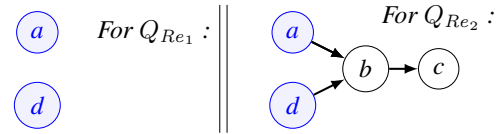
From Th. 3 and 4 follows the next corollary, which shows an equivalence result :

Corollary 1 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ such that $R^2(S)$ is conflict-free, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2} . $F_{\mathcal{A}}(S) \subseteq S$ iff $C_{Re_1}(A', R')$ and $C_{Re_2}(A'', R'')$ are satisfied by S .

Example 9 Let consider \mathcal{A} given in Ex. 1 and $S = \{b, e\}$. There are 3 explanations for proving that S does not satisfy the first reinstatement principle (some unattacked arguments are not in S ; here it is the case for a and d) and one for proving that S satisfies the second reinstatement principle (the arguments defended by S are in S) :



Let consider now $S = \{a, d\}$. There are one explanation for proving that S satisfies the first reinstatement principle (any unattacked argument is in S) and another one for proving that S does not satisfy the second reinstatement principle (some arguments defended by S are not in S ; here it is the case of c) :



3.4 Explanation about Complement Attack

To decide whether a set S of arguments attacks its complement, one must know whether or not all the arguments not in S are attacked by S . Thus, we firstly require our explanation to contain all the arguments of the original framework (S and its complement), and secondly to contain only attacks from S to arguments not in S . However, with only these two constraints, it may happen that no attacks targeting a specific argument outside of S are displayed on the explanation when there are some in the original framework. Hence, we add a third constraint, which is that if an argument not in S is attacked by S , then at least one attack from S to this argument must be present in the explanation.

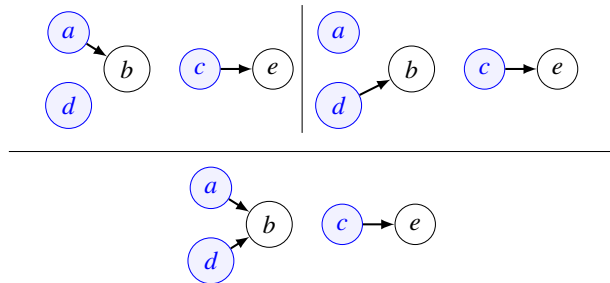
Definition 15 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{(a, b) \in R \mid a \in S, b \notin S\}$. The subgraph (A', R') of \mathcal{A} is an explanation to Q_{CA} iff

- $A' = A$
- $R' \subseteq X$
- $\forall b \in A \setminus S$, if $b \in R^{+1}(S)$, then $\exists (a, b) \in R'$ with $a \in S$

Note that the subgraph G_{CA} recalled in Def. 9 obviously belongs to the class of explanations for the principle of complement attack. Moreover concerning this principle, it was proven in [2] that a set of arguments attacks its complement iff there are no isolated vertices in $G_{CA}(S)$ and the explanation subgraph is always a bipartite graph with the arguments of S being the only possible origins for attacks. We extend these results to our class of explanations for complement attack.

Theorem 5 Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{CA} .
 $A \setminus S \subseteq R^{+1}(S)$ iff $C_{CA}(A', R')$ is satisfied by S .
 (A', R') is a bipartite graph, S can always be one of its parts and all vertices in S are sources in it.

Example 10 Let consider \mathcal{A} given in Ex. 1 and $S = \{a, b, c\}$. There are three explanations to Q_{CA} proving that S satisfies the principle of complement attack :



4 Properties of Explanations

We now turn to the definition of explanation properties and to a formal study of our classes of explanations according to them. This will allow to highlight some particular kinds of explanations, as well as to better understand their behavior. The properties that we will consider are : minimality, maximality, emptiness and uniqueness.

4.1 Some specific explanations

In this section, we identify some specific properties that could be respected by our explanations.

Minimality, Maximality A minimal (resp. maximal) explanation is an explanation which contains the least (resp. all the) possible amount of information. In a sense, a minimal explanation only provides what is required to explain whereas a maximal explanation in fact provides everything that might be relevant to explain, even if it might be redundant.

Definition 16 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. The subgraph (A', R') of \mathcal{A} is a minimal (resp. maximal) explanation that answers Q_π iff there is no subgraph (A'', R'') of \mathcal{A} which is also an explanation that answers Q_π such that (A'', R'') is a strict subgraph of (A', R') (resp. (A', R') is a strict subgraph of (A'', R'')).

Example 7 (cont'd) In this example, the maximal explanation is the first one and the two other ones are minimal.

Emptiness The notion of an empty explanation is one that should be avoided when providing explanations, in the sense that it somewhat represents the incapacity of the system to answer the question that has been asked.

Definition 17 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. The subgraph (A', R') is an empty explanation that answers Q_π iff $(A', R') = (\emptyset, \emptyset)$.

Uniqueness We consider an explanation to be unique when there is only one of its kind. Although we defined classes of explanations in an attempt to represent all the different points of view that could emerge as to how to answer a question, in some situations, there can only be one way to answer that question.

Definition 18 Let $\mathcal{A} = (A, R)$ be a graph. The subgraph (A', R') is a unique explanation that answers Q_π iff there is no subgraph (A'', R'') with $(A'', R'') \neq (A', R')$ which is also an explanation that answers Q_π .

Example 9 (cont'd) In this example, the explanations for the second reinstatement principle are unique (for $S = \{b, e\}$ or $S = \{a, d\}$) whereas the explanation for the first reinstatement principle is unique for $S = \{a, d\}$ but not for $S = \{b, e\}$.

Minimality and uniqueness are seen as explanation principles in [13]. However, these two notions are defined differently in [13], relatively to another concept of explanation based on sets of arguments, not on subgraphs, as we do.

4.2 Properties of specific explanations

Here, we provide the results of our formal study on our explanations using the aforementioned properties. We begin with empty explanations. The results show that, although empty explanations can occur, they only do so in very specific situations.

The following theorem establishes a characterisation of empty explanations, which generalises a similar result given in [2]. Moreover if this empty explanation occurs, it is the only possible one.

Theorem 6 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. (\emptyset, \emptyset) is an explanation that answers

1. Q_π with $\pi \in \{CF, Def, Re_2\}$ iff $S = \emptyset$.
2. Q_{Re_1} iff $\{a \in A \mid R^{-1}(a) = \emptyset\} = \emptyset$.
3. Q_{CA} iff $\mathcal{A} = (\emptyset, \emptyset)$.

If (\emptyset, \emptyset) is an explanation to Q_π with $\pi \in \{CF, Def, Re_1, Re_2, CA\}$, then it is unique.

Now, we turn to our study of maximal explanations. The next theorem states for each principle that there is only one possible maximal explanation.

Theorem 7 Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. If (A', R') is a maximal explanation that answers Q_π with $\pi \in \{CF, Def, Re_1, Re_2, CA\}$, then it is the unique maximal explanation that answers Q_π .

In the worst case, the number of explanations can be exponential in the size of some specific sets of elements, depending on the type of explanation (for instance the set of the attacks between the arguments belonging to the extension S in the case of explanations for the conflict-free principle). Thus considering only minimal explanations is a first step towards a computationally efficient method.

Nevertheless, as it turns out, there can be multiple minimal explanations in general for each principle. The next theorem studies the relation between minimal and maximal explanations and shows that the maximal explanation is exactly the union of all the minimal explanations.

Theorem 8 *Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. Consider $\pi \in \{CF, Def, Re_1, Re_2, CA\}$ and let (A', R') be the maximal explanation that answers Q_π and M be the set of all minimal explanations that answers Q_π . Then, $(A', R') = \bigcup_{G \in M} G$.*

This result opens the way to algorithmic solutions since, for a given principle, a maximal explanation covers all the possible explanations (the minimal ones but also all the intermediate explanations).

5 Computation of Explanations

This section investigates how to compute the maximal and minimal explanations of a class.

Maximal Explanations It turns out that maximal explanations exactly correspond to the explanations defined in [2] (recalled in Def. 9) :

Proposition 3 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. $G_\pi(S)$ is the maximal explanation that answers Q_π .*

This result entails that maximal explanations can be computed using only the graph operators of induced and spanning subgraphs, thus ensuring an efficient computation.

Note that Prop. 3 aggregated with Th. 7 allows to recover a unicity result given in [2].

From Maximal to Minimal Explanations In order to compute the minimal explanations for each principle π , we start from the maximal explanation :

Given $\mathcal{A} = (A, R)$ and $S \subseteq A$, $(A', R') \leftarrow G_\pi(S)$

Then, we gradually remove elements until obtaining a minimal explanation. This leads to five algorithms Alg_π (one for each principle π) that are built following the same schema. They also use the same condition for stopping the removal : “it remains at most one element to remove”. The only differences between these algorithms concern the “nature” of the removed elements :⁵

For CF , removal of attacks between elements of S :

While $|R'| > 1$
 $(x, y) \leftarrow choose(R')$; $R' \leftarrow R' \setminus \{(x, y)\}$

5. Note that these elements are generally attacks except in the case of the principle Re_1 .

For Def , for each attacker of S that is not in S , removal of attacks that target it :

For $y \in R^{-1}(S) \setminus S$
While $|R'^{-1}(y)| > 1$
 $x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

For Re_1 , removal of unattacked arguments not in S :

While $|A' \setminus S| > 1$
 $x \leftarrow choose(A' \setminus S)$; $A' \leftarrow A' \setminus \{x\}$

For Re_2 , for each argument that is an attacker of the arguments S defends and that is not defended by S , removal of attacks that target it :

For $y \in R^{-1}(R^{+2}(S)) \setminus R^{+2}(S)$
While $|R'^{-1}(y)| > 1$
 $x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

For CA , for each argument that is not in S , removal of attacks that target it :

For $y \in A \setminus S$
While $|R'^{-1}(y)| > 1$
 $x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

Our algorithms are sound and complete for the computation of minimal explanations as shown by the following proposition.

Proposition 4 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. Algorithm Alg_π using \mathcal{A} and S as inputs is sound and complete for the computation of a minimal explanation that answers Q_π .*

The computation of minimal explanations thus relies on the computation of maximal explanations, and the removal of some arcs (or arguments) in them. The computation of maximal explanations is already known to be polynomial (see [2]). Moreover the complexity of the removal operation in the worst case is linear in the number of removed elements and this number is either quadratic in the number of vertices in the graph when these elements are attacks (so for any principle except the one for the first part of reinstatement), or linear in the number of vertices in the graph when these elements are vertices (for the first part of reinstatement). From these considerations, our algorithms can be considered as computationally efficient.

Note also that a slight adaptation of these algorithms could produce random intermediate explanations (so neither minimal, nor maximal). This could be done by randomly stopping the removal process after a parametric number of steps. It is also the way to create more specific explanations responding to certain constraints given by users (for instance, explanations containing only x elements of a given type among the $y \geq x$ existing ones).

6 Conclusion and Future Work

This paper has defined *classes of explanations* for principles and semantics for the explanation Verification Pro-

blem XVer in Abstract Argumentation. These classes of explanations have been studied according to general properties such as maximality, minimality, emptiness and uniqueness. They extend and generalize the single explanations of [2], allowing more flexibility in the choice of explanations that could be presented to potential users. Moreover we have established that the explanations of [2] correspond to the maximal explanations of the defined classes, thus providing a way to compute them using graph operators. A procedure to compute minimal explanations from the maximal ones has also been provided and proven sound and complete for each class of explanations.

These results make an implementation of the proposed approach ready to be done. From this implementation, like in any XAI approach, as underlined by [7], an empirical evaluation should be conducted to assess to which extent these visual explanations actually are helpful for human agents to understand the answer to the Verification Problem. This is a first important future work, clearly related with the explainability social process described in [14].

Moreover, this evaluation could also provide a first study about what is a “best explanation” and how to select it. It is therefore also related to a second important future work : how take into account the issue of the “realizability”, or personalization of an explanation. Indeed, one may have in mind parts of an explanation (some arguments, some attacks), but not a correct and complete explanation; determining whether there exists such an explanation, and providing it, would ensure a personalized answer. In order to do so, a deeper investigation of the inner structure of the classes of explanation, and more specifically of the links they could have with lattices, may be of help.

This contribution and its research avenues will be of help in any application which uses computational abstract argumentation [19, 7].

In addition, the approach may be extended in several directions :

- to some semantics that use additional principles like maximality/minimality for set inclusion, for instance, the preferred or grounded semantics; in this case, some new visual criteria must be identified *in order to be able to explain why* a given set is or is not a preferred or a grounded extension; note that the visualization difficulty is not related to the complexity of the underlying problem (since the Ver problem for the grounded semantics is a polynomial problem whereas it is an exponential one for the preferred semantics);
- to contrastive questions : single explanations to such questions have been proposed in [2]; their generalisation to classes of explanations may be studied using the work presented here since, very often, a contrastive question can be viewed as the conjunction of some specific single questions.

Moreover, extending XVer to additional semantics and additional questions can be considered as an attempt to produce a generic approach for the computation of explanations, on the model of the approach of [3].

Finally, more notions of Graph Theory may be investigated in order to provide other kinds of visual explanations. In particular, the notion of graph isomorphism seems of great interest, especially to provide ways of reasoning by association (explaining a result via a structurally identical argumentation framework that one already accepted).

Références

- [1] Ringo Baumann and Markus Ulbricht. Choices and their consequences - explaining acceptable sets in abstract argumentation frameworks. In *Proc. of KR*, pages 110–119, Online event, 2021. IJCAI Organization.
- [2] Philippe Besnard, Sylvie Doutre, Théo Duchatelle, and Marie-Christine Lagasque-Schiex. Explaining semantics and extension membership in abstract argumentation. *Intelligent Systems with Applications*, 16 :200118, 2022.
- [3] Philippe Besnard, Sylvie Doutre, Théo Duchatelle, and Marie-Christine Lagasque-Schiex. Generic logical encoding for argumentation. *Journal of Logic and Computation*, 2022.
- [4] AnneMarie Borg and Floris Bex. A basic framework for explanations in argumentation. *IEEE Intelligent Systems*, 36(2) :25–35, 2021.
- [5] AnneMarie Borg and Floris Bex. Necessary and sufficient explanations for argumentation-based conclusions. In *Proc. of ECSQARU*, volume 12897 of *LNCS*, pages 45–58, Prague, Czech Republic, 2021. Springer.
- [6] Oana Cocarascu, Kristijonas Čyras, Antonio Rago, and Francesca Toni. Explaining with argumentation frameworks mined from data. In *Proc. of DEXAHAI*, Southampton, United Kingdom, 2018.
- [7] Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. Argumentative XAI : A survey. In *Proc. of IJCAI*, pages 4392–4399, Online Event / Montreal, Canada, 2021. IJCAI Organization.
- [8] Sylvie Doutre, Théo Duchatelle, and Marie-Christine Lagasque-Schiex. Classes of Explanations for the Verification Problem in Abstract Argumentation. Research Report IRIT/RR–2022–09–FR, IRIT : Institut de Recherche en Informatique de Toulouse, France, 2022.
- [9] Sylvie Doutre and Jean-Guy Mailly. Quantifying the difference between argumentation semantics. In *Proc. of COMMA*, volume 287, pages 255–262, Potsdam, Germany, 2016. IOS Press.
- [10] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2) :321–357, 1995.
- [11] Wolfgang Dvorák and Paul E Dunne. Computational problems in formal argumentation and their complexity. *Handbook of formal argumentation*, 4 :631–688, 2018.

- [12] Xiuyi Fan and Francesca Toni. On computing explanations in argumentation. In *Proc. of AAAI*, pages 1496–1502, Austin, Texas, USA, 2015. AAAI Press.
- [13] Beishui Liao and Leendert van der Torre. Explanation semantics for abstract argumentation. In *Proc. of COMMA*, volume 326, pages 271–282, Perugia, Italy, 2020. IOS Press.
- [14] Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence*, 267 :1–38, 2019.
- [15] Andreas Niskanen and Matti Järvisalo. Smallest explanations and diagnoses of rejection in abstract argumentation. In *Proc. of KR*, pages 667–671, Rhodes, Greece, 2020. IJCAI Organization.
- [16] Teeradaj Racharak and Satoshi Tojo. On explanation of propositional logic-based argumentation system. In *Proc. of ICAART*, volume 2, pages 323–332, Online Streaming, 2021. SCITEPRESS.
- [17] Zeynep Gozen Saribatur, Johannes Peter Wallner, and Stefan Woltran. Explaining non-acceptability in abstract argumentation. In *Proc. of ECAI*, volume 325, pages 881–888, Santiago de Compostela, Spain, 2020. IOS Press.
- [18] Markus Ulbricht and Johannes Peter Wallner. Strong explanations in abstract argumentation. In *Proc. of AAAI*, pages 6496–6504, Online event, 2021. AAAI Press.
- [19] Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. Argumentation and explainable artificial intelligence : a survey. *The Knowledge Engineering Review*, 36, 2021.
- [20] Srdjan Vesic, Bruno Yun, and Predrag Teovanovic. Graphical representation enhances human compliance with principles for graded argumentation semantics. In *Proc. of AAMAS*, pages 1319–1327, Auckland, New Zealand, 2022. IFAAMAS.

Segmentation de phases de dialogue dans des retranscriptions de conversations de centres d'appels

Guillaume Dubuisson Duplessis, Manon Richard, Anne-Laure Guénet

EDF Commerce, Direction des Systèmes d'Information et du Numérique (DSIN),
CSC datascience & IA, 420 rue Estienne d'Orves, 92700 Colombes

{guillaume.dubuisson-duplessis, anne-laure.guenet}@edf.fr

Résumé

La segmentation de phases de dialogue dans les retranscriptions de conversations de centres d'appels est cruciale pour leur exploitation opérationnelle. Cet article présente une approche d'apprentissage automatique supervisée qui nécessite une charge d'annotation manuelle raisonnable pour la segmentation. Cette approche est basée sur une ingénierie des caractéristiques qui prend en compte la nature conversationnelle des données. Les résultats de l'étude fournissent des enseignements clés et amènent une discussion sur la définition même des phases de dialogue.

Mots-clés

TALN, segmentation de phase de dialogue, retranscription

Abstract

The task of dialogue phase segmentation in call center conversation transcriptions is a crucial step for operational purposes. This research presents a supervised machine learning approach that requires a reasonable amount of manual annotation for segmentation and is based on feature engineering that takes into account the conversational nature of the data. The study describes key insights, including a discussion on the very definition of dialogue phases.

Keywords

NLP, dialogue segmentation, retranscription

1 Introduction

Les appels téléphoniques représentent la majeure partie des contacts de la relation client sur le marché des clients particuliers d'un acteur comme « Électricité de France » (EDF). Les retranscriptions automatiques peuvent nourrir de nombreux cas d'usage [2] comme, par exemple, l'optimisation des appels (aide à la professionnalisation, optimisation du discours des conseillers, optimisation des temps de traitements) et la vérification de la qualité des conversations en fonction de standards établis.

L'exploitation des retranscriptions de conversations de centres d'appels à des fins opérationnelles se fonde généralement sur une tâche clé : la segmentation en phases de dialogue. Le Tableau 1 présente un exemple d'enchaînement de deux phases de dialogue dans un appel de mise

CC :	1	oui d'accord vous pouvez vous connecter sur votre boîte mail on va créer ensemble votre espace client
	2	on va faire une signature dématérialiser de vos contrats donc pas besoin d'imprimer et de scanner d'accord
C :	3	oui
CC :	4	quand tout sera signé je pourrai vous envoyer les documents pour votre bailleur d'accord
C :	5	d'accord ok
...
CC :	6	ensuite pour la date de mise en service
C :	7	oui j'aurais voulu savoir si c'est possible d'avoir l'électricité dès jeudi

TABLE 1 – Un exemple de segmentation de phases de dialogue dans des retranscriptions d'appel téléphonique illustrant l'enchaînement de la phase de signature électronique et la phase de date de mise en service. Il contient sept unités inter-pausales (UIP). CC=« conseiller client », C=« client ».

en service. La tâche de segmentation vise à reconstruire automatiquement les phases de dialogue à partir des retranscriptions. Un dialogue est un objet structuré [4] impliquant des structures locales comme la paire adjacente [11] et des structures plus étendues comme la structure dite « intentionnelle » liée à la tâche sous-jacente [6]. L'expression « phase de dialogue » peut référer à ces structures plus ou moins étendues. Dans ces travaux, nous nous focalisons sur des dialogues orientés par une tâche sous-jacente de mise en service d'un contrat de fourniture d'énergie. Par phase de dialogue, nous référons aux étapes suivies par le conseiller pour mener à bien une mise en service. Schématiquement, les appels suivent une structure enchaînant les étapes suivantes : salutations, description de la demande, description du logement, localisation du logement, choix des offres et services, mise en place du paiement, signature électronique des contrats, rendez-vous technicien et clôture de l'appel. Ces travaux se focalisent sur l'extraction de la phase de signature électronique des contrats à des fins analytiques en vue de mieux connaître celle-ci (temps passé, sujets abordés par le conseiller et le client) et d'identifier des leviers d'op-

timisation via l’analyse des phases anormalement longues. Les contributions de ces travaux sont multiples. Tout d’abord, nous nous intéressons à la tâche de segmentation de retranscriptions d’appels, peu traitée par manque de corpus disponible mais pourtant clé pour l’exploitation de ces données riches [2, 12]. Ensuite, nous présentons une approche d’apprentissage automatique supervisée opérationnelle nécessitant une charge d’annotation manuelle raisonnable. Elle est fondée sur une ingénierie des caractéristiques prenant en compte la nature conversationnelle des données. Enfin, nous présentons les principaux enseignements parmi lesquels une discussion sur la définition même des phases de dialogue.

La suite de l’article se découpe en trois parties. La Section 2 pointe les travaux connexes les plus saillants. La Section 3 forme le cœur de l’article. Elle présente l’approche proposée pour la segmentation de phases de dialogue, les données d’apprentissage utilisées, les principales expérimentations réalisées et discute les résultats obtenus. Enfin, la Section 4 clôt cet article en soulignant les principales conclusions et en identifiant quelques perspectives prometteuses.

2 Travaux connexes

La segmentation en phases de dialogue de retranscriptions de conversations de centres d’appels est une tâche clé permettant l’exploitation des retranscriptions à de nombreuses fins opérationnelles [2]. Compte-tenu du coût important en termes d’annotation de données et de l’indisponibilité de données visant cette tâche, la plupart des approches de segmentation de phases de dialogue se base sur une approche non-supervisée fondée sur une hypothèse de cohérence lexicale et sémantique [7, 1]. Ces approches ont des performances plutôt modestes et limitées pour un usage opérationnel visant à calculer des indicateurs précis sur des phases. Des travaux récents visent à améliorer les performances de ces approches non-supervisées en apprenant automatiquement des modèles évaluant la cohérence d’une paire d’énoncés [12]. Ces travaux combinent les apports de l’apprentissage par transfert via un modèle BERT [3] à la relative facilité de construction d’un corpus d’apprentissage pour entraîner un modèle évaluant la cohérence d’une paire d’énoncés. Nos travaux se positionnent dans un contexte opérationnel de minimisation de l’utilisation de la donnée. Ils se fondent sur une tâche de classification de texte employant des caractéristiques conversationnelles hétérogènes permettant un niveau de performance satisfaisant.

3 Approche proposée

3.1 Vision globale

Afin de détecter une phase dans un appel, la retranscription n’est pas traitée comme un bloc, mais est découpée en unité inter-pausales (UIP). Une UIP est une unité de parole ne contenant pas de pause et provenant d’un seul interlocuteur (cf. Tableau 1). Une retranscription est donc découpée en une multitude d’UIP. La tâche revient alors à une classification binaire : une UIP appartient-elle ou non à la phase d’intérêt ? (cf. Figure 1) Ce découpage permet d’avoir une plus

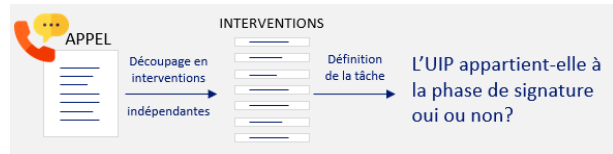


FIGURE 1 – Reformulation du problème de segmentation de phases de dialogue en une tâche de classification au niveau des unités inter-pausales (UIP).

grande volumétrie pour la modélisation. Une fois la classification réalisée, chaque UIP a une prédiction – dans ou hors de la phase – mais il peut parfois manquer de continuité dans la prédiction. Il est alors nécessaire de reconstruire la phase, par exemple en exploitant la densité de prédiction.

3.2 Description du corpus d’apprentissage

Le corpus est composé de 700 retranscriptions d’appels téléphoniques récoltées dans le cadre d’appels de mise en service. Dans ces données, un appel a une durée moyenne de 23min. Les retranscriptions sont donc de taille conséquente : en moyenne un appel comporte 960 UIP. En vertu du règlement général sur la protection des données (RGPD), les retranscriptions ont été désidentifiées et ne comportent aucune donnée à caractère personnel telles que des noms, prénoms, adresses, numéros [5]. Les données à caractère personnel sont substituées par le type de l’entité (e.g., "je suis monsieur [*person*] je vous appelle pour mon logement situé à [*localisation*]"). Les retranscriptions ne font apparaître aucune ponctuation et peuvent comporter des erreurs de retranscription.

Afin de détecter les phases de signature électronique dans les appels, une annotation des retranscriptions a été réalisée sur les 700 retranscriptions par deux personnes, ce qui représente une charge opérationnelle acceptable dans le cadre du projet. L’annotation a consisté à noter le début et la fin de la phase. Le corpus a ensuite été découpé en 3 sous-échantillons : un échantillon *train* composé de 420 appels (257K UIP) qui représente 60% des annotations, un échantillon *dev* (20%) composé de 140 appels (85K UIP) et un échantillon *test* (20%) de 140 appels (90K UIP).

3.3 Expérimentations autour de la classification des unités inter-pausales

3.3.1 Métriques d’évaluation considérées

La phase de signature électronique ne représente qu’une phase mineure dans une retranscription d’appel. L’échantillon est déséquilibré : 15% des UIP font partie de cette phase. Afin de mesurer les performances du modèle de classification, nous avons utilisé la métrique MCC (*Matthews Correlation Coefficient*) afin de prendre en compte ce déséquilibre. D’autres métriques ont également été visualisées à titre indicatif : la précision, le rappel et l’*accuracy*.

3.3.2 Choix de modélisation

Caractéristiques considérées Afin de prédire si une UIP fait partie de la phase considérée ou non, plusieurs variables ont été intégrées dans le modèle. Certaines décrivent direc-

tement l’UIP comme le texte de l’UIP, l’interlocuteur de l’UIP (soit « client » soit « conseiller »), la durée de l’UIP, la position de l’UIP dans l’appel (entre 0 – début – et 1 – fin – de la conversation). D’autres concernent le contexte dans lequel l’UIP a été mentionnée. Dans une fenêtre de T UIP avant et après – taille à définir – la durée des blancs (qui peut marquer un changement de phase dans le dialogue) ainsi que les UIP avant et après (ce qui a été dit par le client et conseiller dans une fenêtre de T UIP avant ou après) sont intégrées en variables supplémentaires dans le modèle. Ainsi, six caractéristiques structurées (interlocuteur, durée, position dans l’appel, durée des blancs avant et après l’UIP) et deux caractéristiques non-structurées (texte de l’UIP et contexte avant/après) sont intégrées en variables explicatives du modèle.

Vectorisations considérées pour les données texte À partir de l’ensemble de ces variables explicatives, nous avons construit un modèle de classification binaire. Afin de traiter les données textuelles, nous avons testé différentes vectorisations. Dans un premier temps, nous avons privilégié un TF-IDF, puis nous avons réduit le nombre de variables du TF-IDF en le combinant à une SVD (*Singular Value Decomposition*). Enfin, nous avons testé GloVe [10] en dimension 100, modèle pré-entraîné sur des UIP de retranscriptions. La représentation au niveau du document est obtenue par *mean* ou *max pooling*. Les données ont certaines spécificités : 50% des UIP ont entre 1-2 *tokens* uniquement et le mot le plus utilisé apparaît uniquement dans 19% des UIP contrairement à un corpus commun où le mot le plus utilisé comme un mot vide apparaît généralement dans plus de 95% des documents. Les hyperparamètres du TF-IDF sont à adapter à cette particularité. En outre, nous n’avons pas utilisé de modèles comme CamemBERT [9] ou FlauBERT [8] pour des raisons de tractabilité sur des conversations qui sont longues. De plus, les résultats rapportés pour le modèle CamemBERT indiquent des performances moindres sur des données de retranscription [9].

Classifieurs considérés Nous avons considéré plusieurs classifieurs : une régression logistique, un SVM (*Support Vector Machine*) avec kernel linéaire et un perceptron multi-couche (MLP) non-linéaire avec une couche cachée de taille 100. Les conversations étant des documents de taille très grande, les classifieurs de type réseau de neurones récurrents n’ont pas été testés pour des raisons de temps d’entraînement. Au cours de nos essais de modélisation, le perceptron multi-couche a produit de meilleures performances que la régression logistique et le SVM avec kernel linéaire. Dans la suite, nous ne reportons que les résultats basés sur les MLP.

3.4 Résultats

Un premier modèle *baseline* a été construit sans prise en compte du contexte de l’UIP. Il est représenté en première ligne du Tableau 2 et est le modèle qui a donné les meilleures performances sans le contexte. Plusieurs essais de modélisation successifs ont permis d’itérer et de constater quelle variable, quelle vectorisation, quel type de modèle impactent les résultats de la classification. Il en ressort

Vectorisation	T	N	MCC CV5	MCC dev
–	–	106	0.454	0.461
TF-IDF + SVD	3	306	0.631	0.656
GloVe	3	206	0.667	0.704
TF-IDF + SVD	6	306	0.7	0.736
GloVe	6	206	0.719	0.732

TABLE 2 – Performances des modèles en fonction de plusieurs approches de vectorisation du contexte et de la taille du contexte considérée en nombre d’UIP avant et après. T = taille du contexte. N = nombre de caractéristiques.

tout d’abord que la prise en compte des variables structurées en plus du texte – telles que la position de l’UIP dans la conversation – augmentent sensiblement la performance du modèle. La vectorisation de l’UIP basée sur le modèle GloVe pré-entraîné fonctionne mieux qu’une vectorisation basée sur TF-IDF et d’autant plus avec du *mean pooling*.

Une fois la *baseline* fixée, le contexte a été rajouté dans la modélisation. Les résultats sont retranscrits dans le tableau 2 et permettent de dégager des enseignements clés. Tout d’abord, le contexte apporte énormément d’information au modèle, en particulier une fenêtre T de taille 6. A noter que différencier le contexte avant du contexte après ne nous a pas apporté de gains de performance. De nouveau, la vectorisation du contexte basée sur notre modèle GloVe pré-entraîné fonctionne mieux qu’une vectorisation basée sur TF-IDF et a l’avantage de limiter le nombre de caractéristiques nécessaires pour une bonne classification. Nous avons également constaté que les UIP sont moins bien prédites lorsqu’elles sont au bord de la phase de signature électronique (mais la vérité terrain contient aussi des erreurs sur les bords).

Pour les besoins opérationnels de notre projet, nous avons retenu le modèle basé sur une vectorisation GloVe du contexte de taille 6 (dernière ligne du Tableau 2). Nous avons vérifié la bonne généralisation de ce modèle sur la partie "test" de notre corpus (MCC=0.736 / accuracy=0.94).

3.5 Reconstruction des phases de dialogue

La reconstruction de la phase de dialogue se fonde sur les prédictions d’appartenance à la phase au niveau des UIP. L’algorithme de reconstruction utilise une fenêtre glissante qui considère le nombre d’UIP prédites comme appartenant à la phase dans la fenêtre pour déterminer la phase de dialogue. Cela permet d’éliminer des UIP isolées faussement détectées comme faisant partie de la phase de signature électronique. Les paramètres qui nous ont donné les résultats les plus satisfaisants sont une taille de fenêtre de 10 UIP et une nombre minimum de voisins prédits dans la phase à 4. Nous avons mesuré la performance de la reconstruction de phase via l’indice de Jaccard (correspondant au calcul de l’intersection entre la phase prédite divisée par l’union de la phase de référence avec celle prédite). L’indice de Jaccard médian est de 0.73. 75% des appels ont un indice de Jaccard entre 0.5 et 0.87.

3.6 Discussion

Cette approche opérationnelle a la particularité de découper le problème en deux tâches (classification des UIP, reconstruction de la phase). Avec un nombre restreint d'annotations, la modélisation a donné des résultats stables sur les échantillons de *dev* et *test* via la prise en compte de caractéristiques conversationnelles hétérogènes.

L'une des difficultés dans la segmentation de phase de dialogue et qui a pu complexifier la reconstruction des phases réside dans deux particularités inhérentes aux phases de dialogue. La première est la présence de sous-dialogues incidents qui peuvent s'immiscer au sein de la phase [4] (e.g., « ok attendez je voulais savoir si je suis sur le principe des heures creuses heures pleines »). La seconde est la présence de phases mêlées (e.g., « donc pendant que vous êtes en train de valider je vais en profiter pour vous donner les conseils en matière d'économie d'énergie »). Dans ces deux cas, la phase segmentée n'est plus unifiée, mais séparée en deux ou plusieurs sous-phases. Cela présente un défi aussi bien de formalisation de la tâche de segmentation que technique pour de futurs travaux.

4 Conclusion et perspectives

Cet article a présenté une approche par apprentissage automatique pour la segmentation d'une phase de dialogue dans de réelles retranscriptions d'appels en centres d'appels ; approche plus générique, rigoureuse, répliquable et maintenable qu'une conception de règles spécifiques. Cette approche supervisée se base sur un corpus d'apprentissage de petite taille et des caractéristiques conversationnelles adaptées à cette tâche. Ces travaux se sont limités à une méthode de reconstruction d'une phase très simple à partir des prédictions au niveau des UIP. Une suite directe vise à améliorer cette méthode de reconstruction de phases. Une autre direction est d'étudier l'adaptation des modèles d'apprentissage profond pour segmenter des phases dans des séquences d'UIP pouvant être très longues. Enfin, une direction intéressante est d'explorer l'opérationnalité d'approches nécessitant pas ou peu de supervision manuelle [12].

Remerciements

Nous remercions chaleureusement toutes les personnes qui sont intervenues de près ou de loin sur ce projet : Gilles Pouëssel, Mélanie Cazes, Laura Rouhier, Sonia Audheon, Marie Hervé, François Raynaud.

Références

- [1] Laurent Bozzi, Philippe Suignard, and Claire Waast-Richard. Segmentation et classification non supervisée de conversations téléphoniques automatiquement retranscrites. In *Actes de la 16ème conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 331–336, 2009.
- [2] Chloé Clavel, Gilles Adda, Frederik Cailliau, Martine Garnier-Rizet, Ariane Cavet, Géraldine Chapuis, Sandrine Courcinous, Charlotte Danesi, Anne-Laure Daquo, Myrtille Deldossi, et al. Spontaneous speech and opinion detection : mining call-centre transcripts. *Language resources and evaluation*, 47(4) :1089–1125, 2013.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4171–4186, 2019.
- [4] Guillaume Dubuisson Duplessis. *Modèle de comportement communicatif conventionnel pour un agent en interaction avec des humains : Approche par jeux de dialogue*. PhD thesis, INSA de Rouen, 2014.
- [5] Guillaume Dubuisson Duplessis, Elliot Bartholme, Sofiane Kerroua, Mathilde Poulain, Ahès Roulier, and Anne-Laure Guénet. Désidentification de données texte produites dans un cadre de relation client. In *Actes de la 27ème conférence Traitement Automatique des Langues Naturelles (TALN) – démonstrations*, pages 10–13, 2020.
- [6] Barbara J Grosz and Candace L Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3) :175–204, 1986.
- [7] Marti A. Hearst. Text tiling : Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1) :33–64, 1997.
- [8] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert : Unsupervised language model pre-training for french. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, 2020.
- [9] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villamonte de la Clergerie, Djamel Seddah, and Benoît Sagot. Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- [10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [11] Emanuel A Schegloff and Harvey Sacks. Opening up closings. 1973.
- [12] Linzi Xing and Giuseppe Carenini. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 167–177, 2021.

Modèles graphiques causaux interactifs pour les données textuelles

A. Ferdjaoui^{1,2}, S. Affeldt², M. Nadif²

¹ SogetiLabs

² Centre Borelli UMR 9010, Université Paris Cité, France

amine.ferdjaoui@etu.u-paris.fr | severine.affeldt@u-paris.fr | mohamed.nadif@u-paris.fr

Résumé

Nous proposons de reconstruire des modèles graphiques causaux à partir de données textuelles via un nouveau package Python, `WordGraph`. Ce package facilite l'exploration de grands corpus de documents par des visualisations interactives sous forme de modèles graphiques de mots. Le pipeline `WordGraph` exploite à la fois les widgets jupyter et le notebook jupyter pour aider les utilisateurs qui n'ont pas d'expérience en Python à maîtriser un pipeline `WordGraph`, qui est entièrement personnalisable. `WordGraph` est disponible via un dépôt GitHub¹ qui fournit également une courte vidéo présentant l'utilisation de notre système.

Mots-clés

co-clustering, réseaux causaux, données textuelles.

Abstract

We propose to reconstruct causal graphical models from textual data via a new Python package, `WordGraph`. This package facilitates the exploration of large document corpora through interactive visualizations in the form of graphical word models. The `WordGraph` pipeline leverages both jupyter widgets and jupyter notebook technologies to help users with no Python experience master a fully customizable `WordGraph` pipeline. `WordGraph` is available via a GitHub¹ repository, which also provides a short video demonstrating the usage of our system.

Keywords

co-clustering, causal network reconstruction, text data.

1 Introduction

1.1 Co-clustering et données textuelles

La tâche consistant à répartir simultanément les objets et les caractéristiques dans des blocs homogènes pour une matrice permet d'obtenir des regroupements de lignes et de colonnes plus précis et plus faciles à interpréter qu'un regroupement unidimensionnel. Depuis les travaux fondateurs de Hartigan [19], le co-clustering (également connu sous le nom de biclustering) a trouvé des applications dans de nombreux domaines tels que la bio-informatique [18], le web mining [14] et le text mining [9]. Le co-clustering

est particulièrement bien adapté aux matrices de données documents×termes, qui sont par essence de haute dimension, peu denses et présentent des caractéristiques directionnelles.

Les algorithmes de co-clustering exploitables sur ces matrices de cooccurrences sont issus de différentes approches. Les méthodes de co-clustering de type spectral, qui traitent la matrice d'entrée comme un graphe bipartite entre les documents et les mots, approximent la coupe normalisée de ce graphe à l'aide d'une relaxation réelle [8]. Les méthodes basées sur des modèles appropriés dérivés de blocs latents ou *Latent Block Models* (LBM) [15], reposent sur des algorithmes de type Expectation-Maximisation [16, 34]. Le co-clustering peut également s'appuyer sur des méthodes basées sur la factorisation matricielle telles que la Factorisation Matricielle Non négative (NMF) [25, 4, 12] ou la Tri-Factorisation (NMTF) [10, 33]. En revanche, les méthodes qui utilisent la théorie de l'information, utilisées avec les tableaux de contingence à deux voies, visent à minimiser la perte d'information mutuelle en regroupant les lignes et les colonnes de la matrice en fonction du co-clustering [9]; il convient de noter que le critère optimisé est associé à un modèle de Poisson de type LBM contraint [27, 17]. Enfin, la copartition est également possible via une version adaptée de la mesure de modularité habituellement utilisée pour les réseaux [24, 3].

Parmi les packages populaires mis à disposition pour le co-clustering de données [23, 5, 11], le package Python `CoClust` est devenu une référence bien connue dans le domaine du co-clustering des matrices de cooccurrence, en particulier pour les matrices document-terme utilisées dans les applications de text mining [32]. `CoClust` fournit les implémentations de trois algorithmes conçus pour traiter efficacement de telles matrices². L'un de ces algorithmes utilise le critère d'information mutuelle, et les deux autres effectuent le co-clustering en maximisant la modularité des graphes bipartites.

Dans les sections suivantes, nous montrons comment le package Python `WordGraph` que nous proposons tire parti de `CoClust` pour permettre l'exploration de grands corpus et du vocabulaire associé à l'aide de modèles graphiques.

1. <https://github.com/MLDS-software/WordGraph>

2. <https://coclust.readthedocs.io/en/v0.2.1/install.html>

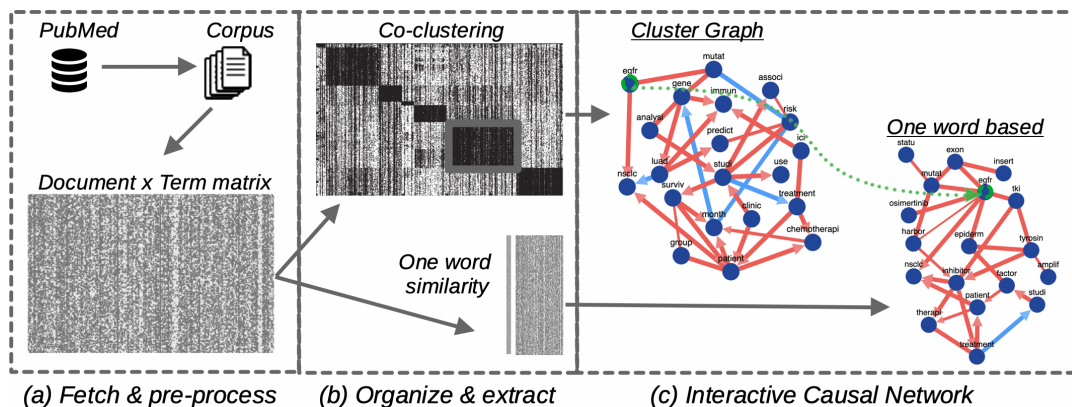


FIGURE 1 – Process WordGraph : de la recherche de documents aux réseaux causaux interactifs entre les mots.

1.2 Inférence de réseaux causaux

La reconstruction de modèles graphiques, pour analyser facilement la quantité toujours croissante des données massives, est devenue une pratique bien connue dans de nombreux domaines. Ces modèles peuvent être appris à partir de données de séries temporelles, d'expériences de perturbation contrôlée ou de données d'observation, comme c'est le cas dans de nombreux contextes biologiques. Les méthodes traditionnelles de reconstruction de modèles graphiques, qui exploitent des données non perturbées, comprennent le *search-and-score* bayésien [20], la *sparse inverse covariance estimation* [13], l'entropie maximale [22] ou les *diffusion maps* [6].

Au-delà de la reconstruction de modèles graphiques, la découverte de la causalité entre plusieurs variables est d'un grand intérêt. Par exemple, elle peut conduire à l'identification de facteurs de transcription ayant un rôle de régulateurs majeurs pour des maladies humaines complexes. Appliquée à des données textuelles, ces reconstructions peuvent fournir des *résumés graphiques* des interactions entre les différents sujets d'un large corpus. Les réseaux de causalité qui en résultent indiqueraient ainsi le flux organisationnel entre les mots par le biais de réseaux dirigés. Cependant, les approches traditionnelles supposent que le modèle sous-jacent est soit un réseau complètement orienté, soit un réseau non orienté, car ces approches ne peuvent pas découvrir la causalité à partir de données d'observation. En revanche, les méthodes basées sur les contraintes [37, 28, 7, 31], qui peuvent identifier les contraintes structurelles correspondant à des arêtes inutiles dans un graphe, peuvent découvrir la causalité à partir de données non perturbées. L'inconvénient des méthodes basées sur les contraintes est qu'elles présentent des problèmes de complexité algorithmique en présence de variables non observées, et qu'elles ne sont généralement pas robustes sur les petits ensembles de données.

Pour contourner ces difficultés, l'algorithme MIIC (Multivariate Information-based Inductive Causation), s'appuie sur une méthode dérivée de la théorie de l'information qui

combine l'apprentissage basé sur les contraintes et le principe de maximisation de la vraisemblance [1, 2, 38]. Plus précisément, MIIC est basé sur l'analyse de l'information multivariée [26, 39], qui étend le concept d'information mutuelle à plus de deux variables. En pratique, l'intégration d'une approche basée sur la découverte des contraintes structurelles dans le cadre de la théorie de l'information améliore considérablement la précision de la prédiction, le temps d'exécution et les capacités de passage à l'échelle en termes de taille d'échantillon et de taille de réseau par rapport aux approches traditionnelles.

Ainsi, alors que MIIC a démontré son intérêt sur une grande variété de données génomiques, à différentes échelles biologiques d'espace et de temps [38], il semble également être particulièrement bien adapté pour traiter les données textuelles de grande dimension, facilitant ainsi l'exploration des thématiques de large corpus. MIIC a été implémenté et mis à la disposition de la communauté dans un package R³. Il est également disponible via un serveur web accessible en ligne⁴ [35].

Dans ce qui suit, nous détaillons l'interface de programmation Python proposée par WordGraph et montrons comment elle combine naturellement les approches de co-clustering, qui facilitent la découverte des thématiques d'un corpus, et la reconstruction de modèles graphiques causaux, qui identifient les interactions dirigées entre les termes.

2 Description générale du pipeline

Notre pipeline se décompose en trois parties. Tout d'abord, il propose la création d'un corpus de documents directement extraits de la base de données en ligne PubMed⁵ (Fig. 1, a), qui comprend plus de 35 millions de citations pour la littérature biomédicale. La création du corpus inclut le prétraitement des données textuelles (par exemple, le *stemming* ou l'élimination des mots peu informatifs)

3. https://github.com/miicTeam/miic_R_package

4. <https://miic.curie.fr>

5. <https://pubmed.ncbi.nlm.nih.gov>

et la construction d'une matrice documents×termes, avec une procédure de pondération TF-IDF (Term Frequency-Inverse Document Frequency).

Ensuite, `WordGraph` permet le co-clustering de la matrice documents×termes pondérée pour explorer les thématiques sous-jacentes. Cette partie s'appuie sur le package `Coclust` et fournit le regroupement simultané des mots et des documents (Fig. 1, b, haut). Dans cette étape, on peut également obtenir un extrait de la matrice documents×termes avec les colonnes correspondant aux termes les plus similaires à un mot donné, sur la base du score de *similarité cosinus* (Fig. 1, b, bas).

Enfin, `WordGraph` fournit une interface de programmation Python du package R `MIIC` qui permet la reconstruction des réseaux causaux à partir de données d'observation. Un modèle graphique peut être obtenu à partir des termes les plus représentatifs - typiquement, les mots les plus fréquents - d'un co-cluster (Fig. 1, c, haut). Les visualisations de graphes, qui utilisent la bibliothèque Python `ipycytoscape`⁶, sont interactives. L'utilisateur peut réorganiser les nœuds, obtenir des vues agrandies et cliquer sur un sommet pour construire un graphe secondaire, *One word similarity*, basé sur un extrait de la matrice document×term (Fig. 1, c, bas). On peut également obtenir ce dernier type de graphe en entrant un mot du vocabulaire dans le champ de saisie *widjet*.

Dans ce qui suit, nous fournissons les détails du code source de `WordGraph` et expliquons les exploitations possibles à plusieurs niveaux. En effet, le package proposé peut être utilisé tel quel, dans une source Python ordinaire, ou bien l'utilisateur peut personnaliser un *notebook jupyter* interactif. Enfin, une application web peut être instantanément générée à partir du notebook pour faciliter le partage en ligne.

3 Un package à plusieurs niveaux

`WordGraph` permet d'accéder aux fonctionnalités de co-clustering de `Coclust` et à la méthode de reconstruction de réseaux causaux de `MIIC` suivant trois niveaux différents afin d'explorer les données textuelles. Dans cette section, nous présentons `WordGraph` comme (i) une interface de programmation Python légère pour le package R `MIIC`, (ii) un *notebook* Python interactif pour la reconstruction de modèles graphiques de mots, et (iii) une application web.

3.1 Une interface de programmation Python pour les réseaux causaux

`WordGraph` est le premier package Python qui propose une interface de programmation au package R `MIIC`. La figure 2 présente les principales fonctionnalités de `WordGraph` et de ses modules.

L'interface de programmation pour `MIIC` est encapsulée dans la méthode `build_graph` (*class* `WordGraph`) et repose sur les fonctions `create_miic` et `preproc_graphML` (module `utils`). Les autres fonctions de `utils` sont dédiées au prétraitement du corpus, tandis que la fonction `parse_pubmed_api`

6. <https://ipycytoscape.readthedocs.io/en/latest/>

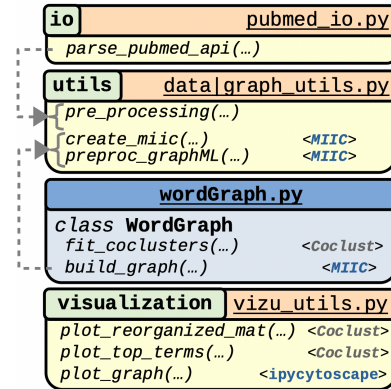


FIGURE 2 – Fonctionnalités principales de `WordGraph`.

offre un accès facile à PubMed (module `io`).

```

1 from wordGraph.wordGraph import WordGraph
2 from wordGraph.io.pubmed_io import
   parse_pubmed_api
3 from wordGraph.visualization.viz_utils import
   plot_top_terms, plot_graph
4
5 # 1. Definition des parametres
6 pm_query = 'cancer'; n_doc = 2500 # Corpus
7 n_clust = 3; n_terms = 10 # Co-clustering
8 g_shf = 100; g_thresh = 0.05 # MIIC
9 # 2. Recuperation des donnees PubMed
10 raw_corpus, clean_corpus=parse_pubmed_api(pm_query
   ,n_doc)
11 # 3. Apprentissage des coclusters
12 wg = WordGraph()
13 wg.fit_coclusters(clean_corpus, n_clust)
14 plot_top_terms(wg, nb_terms=n_terms)
15 # 4. Construction du reseau de mots
16 wg.set_cocluster(cocluster=3, nb_terms=n_terms)
17 wg.build_graph(n_shuffles=g_shf, conf_threshold=
   g_thresh)

```

Listing 1 – Utilisation simple de `WordGraph`

Le Listing 1 présente une utilisation simple de `WordGraph`, depuis la recherche de document via PubMed (l.10) jusqu'au calcul de l'objet graphe `MIIC` (l.17). Nous fournissons dans `WordGraph` un *notebook jupyter* qui intègre cette proposition d'utilisation simple (voir *WordGraph utilisation simple.ipynb*).

3.2 Un notebook interactif d'exploration

`WordGraph` dispose d'un module `visualization` qui utilise le package `ipycytoscape`. En particulier, `plot_graph` affiche des réseaux interactifs dans un *notebook jupyter*, sur lequel la disposition des graphes peut être directement ajustée à l'aide de la souris. Il fournit des histogrammes de fréquence pour les principaux termes des co-clusters et permet ainsi de résumer facilement les thématiques (Fig. 3, à gauche). Un clic sur un nœud du modèle graphique de mots permet l'affichage d'un réseau interactif secondaire, dans lequel les sommets sont choisis en fonction de leur similarité avec le nœud précédemment sélectionné (Fig. 3 droite). De plus, `WordGraph` propose un *notebook jupyter* faci-

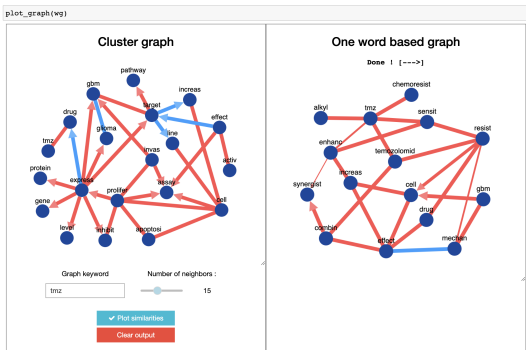


FIGURE 3 – Graphes interactifs dans le notebook jupyter.

lement personnalisable (*WordGraph Custom.ipynb*), dans lequel l'ensemble du pipeline peut être incorporé dans une seule cellule. Le notebook intègre plusieurs éléments ipywidgets (par exemple, un curseur, un bouton, des entrées textuelles et numériques). La figure 4 montre comment le pipeline s'organise dans une seule cellule.

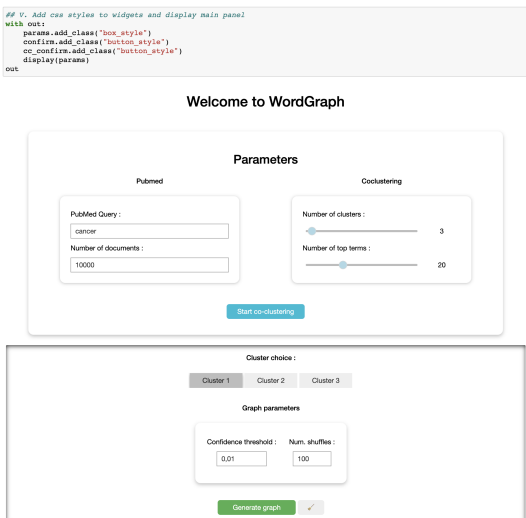


FIGURE 4 – Pipeline interactif dans un jupyter notebook.

La modification ou l'ajout d'un *widget* ne nécessite que quelques lignes de code. Le Listing 2 fournit un exemple d'implémentation pour la mise en place d'un bouton *confirm* et l'appel de sa fonction au moment du clic.

```

1 # Bouton 'Confirm' (label 'Do graph')
2 confirm = widgets.Button(description='Do graph',
3   disabled=False, button_style='success')
4 # Fonction callback via clic souris
5 def on_confirm_button_clicked(button):
6   button.disabled=True
7   # Par exemple: generation du reseau...
8   button.disabled=False
9 # Association de la fonction au bouton
10 confirm.on_click(on_confirm_button_clicked)
    
```

Listing 2 – Personnalisation du notebook interactif

4 Une application web interactive

Le notebook *WordGraph Custom.ipynb* a également été conçu pour être automatiquement transformé en application web, avec le package *voilà*⁷. Ainsi, l'utilisateur peut exploiter le *WordGraph Custom.ipynb* tel quel ou le personnaliser, puis obtenir facilement l'application web correspondante (Fig. 5), prête à l'emploi pour les utilisateurs qui ne sont pas familiers avec la programmation.

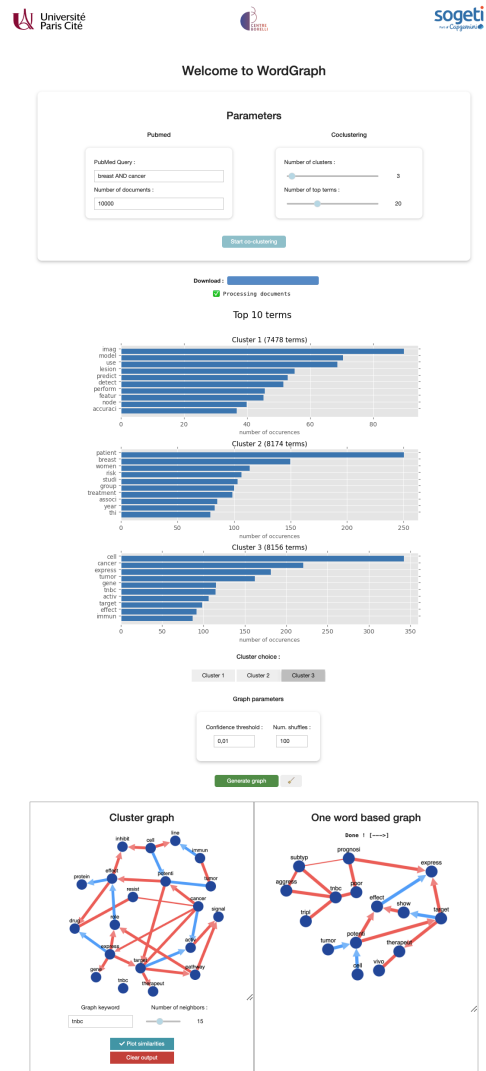


FIGURE 5 – Application web automatiquement générée.

5 Application biomédicale

Nous avons généré trois corpus en lien avec le cancer et d'une taille de 10,000 documents, et avons exploré leurs thématiques avec *WordGraph*.

La Figure 1 se focalise sur le **cancer du poumon**. L'opposition (lien bleu) entre *NSCLC* (Non-Small Cell Lung

7. <https://voila.readthedocs.io/en/stable/using.html>

Cancer) et *LUAD* (LUNG ADenocarcinoma) suggère que les documents traitant de *LUAD* ignorent le sujet *NSCLC*. En effet, même si les cancers de type *LUAD* appartiennent à la famille *NSCLC*, leur pronostic et leur expression génétique sont très distincts [29] et ils sont considérés comme des entités séparées. Le cancer *EGFR*-positive *NSCLC* résulte d'une mutation du gène *EGFR* (récepteur du facteur de croissance épidermique). Notre réseau de mots établit un lien causal entre la présence dans les documents du terme *EGFR* et de l'acronyme *NSCLC*. Le terme *ICI* (Immune Checkpoint Inhibitor) *treatment*⁸ a également une relation de cause à effet avec le nœud *immun*. Le réseau secondaire, basé sur *egfr*, se concentre sur le *treatment*, qui vise généralement à prévenir la croissance anormale de la tumeur en ciblant la Tyrosin Kinase Inhibitor (*TKI*), comme dans le permet le traitement par *osimertinib* [30].

Les modèles graphiques de la Figure 3 correspondent à un corpus centré sur le **glioblastoma cancer**. Le glioblastome multiforme (*GBM*) est le type de cancer du cerveau le plus agressif, avec une augmentation de l'*invasion*, de la *proliferation* et une diminution de l'*apoptosis*, comme le détaille notre graphe. Nous remarquons également le nœud *TMZ*, qui est le médicament (*drug*) chimiothérapeutique prédominant dans le *GBM*. Le graphe secondaire (Figure 3 droite), qui se concentre sur *TMZ*, apporte d'autres informations. En particulier, nous comprenons que la tumeur *GBM* a tendance à développer une *TMZ (temozolomid) resistance* ou *chimoiresistance* [36].

Enfin, la Figure 5 propose une exploration d'un corpus qui concerne le **cancer du sein**. En particulier, le graphe de cluster met l'accent sur la possibilité de *drug resistance* pour ce cancer. Ceci est particulièrement vrai pour le cancer du sein de type Triple Négatif *TNBC*. Séparé du réseau de cluster (Figure 5 gauche), le *TNBC* s'impose comme une sous-thématique, pour laquelle le graphique secondaire (Figure 5 droite) fournit des informations supplémentaires. Plus précisément, il s'agit d'un *subtype aggressive*, dont le *prognosis* est généralement *poor* [21].

6 Conclusion

L'exploration de corpus devient de plus en plus difficile car d'énormes quantités de données textuelles sont disponibles dans de nombreux domaines. Nous avons créé *WordGraph* afin de fournir un package Python facile à exploiter pour les utilisateurs souhaitant extraire un maximum d'informations autour d'un sujet biomédical. Nous avons fait des efforts particuliers pour faire de *WordGraph* une application web autonome pour les utilisateurs n'ayant aucune connaissance de Python ou de *jupyter notebook*, tout en permettant la personnalisation de l'interface. Nos applications d'exploration sur divers corpus autour du cancer démontrent l'intérêt de notre package, qui propose la première interface de programmation Python du package *R MIIIC*, pour la reconstruction de réseaux causaux.

8. Les documents des corpus étant en anglais pour cette application, nous rendons compte des termes également en anglais, tels qu'il apparaissent dans les réseaux. La procédure de *stemming* explique la troncature de certains mots.

Remerciements

Ce travail est soutenu par une subvention de l'Agence Nationale de la Recherche (ANR) (ANR-19-CE23-0002). Il a également reçu la labellisation des pôles de compétitivité *Cap Digital* et *EuroBiomed*. Nous remercions le *Isambert lab* pour son support concernant le package *R MIIIC*.

Références

- [1] Séverine Affeldt and Hervé Isambert. Robust reconstruction of causal graphical models based on conditional 2-point and 3-point information. In *ACI@ UAI*, pages 1–29, 2015.
- [2] Séverine Affeldt, Louis Verny, and Hervé Isambert. 3off2 : A network reconstruction algorithm based on 2-point and 3-point information statistics. In *BMC bioinformatics*, volume 17, pages 149–165. BioMed Central, 2016.
- [3] Melissa Ailem, François Role, and Mohamed Nadif. Co-clustering document-term matrices by direct maximization of graph modularity. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1807–1810, 2015.
- [4] Melissa Ailem, Aghiles Salah, and Mohamed Nadif. Non-negative matrix factorization meets word embedding. In *SIGIR*, pages 1081–1084, 2017.
- [5] Simon Barkow, Stefan Bleuler, Amela Prelić, Philip Zimmermann, and Eckart Zitzler. Bicat : a biclustering analysis toolbox. *Bioinformatics*, 22(10) :1282–1283, 2006.
- [6] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data : Diffusion maps. *Proceedings of the national academy of sciences*, 102(21) :7426–7431, 2005.
- [7] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.
- [8] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001.
- [9] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, 2003.
- [10] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD in-*

- ternational conference on Knowledge discovery and data mining*, pages 126–135, 2006.
- [11] Kemal Eren, Mehmet Deveci, Onur Küçükünç, and Ümit V Çatalyürek. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, 14(3) :279–292, 2013.
- [12] Mickael Febrissy, Aghiles Salah, Melissa Ailem, and Mohamed Nadif. Improving nmf clustering by leveraging contextual relationships among words. *Neuro-computing*, 495 :105–117, 2022.
- [13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441, 2008.
- [14] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.
- [15] Gérard Govaert and Mohamed Nadif. Block clustering with bernoulli mixture models : Comparison of different approaches. *Computational Statistics & Data Analysis*, 52(6) :3233–3245, 2008.
- [16] Gérard Govaert and Mohamed Nadif. *Co-clustering : models, algorithms and applications*. John Wiley & Sons, 2013.
- [17] Gérard Govaert and Mohamed Nadif. Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in data analysis and classification*, 12 :455–488, 2018.
- [18] Blaise Hanczar and Mohamed Nadif. Ensemble methods for biclustering tasks. *Pattern Recognition*, 45(11) :3938–3949, 2012.
- [19] John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337) :123–129, 1972.
- [20] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks : The combination of knowledge and statistical data. *Machine learning*, 20 :197–243, 1995.
- [21] William J Irvin Jr and Lisa A Carey. What is triple-negative breast cancer? *European journal of cancer*, 44(18) :2799–2805, 2008.
- [22] Edwin T Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9) :939–952, 1982.
- [23] Sebastian Kaiser and Friedrich Leisch. A toolbox for bicluster analysis in r. 2008.
- [24] Lazhar Labiod and Mohamed Nadif. Co-clustering for binary and categorical data with maximum modularity. In *2011 IEEE 11th international conference on data mining*, pages 1140–1145. IEEE, 2011.
- [25] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.
- [26] W. McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4) :93–111, 1954.
- [27] Mohamed Nadif and Gérard Govaert. Block clustering of contingency table and mixture model. In *Advances in Intelligent Data Analysis*, pages 249–259, 2005.
- [28] Judea Pearl and Thomas S Verma. A theory of inferred causation. In *Studies in Logic and the Foundations of Mathematics*, volume 134, pages 789–811. Elsevier, 1995.
- [29] Valeria Relli, Marco Trerotola, Emanuela Guerra, and Saverio Alberti. Abandoning the notion of non-small cell lung cancer. *Trends in Molecular Medicine*, 25(7) :585–594, 2019.
- [30] J Remon, CE Steuer, SS Ramalingam, and E Felip. Osimertinib and other third-generation egfr tki in egfr-mutant nsccl patients. *Annals of Oncology*, 29 :i20–i27, 2018.
- [31] Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4) :962–1030, 2002.
- [32] François Role, Stanislas Morbieu, and Mohamed Nadif. Coclust : a python package for co-clustering. *Journal of Statistical Software*, 88 :1–29, 2019.
- [33] Aghiles Salah, Melissa Ailem, and Mohamed Nadif. Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [34] Aghiles Salah and Mohamed Nadif. Directional co-clustering. *Advances in Data Analysis and Classification*, 13 :591–620, 2019.
- [35] Nadir Sella, Louis Verny, Guido Uguzzoni, Séverine Affeldt, and Hervé Isambert. Miic online : a web server to reconstruct causal or non-causal networks from non-perturbative data. *Bioinformatics*, 34(13) :2311–2313, 2018.
- [36] Neha Singh, Alexandra Miner, Lauren Hennis, and Sandeep Mittal. Mechanisms of temozolomide resistance in glioblastoma-a comprehensive review. *Cancer drug resistance*, 4(1) :17–43, 2021.
- [37] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1) :62–72, 1991.
- [38] Louis Verny, Nadir Sella, Séverine Affeldt, Param Priya Singh, and Hervé Isambert. Learning causal networks with latent variables from multivariate information in genomic data. *PLoS computational biology*, 13(10) :e1005662, 2017.
- [39] Raymond W Yeung. A new outlook on shannon’s information measures. *IEEE transactions on information theory*, 37(3) :466–474, 1991.

L'optimisation du covoiturage dynamique multi-saut

C. Fèvre¹, H. Zgaya-Biau¹, P. Mathieu¹, S. Hammadi¹

¹Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRISStAL, F-59000 Lille, France

corwin.fevre@online.fr

Résumé

Dans cet article, nous proposons une approche multi-agent pour résoudre le problème du covoiturage dynamique multi-saut. Les passagers et les conducteurs sont représentés comme des agents autonomes et rationnels en perpétuelle interaction pour satisfaire leurs propres objectifs. Dans la solution proposée, les agents conducteurs et passagers ont une perception modélisée dynamiquement en utilisant des R-Arbres. Nous modélisons leurs préférences en matière de détour et de trajet et montrons l'impact de celles-ci sur la résolution d'une instance de covoiturage dynamique.

Mots-clés

Covoiturage, Simulation, Optimisation, Agents

Abstract

In this paper, we propose a multi-agent approach to solve the dynamic multi-hop ridesharing problem. Passengers and drivers are represented as autonomous, rational agents in perpetual interaction to satisfy their own goals. In the proposed solution, driver and passenger agents have a dynamically modeled perception using R-Trees. We model their detour and route preferences and show the impact of these on the resolution of a dynamic ridesharing instance.

Keywords

Ridesharing, Simulation, Optimization, Agents

1 Introduction

Le covoiturage est devenu un moyen de transport à part entière, que ce soit pour les trajets quotidiens ou pour les voyages plus longs. Outre son intérêt écologique, il permet aux conducteurs de réduire les coûts liés à leurs déplacements tout en offrant la possibilité aux passagers de voyager avec plus de flexibilité que les moyens de transport conventionnels. Il en résulte une utilisation accrue des véhicules déjà déployés sur la route, une réduction de la congestion du trafic et donc une diminution des émissions polluantes. Nous proposons dans ce travail d'étudier le covoiturage dynamique entre particuliers. Un système de covoiturage dynamique prends en compte l'état actuel de l'environnement des utilisateurs pour effectuer des associations de covoiturage. Un tel système collecte en temps réel différents flux d'information dans l'environnement comme les itinéraires et temps de trajets des conducteurs en cours de route ainsi

que les requêtes des passagers. Le caractère dynamique réside alors dans le flux continu de conducteurs et de passagers entrant, évoluant et sortant du système. Cette forme de covoiturage est majoritairement étudiée à l'échelle de grandes métropoles [10, 29] et dans le cadre de trajets courts et spontanés [30, 21].

La flexibilité d'un système de covoiturage dynamique peut être modélisée et quantifiée par deux opérations. D'une part, les conducteurs peuvent choisir de faire des détours pour prendre ou déposer des passagers, et ainsi élargir l'offre de transport du système de covoiturage. D'autre part, les passagers peuvent choisir de passer d'un véhicule à l'autre, et donc d'effectuer des transferts, afin d'arriver à leur destination en plusieurs étapes, si cela est plus pratique pour eux. Un exemple de ces opérations est illustré dans la Figure 1.

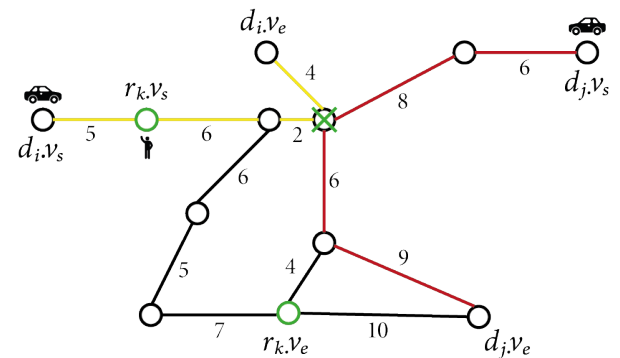


FIGURE 1 – Un exemple de covoiturage multi-saut

Dans cette figure, deux conducteurs d_i et d_j suivent leur itinéraire, respectivement en jaune et rouge, reliant leur nœud de départ v_s et leur nœud d'arrivée v_e . Un passager r_k attend sur son nœud de départ $r_k.v_s$ et veut rejoindre son nœud d'arrivée $r_k.v_e$, tous deux entourés de vert. Les trajectoires des conducteurs se croisent sur le nœud avec une croix verte, un transfert est alors réalisable. Le conducteur d_i peut partager son trajet avec le passager r_k puis, une fois arrivé au nœud de transfert, le passager peut effectuer un transfert avec le conducteur d_j . Le conducteur d_j doit cependant faire un détour pour déposer le passager, ce qui rajoute 5 unités de temps à son temps de trajet.

Le défi d'un tel cadre est d'équilibrer l'impact du covoiturage sur l'offre et la demande. Cet équilibre implique

de nombreux paramètres tels que le ratio conducteurs-passagers ou les détours acceptables par les utilisateurs en termes de temps et de distance. Le problème est alors d'associer au mieux passagers et conducteurs en exploitant leur marge de manœuvre afin de satisfaire les deux parties.

Un système de covoiturage peut être naturellement représenté par un système multi-agent, les conducteurs et les passagers étant des entités autonomes en interaction. C'est pourquoi nous proposons dans cet article un système multi-agent permettant de décrire des comportements spécifiques pour chaque utilisateur, garantissant le caractère hétérogène de notre simulation. La limite de perception de ces agents est définie dynamiquement par leurs préférences spatio-temporelles. Ils disposent ainsi d'une rationalité limitée.

Pour traiter l'important flux d'informations généré par un tel système, nous utilisons une structure d'indexation spatiale nommée R-Arbre [22]. Cette structure, et les algorithmes qui y sont associés, permet de stocker efficacement ces données et d'effectuer des requêtes de superposition sur des points et des zones de l'espace. Les R-Arbres nous permettent aussi de réduire la complexité associée à l'identification du nœud de transfert optimal dans le cas d'un covoiturage multi-saut.

Après un état de l'art sur le problème du covoiturage et ses approches d'optimisation et de simulation dans la section 2, nous définissons notre problème de covoiturage en section 3. Nous proposons en section 4 une solution basée sur un système multi-agent dont le comportement des agents est défini par divers algorithmes. Enfin, en section 5 nous étudions différents comportements typiquement observables dans des scénarios réels de covoiturage. Nous les confrontons à notre modèle et étudions les résultats obtenus sur des cartes réelles.

2 Travaux connexes

À l'occasion d'un covoiturage, un passager peut être acheminé par un seul et unique conducteur, c'est ce que l'on nomme le **covoiturage simple saut** (ou *single-hop* en anglais). Le problème du covoiturage simple-saut consiste à identifier le meilleur conducteur pour acheminer un passager entre deux positions. C'est la première forme de covoiturage étudiée dans la littérature [19, 1, 10] de par ses similarités avec les problèmes d'allocation de ressources.

Il est aussi possible pour un passager d'utiliser plusieurs conducteurs, et donc d'effectuer des transferts afin d'arriver à sa destination. Cette variante s'appelle le **covoiturage multi-saut** (ou *multi-hop* en anglais) ou encore le covoiturage avec transferts [8]. Cette méthode permet d'améliorer les résultats obtenus avec le covoiturage simple-saut en augmentant les possibilités de covoiturage [17, 30]. Le covoiturage multi-saut implique une complexité accrue, car il faut cette fois identifier la meilleure combinaison de conducteurs ainsi que le lieu de transfert optimal pour tous les acteurs du covoiturage. Le problème du covoiturage multi-saut a été associé à de nombreux problèmes NP-Durs et NP-Complets de la littérature [7, 21].

Pour gérer cette complexité, l'espace de recherche doit être

organisé et exploré de manière optimale. De nombreux travaux de recherche hiérarchisent l'espace [23] ou le divisent en zones pour en réduire la complexité [29, 28]. Ces divisions et réductions de l'espace sont cependant la plupart du temps effectuées de manière empirique et non dynamique forçant une approximation importante de l'espace. Dans ce contexte, une structure d'indexation spatiale connue sous le nom de R-Arbre (*R-Tree* [15]) permet d'indexer dynamiquement les objets dans un arbre et d'optimiser les requêtes de superposition de zones ou de nœuds de l'espace [22]. Cette structure a notamment été utilisée pour le partage de taxi et s'est révélée être très efficace [30]. Dans nos travaux, nous proposons d'étendre le champ d'utilisation des R-Arbres au problème du covoiturage entre particuliers.

Il existe de nombreuses façons de modéliser le problème, notamment en ce qui concerne la granularité de la population. D'une part, l'approche centrée-groupe vise à optimiser des objectifs communs à l'ensemble de la population et donc à privilégier le système par rapport à l'individu [10, 1]. Cette méthodologie permet une simulation moins coûteuse en ressources mais s'avère éloignée de la réalité vis-à-vis du caractère hétérogène de la population. En revanche, l'approche centrée-individu cherche à résoudre la demande de chaque utilisateur, au cas par cas, en tenant compte de ses préférences pour définir les objectifs d'optimisation. Ainsi, chaque individu a son propre comportement et sa propre perception. Dans ce contexte, l'approche multi-agent [12] permet de modéliser un tel système en garantissant l'autonomie de chaque utilisateur et en permettant à la simulation d'être plus proche de la réalité [3, 19, 18, 24, 9].

Le problème du covoiturage a d'abord été étudié de manière mono-objectif avec des approches de programmation par contraintes [1, 10]. Des problèmes de covoiturage multi-objectif ont ensuite été étudiés et résolus à l'aide d'heuristiques évolutionnaires et génétiques [17, 16]. Ces méthodes permettent une optimisation plus rapide - bien qu'approximative - et donc plus applicable dans le monde réel. Avec l'avènement du covoiturage multi-saut et afin d'obtenir une solution exacte et multi-objectif, la recherche s'oriente vers une approche du problème de covoiturage via le problème du ou des plus proches voisins d'un groupe (GNN/ANN) [25]. En effet, puisque le multi-saut implique la recherche du nœud de transfert optimal, des heuristiques basées sur des algorithmes de recherche de plus proches voisins ont été développées [31] et ont été améliorées pour être plus efficaces dans un contexte réel [30, 6]. C'est dans cette perspective que nous proposons d'étendre l'algorithme de référence IER (Incremental Euclidean Restriction) [31] pour évaluer conjointement les solutions à sauts multiples et à sauts uniques de manière multi-objectif.

3 Formulation du problème

3.1 Spécification du réseau routier et des agents du système

Le **réseau routier** de notre système est modélisé par un graphe orienté et connexe $G = \langle V, E \rangle$ dans lequel V est l'ensemble des nœuds du graphe et E est l'ensemble des

arêtes liant ces nœuds. Les arêtes représentent des routes et les nœuds représentent des intersections de routes ou des impasses. Un nœud $v_i = (x, y)$ est référencé spatialement par un couple de coordonnées : une longitude x et une latitude y . Différents poids peuvent être attribués aux arêtes désignant le coût du trajet entre deux nœuds, comme le temps de trajet par exemple.

Nous utilisons une simulation en temps discret, le temps évolue à intervalles constants et le pas de simulation courant est dénoté par la variable *time*. À chaque pas de simulation, tous les passagers et conducteurs présents dans le système peuvent effectuer une action en fonction de leur comportement.

Un agent utilisateur du système $u_i = \langle v_s, v_l, v_e, det_{max} \rangle$, avec $u_i \in U$ représentant un passager ou un conducteur dans le système de covoiturage, est initialisé par quatre éléments : son nœud de départ v_s , son nœud de localisation actuel v_l , son nœud d'arrivée v_e et son facteur de détour det_{max} . Le facteur de détour quantifie la volonté de détour d'un agent utilisateur. Ainsi, s'il vaut 0, l'utilisateur ne souhaite pas faire de détour, et s'il vaut 1, l'utilisateur est prêt à doubler son trajet initial pour effectuer des détours. Nous faisons varier ce facteur entre 0 et 1 dans nos expériences. Cet élément permet de définir les limites initiales d'action et de perception d'un agent.

Un agent conducteur $d_i = \langle v_s, v_l, v_e, det_{max}, c_{max} \rangle$, avec $d_i \in D$, est initialisé par cinq éléments. Un conducteur étant un utilisateur, il hérite des quatre premiers paramètres définis précédemment. Le dernier paramètre c_{max} définit le nombre maximal de sièges disponibles dans le véhicule du conducteur. Un conducteur peut effectuer un nombre infini de détours et d'arrêts tant qu'il respecte son objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive. Il suit donc son trajet de nœud en nœud, en prenant systématiquement le chemin le plus court. S'il doit faire un détour pour prendre et déposer un passager, il doit être capable de déterminer si cette action est possible vis-à-vis de son heure d'arrivée la plus tardive. Pour ce faire, cet agent maintient un ordonnanceur de voyage où il stocke différentes informations telles que ses prochains arrêts *stops* (nœuds du graphe de route), les temps de trajets minimum *arr* et maximum *ddl*, les marges de détours *slk* ou les capacités restantes *c* (nombre de sièges disponibles) entre ses arrêts. La modélisation exacte de cet ordonnanceur est détaillée dans nos précédents travaux [13].

Un agent passager $r_i = \langle v_s, v_l, v_e, det_{max}, wt_{max}, pref \rangle$, avec $r_i \in R$, est initialisé par six éléments. Un passager étant un utilisateur du système, il hérite des quatre premiers paramètres définis précédemment. Le paramètre wt_{max} correspond au temps d'attente acceptable au maximum par le passager pour la totalité du trajet. S'il est dépassé, l'agent passager s'impatiente et disparaît du système, ce qui affecte le taux de service global du système de covoiturage. La perception p d'un agent passager représente la zone des conducteurs accessibles. Elle permet d'exclure les conducteurs trop éloignés pour le prendre en charge. Le paramètre *pref* représente le vecteur de préférence du passager, c'est-à-dire son profil/comportement.

Chaque élément $pref_i \in [0, \alpha]$ et la somme des éléments de *pref* doit être égale à α : $sum(pref_i, \dots, pref_k) = \alpha$. On peut alors, par exemple, fixer α à 10 et faire varier la pondération des préférences d'un agent passager entre 0 et 10. De cette manière, nous pouvons attribuer un poids à chaque objectif contenu dans cette liste de préférences et détailler différents profils de passagers. Ceci implique que, si $pref_i > pref_j$, alors le passager accorde plus d'importance à la préférence $pref_i$ qu'à la préférence $pref_j$ lors de l'optimisation de son trajet. Un passager a donc pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive tout en sélectionnant le ou les meilleurs covoiturages en fonction de ses préférences.

Un agent service de transport $t_{sa_i} = \langle RT_p \rangle$ est responsable de la connexion des agents utilisateurs. Il remplit deux fonctions. La première consiste à stocker et à mettre à jour toutes les perceptions des agents conducteurs, à leur demande, lorsqu'un événement entraîne une modification de leur perception. La seconde fonction est de répondre aux requêtes spatiales des agents passagers du système qui souhaite identifier des candidats au covoiturage. L'agent service de transport est donc implémenté comme un "tableau noir" afin de mettre à jour et de transmettre les informations des agents du système lorsque cela est demandé.

3.2 Contraintes et Formulations

Nous limitons notre étude au covoiturage à un et deux sauts, c'est-à-dire au covoiturage limité à deux conducteurs pour un transfert. En effet, plusieurs articles comme [23] ont montré que l'augmentation du nombre de transferts possibles améliore à peine les performances d'un système de covoiturage tandis que la complexité du problème explose exponentiellement. Dans cet article, chaque requête de covoiturage est associée à un seul et unique passager.

Pour qu'un covoiturage entre un passager et un conducteur soit réalisable, nous définissons plusieurs contraintes spatio-temporelles :

- **Contrainte temporelle** : Les conducteurs et les passagers doivent arriver à leur destination avant leur heure d'arrivée souhaitée. Un passager doit être pris en charge avant que son temps d'attente maximum ne soit atteint.
- **Contrainte de capacité** : La capacité courante d'un conducteur ne doit jamais dépasser la capacité maximale de son véhicule.
- **Contrainte de l'ordre des arrêts** : Un conducteur partageant son itinéraire avec un passager doit d'abord passer par le nœud de prise en charge du passager avant de visiter le nœud de dépôt de ce dernier.

Une association ("*a match*" en anglais) $m \in r_i.M$ est définie par plusieurs variables. Un ou deux conducteurs $d_j, d_k \in D$ (simple-saut : $j = k$, multi-saut : $j \neq k$); un nœud de transfert v_{tsf} s'il y a un transfert; et enfin quatre critères : un nombre de transferts nb_{tsf} (0 ou 1 dans cette étude), un temps d'attente total $total_wt$, un temps d'arrivée arr_time , et une distance additionnelle de

détour add_dist . Chaque passager a pour but de choisir l'association la plus optimale dans M selon ses préférences $pref$. Le problème étudié est donc multi-objectif en impliquant la minimisation simultanée de ces quatre critères.

La complexité de notre problème peut être résumée par : quel est le nombre de combinaisons de covoiturage possible ? Avec une approche individu-centrée cette complexité peut être formalisée comme suit :

$$O(R * H^{D*N}) \tag{1}$$

avec H : le nombre de sauts possibles; D : le nombre d'agents conducteurs; R : le nombre d'agents passagers; N : le nombre de nœuds du graphe de route. Ce problème est ainsi associé aux problèmes de classe de complexité exponentielle : $O(2^n)$.

4 Le système multi-agent proposé

À chaque étape de la simulation, le tour d'un agent utilisateur est divisé en deux phases : la phase de mise à jour de sa perception de l'environnement et la phase de prise de décision.

4.1 La mise à jour de la perception des agents

La perception d'un agent utilisateur représente ce qu'il est capable d'utiliser dans l'environnement pour atteindre son objectif. Dans nos travaux, cette perception est dynamique et doit être mise à jour régulièrement.

Pour établir sa perception, l'agent utilisateur collecte d'abord les nœuds du plus court chemin entre son origine et sa destination. Il récupère ensuite les latitudes et longitudes minimales et maximales de tous ces nœuds pour définir le rectangle de délimitation minimum (MBR) de sa perception. Cette perception initiale est étendue en y ajoutant la marge de temps de détour ou d'attente restante de l'agent afin de former le rectangle englobant (BBOX) de sa perception. Ce processus est basé sur nos travaux antérieurs [13] et est illustré dans la Fig.2.

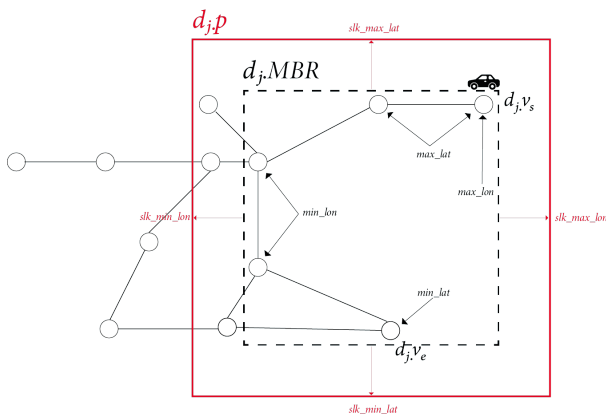


FIGURE 2 – Exemple de calcul de la perception d'un agent conducteur.

Cette méthode est utilisée pour la perception d'un conducteur, entre chaque arrêt de son ordonnanceur, et pour la perception d'un passager, de son point de départ à son point

d'arrivée. Cette perception est approximative, ainsi, un test de faisabilité plus rigoureux, basé sur les contraintes de la section 3.2, est nécessaire avant tout covoiturage.

Un agent conducteur ne met à jour sa perception que dans deux cas : (1) s'il atteint un nœud du graphe de route; (2) si un contrat de covoiturage impliquant un détour est conclu avec un passager. Un agent passager met à jour sa perception à chaque étape de la simulation. Si un nouveau conducteur apparaît dans sa perception, il cherche à déterminer s'il peut permettre de rejoindre sa destination.

4.2 L'identification du covoiturage optimal

Le processus d'identification du covoiturage optimal est résumé dans l'algorithme 1. Comme expliqué précédemment, un conducteur peut permettre à un passager de rejoindre sa destination de deux façons : (1) simple-saut : en assurant la totalité du trajet (2) multi-saut : en lui offrant la capacité d'effectuer un transfert avec un autre conducteur. Pour identifier les conducteurs candidats à un covoiturage, un passager effectue des requêtes à l'agent service de transport. Un conducteur candidat au covoiturage est un conducteur dont au moins l'une des perceptions chevauche la perception du passager ainsi qu'au moins l'un de ces deux éléments : le nœud de départ et/ou d'arrivée du passager (PMO L1-3). Ainsi, un conducteur dont la perception chevauche les deux nœuds est candidat au simple-saut, tandis que si sa perception ne chevauche que l'un des nœuds, il est nécessaire de trouver un autre conducteur complémentaire pour espérer effectuer covoiturage multi-saut. Un conducteur est complémentaire à un autre s'il chevauche une partie de sa perception, la zone de rencontre des perceptions des deux conducteurs et du passager est appelée zone de transfert (voir Figure 3).

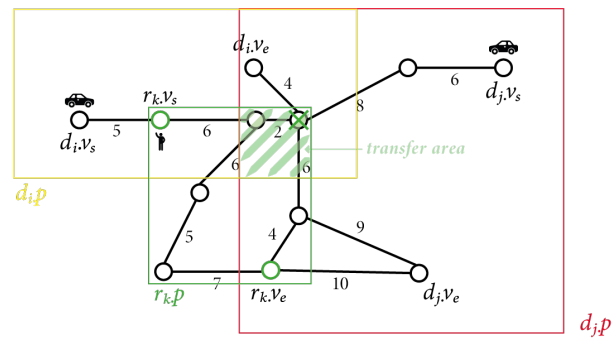


FIGURE 3 – Identification des candidats au covoiturage à l'aide des différentes perceptions.

L'évaluation d'un covoiturage simple saut est peu complexe. Le passager évalue les différents objectifs relatifs au trajet et établit une valeur indiquant la performance de cette solution en fonction de ses préférences (PMO L7-10). La fonction multi-objectif responsable de cette évaluation est détaillée plus tard dans cet article. À contrario, l'évaluation d'un covoiturage multi-saut induit une complexité très importante. Il faudrait effectuer cette même méthode pour chaque nœud de transfert possible pour les conducteurs. Il

n'est pas envisageable de couvrir toutes ces possibilités. C'est pourquoi nous avons développé une heuristique, nommée CIER *Constrained Incremental Euclidean Restriction* visant à prioriser l'évaluation des nœuds de transferts les plus proches du passager et des conducteurs (PMO L12-17)[14].

Cette heuristique est basée sur l'algorithme IER (*Incremental Euclidean Restriction*) [26] permettant la résolution du problème des plus proches voisins d'un groupe (ANN/GNN) [25]. Ce problème est semblable au problème de l'identification du nœud de transfert optimal auquel nous sommes confrontés. L'objectif est d'identifier le(s) nœud(s) d'un graphe le(s) plus proche(s) d'un groupe d'autres nœuds de ce même graphe. Un exemple simple est le problème du restaurant : quel est le restaurant le plus proche de trois individus éparpillés dans une ville? Les nœuds de transferts sont alors les restaurants et les conducteurs et le passager sont les individus. La notion du plus proche peut alors être complexe : minimisant la somme des distances (utilitaire), la moyenne des distances (égalitaire)...

Nous avons choisi une approche utilitariste ayant pour objectif de minimiser la distance additionnelle de détour occasionnée par le covoiturage. L'intérêt de cette approche est de limiter l'impact du covoiturage sur les conducteurs afin de conserver leur marge de détour pour les autres passagers du système. L'étude et la comparaison d'autres politiques feront certainement l'objet d'une étude de notre part. Une fois le nœud de transfert optimal identifié, le passager évalue la solution de la même façon que pour un covoiturage simple-saut.

Pour diversifier l'offre de covoiturage proposée à un passager, nous proposons une solution basée sur le mécanisme de dominance de Pareto [27]. L'idée générale est de maintenir une liste des meilleures solutions de covoiturage *best_matches*. Une meilleure solution de covoiturage est une solution minimisant au moins un des objectifs présentés dans la section 3.2 (temps d'attente, heure d'arrivée, distance additionnelle et nombre de transferts). Elle est, dans ce cas, Pareto dominante et est ajoutée à la liste *best_matches*. De cette façon, nous préservons des solutions optimales pour chaque objectif et le passager peut, selon ses préférences, choisir celle qui lui convient le mieux. Si une solution de covoiturage optimise tous les objectifs, alors elle est Pareto optimale, la liste est alors vidée et cette solution lui est ajoutée (PMO, L19-24). Le nombre de solutions optimales est ainsi compris entre 0, quand il n'y a pas de solution, et 4 si il existe une solution pareto dominante par objectif.

En plus de cette liste des meilleures solutions, nous maintenons deux autres listes : la liste des pires valeurs *worst_goals* et des meilleures valeurs *best_goals* atteintes pour chaque objectif. Ces listes servent d'intervalles de référence à un agent passager pour évaluer la performance d'une solution. En effet, en effectuant une différence proportionnelle pondérée par son vecteur de préférence, l'agent passager obtient un pourcentage de performance pour chaque objectif ainsi qu'en moyenne. Ce pour-

centage est calculé à l'aide de la formule 2 :

$$\frac{\sum_i^k (1 - \frac{abs(candidate_goals_i - worst_goals_i)}{worst_goals_i - best_goals_i} * 100) * pref_i}{k + sum(pref)} \quad (2)$$

Avec k le nombre d'objectifs à considérer. De cette façon, un objectif plus important pour le passager a plus d'impact sur l'évaluation finale de la solution. Le passager sélectionne enfin la solution la plus performante (PMO, L28).

Algorithm 1 PMO - Preferential Matching Optimization

Input : $G =$ road infrastructure graph, $r_i =$ a rider agent
Output : *best_match* = the best match

- 1: *candidates_s* : drivers which perception overlaps starting node
- 2: *candidates_e* : drivers which perception overlaps ending node
- 3: *best_add_dist*, *best_wt*, *best_arrival_time*, *best_nb_tsf* := ∞ ,
- 4: *best_matches* := *list()* # $r_i.M$
- 5: **for all** $d_s \in candidates_s$ **do**
- 6: **for all** $d_e \in candidates_e$ **do**
- 7: **if** $d_s.id = d_e.id$ and match is feasible **then**
- 8: #single hop
- 9: eval $m.total_wt$, $m.arr_time$, $m.nb_tsf$, $m.add_dist$
- 10: set match driver $m.d = d_s$
- 11: **else**
- 12: #multi hop, CIER heuristic
- 13: Q : set of drivers et rider position nodes
- 14: T : set of candidate transfer nodes
- 15: find : $t \in T$ minimizing aggregated distance from Q and feasible
- 16: eval : $m.total_wt$, $m.arr_time$, $m.nb_tsf$, $m.add_dist$
- 17: set match drivers : $m.d_0 = d_s$, $m.d_1 = d_e$, $m.v_tsf = t$
- 18: **end if**
- 19: **if** m minimizes all current best values of the objectives (pareto optimal) **then**
- 20: overwrite *best_matches* := *set(m)* #the only element of the set is now m
- 21: update all objectives : *best_add_dist*, *best_wt*, *best_nb_tsf*, *best_arrival_time*
- 22: **else if** m minimizes at least one objective (pareto dominant) **then**
- 23: add m to *best_matches*
- 24: update concerned objectives
- 25: **end if**
- 26: **end for**
- 27: **end for**
- 28: *best_match* = best match among *best_matches* according rider preferences
- 29: **return** *best_match*

5 Expérimentations et résultats

Dans cette section, nous proposons de simuler différents comportements de passagers et de conducteurs à l'aide du modèle agent présenté dans cet article. Les résultats présentés montrent l'impact de ces comportements sur des instances de problème de covoiturage.

5.1 Création des profils utilisateurs

Nous appelons "un profil utilisateur" un ensemble de préférences visant à simuler un comportement réel. Pour les conducteurs, nous faisons essentiellement varier la préférence de détour det_{max} entre 0 (c'est-à-dire pas de détour) et 1, correspondant à une capacité de détour de la taille du trajet initial du conducteur. Nous pouvons ainsi générer tout un spectre de comportement de conducteurs, allant du plus égoïste au plus altruiste. Concernant les passagers, nous définissons 5 profils d'utilisateurs présentés dans la Table 1 en fonction des 4 objectifs présentés en section 3.2.

Chaque valeur de ce tableau représente le poids associé à chaque objectif : plus le poids est élevé, plus le profil a de l'intérêt pour l'objectif concerné. Concrètement, on

TABLE 1 – Tableau des différents profils des agents passagers

Profil	total_wt	add_dist	arr_time	tsf_nb
Équilibré	3	3	3	1
Pluie	7	1	1	1
Écologique	1	7	1	1
Pressé	1	1	7	1
Confort	1	1	1	7

peut imaginer un curseur ajustant les poids de ces objectifs dans une application smartphone de covoiturage dynamique. Le profil "équilibré" représente un passager moyen, sans réelles préférences, et sert de profil de référence. Le profil "pluie" représente un agent passager sous la pluie qui souhaite être pris en charge rapidement. Il donne alors la priorité à la minimisation du temps d'attente total du trajet. Le profil "écologiste" représente un passager préoccupé par l'impact environnemental lié à la conduite d'une voiture. Ce profil privilégie le partage de trajet qui implique peu de détours et cherche donc à minimiser la distance additionnelle de détour liée au covoiturage. Le profil "pressé" représente un passager qui souhaite atteindre sa destination le plus rapidement possible. Ce profil donne la priorité à la minimisation de l'heure d'arrivée à sa destination. Enfin, le profil "confort" souhaite éviter le désagrément de devoir changer de véhicule au cours d'un trajet. Ce profil donne la priorité aux covoiturages simple-sauts et minimise donc le nombre de transferts de véhicules.

5.2 Protocole expérimental

Nous avons privilégié l'étude de petites instances à de nombreuses reprises afin d'obtenir des résultats reproductibles et fiables. De cette manière, les biais liés aux paramètres du système sont contrôlés. Nous effectuons une simulation avec une génération continue de conducteurs nous permettant de reproduire une situation réelle où il y a un flux entrant et sortant de conducteurs dans le système. En variant la densité de ce flux, nous pouvons simuler tout un panel de trafics routiers et évaluer les comportements en situation de rareté ou d'abondance de l'offre de covoiturage. La résolution de l'instance se termine lorsque tous les agents passagers ont quitté le système : en atteignant leur destination ou en s'impatiantant.

Nos instances sont composées d'un graphe de 150 nœuds provenant d'un réseau routier réel de la ville de San Francisco, extrait d'Open Street Map et modélisé à l'aide de la bibliothèque OSMNX [5]. Ce graphe est illustré dans la Figure 4. Sur ce graphe en grille, 30 conducteurs et 20 passagers interagissent. Leurs nœuds de départ et de destination sont générés de manière aléatoire et uniforme. Les itinéraires initiaux des conducteurs sont dérivés du plus court chemin entre leurs nœuds de départ et d'arrivée. Nous supposons que les passagers ont un temps d'attente maximal de 3 minutes et un facteur de détour fixé à 0,2 pour créer une situation difficile.

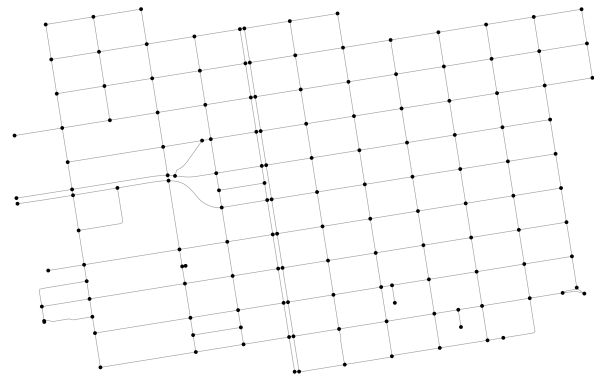


FIGURE 4 – Le graphe étudié : une section du côté nord de la ville de San Francisco

5.3 Résultats

Dans cette sous-section, nous détaillons les résultats obtenus avec notre modèle. L'objectif est de mettre en évidence l'impact des différentes stratégies sur des indicateurs tels que : le temps d'attente moyen, le taux de service (succès du covoiturage), la distance additionnelle moyenne et le nombre moyen de véhicules nécessaires au covoiturage. Chaque figure comprend plusieurs courbes correspondant aux profils des passagers mentionnés précédemment. Chaque point d'une courbe est le résultat de 50 expériences moyennées. Ces expériences correspondent aux 50 mêmes expériences pour chaque profil de passager. Enfin, l'axe des abscisses est indexé par le facteur de détour des conducteurs.

5.3.1 L'impact du profil "écologiste"

La Figure 5 montre comment le profil "écologiste" (courbe verte/triangles) minimise la distance additionnelle moyenne de détour des conducteurs par rapport aux autres profils. Ce phénomène est amplifié par l'augmentation du facteur de détour des conducteurs. On observe ainsi des détours 2 à 3 fois moins importants avec ce profil qu'avec les profils concurrents. En conséquence, les autres objectifs sont très largement négligés par ce profil (Figures 6, 7, 8).

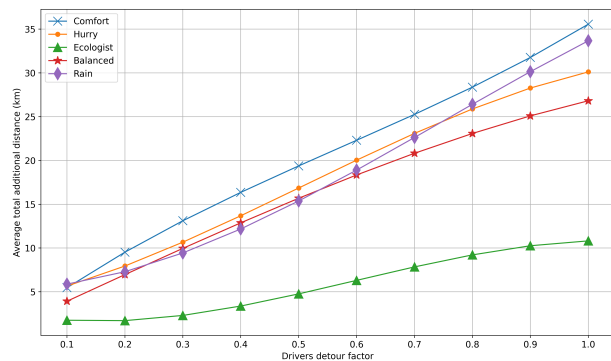


FIGURE 5 – Distance additionnelle moyenne de détour des conducteurs

5.3.2 L'impact du profil "confort"

Le profil "confort" (courbe bleue/croix) dans la Figure 6, vise à minimiser le nombre de transferts (changements de véhicules) pour atteindre la destination cible. On constate que ce profil est capable de minimiser sa cible de plus en plus (environ de 1.4 à 1.1 véhicule impliqué en moyenne) avec l'augmentation du facteur de détournement des conducteurs. Il est également bien meilleur que les autres profils en montrant entre 20% et 70% de performance supplémentaire sur cet objectif. En revanche, il est le plus faible pour la distance de détournement supplémentaire (Figure 5) et est similaire à la majorité des autres profils pour les autres objectifs (Figures 7 et 8). Nous pouvons conclure ici que la réduction du nombre de transferts entre véhicules et la favorisation du covoiturage simple-saut augmente la distance de détournement.

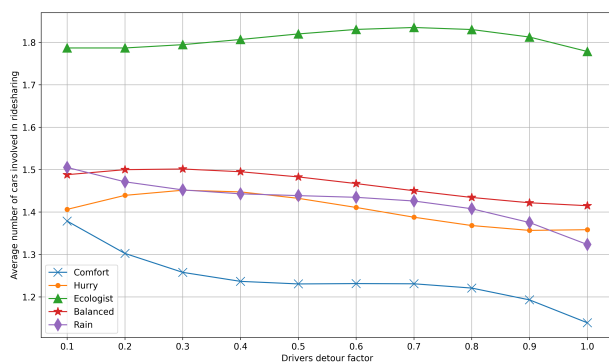


FIGURE 6 – Nombre moyen de véhicules impliqués dans le covoiturage

5.3.3 L'impact des profils "pluie", "pressé" et "équilibré"

Nous avons choisi de traiter ces trois profils conjointement, car, dans la majorité des cas, la tendance des courbes et leurs résultats sont similaires. Cela s'explique par la difficulté de dissocier le temps de parcours, le temps d'attente et la distance de parcours. Ces derniers sont en effet étroitement liés et interdépendants. Néanmoins, le profil "pluie" (courbe mauve/losanges) réussit à réduire son temps d'attente (Figure 7) quand les conducteurs sont peu disposés à effectuer des détours (facteur inférieur à 0.5). Il se confond ensuite avec les autres profils.

Le profil "Pressé", cherchant à arriver le plus rapidement possible, vise indirectement à réduire son temps d'attente. C'est pourquoi nous le voyons se positionner très légèrement comme le leader de la minimisation du temps d'attente sur la Figure 7 pour un facteur de détournement du conducteur de 0.5 à 0.9. Notre profil "équilibré" remplit parfaitement son objectif d'être le juste milieu entre les différents profils et leurs objectifs, comme on peut le voir sur la totalité des figures.

5.3.4 La performance globale des profils : le taux de service

Pour conclure sur l'analyse de ces résultats, nous pouvons considérer la performance des profils du point de vue du

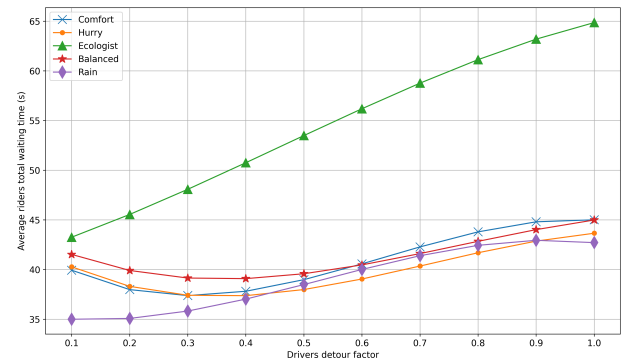


FIGURE 7 – Temps d'attente moyen d'un passager

taux de service moyen (Figure 8). Le taux de service représente la part des passagers qui ont réussi à se rendre à leur destination en covoiturant.

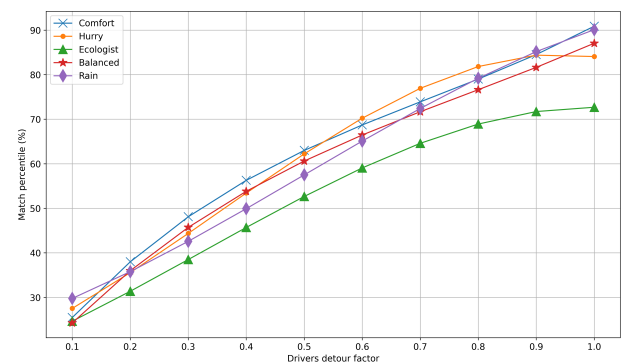


FIGURE 8 – Taux de service moyen du système

On constate tout de suite que plus le facteur de détournement augmente, plus il existe de solutions potentielles et donc plus le taux de service augmente. Ce phénomène est normal et démontre le bon fonctionnement de notre modèle. Nous remarquons que le profil "écologiste" (triangles verts) présente un taux de service relativement faible par rapport au reste des profils avec une différence moyenne de 5 à 15%. Aussi, il représente le pire taux de service atteint avec environ 72% pour un facteur de détournement des conducteurs de 1. Si on émet l'hypothèse que les utilisateurs qui souhaitent être passagers ont finalement pris leur véhicule pour rejoindre leur destination, il se peut que ce profil ne permette pas de réduire, autant qu'espéré, les émissions globales du système, curieux paradoxe !

6 Conclusion et perspectives

Dans cet article, nous proposons un système multi-agent pour optimiser le covoiturage dynamique en fonction des préférences des conducteurs et des passagers. Nous modélisons les différents acteurs d'un système de covoiturage et leur spectre de comportements possibles. Nous apportons une approche originale, basée sur des techniques d'indexation spatiale, pour modéliser la perception de nos agents et pour réduire le coût computationnel associé à la

recherche du covoiturage optimal. Les expériences présentées ont montré que l'approche proposée permet une optimisation individuelle et efficace.

Cette approche révèle certaines limitations pour les profils favorisant le temps d'attente et le temps de trajet. En effet, l'optimisation par agrégation des pondérations semble insuffisante lorsque les objectifs sont interdépendants. Pour répondre à ce constat, d'autres techniques peuvent être envisagées, comme l'agrégation par l'intégrale de Choquet, qui s'est avérée efficace pour l'optimisation multicritère [4]. Une autre perspective consiste à rendre le comportement des agents dynamique. En effet, le comportement des agents est pour l'instant défini à leur apparition et ne change pas au cours du temps. En fonction de l'état de leur environnement (trafic, météo, pénurie d'offres ou de demandes, etc.), leur comportement pourrait évoluer et les mener à changer leurs objectifs principaux [2]. En étendant notre approche à des agents au comportement dynamique, nous nous rapprocherions encore davantage du monde réel et de ses caractéristiques.

Des critères et mécanismes tels que le nombre d'arrêts maximum souhaité par un conducteur ou encore la possibilité de réserver un seul covoiturage pour plusieurs personnes sont essentiels dans le cadre d'une mise en application de notre solution. Il seront inclus et étudiés dans nos prochains articles.

Enfin, il serait fructueux de considérer également la complémentarité avec d'autres modalités de transport, en particulier les transports publics tels que le bus et le métro [20, 11]. Nous passerions alors de la mono-modalité à la comodalité, c'est-à-dire la multimodalité associée au covoiturage.

Références

- [1] Niels Agatz, Alan L. Erera, Martin W.P. Savelsbergh, and Xing Wang. Dynamic Ride-Sharing : A Simulation Study in Metro Atlanta. *Procedia - Social and Behavioral Sciences*, 17 :532–550, 2011.
- [2] Flavien Balbo and Suzanne Pinson. Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *Decision Support Systems*, 41(1) :131–146, 2005.
- [3] H. A. N. C. Bandara and Dileeka Dias. A multi-agent system for dynamic ride sharing. In *2009 International Conference on Industrial and Information Systems (ICIIS)*, page 199–203. IEEE, Dec 2009.
- [4] Sondes Ben Cheikh-Graïet, Mariagrazia Dotoli, and Slim Hammadi. A Tabu Search Based Metaheuristic for Dynamic Carpooling Optimization. *Computers & Industrial Engineering*, February 2020.
- [5] Geoff Boeing. OSMnx : New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65 :126–139, September 2017.
- [6] Zhongpu Chen, Bin Yao, Zhi-Jie Wang, Xiaofeng Gao, Shuo Shang, Shuai Ma, and Minyi Guo. Flexible Aggregate Nearest Neighbor Queries and its Keyword-Aware Variant on Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.
- [7] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem : Models and algorithms. *Ann Oper Res*, page 18, 2007.
- [8] Cristián Cortés, Martín Matamala, and Claudio Contardo. The pickup and delivery problem with transfers. *European Journal of Operational Research*, 200(3) :711–724, February 2010.
- [9] Alaa Daoud, Flavien Balbo, Paolo Gianessi, and Gauthier Picard. Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande. In *JFSMA29*, pages 127–136. Cépaduès, 2021.
- [10] A. Di Febbraro, E. Gattorna, and N. Sacco. Optimization of Dynamic Ridesharing Systems. *Transportation Research Record : Journal of the Transportation Research Board*, 2359(1) :44–50, January 2013.
- [11] Azise Oumar Diallo, Guillaume Lozenguez, Arnaud Doniec, and René Mandiau. Usage des parkings relais dans les comportements de déplacements intermodaux. In *JFSMA29, June 28-30, 2021*, pages 83–92. Cépaduès, 2021.
- [12] Jacques Ferber. Multi-agent systems : An introduction to distributed artificial intelligence, 1999.
- [13] Corwin Fevre, Hayfa Zgaya-Biau, Philippe Mathieu, and Slim Hammadi. Multi-agent Systems and R-Trees for Dynamic and Optimised Ridesharing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1352–1358, October 2021.
- [14] Corwin Fevre, Hayfa Zgaya-Biau, Philippe Mathieu, and Slim Hammadi. Preferential Optimization of Multi-hop Dynamic Ridesharing based on R-Trees and Multi-Agent Systems. *Under review*, 2022.
- [15] Antonin Guttmann. *R Trees : A Dynamic Index Structure for Spatial Searching*, volume 14. January 1984.
- [16] Wesam Herbawi and Michael Weber. Evolutionary Multiobjective Route Planning in Dynamic Multi-hop Ridesharing. In *Evolutionary Computation in Combinatorial Optimization*, pages 84–95, Berlin, Heidelberg, 2011. Springer.
- [17] Wesam Herbawi and Michael Weber. Modeling the Multihop Ridematching Problem with Time Windows and Solving It Using Genetic Algorithms. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 1 :96, November 2012.
- [18] Karama Jeribi, Hinda Mejri, Hayfa Zgaya, and Slim Hammadi. Distributed graphs for solving co-modal transport problems. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1150–1155, October 2011.
- [19] Ece Kamar and Eric Horvitz. Collaboration and Shared Plans in the Open World : Studies of Ridesharing. In *IJCAI International Joint Conference on Artificial Intelligence*, page 187, January 2009.
- [20] Jane Lin, Sandeep Sasidharan, Shuo Ma, and Ouri Wolfson. A Model of Multimodal Ridesharing and Its Analysis. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, pages 164–173, Porto, June 2016. IEEE.
- [21] Shuo Ma and Ouri Wolfson. Analysis and evaluation of the slugging form of ridesharing. In *SIGSPATIAL'13*, pages 64–73, Orlando, Florida, 2013. ACM Press.

- [22] Yannis Manolopoulos, Yannis Theodoridis, and Vassilis J. Tsotras. Spatial Indexing Techniques. In *Encyclopedia of Database Systems*, pages 1–7. Springer New York, New York, NY, 2014.
- [23] Neda Masoud and R. Jayakrishnan. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. *Transportation Research Part B : Methodological*, 106 :218–236, December 2017.
- [24] Philippe Mathieu and Antoine Nongaillard. Effective evaluation of autonomous taxi fleets. In *ICAART (1)*, pages 297–304, 2018.
- [25] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems*, 30(2) :529–576, June 2005.
- [26] Dimitris Papadias, Jun Zhang, Nikos Mamoulis, and Yufei Tao. Query Processing in Spatial Network Databases. In *Proceedings 2003 VLDB Conference*, pages 802–813. Elsevier, 2003.
- [27] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1964.
- [28] Shuo Ma, Yu Zheng, and O. Wolfson. T-share : A large-scale dynamic taxi ridesharing service. In *ICDE29'*, pages 410–421, Brisbane, QLD, April 2013. IEEE.
- [29] Amirmahdi Tafreshian and Neda Masoud. Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C : Emerging Technologies*, 114 :532–553, May 2020.
- [30] Yixin Xu, Lars Kulik, Renata Borovica-Gajic, Abdullah Aldwyish, and Jianzhong Qi. Highly Efficient and Scalable Multi-hop Ride-sharing. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 215–226, Seattle WA USA, November 2020. ACM.
- [31] Man Yiu, Nikos Mamoulis, and Dimitris Papadias. Aggregate nearest neighbor queries in road networks. *IEEE Transactions on Knowledge and Data Engineering*, 17 :820–833, July 2005.

Champ neuronal et apprentissage profond de topologies pour la fusion multimodale

S. Forest^{1,2}, J.-C. Quinton¹, M. Lefort²

¹ Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, UMR 5224, F-38000, Grenoble, France

² Univ. Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR 5205, F-69622, Villeurbanne, France

{simon.forest, quintonj}@univ-grenoble-alpes.fr, mathieu.lefort@univ-lyon1.fr

Résumé

Des agents artificiels tels que des robots ont souvent intérêt à fusionner des données issues de différentes modalités. Pour cela, il peut être pertinent de prendre en compte les variations dans la structure et la résolution des topologies sous-jacentes aux espaces sensoriels. Nous proposons d'utiliser un champ neuronal dynamique pour sélectionner des stimuli dans un contexte multimodal. Nous avons adapté le modèle à des topologies apprises (par des gaz neuronaux croissants notamment) pour la fusion, nous l'étendons maintenant en insérant un auto-encodeur pour réduire la dimensionnalité des données d'entrée.

Mots-clés

Fusion multimodale, champ neuronal dynamique, apprentissage de variétés, auto-encodeur, gaz neuronal croissant

Abstract

Artificial agents such as robots can often benefit from merging data from different modalities. For this purpose, it may be relevant to take into account the variations in the structure and resolution of the topologies underlying the sensory spaces. We propose to use a dynamic neural field to select stimuli in a multimodal context. We had adapted the model to learned topologies (with growing neural gas in particular) for fusion, we now extend it by inserting an auto-encoder to reduce the dimensionality of the input data.

Keywords

Multimodal fusion, dynamic neural field, manifold learning, auto-encoder, growing neural gas

1 Introduction

Quand on parle de traitement de l'information et de prise de décision comportementale, la façon dont on fusionne les données issues de différentes sources n'est pas à négliger. Prenons un exemple : un robot a la tâche de trouver et atteindre un réveil lorsqu'il se met à sonner. Au début, le robot peut faire face à plusieurs objets ressemblant à un réveil, qu'il ne devrait avoir aucune difficulté à distinguer. Lorsqu'une sonnerie se fait entendre, le robot devrait être capable de localiser son origine, mais avec généralement une faible précision. Avant d'entreprendre une action, le

robot doit sélectionner un objet. Dans ce cas, il s'agit de l'objet, parmi ceux qui ressemblent à un réveil, qui coïncide le plus avec la localisation de la source sonore. Mais la manière dont les modalités visuelle et auditive doivent être pondérées dépend non seulement de la tâche (une horloge visible de face est moins importante qu'un son provenant d'un côté), mais aussi de la fiabilité des capteurs (la réverbération de la pièce peut rendre l'orientation du son moins décisive).

La tâche dans cet exemple est confrontée à de multiples défis, notamment la fusion de modalités sensorielles de disponibilité et de fiabilité différentes, et la sélection de (et l'attention vers) la cible. Pour résoudre ces problèmes, beaucoup de modèles actuels se basent exclusivement sur l'apprentissage profond. Dans cet article, nous prenons l'apprentissage profond comme un outil de pré-traitement de données et de réduction de dimension. Pour la fusion, nous nous reposons sur une autre approche à partir d'un champ neuronal dynamique (*dynamic neural field*, DNF), un modèle bio-inspiré d'activité neuronale [2]. Il s'agit d'un réseau récurrent en temps continu placé dans une topologie, où les poids sont connus et dépendent de la distance entre les neurones. Avec un mélange d'excitation à courte portée et d'inhibition à longue portée, les stimuli d'entrée sont mis en compétition jusqu'à ce qu'une bulle d'activité émerge, qui peut être interprétée comme une décision, décentralisée et dynamique, de sélection d'une cible et/ou d'action. De plus, la dynamique donne un lissage temporel à la bulle d'activité, malgré les fluctuations des entrées et les distracteurs potentiels. Le DNF a connu diverses applications, notamment en robotique [14, 34, 39]. En particulier, les propriétés d'interaction du DNF le rendent très adapté à la fusion multimodale [11, 35].

Les premières implémentations de DNF trouvent une limite dans la nature de la variété (*manifold* en anglais) sur laquelle elles évoluent. La plupart des applications de la littérature supposent l'existence d'une topologie régulière sous-jacente, le plus souvent 1D ou 2D [36]. Mais elle n'est guère représentative des disparités de l'espace sensoriel, disparités qui deviennent cruciales lors de la fusion multimodale. En effet, intéressons-nous à la forme des stimuli perçus dans l'environnement. La quantité d'informations disponibles est énorme, et les données qu'un agent reçoit

de ses capteurs n'en sont qu'une projection dans quelques dimensions données. Équipé d'une caméra standard, un robot recevra une projection en 2D de la partie de l'environnement à laquelle il fait face. Avec un seul microphone, il peut détecter des sons provenant de n'importe où autour de lui, mais il peut difficilement les localiser. Deux microphones peuvent permettre une certaine localisation sonore 1D le long de l'axe sur lequel ils sont alignés, généralement azimutal (grâce à la différence de temps ou d'intensité interaurale), et même un peu de 2D ou 3D en exploitant la forme des pavillons des oreilles, avec une fonction de transfert liée à la tête (*head-related transfer function*, HRTF) [4]. Nous devons d'abord tenir compte des spécificités de chaque modalité sensorielle avant de créer des comportements qui l'exploitent au mieux. De plus, nous devons trouver un moyen de faire correspondre des informations complémentaires provenant de différentes modalités, ce qui revient généralement à projeter des stimuli sur une variété commune.

Dans [10], nous avons proposé une manière d'adapter le DNF à des variétés de dimensionnalité et structure arbitraires. À l'aide de gaz neuronaux croissants (*growing neural gas*, GNG), une topologie multimodale est créée à l'intérieur de laquelle le DNF peut fusionner et sélectionner des stimuli. Les applications testées dans ce précédent article étaient limitées par la faible capacité d'apprentissage du GNG : s'il permet de faire ressortir un espace sous-jacent dans des données de plus grande dimension, il ne permet pas de réaliser des tâches plus complexes dans des espaces de très grande dimension. Localiser un stimulus visuel à partir d'une photographie, ou un stimulus auditif à partir d'un enregistrement audio brut, n'est pas possible avec un GNG seul. Il est nécessaire au préalable de réduire la dimensionnalité des entrées en apprenant des projections vers un espace plus facile à exploiter. Cette solution peut être apportée par des réseaux de neurones. Dans cet article, nous étendons la précédente contribution en ajoutant une méthode d'apprentissage profond de variétés, à savoir un auto-encodeur de Wasserstein en coupes (*sliced Wasserstein auto-encoder*, SWAE), en amont du modèle. Notre objectif est de vérifier si les propriétés du modèle précédent se maintiennent lorsqu'on ajoute une opération visant à réduire les dimensions de l'espace d'entrée, au risque de dégrader la réelle topologie sous-jacente pendant l'apprentissage de l'encodeur.

Cet article est structuré comme suit. Dans la section 2, nous présentons des travaux existants sur l'apprentissage de variétés et le DNF, et en particulier leurs applications à la fusion multimodale. Puis nous décrivons notre modèle complet dans la section 3, et montrons sa robustesse, ses performances et ses propriétés dans la section 4. Nous concluons et ajoutons des perspectives additionnelles dans la section 5.

2 Travaux existants

2.1 Apprentissage de variétés

Les capteurs fournissent des échantillons de haute dimension de l'environnement, mais les espaces sensoriels cor-

respondent souvent à des variétés de dimension inférieure. L'apprentissage profond est particulièrement adapté à la génération de telles variétés (voir [5] pour une revue). Par exemple, il a été démontré que les dernières couches d'un réseau neuronal profond contiennent une dimensionnalité intrinsèque inférieure au nombre de descripteurs dans les données [3]. Des méthodes spécifiques telles que les auto-encodeurs variationnels [16] peuvent apprendre une structure sous-jacente de manière non supervisée. D'autres types d'auto-encodeurs existent, notamment le SWAE [18]. Ce dernier utilise dans l'apprentissage une distance de Wasserstein, qui compare la distribution des données encodées dans l'espace latent à une distribution choisie. On peut ainsi imposer, par exemple, que l'espace latent suive une distribution uniforme en 2D. Il est ensuite possible d'exploiter les propriétés géométriques de la topologie ainsi encodée [19].

Une approche moins contraignante, privilégiée dans notre précédente contribution [10], repose sur les méthodes d'auto-organisation. Dans les cartes auto-organisatrices (*self-organizing maps*, SOM), par exemple le modèle de Kohonen [17], chaque neurone représente une entrée prototypique dans l'espace sensoriel à haute dimension, de sorte que l'espace d'entrée est projeté sur un treillis neuronal de forme et de taille fixes. Dans le cas du gaz neuronal (*neural gas*, NG), les neurones ne sont pas disposés sur un treillis, mais sont connectés selon une règle Hebbienne, de sorte que les neurones de prototypes proches sont reliés entre eux [25]. Le gaz finit par remplir l'espace d'entrée, d'une manière qui imite la distribution des stimuli. Le gaz neuronal croissant (*growing neural gas*, GNG) [12] est un dérivé du NG, dans lequel des neurones sont ajoutés (ou supprimés) au fur et à mesure jusqu'à ce qu'une condition spécifique soit remplie, s'adaptant ainsi à la topologie indéterminée de l'espace d'entrée.

Variétés en fusion multimodale

De multiples articles ont montré des résultats prometteurs en fusion multimodale à l'aide d'apprentissage profond. L'apprentissage profond non supervisé peut être utilisé pour projeter des données multimodales sur une variété de faible dimension pour une utilisation en robotique [8, 21]. Les entrées peuvent être mélangées pendant l'entraînement du réseau de neurones pour exploiter les corrélations entre les modalités [40, 43]. Récemment, des extensions du réseau Transformer ont été proposées, permettant de recevoir des entrées multimodales pondérées par un module d'attention [15, 28]. Cependant, la plupart de ces travaux partent du principe que toutes les données multimodales sont corrélées entre elles. De plus, les architectures profondes sont dédiées à une tâche spécifique et aucun paradigme générique n'émerge [29].

Nous cherchons à créer une nouvelle topologie multimodale sur laquelle de nouvelles propriétés dynamiques pourraient être appliquées. Dans un premier temps, l'auto-organisation offre des solutions pour un coût bien moindre [13, 20, 22, 27, 31, 41]. Les SOM et leurs dérivés sont utilisés depuis longtemps comme modèles de fusion multimodale,

mais les façons de combiner les modalités peuvent être très diverses. Les architectures composées de SOM peuvent être divisées en deux catégories. Dans la première, une SOM est formée pour chaque modalité, puis toutes les cartes unimodales sont connectées en fonction d'une règle d'apprentissage spéciale [13, 22]. Dans la seconde, les cartes unimodales sont reliées à une nouvelle SOM [20, 27] ou un NG [41] multimodal qui combine toutes les informations. Des couches supplémentaires de SOM peuvent également être envisagées pour créer un flux hiérarchisé d'informations [31]. De plus, les modèles peuvent être rendus plus adaptatifs aux tâches dépendantes du temps à l'aide de cartes « croissantes au besoin » [31], une alternative au GNG conçue pour les distributions d'entrées dynamiques [24]. Certains de ces modèles ont déjà été testés pour des modalités visuelles, auditives et/ou proprioceptives sur des robots [13, 20].

2.2 Champ neuronal dynamique

Après l'apprentissage de cartes multimodales et/ou de cartes unimodales interconnectées, nous avons besoin d'un paradigme pour dicter la manière dont la perception va se produire. La perception multimodale peut être considérée comme une forme de décision prenant en compte des entrées sensorielles de fiabilité et de pertinence variables. Nous suivons le choix d'architecture fait dans [27] et [22], où le champ neuronal dynamique (DNF) est utilisé comme paradigme qui régit la fusion ou la ségrégation des stimuli dans l'espace topologique multimodal. Le DNF a de nombreuses propriétés utiles pour la perception multimodale.

Originaire du domaine des neurosciences, le DNF a diverses applications en robotique [36]. Par exemple, l'attention visuelle peut être cumulée avec un contrôle moteur pour qu'un robot fixe de manière autonome les objets de son environnement et apprenne une carte sensori-motrice [34]. Le DNF repose sur une population d'unités connectées topologiquement, à une échelle mésoscopique, où l'activité apparente (ou potentiel membranaire moyenné sur des groupes de neurones) peut être lue pour interpréter des décisions à un niveau comportemental. L'activité évolue dans le temps en fonction d'une somme de stimulations externes et d'interactions latérales entre les neurones. Les neurones stimulés vont envoyer une forte excitation à leurs voisins les plus proches, et une inhibition modérée à leurs voisins plus éloignés, conduisant à l'émergence d'une bulle d'activité stable. En fonction du paramétrage, cela peut conduire à plusieurs types de comportements [36]. Avec une forte excitation locale, la bulle peut être auto-entretenu, agissant comme une mémoire à long terme [34]. L'inhibition à longue portée créera une compétition entre les stimuli conflictuels, jusqu'à ce que l'un d'entre eux domine les autres, ou qu'ils soient fusionnés en une seule bulle à une position interpolée [11, 38]. Ensuite, la bulle auto-entretenu peut être utilisée pour une attention sélective robuste, capable d'ignorer le bruit et les faibles distracteurs [9]. Enfin, la sortie du DNF peut être directement exploitée pour générer une commande motrice [33, 34].

Les propriétés du DNF peuvent être très utiles à la fusion

multimodale. Il fournit des moyens non seulement pour améliorer la robustesse des décisions lorsque les modalités sont congruentes [35], mais aussi pour résoudre les conflits entre modalités [11]. C'est là que le choix de la variété sous-jacente peut être très important, car tout le processus de décision se base dessus, et sa structure peut déterminer entre autres la fiabilité des stimuli, qui peut avoir une influence sur la sélection.

La grande majorité des travaux utilisant le DNF supposent que la dynamique se déroule sur une topologie complètement régulière, par exemple un treillis 2D dans le cas de la vision. Cependant, il n'existe pas de moyen clair de projeter deux modalités ou plus sur le même treillis. Dans [35] et [11], des hypothèses fortes sont faites sur la forme des stimuli dans une modalité afin qu'ils s'intègrent dans la topologie de l'autre. Pour résoudre ce problème, [22] propose d'utiliser des variétés distinctes pour chaque modalité, chacune apprise par une SOM, et d'appliquer un DNF sur chacun d'eux. La communication entre les modalités est assurée par un ensemble de connexions topographiques.

Une contribution intéressante de cette dernière référence est l'utilisation d'une variété apprise comme siège de la dynamique neuronale. Par ailleurs, certaines tentatives visant à modifier la projection des entrées dans la variété ont donné des résultats satisfaisants : [38] et [11] ont ainsi reproduit des comportements biologiques après avoir appliqué aux stimuli visuels une transformation logpolaire, qui modélise les variations de résolution de la rétine humaine [30]. Dans le cas de [22], les projections reçues par les neurones sont modifiées, bien qu'elles soient toujours organisées en un réseau rectangulaire. Étant donné que le DNF est fortement dépendant de la topologie et qu'il repose sur un noyau d'interaction symétrique¹, rompre la régularité de la topologie sous-jacente doit être fait avec prudence. Une démonstration de ceci a été faite dans [10], où le DNF est adapté à des variétés de forme et de dimension non contraintes, avec des résultats probants. Cependant, les topologies qui y sont présentées sont relativement simples, et le DNF n'a pas encore été testé sur des espaces réduits par apprentissage profond, qui risquent d'être beaucoup plus irréguliers. C'est l'objet de cet article.

3 Méthodes

Cet article reprend en partie les méthodes développées dans [10], où nous utilisons des GNG pour apprendre des variétés de l'espace sensoriel dans chaque modalité (sous-section 3.3), avant de les assembler en un graphe multimodal (3.4), puis d'utiliser un DNF pour créer des comportements dans cette nouvelle topologie (3.5). Nous ajoutons une étape préalable d'apprentissage d'un auto-encodeur, dont l'encodeur servira à réduire la dimension des données d'entrées avant l'apprentissage des graphes (3.2), et le décodeur servira à l'analyse des sorties du DNF (3.6). Les

1. Il a été proposé de rompre la symétrie du côté du DNF, soit par des noyaux asymétriques [6], soit par des distorsions de la topologie à l'aide de renforcements prédictifs [32], mais les deux requièrent une étape d'apprentissage supplémentaire.

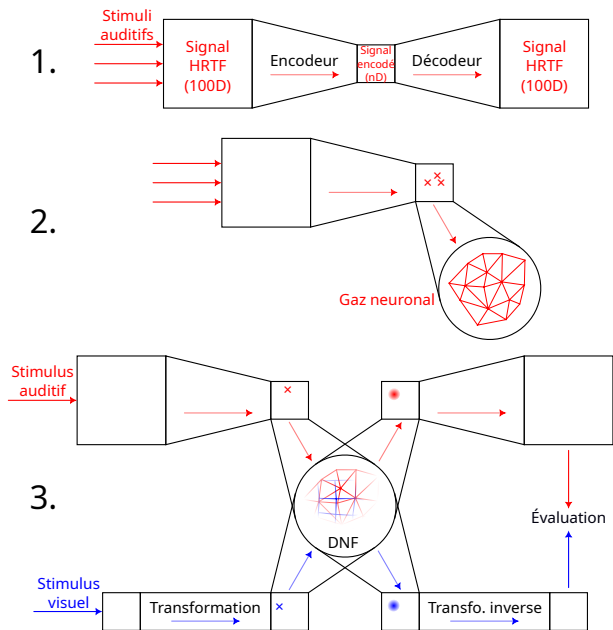


FIGURE 1 – Résumé des principales étapes de cette contribution. **1.** Apprentissage de l’auto-encodeur (sous-section 3.2). **2.** Apprentissage des graphes topologiques (sous-sections 3.3 et 3.4). **3.** Stimulation, émergence d’une activité dans le graphe bimodal et évaluation (sous-sections 3.5 et 3.6).

grandes étapes de cette contribution sont synthétisées dans la figure 1.

3.1 Données

Pour cette contribution, nous nous concentrons sur une architecture bimodale, même si ces travaux pourraient également être applicables à trois modalités ou plus. Nous prenons l’exemple d’un robot devant localiser un signal (e.g., la position d’une personne qui l’appelle) en recevant simultanément des données visuelles (détection de visage) et auditives (spectre de fréquences sonores).

Audition. Une façon de localiser les sources sonores pour les robots consiste à calculer une HRTF, une fonction qui associe des caractéristiques fréquentielles (causées par les interférences de la tête et des pavillons des oreilles sur le signal) à l’orientation de l’origine du signal [4]. Les données procurées par [1] comprennent les réponses enregistrées par un robot équipé de pavillons artificiels, à un son émis depuis différents angles. Étant donnée la position d’un stimulus externe en 2D, nous pouvons interpoler les réponses reçues par les deux oreilles robotiques. Nous calculons ensuite leur transformée de Fourier et faisons la différence entre les oreilles pour obtenir une HRTF. À la fin, chaque entrée audio est à 100 dimensions.

Cette HRTF 100D porte implicitement les informations 2D de la localisation du signal dans le référentiel de la tête du robot : azimut et élévation. C’est le signal 100D que nous donnerons en entrée de l’auto-encodeur.

Vision. La vision artificielle a très généralement une résolution homogène. Cependant, nous pouvons concevoir des cas où la perception visuelle n’est pas parfaitement régulière, par exemple à cause d’une tâche sur l’objectif de la caméra. Nous n’ajoutons pas d’étape d’apprentissage pour la vision ici. Mais, pour tester la robustesse de la fusion dans des modalités de résolution changeante, nous traitons les stimuli visuels comme des points dans un espace 2D, auxquels nous appliquons une transformation bio-inspirée, pour rester dans le même niveau de réalisme que la HRTF. Nous modifions donc l’espace visuel par une transformation logpolaire. Cette transformation a originellement été utilisée pour décrire chez l’humain la façon dont un stimulus capté par la rétine est projeté sur le colliculus supérieur, une région du cerveau impliquée dans la génération de mouvements oculaires [30]. Elle permet notamment de reconstituer la différence de résolution entre le centre de la rétine et sa périphérie. Elle a déjà été appliquée à des systèmes artificiels, par exemple pour améliorer le contrôle du regard chez les robots [23], ou pour renforcer l’apprentissage d’un réseau de neurones sur des données visuelles [7]. Et elle a déjà été couplée avec un DNF [11, 38].

3.2 Encodage

La réduction de données est effectuée par un auto-encodeur. Nous utilisons en particulier un SWAE (*sliced Wasserstein auto-encoder*), car il permet de former un espace latent euclidien muni d’une distance directement exploitable par le GNG. Il est entraîné sur les données HRTF 100D, avec un espace latent 2D, 5D ou 20D, dans lequel les entrées encodées doivent suivre une distribution uniforme. L’encodeur et le décodeur sont faits d’un réseau de neurones entièrement connecté. Les couches cachées sont séparées par un ReLU pour l’encodeur, et un LeakyReLU pour le décodeur. Le nombre de neurones dans chaque couche est listé dans la table 1.

TABLE 1 – Taille des couches de neurones des encodeurs de SWAE. Les mêmes hyper-paramètres sont utilisés pour le décodeur, en sens inverse.

Entrée	Couches cachées			Sortie
100	90	70	50	20
100		50	20	5
100	64	32	16	2

3.3 Topologies unimodales

L’objectif de cette étape est de créer des graphes représentatifs de l’espace latent de chaque modalité. Ces graphes seront ensuite fusionnés pour servir de support à la prise de décision dans un environnement multisensoriel.

Le choix de l’auto-encodeur peut permettre d’imposer la structure de l’espace latent et la distribution des stimuli encodés dans cet espace. Connaissant ceci, il serait aisé d’échantillonner la distribution par un ensemble de nœuds, et de les connecter en fonction de la structure de la topologie sous-jacente. Il n’est pas indispensable d’ajouter une

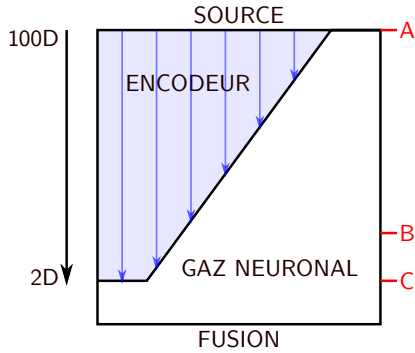


FIGURE 2 – Pré-traitement des données de HRTF avant la fusion. Plusieurs degrés de compression par l’encodeur sont possibles avant la création d’un graphe (ici, par gaz neuronal) utilisé pour la fusion et la prise de décision. Nous testons différents niveaux dans cet article. A : Pas d’encodage. C : Réduction maximale du nombre de dimensions de l’espace d’entrée. B : Réduction intermédiaire, par exemple 20D ou 5D.

étape d’apprentissage. Cependant, nous reprenons le choix de [10] de créer ces graphes à l’aide de GNG. L’algorithme en question n’introduit pas d’hypothèse supplémentaire sur la structure de l’espace latent. Il a l’avantage de fonctionner à la fois sans et avec auto-encodeur, pouvant aussi apprendre une structure sous-jacente de l’espace latent qui diffère de celle imposée dans l’apprentissage du SWAE.

L’algorithme complet du GNG est décrit dans [12]. Pour résumer, le GNG se forme à partir de stimuli tirés aléatoirement. À chaque fois, les deux nœuds dont l’entrée prototypique correspond le mieux au stimulus sont connectés entre eux. Ensuite, l’unité qui correspond le mieux (*best-matching unit*, BMU) et ses voisins topologiques directs voient leur prototype déplacé vers le stimulus. Les connexions qui n’ont pas été mises à jour depuis longtemps sont supprimées, de même que les nœuds isolés. Puis, à intervalles fixes, un nouveau nœud est inséré². Son entrée prototypique va au milieu de la connexion la plus activée.

Cette étape diffère de l’article précédent dans la mesure où le GNG peut recevoir des données plus ou moins compressées par l’encodeur (figure 2). Dans le cas des signaux auditifs, le GNG peut opérer aussi bien sur des HRTF brutes en 100 dimensions que sur un encodage en 2D dans le cas le plus extrême. Pour les signaux visuels, l’encodage est remplacé par un pré-traitement manuel.

3.4 Topologie multimodale

Cette étape est directement reprise de [10] : nous créons un nouveau graphe bimodal qui contient tous les nœuds et arêtes d’une modalité et de l’autre. Pour créer de nouvelles arêtes bimodales, nous connectons des neurones de chaque modalité qui s’activent ensemble, d’une façon inspirée d’un apprentissage Hebbien. Plus précisément, nous tirons une entrée multimodale aléatoire, et si elle se trouve

2. Dans cet article, le nombre de nœuds est plafonné à 500 dans chaque modalité.

à portée des deux modalités, nous récupérons la BMU de chaque GNG et nous les connectons. Un seul changement est fait par rapport à l’article précédent : afin d’éviter que des nœuds trop épars dans un des GNG se connectent à énormément de nœuds de l’autre modalité, le nombre de connexions intermodales est limité à deux par nœud.

3.5 Sélection d’une activité

L’adaptation du DNF aux graphes créés précédemment est une des contributions centrales de [10]. Nous la récapitulons dans cette sous-section.

Une fois le graphe bimodal créé, les neurones qui lui sont associés peuvent être stimulés par des entrées sensorielles (via leur modalité respective), et nous pouvons utiliser un DNF pour sélectionner un stimulus. Le DNF s’exprime généralement sous la forme d’une équation intégrodifférentielle dans un champ continu de neurones, qui est ensuite discrétisée et calculée par la méthode d’Euler (voir eq. 2). L’intégration d’un DNF est comparable à la simulation de réseaux de neurones récurrents en temps continu. Dans le DNF, la distance entre les neurones joue un rôle important, car elle détermine s’ils vont s’exciter ou s’inhiber mutuellement. Notre modèle diffère des autres modèles de la littérature dans la mesure où tous les neurones ne partagent pas un système de coordonnées commun. Nous devons donc adapter l’équation du DNF, afin que les distances soient définies sur le graphe, et seulement cela. Nous nous basons sur la distance standard de la théorie des graphes, c’est-à-dire le nombre d’arêtes sur le chemin le plus court entre deux sommets quelconques [42].

Dans notre modèle, chaque neurone est lié à une modalité spécifique. Ainsi, l’entrée externe reçue individuellement sera spécifique à la modalité (bien que le reste des opérations du DNF ne le soit pas). Pour s’assurer que la quantité totale de stimulation externe soit indépendante de la résolution locale d’une modalité, nous rangeons tous les neurones d’une modalité par ordre de proximité au stimulus (en utilisant la distance euclidienne dans le système de coordonnées de cette modalité), et les excitons en fonction de leur rang par ordre décroissant. Pour chaque neurone indexé k , étant donné un stimulus indexé i , on note $r_{k,i}$ le rang de proximité entre l’entrée prototypique de k et les coordonnées de i . La stimulation externe I_k reçue par k est donnée par :

$$I_k = \lambda_{m,i} e^{-\frac{r_{k,i}^2}{2\sigma_i^2}} \quad (1)$$

où $\lambda_{m,i}$ est l’intensité du stimulus i par rapport à la modalité m de k . Un neurone ne peut recevoir que des entrées externes provenant de sa propre modalité.

Ensuite, nous calculons l’évolution de l’activité dans le graphe au cours du temps. Ce qui suit est complètement indépendant de la modalité. Le potentiel U_k du neurone k est initialisé à 0 et mis à jour de façon incrémentale par³ :

3. Dans cette équation, seul U_k est incrémenté dans le temps, et les entrées I_k sont statiques. Cependant, aucune de nos hypothèses n’empêche les entrées d’évoluer au cours du temps. Nous faisons ce choix car les entrées dynamiques ne sont pas nécessaires pour les résultats présentés dans cet article. Sinon, l’équation (2) pourrait être réécrite en exprimant $U_k(t)$ comme une fonction de $U_*(t - \Delta t)$ et $I_k(t)$.

$$\Delta U_k = \frac{\Delta t}{\tau} \left(-U_k + I_k + \sum_{k'} W(\langle k, k' \rangle) f(U_{k'}) + h \right) \quad (2)$$

où Δt est le temps de simulation entre chaque étape, τ une constante de temps qui détermine la vitesse de mise à jour du DNF, f une fonction d'activation (ReLU), et h un potentiel de repos négatif. $\langle \cdot, \cdot \rangle$ désigne la distance minimale en nombre d'arêtes entre deux nœuds dans le gaz neuronal bimodal, et W est une fonction de poids exprimée comme suit :

$$W(\delta) = \lambda_+ e^{-\frac{\delta^2}{2\sigma_+^2}} - \lambda_- e^{-\frac{\delta^2}{2\sigma_-^2}} \quad (3)$$

avec des amplitudes $\lambda_+ > \lambda_- > 0$ et des largeurs $\sigma_+ < \sigma_-$. W peut être vu comme un noyau en forme de chapeau mexicain [2].

3.6 Évaluation

Une façon possible d'interpréter la décision est de lire la sortie $f(U)$. Il est courant de prendre un barycentre de la sortie comme estimateur de la position sélectionnée par le modèle. En l'occurrence, les coordonnées des nœuds sur lesquels le DNF évolue ne sont pas directement exploitables. Nous devons d'abord décoder ces coordonnées, soit en inversant la transformation logpolaire pour la modalité visuelle, soit en utilisant le décodeur appris précédemment pour la modalité auditive (figure 1, étape 3). Les HRTF décodées sont reliées à des coordonnées 2D par interpolation dans la base de données de signaux audio. La somme de toutes les coordonnées 2D pondérées par l'activation $f(U)$ donne la position perçue du signal.

À des fins d'évaluation, nous comparons la position perçue à la position réelle du stimulus. Ceci nous donne un indicateur de la précision du modèle de fusion, même si cette étape n'est pas indispensable à ce stade. Le modèle pourrait fonctionner sans que nous ne possédions de manière supervisée de replacer, pour chaque modalité, chaque nœud de l'espace latent dans un système de coordonnées 2D intelligible (azimut-élévation). En effet, si l'espace latent contient des représentations internes de l'espace de décision, une décision prise par le DNF pourrait être transformée directement en action sur l'environnement, sans décodage explicite. Le décodage n'est pas non plus nécessaire à la fusion. Nous le faisons donc principalement pour l'évaluation et la visualisation.

4 Résultats

4.1 Topologies apprises

Un graphe est créé à partir de données audio encodées par un SWAE (figure 3). Plusieurs niveaux de compression des dimensions ont été testés (voir figure 2) : pas d'encodage (A), une réduction intermédiaire (B) vers 20 ou 5 dimensions, ou une réduction maximale (C) vers 2 dimensions.

Le premier GNG est le même qu'obtenu dans [10] (figure 3, première ligne). Une fois replacé dans des coordonnées 2D, il paraît assez régulier. L'allongement apparent du graphe le

long de la direction azimutale est dû aux conditions matérielles de la perception des sons, qui rendent une discrimination gauche/droite plus facile à mener qu'une discrimination haut/bas. Une analyse plus poussée est proposée dans l'article cité.

En ajoutant un SWAE avec un espace latent imposé en 20 dimensions (figure 3, deuxième ligne), on constate que le GNG produit est très similaire au premier. Les données pertinentes dans notre évaluation (azimut et élévation) n'ont presque pas été dégradées. Cela signifie qu'il n'y a aucune perturbation à anticiper dans les propriétés du modèle de fusion, et ce, même après l'insertion d'un réseau de neurones qui transforme les données sans avoir connaissance des descripteurs les plus pertinents dans cette tâche.

Cependant, la conservation de ces propriétés n'est pas systématique en fonction de l'encodage employé. La forte dégradation des GNG 5D (figure 3, troisième ligne) et 2D (quatrième ligne) montre qu'une compression trop forte peut aboutir à une perte partielle des informations d'azimut et d'élévation. En effet, le SWAE n'a pas de raison de privilégier ces deux caractéristiques. Il sélectionne seulement les descripteurs qui permettent de mieux encoder les données en quelques dimensions et de les reconstruire. Dans les cas 5D et 2D, on voit que le GNG reste assez régulier en périphérie, et est particulièrement dégradé entre -40 et 40 degrés environ. C'est cohérent avec le fait que la localisation des sons est plus facile sur les côtés de la tête. Le SWAE a ainsi appris que la HRTF était mieux encodée par les informations de localisation sur les côtés, mais pas au centre du champ perceptif, où ces informations sont perdues au profit d'autres caractéristiques, moins pertinentes dans notre tâche, mais plus utiles pour la reconstruction des données par le décodeur.

Cette dégradation involontaire ne rend pas le nouveau GNG inexploitable, d'autant plus dans un contexte de fusion de données, où une autre modalité peut apporter des informations complémentaires. Nous le testons avec l'audition encodée en 2D (le cas le plus critique) et la vision. Comme le GNG 5D est qualitativement proche du GNG 2D, nous nous attendons à ce que les performances soient similaires avec ce nombre de dimensions. De même, nous n'attendons pas de différence dans le comportement du modèle entre des données encodées en 20D et des données non encodées, puisque les GNG obtenus sont qualitativement très proches.

4.2 Évaluation en deux dimensions

Pour cette expérience, nous apprenons un second GNG à partir de données visuelles 2D altérées par transformation logpolaire (figure 4). Comme prévu, la résolution est beaucoup plus élevée au centre du champ de vision⁴ qu'en périphérie. Ensuite, comme décrit en section 3.4, nous connectons ce GNG visuel au GNG auditif 2D pour former un nouveau graphe bimodal (figure 5).

Sur ce graphe multimodal, nous pouvons envoyer des stimuli audiovisuels et faire opérer un DNF pour sélection-

4. L'absence de nœuds autour d'un azimut de zéro est due à la présence d'un log dans la transformation. Le modèle sur lequel elle s'appuie est originellement prévu pour deux hémisphères disjoints.

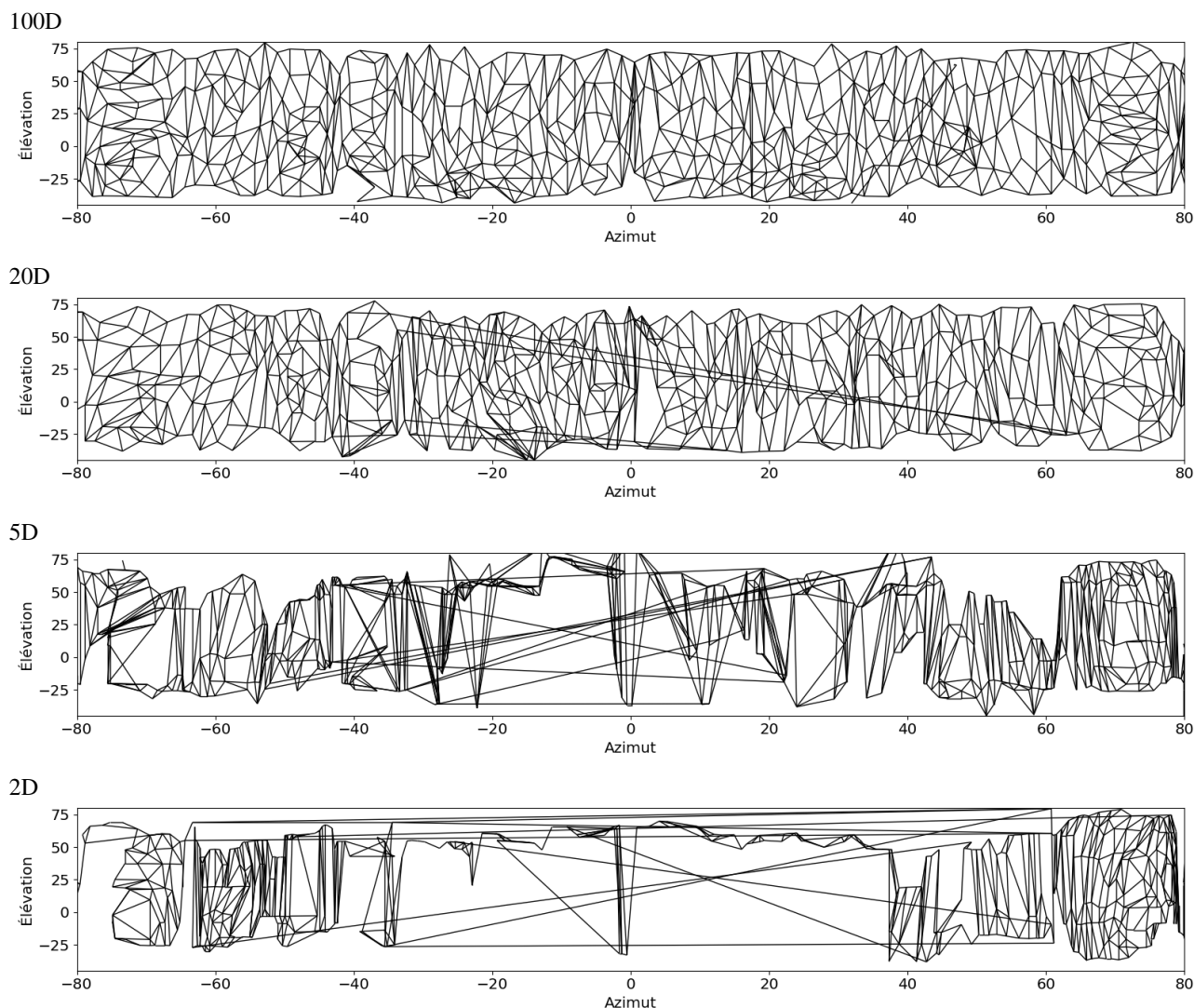


FIGURE 3 – GNG créés à partir de données audio. Le premier est appris à partir des données HRTF 100D sans encodage. Les autres sont appris à partir d’un encodage vers un espace latent 20D, 5D ou 2D. L’affichage des nœuds en 2D est réalisé en décodant leurs entrées prototypiques avec le SWAE le cas échéant, puis en interpolant les azimut et élévation d’après la base de données de HRTF. Notez que dans cet affichage, les deux axes n’ont pas la même échelle.

ner une localisation. Nous évaluons la distance entre la position trouvée et la position réelle de la source pour plusieurs azimuts (figure 6). Afin de mitiger l’effet du choix de l’élévation sur la sélection (la précision peut varier selon que la position du stimulus coïncide par hasard avec l’entrée prototypique d’un des nœuds, ou qu’au contraire elle soit très éloignée de la BMU), nous testons les élévations $[-30, -25, -20, \dots, 30]$ et gardons la valeur de distance moyenne.

Les performances dans les cas unimodaux confirment nos observations précédentes. Hormis un artefact au centre, la perception visuelle est plus précise vers le centre qu’en périphérie, et inversement pour l’audition. Dans le cas bimodal, la perte est généralement entre les pertes subies par chaque modalité seule. Même si l’amélioration n’est pas franche, elle reste intéressante étant donnée la forte dégra-

vation de la topologie par un SWAE qui a comprimé la HRTF en 2D sans avoir de raison explicite de conserver les informations utiles à la localisation.

5 Conclusion et perspectives

Notre modèle étend une précédente contribution, en ajoutant un SWAE pour encoder des données en amont de la création d’un graphe multimodal, sur lequel un DNF peut faire de la fusion. Dans une certaine mesure, il est possible de réduire la dimensionnalité des données d’entrée par un auto-encodeur sans aucun impact à anticiper sur l’efficacité du modèle de fusion. Toutefois, il apparaît une borne à la force de la compression, au-delà de laquelle des informations pertinentes risquent d’être perdues. Mais la perte sera en partie compensée par l’autre modalité après la fusion.

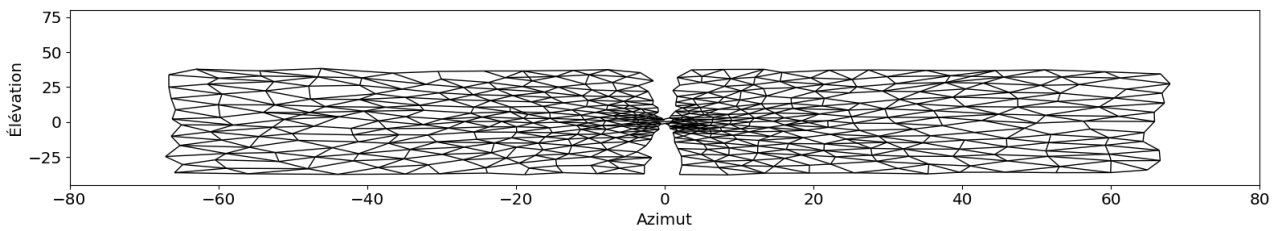


FIGURE 4 – GNG appris sur des données visuelles 2D déplacées par une transformation logpolaire

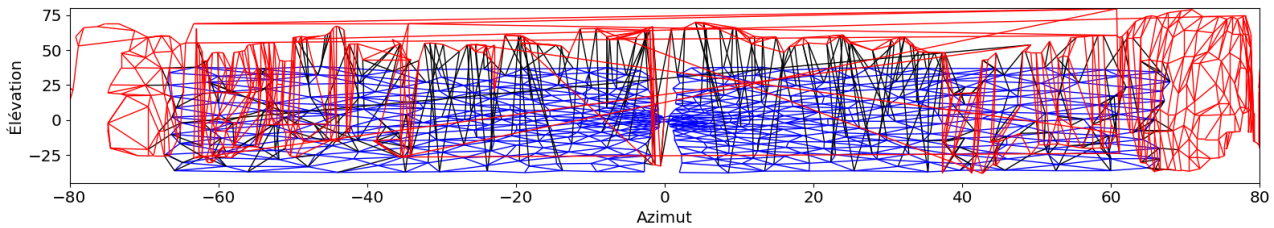


FIGURE 5 – Graphe audiovisuel obtenu en associant un GNG visuel et un GNG auditif 2D. Les connexions intra-visuelles sont affichées en bleu, intra-auditives en rouge, et inter-modales en noir.

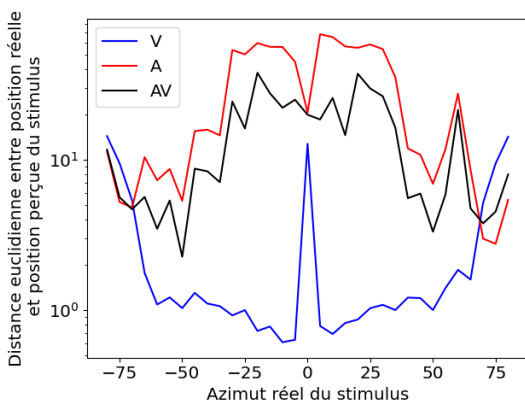


FIGURE 6 – Perte de précision dans la localisation d’un stimulus dans une topologie uniquement visuelle (V), uniquement auditive (A), ou bimodale (AV). L’ordonnée représente la distance en échelle logarithmique entre les positions réelle et perçue du stimulus. Pour chaque azimut, la distance est moyennée sur plusieurs élévations.

Les résultats présentés ici pourraient être améliorés de plusieurs façons. D’abord, dans la figure 6, il aurait été souhaitable que la fonction de distance dans le cas audiovisuel suive la meilleure des deux modalités, voire fasse mieux que les deux. Les résultats présentés précédemment dans [10] tendaient à montrer qu’il était possible de conserver les meilleures propriétés des deux modalités. Mais il semblerait que la dégradation des informations de localisation auditives par le SWAE, quand l’espace latent est imposé en 2D sans supervision sur la pertinence des dimensions à conserver, soit trop détrimentale pour que la fusion des GNG rende les modalités vraiment complémentaires.

C’est pour cette raison que nous avons eu besoin de limiter le nombre de connexions intermodales pour chaque nœud, car sinon les nœuds situés en haut du GNG auditif se connecteraient à toute une colonne de nœuds visuels, effaçant indirectement la perception de l’élévation dans tout le centre du champ de vision. Bien entendu, il serait envisageable de contraindre la structure de l’espace latent pour que, même en 2D, le SWAE apprenne à garder les propriétés de localisation sonore qui nous intéressent. L’optimisation du réseau de neurone pour accomplir cette tâche précise serait une perspective de prolongement de nos travaux. Cependant, des tests préliminaires que nous avons menés sur d’autres types d’auto-encodeurs (notamment VAE) semblent confirmer que les propriétés géométriques de l’espace latent formé par un SWAE sont bien indispensables à la création d’un graphe compatible avec notre méthode de fusion, même s’il se posera inévitablement la question de la superposition d’espaces latents de modalités différentes.

La principale nouveauté de notre modèle est que nous sommes désormais capables de coupler, d’une part, un apprentissage profond, à, d’autre part un modèle de fusion peu coûteux en apprentissage et ayant accès à des propriétés intéressantes (i.e., les capacités de sélection, attention, mémoire, etc. qui ont longuement été développées dans la littérature du DNF). L’ajout de réseaux de neurones ouvre la porte à la manipulation d’espaces d’entrée bien plus complexes. Un exemple serait la reconnaissance d’émotion. Les informations sur l’émotion d’un individu peuvent être perçues par plusieurs canaux : reconnaissance visuelle des expressions du visage, reconnaissance auditive du timbre de la voix, traitement du langage naturel... De nombreux travaux proposent de fusionner ces modalités dans le domaine de l’apprentissage profond, mais pas avec un DNF, car il

n'est pas capable d'intégrer des données aussi complexes. Notre méthode crée une opportunité de le faire.

Parmi les extensions possibles, et notamment dans le cas de tâches aussi complexes que la reconnaissance d'émotions, il serait intéressant d'étudier l'apprentissage simultané de plusieurs auto-encodeurs pour des modalités différentes. Une première piste, qui serait plus utile dans notre exemple de localisation, serait d'utiliser une modalité pour superviser l'autre. Nous entraînerions un auto-encodeur sur les données auditives, pour se conformer non pas à une distribution fixée arbitrairement, mais à la distribution latente des données visuelles, dont on sait qu'elles sont apprises avec une meilleure précision. Ce type de solution a déjà été exploré dans la littérature [26,37]. Ce serait une manière de privilégier lors de l'encodage les informations qui correspondent d'une modalité à une autre.

Cependant, il ne faut pas compter sur une correspondance systématique entre les modalités. Par exemple, il n'est pas garanti que tous les descripteurs d'une émotion soient accessibles aussi bien par reconnaissance visuelle que par traitement du langage : dans le langage seul, il est notamment difficile de distinguer si un compliment est sarcastique ou non, là où l'expression du visage permet de faire plus facilement la distinction entre un sentiment heureux ou en colère. En forçant une corrélation visuo-langagière, nous risquerions d'entraîner un auto-encodeur (côté traitement du langage) à reconnaître dans du bruit un descripteur (existant en vision) auquel il n'a en fait pas accès. Une piste alternative serait de modifier la distance de Wasserstein pour qu'à l'entraînement, un auto-encodeur accorde moins d'importance à une dimension sur laquelle une autre modalité est plus efficace. En entraînant les modalités en parallèle, nous pourrions nous assurer de leur bonne complémentarité, et laisser ensuite le DNF réaliser la fusion et la prise de décision sans apprentissage supplémentaire.

Remerciements

Ces travaux ont été soutenus par la région Auvergne-Rhône-Alpes via l'initiative Pack Ambition Recherche (projet AMPLIFIER), ainsi que l'Agence Nationale de la Recherche – institut 3IA – MIAI@Grenoble Alpes (ANR-19-P3IA-0003).

Une partie des calculs présentés dans cet article ont été réalisés grâce aux infrastructures de GRICAD (<https://gricad.univ-grenoble-alpes.fr>), soutenue par les communautés de recherche de Grenoble.

Références

- [1] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano. The CIPIC HRTF database. In *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, pages 99–102. IEEE, 2001.
- [2] S.-I. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2) :77–87, 1977.
- [3] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan. Intrinsic dimension of data representations in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [4] S. Argentieri, P. Danès, and P. Souères. A survey on sound source localization in robotics : From binaural to array processing methods. *Computer Speech & Language*, 34(1) :87–112, 2015.
- [5] Y. Bengio, A. Courville, and P. Vincent. Representation learning : A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8) :1798–1828, 2013.
- [6] M. Cerda and B. Girau. Asymmetry in neural fields : a spatiotemporal encoding mechanism. *Biological cybernetics*, 107(2) :161–178, 2013.
- [7] G. Dabane, L. U. Perrinet, and E. Dauce. What you see is what you transform : Foveated spatial transformers as a bio-inspired attention mechanism. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.
- [8] A. Droniou, S. Ivaldi, and O. Sigaud. Deep unsupervised network for multimodal perception, representation and classification. *Robotics and Autonomous Systems*, 71 :83–98, 2015.
- [9] J. Fix, N. Rougier, and F. Alexandre. A dynamic neural field approach to the covert and overt deployment of spatial attention. *Cognitive Computation*, 3(1) :279–293, 2011.
- [10] S. Forest, J.-C. Quinton, and M. Lefort. Combining manifold learning and neural field dynamics for multimodal fusion. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.
- [11] S. Forest, J.-C. Quinton, and M. Lefort. A dynamic neural field model of multimodal merging : application to the ventriloquist effect. *Neural Computation*, 34(8) :1701–1726, 2022.
- [12] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1995.
- [13] N. Gonnier, Y. Boniface, and H. Frezza-Buet. Input prediction using consensus driven SOMs. In *2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 38–42. IEEE, 2021.
- [14] Q. Houbre, A. Angleraud, and R. Pieters. Balancing exploration and exploitation : a neurally inspired mechanism to learn sensorimotor contingencies. In *Human-Friendly Robotics 2020 : 13th International Workshop*, pages 59–73. Springer, 2021.
- [15] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver : General perception with iterative attention. In M. Meila and T. Zhang,

- editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021.
- [16] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.
- [17] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1) :59–69, 1982.
- [18] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019.
- [19] S. Krishnagopal and J. Bedrossian. Preserving data manifold structure in latent space for exploration through network geodesics. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.
- [20] S. Lalle and P. F. Dominey. Multi-modal convergence maps : from body schema and self-representation to mental imagery. *Adaptive Behavior*, 21(4) :274–285, 2013.
- [21] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch : Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.
- [22] M. Lefort, Y. Boniface, and B. Girau. SOMMA : Cortically inspired paradigms for multimodal processing. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.
- [23] L. Manfredi, E. S. Maini, and C. Laschi. Neurophysiological models of gaze control in humanoid robotics. In B. Choi, editor, *Humanoid Robots*, chapter 10. IntechOpen, Rijeka, 2009.
- [24] S. Marsland, J. Shapiro, and U. Nehmzow. A self-organising network that grows when required. *Neural networks*, 15(8-9) :1041–1058, 2002.
- [25] T. Martinetz and K. Schulten. A “neural-gas” network learns topologies. *Artificial neural networks*, 1 :397–402, 1991.
- [26] S. Moon, S. Kim, and H. Wang. Multimodal transfer deep learning with applications in audio-visual recognition. *arXiv preprint arXiv :1412.3121*, 2014.
- [27] O. Ménard and H. Frezza-Buet. Model of multimodal cortical processing : Coherent learning in self-organizing modules. *Neural Networks*, 18(5) :646–655, 2005.
- [28] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34 :14200–14213, 2021.
- [29] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.
- [30] F. P. Ottes, J. A. V. Gisbergen, and J. J. Eggermont. Visuomotor fields of the superior colliculus : A quantitative model. *Vision Research*, 26(6) :857–873, 1986.
- [31] G. I. Parisi, J. Tani, C. Weber, and S. Wermter. Emergence of multimodal action representations from neural network self-organization. *Cognitive Systems Research*, 43 :208–221, 2017.
- [32] J.-C. Quinton and B. Girau. Predictive neural fields for improved tracking and attentional properties. In *The 2011 International Joint Conference on Neural Networks*, pages 1629–1636. IEEE, 2011.
- [33] J.-C. Quinton and L. Goffart. A unified dynamic neural field model of goal directed eye movements. *Connection Science*, 30(1) :20–52, 2018.
- [34] Y. Sandamirskaya. Dynamic neural fields as a step toward cognitive neuromorphic architectures. *Frontiers in Neuroscience*, 7 :276, 2014.
- [35] C. Schauer and H. M. Gross. Design and optimization of Amari neural fields for early auditory-visual integration. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 4, pages 2523–2528, 2004.
- [36] G. Schöner, J. Spencer, and DFT Research Group. *Dynamic Thinking : A Primer on Dynamic Field Theory*. Oxford Series in Developmental Cognitive Neuroscience. Oxford University Press, 2015.
- [37] S. Stojanov, A. Thai, and J. M. Rehg. Using shape to categorize : Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021.
- [38] W. Taouali, L. Goffart, F. Alexandre, and N. P. Rougier. A parsimonious computational model of visual target position encoding in the superior colliculus. *Biological Cybernetics*, 109(4) :549–559, 2015.
- [39] J. Tekülve, A. Fois, Y. Sandamirskaya, and G. Schöner. Autonomous sequence generation for a neural dynamic robot : scene perception, serial order, and object-oriented movement. *Frontiers in neurorobotics*, 13 :95, 2019.
- [40] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018.
- [41] M. Vavrečka and I. Farkaš. A multimodal connectionist architecture for unsupervised grounding of spatial language. *Cognitive Computation*, 6(1) :101–112, 2014.
- [42] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [43] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo. Deep multimodal representation learning from temporal data. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5066–5074, 2017.

Change-Relaxed Active Fairness Auditing

A. Godinot^{1,2,3}, E. Le Merrer², C. Penzo³, F. Taïani¹, G. Tredan⁴

¹ Université de Rennes

² Centre Inria de l'Université de Rennes

³ Pôle d'Expertise de la Régulation Numérique

⁴ LAAS/CNRS

Abstract

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. This paper focuses on a recent theoretical advance named manipulation-proofness. It aims at guaranteeing successive audits of a platform cannot be gamed by the platform, provided the labels returned on the audit dataset do not change.

While this constitutes a decisive step for reliable audits, it is too restrictive, as models naturally evolve with time in practice. This paper thus explores how manipulation-proofness can be adapted to better fit actual scenarios, by studying the effects of relaxing the constraint on the amount of change the remote model can operate while being audited. Our results on the COMPAS dataset demonstrate a request gain in one of the two models considered, while also noticing the surprisingly good performance of the random strawman approach. We believe this constitutes an interesting step for further attempts to improve reliable and manipulation proof audits.

Keywords

Artificial Intelligence, Algorithmic Auditing, Black-box models, Active Learning

1 Introduction

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. The advent of highly publicized audits, such as ProPublica's story on COMPAS [12] or Reuters study on Amazon's recruiting tool [8], has led to the algorithmic audit field gaining significant traction. For the public to trust Artificial Intelligence (AI) systems, and more broadly algorithmic decision systems, we need methods to explain the decision of such systems [19, 13], certify their implementation [22, 20] and automatically and robustly detect misconduct [14, 18].

Inspired by "traditional" financial audits, we focus in this work on *external certification audits*. In this type of audit, an external auditor (e.g. a regulator, or an auditing com-

pany) is commissioned by a platform to certify some desirable property (the absence of bias, for example) of its system. The system consists in a Machine Learning (ML) model h^* (see subsection 3.1) which is accessed by users through an interface (e.g. a web-page or an Application Programming Interface). To restrict the scope of this work, we consider that h^* is a binary classifier. Furthermore, we assume that the answers presented through the interface are faithful to that of the model h^* . We assume that the platform does not give access to the weights or implementation of the model h^* . The goal of the auditor is thus to certify the system as implemented and as seen by the users. The only information the auditor knows about the audited system is the hypothesis class \mathcal{H} of the model $h^* \in \mathcal{H}$. We dub this setting *remote black-box certification*. Yan and Zhang [22] recently proposed a theoretical framework to model the problem of remote black-box auditing. They provide an algorithm to select a minimal set of points S to estimate a property $\mu(h^*)$ (demographic parity for example) of the remote model h^* . While the model h^* behind the API is allowed to change after the audit, the auditor is guaranteed that the value $\mu(h)$ of any model $h \in \mathcal{H}$ that agrees with h^* on S will be close to their estimation $\hat{\mu}(h^*)$. This new estimation problem coined *manipulation-proof estimation* by Yan and Zhang is a step towards robust auditing as it provides a framework amenable to theoretical analysis. In practice, even if the type of model stays the same, because of retraining, arrival of new users or small tweaks, models served by platforms change over time. Thus, the requirement that the output of the API on the audit points does not change is too restrictive in practice. Moreover, Yan and Zhang only experimented with linear models on small datasets. In this work, we relax this recent formalization and empirically analyze its performance.

Contributions. This paper makes the following contributions. It first reviews the algorithmic audit setup, and recalls the concept and shortcomings of manipulation proofness (Sections 1 and 3). It then proposes a relaxation (coined *r-AFA+*) on the tolerated audit errors. We then evaluate this relaxation in Section 5, with two model classes and the COMPAS dataset, before we conclude.

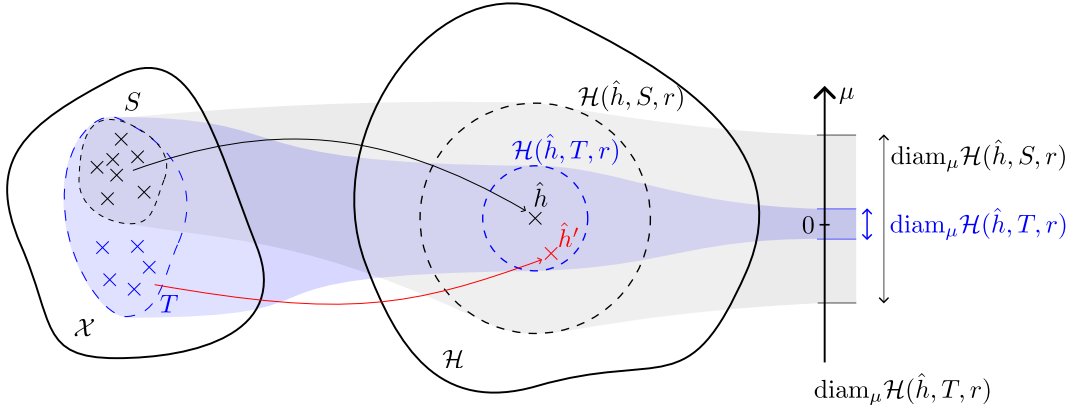


Figure 1: Schematic overview of the r -AFA+ algorithm. On the left, S and T are subsets of the input set \mathcal{X} . The goal of the approach is to identify a small set of inputs (left) that quickly reduce the version space (center) to models having close μ -values (right). Those inputs are iteratively improved to refine an estimate of the property of interest (μ -value).

At each outer iteration (line 2), the algorithm consists in three steps. **1. (black)** Train a surrogate \hat{h} on the current labeled dataset $(S, h^*(S))$. **2. (blue)** Find dataset T that minimizes the μ -diameter of the version space. **3. (red)** Merge T in S , query the API on T to train a new surrogate \hat{h}' .

2 Related work

AI audits. The AI audit field seeks to understand Artificial Intelligence (AI) algorithms as part of a larger socio-technical system. Most of the published audits include two phases (see for example [2, 15]). First, the auditor analyses the context of the algorithm: the training data, the users or the team who built said system. Then, they typically perform a statistical study to discover potential biases in the algorithm’s output. Recently, efforts have been made to formalize requirements dictated by regulatory bodies (such as the Data Minimization Principle [18]) and provide algorithms to help their enforcement. One challenge of remote black-box auditing is to limit the number of queries used to perform the audit. Issuing too few queries prevents any meaningful analysis but if the auditor requests large bursts of queries, they risk being blacklisted.

Robust auditing. Audits relying on statistical studies often make the simplifying assumption that the audited platform is cooperative and honest. While very practical, this is overly optimistic since there were examples of companies trying to evade high-stakes audits in the past [11]. Without this assumption, many simple audits become theoretically impossible [21]. To overcome this limit, the notion of manipulation-proof estimation has recently emerged [22]. Intuitively, this approach aims at constructing an auditing procedure that is resilient to arbitrary manipulation by the auditee while making as few assumptions as possible on the audited target.

Distribution testing. The field of tolerant distribution testing is interested in answering the question: given samples from an unknown distribution p , is this distribution ϵ_1 -close ($\min_{q \in \mathcal{P}}(d, q) \leq \epsilon_1$) to the set of distributions \mathcal{P} or is it ϵ_2 -far ($\min_{q \in \mathcal{P}}(d, q) \geq \epsilon_2$) from it? Some fairness measures (such as demographic fairness) can be formulated as the independence between the output of positive labels (e.g.

granting a loan, recruiting a new employee) and sensitive attributes (e.g. gender, ethnicity, religious beliefs, political views). Thus, as a specific tolerant distribution test, testing for independence could certify demographic parity. For an introduction to distribution testing and its extension to tolerant distribution testing, refer to [4] and [5]. It is however not straightforward how to certify other fairness properties.

Active learning. Active learning is a form of interactive learning where the learning algorithm can iteratively select the examples to train on. At each training step, the learning algorithm can use the performance of the trained model and past training examples to decide which training example to select next in order to optimize the learning process. The literature on active learning proved that interacting with the trained model could dramatically reduce learning sample complexity [10]. Even if the trained model is treated as a black box, it is possible to iteratively select training points based on the model’s outputs to reduce the number of training points needed [7]. These methods cannot be directly applied to black-box remote auditing because they need to probe the model on the whole dataset at each iteration, whereas we try to minimize the number of queries to said model. Yet, the method we present in this work builds on this idea to select the audit dataset by interacting with a *surrogate* of the API instead of the API itself. (We discuss this notion of surrogate model in more detail in Section 3.)

3 Manipulation proofness with AFA

In this section, we present in more detail the notion of *manipulation-proof estimation* introduced by Yan and Zhang [22] in the context of *remote black-box auditing*. We first introduce some key notations and assumptions and formalize the auditing process as a game between the auditor and the platform. We then define the notion of manipulation-proof estimation and present the general intu-

ition behind the AFA algorithm proposed by Yan and Zhang to solve the auditing game in a manipulation-proof manner.

3.1 Notations and assumptions

We consider a platform that seeks to solve a classification task (e.g. whether or not to grant a loan) based on user features (some information regarding the prospective borrower) grouped in a vector $x \in \mathcal{X}$. We assume that the space of all possible inputs—the *sample space* \mathcal{X} —is finite. Should \mathcal{X} not be finite, it suffices to sample a fixed number of instances in \mathcal{X} and treat them as a finite sample space. As explained in [7], it is then possible to adapt the bounds obtained for a finite \mathcal{X} .

When the platform trains its model, it effectively chooses a hypothesis $h^* \in \mathcal{H}$ in a set of possible (deterministic) models—the *hypothesis space* \mathcal{H} . The hypothesis space could for example be the set of linear binary classifiers on \mathcal{X} . Since the platform solves a classification task, the set of possible outputs—the *output space* \mathcal{Y} —is also finite. Therefore, because \mathcal{X} and \mathcal{Y} are finite, the space $\mathcal{Y}^{\mathcal{X}}$ of all functions from \mathcal{X} to \mathcal{Y} (and by extension $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$) is also finite. For any set S , we write $|S|$ its cardinal and $\mathcal{P}(S)$ the set of all its subsets.

The auditor seeks to test whether the model h^* used by the platform respects some desirable property $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \rightarrow \mathbb{R}$. For simplicity, we use the notation abuse $\mu(h) = \mu(\mathcal{X}, h)$. Define the *demographic parity* μ_{DP} as

$$\mu_{\text{DP}}(S, h) = \frac{1}{|S \cap A|} \sum_{x \in S \cap A} \mathbb{1}\{h(x) = 1\} - \frac{1}{|S \cap A^c|} \sum_{x \in S \cap A^c} \mathbb{1}\{h(x) = 1\} \quad (1)$$

where $\mathbb{1}\{P\}$ is the indicator function for the predicate P , A is the set of samples in \mathcal{X} with a positive sensitive attribute and A^c its complementary in \mathcal{X} .

We define the Hamming distance $d_H(h(S), h^*(S)) = |x \in S : h(x) \neq h^*(x)|$. Simply put, the Hamming distance is the number of points in S on which two hypotheses (or models) h and h^* disagree. Finally, for any subset of the hypothesis class $V \subset \mathcal{H}$, the μ -*diameter* $\text{diam}_{\mu}(V)$ is the largest difference in the value of μ between any two models in V .

$$\text{diam}_{\mu}(V) = \max_{h, h' \in V} |\mu(h) - \mu(h')| \quad (2)$$

3.2 The Auditing Game

The auditing process can be modeled as a game between the auditor and the audited platform. First, the auditor decides on the fairness measure $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \rightarrow \mathbb{R}$ on which they want to evaluate the platform. We assume the auditor can directly query the platform's output on a given input $x \in \mathcal{X}$, either through an API or through scraping. By gathering outputs on well-chosen inputs, the auditor seeks to construct an *audit dataset* $S \subseteq \mathcal{X}$, which the auditor will then use to estimate how well the platform respects the fairness measure μ , by using $\hat{\mu} = \mu(S, h^*)$ as an estimation of the true value $\mu(\mathcal{X}, h^*)$. In practice, platforms

regularly retrain their model h^* , for instance to account for new users or to improve it. As a result, h^* is likely to evolve after it has been audited. Thus, constructing a robust estimator intuitively means constructing an estimator that does not change too much even if the model is slightly modified after the audit. More formally, this auditing game can be described as follows:

Phase 1. At time t_0 , the auditor constructs an audit dataset $S \subset \mathcal{X}$ to build its estimator $\hat{\mu}(S, h_{t_0}^*)$ by interacting with the model $h_{t_0}^*$ served by the platform. The time needed to construct the audit dataset is supposed to be negligible and the remote model is assumed not to change during this phase.

Phase 2. At any time $t > t_0$ after the audit, and for any reason (retraining, new user or even adversarial change), we allow the model to change slightly. By re-querying the new model h_t^* on the same dataset S , we verify that answers to queries in S have not changed $d_H(h_t^*(S), h_{t_0}^*(S)) = 0$. The auditor's goal is to detect through their estimation $\hat{\mu}(S, h_t^*)$ when $\mu(\mathcal{X}, h_t^*)$ deviates too much from some target boundary, in which case the certificate must be revoked.

3.3 Manipulation-proof estimation.

The auditor's algorithm solves the above auditing game if it can produce an auditing set S such that for all $h \in \mathcal{H}$, if $d(h(S), h_t^*(S)) = 0$, then the μ -value of h cannot be at a distance larger than ϵ to $h_{t_0}^*$. To formalize the auditor's goal, we first define the notion of *version space*. It is the set of models h whose output agree with that of h^* on S .

$$\mathcal{H}(S, h^*) = \{h \in \mathcal{H} : d(h(S), h^*(S)) = 0\} \quad (3)$$

Then, an estimator $\hat{\mu}(S, h^*)$ of $\mu(\mathcal{X}, h^*)$ is said (r, ϵ) -*manipulation-proof* i.i.f.

$$\text{diam}_{\mu} \mathcal{H}(S, h^*) < \epsilon \quad (4)$$

The auditor only queries the labels $h^*(x)$ of points $x \in S$ therefore, they can only base their estimation $\hat{\mu}(S, h^*)$ of μ on $(S, h^*(S))$. Multiple models in \mathcal{H} can have the same answers on S and the auditor does not have any means to know which one of them is behind the API. Thus, there is an uncertainty on the true value $\mu(\mathcal{X}, h^*)$. The μ -diameter evaluates how well different audit datasets S might lead to a smaller/larger uncertainty on $\mu(\mathcal{X}, h^*)$. In their paper, Yan and Zhang frame the auditing game as a minimax game and prove a lower-bound on the number of queries required to reach ϵ -manipulation proofness. Inspired by the *Multiplicative Weight Update* method [1], they provide a randomized approximate algorithm AFA (Active Fairness Auditing, see our adapted version algorithm 1) to compute a solution with a query competitive ratio of $\mathcal{O}(\log(\mathcal{H}) \log(\mathcal{X}))$.

We present in algorithm 1 the core structure of the algorithm proposed in [22] with the modifications discussed in Section 4. The intuition behind this algorithm is to use the black-box teaching algorithm introduced in [7]. To avoid probing the API on the entire \mathcal{X} , we assume that we have access to an oracle $\mathcal{O} : \mathcal{P}(\mathcal{X}) \rightarrow \mathcal{H}$ providing surrogates of

h^* trained on S (line 4). The oracle \mathcal{O} is assumed to be mistake bounded, that is there exists $M > 0$ such that for any sequence $(x_i)_i$ of points from \mathcal{X} and their corresponding labels $(y_i)_i$, $\sum_{k=1}^{+\infty} \mathbb{1} \{ \mathcal{O}((x_1, \dots, x_k))(x_k) \neq y_k \} \leq M$. Example of such oracles include the perceptron algorithm [16] and the halving algorithm [3]. This surrogate is then used with the black-box teaching algorithm in [7] to find a subset T of \mathcal{X} maximizing $\text{diam}_\mu \mathcal{H}(\mathcal{O}(S), T)$.

4 Robust auditing in practice: giving some slack on the radius

Because our goal is to systematically analyze the performance of [22] in practice, we had to modify it to account for more realistic settings. The original AFA algorithm requires that after the audit, the labels of the queried points must not change. We argue that this assumption needs to be relaxed for two reasons. First, as we said for practical reasons the model might change slightly over time, modifying a small fraction of labels. Second, the auditor might not have access to an exact description of \mathcal{H} . This implies that the hypothesis space used for the audit $\mathcal{H}_{\text{surrogate}}$ does not match the one used by the platform \mathcal{H}_{API} . Because of this mismatch, the return condition line 19 might never be met. For these reasons, we relaxed the condition to $d(h(S), h^*(S)) < r$ in the definition of the version space (Equation 3) and adapted the algorithm accordingly. We name this method r -AFA+ (r -radius Active Fairness Auditing) and provide the pseudo-code in algorithm 1.

$$\mathcal{H}(S, h^*, r) = \{h \in \mathcal{H} : d(h(S), h^*(S)) \leq r\} \quad (5)$$

Theoretically, it is still unclear how these allowed errors might influence the query complexity of r -AFA+ compared to AFA. On one hand it definitely increases the cardinal of the version space, potentially increasing the μ -diameter for a given budget, requiring a larger T at each inner iteration. On the other hand, the exit condition of the algorithm (line 19) is less restrictive and decreases the number of outer iterations (and thus the total number of queries in S). As this is still preliminary work, we leave a more in-depth analysis of r -AFA+ for future work, and discuss the empirical results that follow from this relaxation in the next section.

5 Evaluation

We now quantify the impact of relaxing AFA with a tolerance radius of r changes, with r -AFA+.

Implementation Based on the notebooks provided in the supporting material of [22] we reimplemented algorithm 1 (with our modifications discussed in section 4). The implementation differs from the pseudocode in two ways. First, we modify the termination of the algorithm. The joint requirements for termination of estimated μ -diameter smaller than ϵ (line 10) and surrogate/API agreement (line 19) are replaced by the condition that $|S|$ does not exceed the budget. Second, instead of querying all the points in T (line 17) we only query one of them and re-enter the inner loop.

Algorithm 1 Remote black-box certification with r -AFA+

Require: Hypothesis class \mathcal{H} , mistake M -bounded oracle \mathcal{O} , target error ϵ , property μ , confidence δ , radius r

Ensure: audit dataset S

```

1:  $S \leftarrow \emptyset$ 
2: while true do
3:    $T \leftarrow \emptyset$ 
4:    $\hat{h} \leftarrow \mathcal{O}(S)$ 
5:    $w(x) \leftarrow \frac{1}{|\mathcal{X}|}, \forall x \in \mathcal{X}$ 
6:    $\tau(x) \sim \text{Exp} \left( \ln \left( \frac{M}{\delta} |\mathcal{H}|^2 \right) \right)$ 
7:   while true do
8:      $\triangleright$  Estimate the  $\mu$ -diameter of the current version space  $\triangleleft$ 
9:      $(h_{\min}, h_{\max}) \leftarrow \arg \min / \max_h \mu(h)$ 
10:    s.t.  $d(h(T), \hat{h}(T)) \leq r$ 
11:    if  $\mu(h_{\max}) - \mu(h_{\min}) < \epsilon$  then
12:       $\Delta(h_{\max}, h_{\min}) =$ 
13:       $\{x \in \mathcal{X} : h_{\max}(x) \neq \hat{h}(x) \text{ or } h_{\min}(x) \neq \hat{h}(x)\} \triangleleft$ 
14:       $\triangleright$  Multiplicative weight update
15:      while  $\sum_{x \in \Delta(h_{\max}, h_{\min})} w(x) \leq 1$  do
16:         $w(x) \leftarrow 2w(x), \forall x \in \Delta(h_{\max}, h_{\min})$ 
17:       $T \leftarrow \{x \in \mathcal{X} : w(x) \geq \tau(x)\}$ 
18:   query  $h^*$  on  $T$ 
19:    $S \leftarrow S \cup T$ 
20:   if  $\hat{h} \in \mathcal{H}(h^*, S, r)$  then
21:     return  $S$ 

```

Dataset We run our experiments on three datasets : student performance [6], COMPAS [12] and the reconstructed adults dataset [9]. In this preliminary version of our work, we only showcase results on the COMPAS dataset. COMPAS is a tool used by the US Department of Justice to evaluate the risk of recidivism among defendants, based on individual features such as age, gender, localization, origins amongst others. The COMPAS dataset consists in a list of 6172 defendants with their individual features and recidivism status.

Classifier model We run our experiments with multiple API hypotheses classes adapted to the classification task on tabular data: linear regression, support vector machines, decision trees and gradient boosted decision trees. Again, because it is a preliminary version of our work, we only analyse here the case of decision trees and linear classifiers, as implemented in `scikit-learn` [17]. We perform classical hyperparameter optimization with 5-fold validation to train the model behind the API.

Auditing algorithms The simplest algorithm we test is a random sampling baseline. Given a budget b , b points are uniformly sampled in \mathcal{X} without replacement to form the audit dataset S . The second baseline is the AFA algorithm. Then we test our method r -AFA+ with two values of r .

Evaluation results In Figure 2, we plot the value of the μ -diameter $\text{diam}_\mu \mathcal{H}(h^*, S, 5)$ against the audit budget $|S|$. On

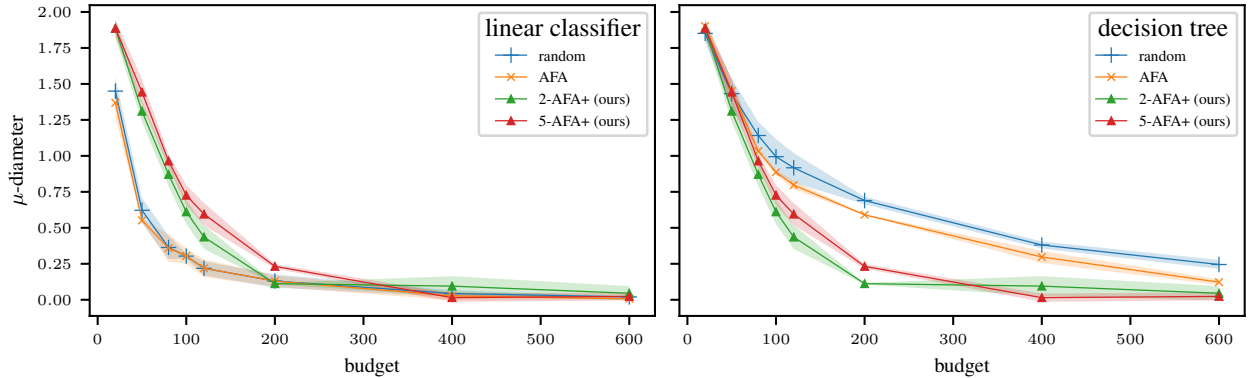


Figure 2: μ -diameter of the version space for budgets ranging from 200 to 600 requests. On the left, \mathcal{H} is the set of linear classifiers. On the right, \mathcal{H} is the set of decision trees. The figure compares two baselines (random sampling and original AFA) against r -AFA+ for two radius values ($r = 2$ and $r = 5$).

the right, \mathcal{H} is the set of decision trees. The figure compares the two baselines (random sampling and original AFA) and our method. Note that we are evaluating the audit algorithms in the more realistic setting with $r = 5 > 0$.

In the case of linear classifiers, both our methods and the baselines tend to a null μ -radius. Our methods even reduce slightly the convergence speed of the μ -radius to 0 (2-AFA+ needs ~ 50 more queries than random and AFA to reach a μ -diameter of 0.25). Yet, after 200 queries, all methods become equivalent in terms of μ -diameter. In addition, this plot highlights the performance of the simplest random baseline: it is the best performing method on this (dataset, api model) combination.

The second situation gives a totally different picture of the comparison between AFA and r -AFA+. While AFA performs as well as the random baseline, our methods allow to save up to 400 queries ($\sim 66\%$) to reach a μ -diameter of 0.25. The intuition behind the performance gap of r -AFA+ between the decision tree and linear APIs is linked to the regularity of the decision function. If the decision boundary is smoother (as is the case for linear models), two models that do not agree on a given set of points would not agree on the remaining points with a high probability. On the other hand, if the decision boundary is very irregular (as is the case for decision trees), two models that do not agree on a set of points might still be very close on the other points. Thus in this case, it seems that allowing for more disagreement (a.k.a. increasing r) between \hat{h} and the μ -optimal model h in line 9 helps to include models similar to the API h^* even if \hat{h} is far from it in the beginning.

The takeaways from this evaluation are that i) the gains in terms of budget are highly hypothesis dependant, 2) AFA is never substantially better than random, which questions its utility (high complexity w.r.t. random selection), and 3) r -AFA+ is at least as competitive as random and AFA on the long run (i.e. for small diameters).

6 Conclusion

Being robust to slight model changes is a practical requirement to take into account the practices of deployed ML systems that often evolve. In this context, the promising auditing approach of producing one-shot certificates might frequently require auditors to re-audit the target model after each slight update. This paper explores how certificates can be designed to be robust to such modifications and presented preliminary results that support this direction.

We have empirically shown that the r -AFA+ relaxation can provide an interesting gain over AFA in one scenario, and that the random and computationally cheap strawman approach is also to be considered. We leave to futurework a full characterization of the model families on which these observations generalize. Futurework also includes the study of the impact of removing the assumption that the hypothesis class is known by the auditor. More precisely, allowing for a restricted hypothesis space while preserving the accuracy of audits seems like an important next step for reliable and practical audits.

As a final remark, throughout this work, we used the term "Active Auditing" coined by the authors of [22]. Yet since, this algorithm guarantees that *if* the platform does not change \mathcal{H} *then* we can "easily" verify that our estimated value still holds. Thus, a more accurate term would be "active certification". This splits the goal of algorithmic auditing: trying to build certificates for platforms to defend themselves, or finding estimators that are able to uncover misconduct robust to concealment attempts from the platform.

Acknowledgements

We would like to thank Tom Yan (co-author of [22]) for the exchange we had upon implementing their algorithm.

References

- [1] Sanjeev Arora, Elad Hazan, and Satyen Kale. "The Multiplicative Weights Update Method: A Meta-

- Algorithm and Applications”. In: *Theory of Computing* 8.6 (May 1, 2012), pp. 121–164.
- [2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. “Fairness in Machine Learning”. In: *Nips tutorial 1* (2017).
- [3] Ja M. Barzdin. “On the Prediction of General Recursive Functions”. In: *Soviet Mathematics Doklady*. Vol. 13. 1972, pp. 1224–1228.
- [4] Clément L. Canonne. *A Survey on Distribution Testing: Your Data Is Big. But Is It Blue?* Graduate Surveys 9. Theory of Computing Library, Aug. 15, 2020. 100 pp.
- [5] Clement L. Canonne et al. “The Price of Tolerance in Distribution Testing”. In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Conference on Learning Theory. PMLR, June 28, 2022, pp. 573–624.
- [6] Paulo Cortez and Alice Silva. “Using Data Mining to Predict Secondary School Student Performance”. In: *EUROSIS* (Jan. 1, 2008).
- [7] Sanjoy Dasgupta et al. “Teaching a Black-Box Learner”. In: *Proceedings of the 36th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, May 24, 2019, pp. 1547–1555.
- [8] Jeffrey Dastin. “Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women”. In: *Reuters. Retail* (Oct. 10, 2018).
- [9] Frances Ding et al. “Retiring Adult: New Datasets for Fair Machine Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 6478–6490.
- [10] Steve Hanneke. “Theory of Disagreement-Based Active Learning”. In: *Foundations and Trends® in Machine Learning* 7.2-3 (June 11, 2014), pp. 131–309. ISSN: 1935-8237, 1935-8245.
- [11] Russell Hotten. “Volkswagen: The Scandal Explained”. In: *BBC News. Business* (Sept. 22, 2015).
- [12] Jeff Larson et al. “How We Analyzed the COMPAS Recidivism Algorithm”. In: *ProPublica* (May 23, 2016).
- [13] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.
- [14] J. Nathan Matias, Austin Hounsel, and Nick Feamster. “Software-Supported Audits of Decision-Making Systems: Testing Google and Facebook’s Political Advertising Policies”. In: *Proceedings of the ACM on Human-Computer Interaction* 6 (CSCW1 Apr. 7, 2022), 118:1–118:19.
- [15] Danaë Metaxa et al. “Auditing Algorithms: Understanding Algorithmic Systems from the Outside In”. In: *Foundations and Trends® in Human-Computer Interaction* 14.4 (2021), pp. 272–344. ISSN: 1551-3955, 1551-3963.
- [16] Albert B. Novikoff. *On Convergence Proofs for Perceptrons*. STANFORD RESEARCH INST MENLO PARK CA, 1963.
- [17] Fabian Pedregosa et al. “Scikit-Learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928.
- [18] Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. “Auditing Black-Box Prediction Models for Data Minimization Compliance”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20621–20632.
- [19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““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. New York, NY, USA: Association for Computing Machinery, Aug. 13, 2016, pp. 1135–1144. ISBN: 978-1-4503-4232-2.
- [20] Ali Shahin Shamsabadi et al. “Confidential-PROFIT: Confidential PROof of FaIr Training of Trees”. In: The Eleventh International Conference on Learning Representations. Feb. 1, 2023.
- [21] Ali Shahin Shamsabadi et al. “Washing The Unwashable : On The (Im)Possibility of Fairwashing Detection”. In: *Advances in Neural Information Processing Systems*. Oct. 31, 2022.
- [22] Tom Yan and Chicheng Zhang. “Active Fairness Auditing”. In: *Proceedings of the 39th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, June 28, 2022, pp. 24929–24962.

Génération de données synthétiques de marche : application au cas de patients atteints de sclérose en plaques

K. Le Gall¹, L. Bellanger¹, A. Stamm¹, D.A. Laplaud²

¹ Laboratoire de Mathématique Jean Leray, UMR CNRS 6629, Nantes Université, France

² CR2TI, INSERM U1064, CHU de Nantes, Nantes Université, France

Klervi.Legall@univ-nantes.fr

Résumé

L'objectif de ce travail est de générer des séries temporelles de quaternions synthétiques (QTS). L'approche proposée permet de construire un jeu de données synthétiques en mêlant ACP fonctionnelle et proches voisins et permet une bonne conservation de la géométrie des données. Nous montrerons la pertinence de notre approche à l'aide d'un échantillon de séries temporelles de quaternions issu de données de marche de 27 patients atteints de sclérose en plaques issu d'une étude clinique menée en collaboration avec l'équipe de neurologie du CHU de Nantes.

Mots-clés

Analyse en Composantes Principales fonctionnelle, Séries temporelles de Quaternions unitaire, Marche, Sclérose en plaques, Données Synthétiques.

Abstract

The objective of this work is to generate synthetic quaternion time series (QTS). The proposed approach allows to build a synthetic dataset by mixing functional PCA and nearest neighbours and allows a good preservation of the data geometry. We will show the relevance of our approach using a sample time series of quaternions from the gait data of 27 multiple sclerosis patients from a clinical study conducted in collaboration with the neurology team of the Nantes University Hospital.

Keywords

Functional Principal Component Analysis, Unit Quaternion Time Series, Walking, Multiple Sclerosis, Synthetic Data.

1 Introduction

Les séries temporelles de quaternions unitaires (QTS) permettent de caractériser les rotations et sont donc présentes dans de nombreux domaines tels que la robotique, les jeux vidéo ou la santé. L'un des principaux défis dans certains domaines tels que la médecine est la taille réduite des échantillons en raison de la difficulté à mener de nombreuses expériences coûteuses. A ce jour, il n'existe pas de méthode pour générer des séries temporelles de quaternions unitaires proches des données originales. La création

de données synthétiques permet remédier à ce problème. Les données synthétiques sont "toute donnée de production applicable à une situation donnée qui n'est pas obtenue par mesure directe", selon le McGraw-Hill Dictionary of Scientific and Technical Terms [7].

L'approche que nous proposons pour générer des QTS synthétique est inspirée de la méthode avatar [6], une méthode d'anonymisation de données basée sur l'individu et ses proches voisins, qui nous permet de respecter la géométrie de nos données. La méthode proposée peut également servir de méthode d'anonymisation des données afin de respecter le Règlement général sur la protection des données (RGPD) et ainsi protéger les données personnelles des individus.

Nous illustrerons notre approche en utilisant un échantillon de 27 patients atteints de sclérose en plaques issus d'une étude clinique menée en collaboration avec l'équipe de neurologie du CHU de Nantes pour lesquels nous avons mesuré un biomarqueur appelé Individual Gait Pattern qui caractérise la rotation de la hanche d'un individu au cours d'un cycle de marche moyen en utilisant des séries temporelles de quaternions unitaires. L'augmentation de notre ensemble de données nous permettra également de tester la robustesse des algorithmes de classification précédemment mis en œuvre.

2 Méthode de génération de séries temporelles de quaternions unitaires synthétiques

2.1 Séries temporelles de quaternions unitaires

Un quaternion est un élément $(w,x,y,z) \in \mathbb{R}^4$, qui est une extension des nombres complexe et qui s'écrit comme suit :

$$\mathbf{q} = (w, x, y, z)^T = w + ix + jy + kz \in \mathbb{R}^4 \quad (1)$$

où i, j et k respectent $i^2 = j^2 = k^2 = ijk = -1$. [12]

Les quaternions unitaires sont ceux pour lesquels $\|\mathbf{q}\| = w^2 + x^2 + y^2 + z^2 = 1$ de sorte que leur norme soit égale à 1. Le groupe des quaternions unitaires \mathbb{H}_u forme un groupe de Lie isomorphe au groupe unitaire spécial $SU(2)$

qui couvre deux fois le groupe des matrices de rotations en 3-Dimensions [2].

Les quaternions unitaires décrivent une rotation d'un angle θ autour d'un axe $\mathbf{u} = (u_x, u_y, u_z)^\top \in \mathbb{S}^2$ où \mathbb{S}^2 est la 2-sphère qui peut être exprimée comme suit :

$$\mathbf{q} = \left(\cos \frac{\theta}{2} + \mathbf{u}_x \sin \frac{\theta}{2} i + \mathbf{u}_y \sin \frac{\theta}{2} j + \mathbf{u}_z \sin \frac{\theta}{2} k \right)^\top \quad (2)$$

Série temporelle de quaternion unitaires

Une série temporelle de quaternions unitaires (QTS) est un ensemble de quaternions unitaires suivant une grille temporelle $t_{i,1}, \dots, t_{i,n}$. On note une QTS comme : $\mathbf{Q}_i = (\mathbf{q}_{i,1}, \dots, \mathbf{q}_{i,n})$. Elle représente des rotations 3D consécutives dans le temps.

2.2 Génération de QTS synthétiques

Le logarithme d'un quaternion unitaire est déterminé à partir de sa forme polaire :

$$\ln(\mathbf{q}) = \ln(\exp(\tilde{\mathbf{u}} \frac{\theta}{2})) = \tilde{\mathbf{u}} \frac{\theta}{2} = (0, u_x \frac{\theta}{2}, u_y \frac{\theta}{2}, u_z \frac{\theta}{2})^\top \quad (3)$$

Cette transformation logarithmique est une application entre l'espace des quaternions unitaires et l'espace tangent au point $q = (1, 0, 0, 0)$ [8]. Cela signifie que si les données sont préalablement centrées, on peut les traiter dans \mathbb{R}^3 . Cet outil est particulièrement intéressant lorsque les points dans \mathbb{H}_u sont proches les uns des autres, cela garanti une conservation suffisante des distances entre les points dans l'espace tangent.

En centrant les séries temporelles de quaternions et appliquant cette transformation logarithmique, nous obtenons trois séries temporelles qui peuvent être traitées comme des données fonctionnelles.

Lorsque l'on travaille avec des données fonctionnelles, les observations sont des fonctions qui varient selon certaines variables continues, par exemple le temps. L'ensemble de données est alors une collection de n fonctions $X_i(t)$, $i = 1, \dots, n$.

La méthode que nous proposons repose sur la possibilité de décomposer des données fonctionnelles sous forme de fonctions et de scores à l'aide d'une *analyse en composantes principales fonctionnelle multivariée* (MFPCA) [9] tel que :

$$X_i(t) = \sum_{k=1}^K F_i^k \mathbf{u}_k(t) \quad (4)$$

Ainsi, pour chaque individu i , à chaque temps t , on peut décomposer une log-série temporelle de quaternions selon trois fonctions $\mathbf{u}_k(t) \in \mathbb{R}^3$ et un score $F_i^k \in \mathbb{R}$ pour chaque composante principale $k \in 1 : K$.

Cette formule nous permet donc également d'obtenir de nouvelles données, en changeant le score F_i^k , mais en conservant les fonctions principales.

Dans la méthode proposée, de nouveaux scores synthétiques $F_{i \text{ new}}^k$ sont calculés pour chaque individu i . A partir des scores obtenus par MFPCA, on détermine les L plus proches voisins de l'individu i sur les $T \leq K$ premières

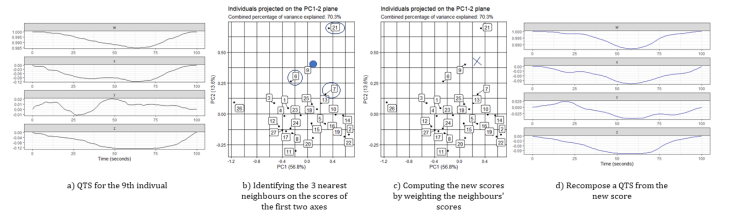


FIGURE 1 – Génération d'une série temporelle de quaternions unitaires synthétique

composantes. Les scores de ces voisins sont ensuite pondérés en prenant compte d'une distribution aléatoire et la distance à l'individu i et ses L voisins afin d'obtenir ce score synthétique.

Les données sont ensuite recomposées pour obtenir des séries temporelles de quaternions unitaires $\mathbf{Q}_{i \text{ new}}$ tel que :

$$\mathbf{Q}_{i \text{ new}} = \exp\left(\sum_{k=1}^K (F_{i \text{ new}}^k \mathbf{u}_k(t))\right) \quad (5)$$

Enfin, les séries temporelles de quaternions unitaires sont décentrées.

Nous proposons également des métriques qui permettent de contrôler la qualité des QTS synthétiques obtenues. Afin de vérifier le respect de la géométrie, nous calculons des distances de Frobenius entre les matrices d'adjacences des graphes des plus proches voisins [1, 5]. La conservation des des informations est indiquée par le coefficient RV [10], et si la méthode est utilisée pour de l'anonymisation, le risque de ré-identification est évaluée par le local cloaking [6].

La figure 1 illustre la méthode proposée.

3 Application aux données de marche

Les données sur lesquelles nous nous appuyons sont issues d'une étude menée en partenariat avec l'équipe de neurologie du Centre d'Investigation Clinique de Nantes, comprenant le Pr. Laplaud, (Neurologue et PU-PH au CHU de Nantes) et le Pr. Gourraud (PU-PH au CHU de Nantes). L'étude MYO porte sur le signal nerveux des patients atteints par la sclérose en plaques (SEP) mesuré par le bracelet électronique MYO en 2018 et un amendement a permis d'ajouter à cette étude principale une étude ancillaire pour mesurer la marche des patients via un capteur positionné à la hanche droite. Nous avons pu recueillir des données pour 27 patients.

Nous travaillons sur l'analyse des données de ce capteur afin de comprendre les troubles de la marche dans le contexte de la sclérose en plaques. Le système de capteurs transmet le vecteur d'orientation absolue du dispositif sous la forme d'un quaternion unitaire toutes les 0,01 secondes, une série temporelle de quaternions unitaires est donc recueillie pour chaque patient. Ces données brutes sont ensuite segmentées en cycles de marche et la signature de marche (SdM) d'un individu correspond au centre

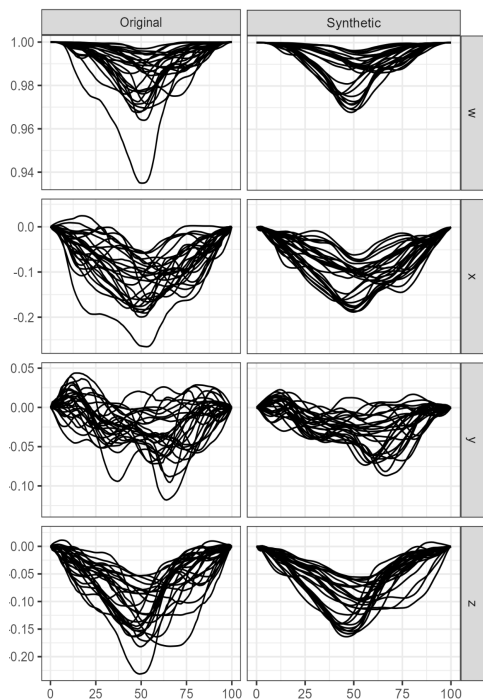


FIGURE 2 – Signatures de marche des 27 patients de l'étude ancillaire MYO et signatures de marches synthétiques obtenues à partir de ces données avec $L = 4$ et $T = 10$

des cycles de marche détectés, ce centre est obtenu avec une méthode de k-means alignement [11] pour lequel le nombre de cluster est égal à 1. La série temporelle est ensuite exprimée en pourcentage de la durée totale. Le temps 1 (0%) est la référence à partir de laquelle les orientations observées pendant le cycle de marche sont calculées. Cela permet d'obtenir une SdM dont le premier et le dernier élément sont des quaternions identité $\mathbf{q} = (1, 0, 0, 0)$. Ces SdM sont des QTS unitaires et décrivent les rotations "moyennes" de la hanche pendant un cycle de marche [4].

La figure 2 représente les SdM des patients de l'étude ancillaire MYO, ainsi qu'une génération de données synthétiques possible à partir de ces SdM.

Deux éléments se dégagent de cette figure, on remarque tout d'abord que les données sont davantage centrées, cela est lié à la pondération des proches voisins pour créer les scores synthétiques qui permet de donner moins d'importance à de potentiels cas extrêmes. On observe également des groupes d'amplitudes différentes, en partie expliqués par la pathologie. Le coefficient RV est de 0.75, ce qui montre une bonne conservation des données, la géométrie a également été vérifiée.

Cette approche pourra aussi permettre de tester la robustesse des algorithmes de classification précédemment mis en œuvre [3].

Remerciements

Les auteurs remercient la fondation ARSEP (fondation pour l'Aide à la Recherche sur la Sclérose En Plaques) et l'AMIES (Agence pour les Mathématiques en Interaction avec l'Entreprise et la Société) pour le financement des études cliniques qui ont menées à l'obtention des données ainsi que le groupe de l'Observatoire Français de la Sclérose en Plaques (OFSEP) et les CHU de Rennes et de Nantes. Ce travail s'inscrit dans une thèse co-financée par l'ANR AIBY4 (ANR-20-THIA-0011) ainsi que Nantes Université.

Références

- [1] R. Balakrishnan, K. Ranganathan, *A Textbook of Graph Theory*, Springer New York, 2012.
- [2] M.S. Dijkhuizen, *The double covering of the quantum group $SO_q(3)$* , Proceedings of the Winter School "Geometry and Physics". Circolo Matematico di Palermo, Vol. 37, pp. 47-57, 1994.
- [3] P. Drouin, A. Stamm, L. Chevreuil et al, *Semi-supervised clustering of quaternion time series : application to gait analysis in multiple sclerosis using motion sensor data*, accepted, 2022.
- [4] P. Drouin, A. Stamm, L. Chevreuil et al, *Gait impairment monitoring in multiple sclerosis using a wearable motion sensor*, Medical Case reports and Reviews, Vol. 5, pp. 1-5, 2022.
- [5] D.Eppstein, M.S Paterson, F.F Yao, *Discrete and Computational Geometry*, Springer New York, pp. 263-282, 1997.
- [6] M. Guillaudeux, O. Rousseau, J. Petot et al, *Patient-centric synthetic data generation, no reason to risk re-identification in the analysis of biomedical pseudonymised data*, PREPRINT(V1), 2022.
- [7] S.P. Parker, *McGraw-Hill Dictionary of Scientific and Technical Terms. 7e éd.*, Cambridge University Press, 2009.
- [8] M. Piórek, « *Analysis of Chaos for Quaternion Time Series* » In : *Analysis of Chaotic Behavior in Non-linear Dynamical Systems*, Springer, pp. 73-88, 2019.
- [9] J.O. Ramsay , B.W. Silverman, « *Principal components analysis for functional data* » In : *Functional Data Analysis*, Springer, pp. 147-172, 2005.
- [10] P. Robert, Y. Escoufier, *A Unifying Tool for Linear Multivariate Statistical Methods : The RV- Coefficient*, Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol.25, 1976.
- [11] L.M Sangalli, P. Secchi, S. Vantini et al, *k-mean alignment for curve clustering*, Computational Statistics Data Analysis, Vol. 54, pp. 1219-1233, 2010.
- [12] J. Voight, *Quaternion Algebras*, Springer Nature, 2005.

Jouer avec des Cryptarithmes en Programmation par Contraintes

Arnaud Malapert¹, Margaux Schmied², Davide Fissore², Marie Pelleau¹, Ambre Picard Marchetto²

¹ Université Côte d'Azur, CNRS, I3S, France

² Université Côte d'Azur, France

¹ `firstname.lastname@univ-cotedazur.fr` ² `firstname.lastname@etu.univ-cotedazur.fr`

Résumé

Un cryptarithme est un casse-tête mathématique et logique dans lequel des mots forment une équation où les lettres représentent des chiffres à déterminer dans une base donnée. Ce problème est populaire en mathématiques récréatives, dans l'enseignement, et en programmation par contraintes. Nous proposons une approche générale, efficace et simple d'utilisation pour résoudre ce problème NP-Complet. La suite naturelle est une approche hiérarchique pour leur génération. Les évaluations expérimentales ont engendré une vaste collection, variée et remarquable.

Mots-clés

Cryptarithme ; puzzle ; résolution ; génération.

1 Introduction

Un cryptarithme est un casse-tête mathématique et logique dans lequel un ensemble de mots est écrit sous la forme d'une équation où les lettres représentent des chiffres à déterminer dans une base donnée. Il faut remplacer les lettres par des chiffres en respectant les règles suivantes :

Règle i) Chaque lettre est associée à un seul chiffre.

Règle ii) L'équation est vérifiée dans une base donnée en remplaçant les lettres par les chiffres.

Règle iii) Les lettres représentent des chiffres distincts.

Règle iv) Le chiffre de poids fort des mots est non nul.

Les deux dernières règles sont quelquefois relâchées pour obtenir une solution. Idéalement, il existe une solution unique.

Si l'invention des cryptarithmes remontent à la Chine antique, le cryptarithme le plus connu a été publié en juillet 1924 dans le *Strand Magazine* [8] par H.E. Dudeney :

$$\begin{array}{r} \text{SEND} \qquad \qquad 9567 \\ + \text{MORE} \qquad + \text{1085} \\ \hline \text{MONEY} \qquad \qquad 10652 \end{array}$$

$$S = 9; E = 5; N = 6; D = 7; M = 1; O = 0; R = 8; Y = 2$$

Malgré son apparente simplicité, le problème est NP-complet [9] ce qui est prouvé par une réduction de 3-SAT. Dans sa forme originelle, un cryptarithme est une addition avec un second membre réduit à un seul terme, et les nombres sont écrits en base 10. Mais, il existe d'autres variantes dignes d'intérêt présentées ultérieurement.

Ce puzzle est populaire en mathématiques récréatives comme en attestent les différentes publications et ressources glanées sur internet. Il existe plusieurs livres classiques [3, 10, 13] ou plus récents [7, 14] au contenu similaire avec un catalogue de puzzles classés par niveau avec leurs solutions accompagnés de conseils et méthodes de résolution « à la main ». Les cryptarithmes sont aussi présents dans des magazines, par exemple le journal *Sphinx* publié en Belgique en langue Française dans les années 1930 qui organisait aussi des compétitions. De nos jours, ils sont toujours utilisés dans les compétitions de la fédération Française des jeux mathématiques. Il existe aussi plusieurs sites web [6, 12, 19–21] consacrés à ce puzzle sur lesquels on trouvera des contenus similaires aux livres souvent avec des solveurs ou générateurs de cryptarithmes. Les solveurs disponibles utilisent le plus souvent des algorithmes de recherche exhaustive plus ou moins avancés et efficaces.

Ce puzzle est aussi populaire en enseignement, de l'école primaire à l'université. En primaire et au collège, il permet d'acquérir des compétences en résolution de problème, logique et calcul ce qui a fait l'objet de travaux en sciences de l'éducation [4, 22]. Au lycée et à l'université, c'est un problème simple et structuré qui est donc adapté pour l'apprentissage et la programmation de recherches exhaustives. La difficulté de programmer des recherches exhaustives générales et efficaces permet de présenter les avantages de la modélisation à l'université. C'est particulièrement vrai en programmation par contraintes [18] où $\text{SEND} + \text{MORE} = \text{MONEY}$ apparaît dans la majorité des tutoriels et manuels des solveurs de contraintes. Cependant, le problème n'a jamais fait l'objet d'une publication et n'apparaît pas non plus dans la CSPLib [1].

Les approches actuelles pour résoudre un cryptarithme montrent plusieurs limites. Résoudre un cryptarithme en décimal est un problème difficile pour les humains qui devient extrêmement difficiles dans les autres bases où nos capacités de calcul sont bien moindres.

Programmer une recherche exhaustive générale et efficace est une tâche complexe et la plupart des solveurs disponibles imposent des restrictions sur les instances du problème. Deux restrictions courantes sont de se limiter à une addition ou à la base décimale. Une autre restriction fréquente est de se limiter à l'arithmétique simple précision ce qui limite la taille des mots de l'équation.

Écrire un modèle spécifique est une tâche plus aisée, mais

cela ne résout pas le problème, seulement une instance ! Écrire un modèle général est une tâche plus difficile qui n'a jamais été réalisée à notre connaissance.

Finalement, il est temps de mettre à l'épreuve la réputation de facilité du problème et les performances de la programmation par contraintes. Les deux objectifs principaux sont de participer à la diffusion scientifique de la programmation par contraintes et d'être utilisé pour la création de contenu pédagogique dans l'enseignement.

La première contribution est de proposer un solveur de cryptarithme basé sur la programmation par contraintes qui soit général, efficace, et facile d'utilisation. Les solveurs disponibles satisfont au mieux deux de ces trois critères, abandonnant le plus souvent la généralité.

Assez naturellement, la seconde contribution est de proposer des générateurs de cryptarithmes, et de variantes, à partir d'une liste de mots. En effet, la découverte « à la main » d'un nouveau cryptarithme reste réservée aux experts et les générateurs disponibles sont plus limités que les solveurs. La génération de cryptarithmes constitue un défi plus difficile que leur simple résolution.

Ces contributions prennent la forme d'une bibliothèque Java CRYPTATOR¹ sous licence libre basée sur le solveur Choco [5] proposant deux applications en ligne de commandes.

La suite de ce document est structurée de la manière suivante. La section 2 introduit une forme générale de cryptarithme définie par une grammaire et présente plusieurs variantes du puzzle. La section 3 décrit un modèle pour la résolution de cryptarithme en arithmétique simple précision, et un modèle restreint aux additions en arithmétique multiprécision. La section 4 décrit plusieurs modèles avec un socle commun pour la génération de différentes variantes à partir d'une liste de mots. La section 5 présente l'évaluation expérimentale des performances des solveurs. La section 6 présente les expériences pour créer une vaste collection de cryptarithmes aux propriétés variées et remarquables. La section 7 décrit les choix technologiques et des éléments d'architecture de la bibliothèque, et présente quelques cas d'usage.

2 Définition d'un cryptarithme

Un cryptarithme est défini par une équation mathématique et une base arithmétique. Une équation est définie par une grammaire présentée en section 2.1 pour la facilité d'utilisation. Par défaut, on suppose que les règles i, ii, iii, et iv de la section 1 doivent être satisfaites par une solution. Cependant, nous verrons en section 3 que les règles iii et iv peuvent être relâchées.

À partir de ces entrées, on peut poser plusieurs questions : existence d'une solution ; existence d'une solution unique ; énumération des solutions. Historiquement, par un souci d'élégance, l'existence d'une solution unique est la question fondamentale.

La section 2.2 introduit plusieurs variantes de cryptarithmes considérées dans ce travail.

1. <https://github.com/arnaud-m/cryptator>

2.1 Grammaire d'une équation

La grammaire sert à simplifier la définition d'une équation. Elle est exprimée en *Extended Backus-Naur Form* (EBNF), afin de ne pas allourdir cet article nous ne donnons pas ici la grammaire.

La grammaire permet la reconnaissance d'une forme infixe, la forme classique que les humains manipulent, mais elle n'est pas nécessaire pour la reconnaissance des formes préfixe ou suffixe. Elle permet aussi de détecter et d'expliquer des erreurs lors de la saisie d'une équation.

La représentation d'un cryptarithme se base sur la construction d'un arbre syntaxique réalisé à partir d'un parseur. Ce parseur est capable de reconnaître les cryptarithmes classiques, mais il est plus général pour capturer ou imaginer des variantes. La grammaire est une grammaire hors contexte dont les règles de dérivation suivent la structure suivante $X \rightarrow \alpha$ où X est un symbole non terminal et α est une suite de symboles terminaux ou non terminaux. La vérification de la structure d'une grammaire est subdivisée en deux parties essentielles : un *lexeur*, ayant le rôle de renvoyer sous forme de *tokens* les mots reconnus à partir du texte entré et un *parseur* qui vérifie la bonne structure syntactico-sémantique de la suite des *tokens* reçus par le *lexeur*.

En particulier, le lexeur distingue les *termes* (des suites alphanumériques composées des symboles UTF-8), les *comparateurs* tels que l'égalité et l'inégalité stricte ou large, et les *opérateurs* tels que l'addition, la multiplication, la soustraction, la division, l'élévation à la puissance et le modulo. De plus, l'opérateur de conjonction de cryptarithmes `&&` ou `;` permet de séparer une équation en plusieurs cryptarithmes. Tous ces opérateurs suivent les règles de priorités usuelles. Le parseur accepte les parenthèses pour permettre la manipulation des priorités d'opération. Tous les espaces, tabulations ou sauts de ligne sont autorisés et ignorés pour procurer un confort d'utilisation.

Le lexeur distingue deux types de terme. Un *mot* est une séquence de lettres à remplacer par des chiffres. Un *nombre* est une séquence de chiffres décimaux dont la valeur décimale est utilisée pour vérifier l'équation. Dans la grammaire, un nombre est délimité par des guillemets simples ou doubles. Ainsi, le cryptarithme "11" + 89 = "40" contient deux constantes, 11 et 40, et un mot 89. Dans la solution, les lettres (8 et 9) peuvent prendre des valeurs différentes de leur propre sémantique (2 et 9).

L'opérateur de conjonction et les nombres ont été introduits pour capturer certaines variantes introduites ci-dessous.

2.2 Variantes

Au cours de son histoire, de nombreuses variantes de cryptarithmes ont été introduites. Ici, nous nous limiterons à celles présentes dans la collection de la section 6.

2.3 Doublement vrai

Un cryptarithme est doublement vrai si chaque mot est un nombre écrit en toute lettre et que l'équation textuelle est elle aussi vérifiée. Ci-dessous, l'équation textuelle est à gauche, le cryptarithme au milieu, et sa solution droite.

0	CERO	8027
+ 6	+ SEIS	+ 3013
+ 7	+ SIETE	+ 31040
13	TRECE	42080

2.4 Mots croisés

Un mots croisés est une variante dans laquelle plusieurs équations sont écrites dans une grille et une seule affectation des chiffres aux lettres doit permettre de vérifier toutes les équations.

AN	+	TA		DOL	87	+	38		125
+		+		+	+		+		+
ODE	+	TEL		LAD	216	+	365		581
TUT	+	SUT		NUE	303	+	403		706

Donc, chaque ligne et chaque colonne représente une équation et il faut résoudre la conjonction de ces cryptarithmes donnée dans la grammaire ci-dessous.

$$\begin{aligned} \text{AN} + \text{TA} &= \text{DOL}; \text{ODE} + \text{TEL} = \text{LAD}; \text{SUT} + \text{TUT} = \text{NUE}; \\ \text{AN} + \text{ODE} &= \text{TUT}; \text{TA} + \text{TEL} = \text{SUT}; \text{DOL} + \text{LAD} = \text{NUE} \end{aligned}$$

2.5 Multiplications

On va distinguer plusieurs types de multiplications. D'abord, la forme classique de la multiplication courte a un seul terme dans le second membre.

$$\begin{aligned} \text{GREY} \times \text{BLUE} &= \text{DARKBLUE} \\ 8601 \times 3450 &= 29673450 \end{aligned}$$

Il y a aussi les multiplications courtes doublement vrai.

$$\begin{aligned} \text{CINQ} \times \text{SIX} &= \text{TRENTE} \\ 5409 \times 142 &= 768078 \end{aligned}$$

Et, il y a des multiplications courtes avec plusieurs termes dans le second membre comme dans $\text{ORC} \times \text{FREAK} = \text{ELF} \times \text{FAIRY}$.

Une dernière variante amusante est la multiplication longue, une méthode de calcul classique enseignée à l'école.

	MU		16
×	MU	×	16
	NU		96
+	MU•	+	160
	TAU		256

Une multiplication longue décompose le produit d'une multiplicande et d'un multiplicateur en une somme des produits du multiplicande avec la valeur de chaque chiffre du multiplicateur. Une multiplication longue est divisée en trois parties : le produit du multiplicande et du multiplicateur ; la somme des produits partiels, et le résultat du produit.

Pour définir un tel cryptarithme, la grammaire a besoin de l'opérateur de conjonction comme pour les mots croisés, mais aussi de la multiplication par un nombre, ici une puissance de la base.

$$\begin{aligned} \text{MU} * \text{MU} &= \text{TAU}; \text{MU} * \text{M} = \text{MU}; \text{MU} * \text{U} = \text{NU}; \\ \text{NU} * '1' + \text{MU} * '10' &= \text{TAU}; \end{aligned}$$

3 Résolution d'un cryptarithme

Nous expliquons ici la compilation d'un cryptarithme vers un modèle en programmation par contraintes. Cette explication est divisée en trois parties. La section 3.1 modélise l'affectation des chiffres aux lettres, soit l'application des règles **i**, **iii**, et **iv**. La section 3.2 modélise l'équation en arithmétique simple précision, soit l'application de la règle **ii**. Et, la section 3.3 modélise l'équation en arithmétique multiprécision, mais est restreint aux additions. En base décimale, la valeur d'un mot de dix caractères n'est plus représentable en simple précision.

3.1 Affectation des chiffres aux lettres

Un mot w de longueur $|w|$ est représenté par une suite finie de lettres $w = l_{|w|-1}l_{|w|-2} \dots l_1l_0$. L'entrée d'un cryptarithme définit un ensemble W de mots et un alphabet A , l'union des lettres des mots.

$$A = \bigcup_{w \in W} \bigcup_{i=0}^{|w|-1} \{l_i\}$$

Soit b la base arithmétique du cryptarithme, la variable entière $x_l \in [0, b-1]$ représente le chiffre associé à la lettre $l \in A$ imposant ainsi la règle **i**.

Le nombre d'occurrences de chaque chiffre parmi les lettres est imposé par la contrainte *global cardinality* [17]. On généralise la règle **iii** pour traiter le cas où il y a plus de lettres que de chiffres dans la base.

$$\left\lfloor \frac{|A|}{b} \right\rfloor \leq |\{l \in A \mid x_l = v\}| \leq \left\lceil \frac{|A|}{b} \right\rceil \quad (1)$$

La règle **iv** est imposée par une contrainte arithmétique.

$$x_{l_{|w|-1}} > 0 \quad \forall w \in W \quad (2)$$

3.2 Arithmétique simple précision

L'équation est construite en créant des variables auxiliaires associées aux valeurs des mots et en utilisant l'arithmétique du solveur pour vérifier l'équation (règle **ii**).

La variable auxiliaire V_w représente la valeur du mot $w \in W$ dans la base b . Elle peut être définie par la méthode d'exponentiation.

$$V_w = \sum_{i=0}^{|w|-1} b^i \times x_{l_i} \quad \forall w \in W \quad (3)$$

Une alternative est la méthode de Ruffini-Horner.

$$\begin{aligned} V_w &= ((\dots((bx_{l_{|w|-1}} + x_{l_{|w|-2}})b + x_{l_{|w|-3}})b \\ &\quad + \dots)b + x_{l_1})b + x_{l_0} \quad \forall w \in W \quad (4) \end{aligned}$$

3.3 Arithmétique multiprécision

En multiprécision, il n'est plus possible d'utiliser une variable auxiliaire représentant la valeur d'un mot pour éviter un dépassement de capacité. Il est quelquefois possible de changer l'arithmétique d'un programme, mais c'est plus

compliqué pour un solveur. Il n'existe pas non plus de solveurs multiprécision à notre connaissance. Donc, la seule solution est de proposer un modèle pour le calcul multiprécision. La difficulté est que le résultat d'une opération devient le résultat d'un algorithme et non plus d'une opération gérée par le processeur. Pour le moment, le modèle est restreint à l'addition qui est l'opération la plus facile, mais aussi la plus importante pour un cryptarithme.

Le modèle calcule la somme des chiffres des opérandes (mots ou nombres) à chaque position, puis propage les retenues pour calculer le résultat final de l'addition. Soit W^1 (resp. W^2) la suite des termes du membre gauche (resp. droit) de l'addition. Chaque mot ou nombre peut être répétés dans plusieurs termes. Soit $m = \max_W |w| - 1$ la position maximale dans un mot, la variable $S_i^j \in \mathbb{N}^+$ représente la somme des chiffres en position i des opérandes du membre j .

$$\sum_{\substack{w \in W^j \\ |w| > i}} x_{li} = S_i^j \quad \forall i \in [0, m], \forall j \in [1, 2] \quad (5)$$

Ensuite, soit la variable auxiliaire $D_i^j \in [0, b - 1]$ représentant le chiffre à la position i du résultat de l'addition du membre j , et la variable auxiliaire $C_i^j \in \mathbb{N}^+$ représentant la retenue à la position i . Les chiffres du résultat sont calculés en propageant les retenues de l'addition sauf la dernière.

$$b \times C_i^j + D_i^j = S_i^j + C_{i-1}^j \quad \forall i \in [1, m], \forall j \in [1, 2] \quad (6)$$

$$b \times C_0^j + D_0^j = S_0^j \quad \forall j \in [1, 2] \quad (7)$$

La dernière retenue n'est pas propagée et la valeur de C_m^j peut donc être supérieure à la base b .

Pour vérifier l'égalité des résultats, tous les chiffres des résultats et la dernière retenue des membres droit et gauche doivent être égaux.

$$D_i^1 = D_i^2 \quad \forall i \in [1, m] \quad (8)$$

$$C_m^1 = C_m^2 \quad (9)$$

4 Génération de cryptarithmes

La génération de cryptarithmes est une tâche bien plus ardue que leur résolution, car l'explosion combinatoire du choix des mots et de l'équation est considérable. Une méthode naturelle pour la contrôler est de restreindre le dictionnaire et la forme de l'équation.

Les méthodes de génération prennent en entrée une liste de mots et énumèrent sous contraintes des cryptarithmes candidats. Chaque candidat est résolu et la décision est prise de le conserver ou non.

Certaines restrictions sont assez naturelles. Un cryptarithme n'est réellement amusant qu'avec des mots d'une langue naturelle. De plus, la difficulté pour le construire est bien moindre avec des mots quelconques, car le calcul devient facile. Résoudre un cryptarithme dans une base non décimale est plus difficile et moins amusant. L'élégance demande à ce que la solution soit unique. Sauf mention du contraire, les restrictions suivantes sont appliquées :

1. Les mots sont en langue naturelle.
2. Il n'y a pas de répétition de mots.
3. Les lettres représentent des chiffres distincts.
4. La base est décimale.
5. La solution est unique.

Chaque modèle est spécifique au type de cryptarithme, mais ils partagent un socle commun pour la sélection des mots. Chaque modèle spécifique impose les restrictions communes et des contraintes supplémentaires sur les longueurs des mots du cryptarithme. En effet, chaque modèle intègre des raisonnements sur la longueur du résultat d'une opération en fonction de la longueur de ses opérandes. Ces raisonnements ne seront pas présentés en détail, car les explications seraient trop longues et ils manquent de généralité. Remarquez qu'aucun modèle de génération ne considère l'affectation, partielle ou totale, des chiffres aux lettres du candidat qui est laissée entièrement à la charge de la résolution. En d'autres termes, lors de la génération rien ne certifie qu'il puisse exister une solution et que celle-ci est unique.

Prouver qu'un cryptarithme ne provoque pas de dépassement d'entier n'est pas trivial. À titre exemple, le cryptarithme suivant admet une solution unique.

$$\begin{aligned} T^E \times S^T &= \text{TEST} \\ 2^5 \times 9^2 &= 2592 \end{aligned}$$

Le membre de droite étant sur quatre caractères cela nous assure de ne pas avoir de dépassement d'entier. Cependant si on ne considère que le membre de gauche, celui-ci peut provoquer un dépassement d'entier. L'ordre dans lequel les contraintes associées à cette équation vont être construites dépend du solveur sous-jacent. C'est pourquoi nous n'intégrons pas de vérification au moment de la génération et nous nous reposons uniquement sur le solveur.

4.1 Sélection d'un ensemble de mots

Le modèle de sélection des mots joue un rôle central. Il définit des variables booléennes de décision y_w qui indiquent la présence d'un mot w de l'ensemble W . Les variables auxiliaires N et L représentent le nombre d'opérandes et la longueur du plus long mot.

$$N = \sum_W y_w \quad (10)$$

$$L = \max_W |w| \times y_w \quad (11)$$

Le modèle déclare des contraintes pour que les lettres représentent des chiffres distincts. Il est nécessaire que le nombre de lettres distinctes soit inférieur ou égal à la base b . La variable booléenne X_l indique la présence de la lettre $l \in A$. Soit $W_l \subseteq W$ l'ensemble des mots avec la lettre l .

$$W_l = \{w \in W \mid \exists i \in [0, |w| - 1], l_i = l\} \quad \forall l \in A$$

La contrainte (12) impose que la variable X_l soit vraie si et seulement si un mot de W_l est présent. Soit b la base arithmétique du cryptarithme, la contrainte (13) impose que

les lettres représentent des chiffres distincts.

$$X_l = \bigvee_{w_l} y_w \quad \forall l \in A \quad (12)$$

$$\sum_A X_l \leq b \quad (13)$$

4.2 Addition et multiplication

La variable de décision y_w^1 (resp. y_w^2) indique la présence d'un mot $w \in W$ dans le membre gauche (resp. droit). Les variables auxiliaires associées sont aussi définies par les contraintes (10) et (11). Par contre, les contraintes (12) et (13) pour les lettres distinctes ne sont pas définies, car elles seraient redondantes.

La contrainte (14) interdit la répétition des mots à gauche et à droite. La contrainte (15) brise partiellement les symétries.

$$y_w = y_w^1 + y_w^2 \quad (14)$$

$$L^1 \leq L^2 \quad (15)$$

Des contraintes supplémentaires sur la longueur du résultat en fonction de la longueur des opérandes sont aussi déclarées. Elles ne sont pas présentées ici, car les explications seraient trop longues et elles manquent encore de généralité.

4.3 Doublement vrai

On va simplement étendre les modèles pour l'addition et la multiplication. Soit v_w la valeur du nombre écrit dans le mot $w \in W$, on ajoute la contrainte (16) pour l'addition.

$$\sum_W v_w \times y_w^1 = \sum_W v_w \times y_w^2 \quad (16)$$

Pour la multiplication, poser la contrainte de produit équivalente est difficile en programmation par contraintes à cause de l'arithmétique simple précision. Passer au logarithme nécessite la gestion des domaines réels ou flottants ce qui n'est pas fréquent. En pratique, on calcule une approximation naïve de la contrainte (17) en normalisant et arrondissant au plus près le logarithme.

$$\left| \sum_W \log(v_w) \times (y_w^1 - y_w^2) \right| \leq \epsilon \quad (17)$$

4.4 Mots croisés et multiplication longue

Nous ne détaillerons pas les modèles de mots croisés et de multiplication longue. Le modèle de mots croisés inclut un modèle d'affectation des mots dans la grille couplé à un modèle d'addition pour chaque ligne et chaque colonne. Le modèle de multiplication longue est complètement spécifique, car les contraintes sur la longueur des opérandes sont très fortes.

5 Évaluation expérimentale

Nous présentons ici les expériences menées pour évaluer nos approches de résolution et de génération de

cryptarithmes et les comparer avec le solveur spécialisé CRYPT [21]. Le solveur CRYPT écrit en C utilise un algorithme de retour arrière. Il se restreint aux cryptarithmes en simple précision, avec additions décimales et des lettres minuscules et majuscules de l'alphabet latin. Seules les additions seront évaluées, car c'est la forme la plus classique et pour comparaison avec CRYPT qui n'accepte que des additions.

La section 5.1 compare les performances de résolution d'additions de nos solveurs PPC et du solveur CRYPT. La section 5.2 étudie le passage à l'échelle de notre solveur lorsque la base arithmétique augmente. La section 5.3 évalue et compare les performances pour la génération d'additions de notre approche et du solveur CRYPT.

Les expériences ont eu lieu sur une machine Dell avec 256 GB de RAM et 4 Intel E7-4870 2.40 GHz processeurs sous CentOS Linux 7.9 (chaque processeur a 10 cœurs).

5.1 Performances de la résolution

Nous analysons les performances de la résolution sur les additions de la collection de la section 6. Toutes les instances ont donc une solution unique. Notre jeu d'instances est composé de 35800 additions dont les mots ont 8 lettres ou moins. Pour résoudre un cryptarithme, un solveur trouve une solution, puis prouve qu'aucune autre existe. Ce choix est guidé par la difficulté et le moindre intérêt d'une collection de cryptarithmes sans ou avec plusieurs solutions.

Nous évaluons quatre solveurs :

SCALAR le modèle simple précision de la section 3.2 avec la contrainte (3);

HORNER le même modèle avec la contrainte (4);

BIGNUM le modèle multiprécision de la section 3.3;

CRYPT un solveur spécialisé [21] écrit en C.

Les solveurs PPC résolvent les 35800 instances. Le solveur CRYPT résout seulement les 35104 en alphabet latin.

La figure 1 donne le nombre d'instances résolues en fonction du temps (en secondes) pour chaque solveur. Le premier constat est sévère puisque le solveur spécialisé CRYPT (à gauche) est extrêmement rapide en résolvant chaque instance en moins d'une demi-seconde. De manière symétrique, la contrainte HORNER (à droite) est de loin la plus inefficace et ne permet même pas de résoudre toutes les instances dans le temps imparti (la ligne verticale à l'extrémité droite). Au milieu, le modèle simple précision SCALAR et multiprécision BIGNUM sont presque 100 fois moins rapides que CRYPT. Après un démarrage plus lent, le modèle SCALAR prend un léger avantage sur BIGNUM et résout plus de 90% des instances en moins d'une seconde. Le temps de résolution moyen de SCALAR est trois fois celui de BIGNUM, mais les temps médians sont quasiment égaux.

Enfin, la complexité d'un solveur de contraintes induit un surcoût non négligeable par rapport à un algorithme très spécialisé sur des additions décimales. Cependant, les performances sont acceptables pour une utilisation interactive du solveur et sont compensées par la généralité de l'approche PPC.

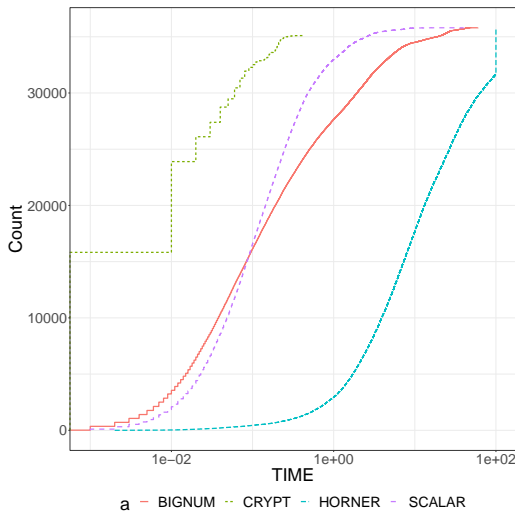


FIGURE 1 – Instances résolues en fonction du temps.

5.2 Passage à l'échelle de la résolution

Nous n'avons trouvé que trois instances non décimales dans notre bibliographie, deux en base 15 et une en base 16. Seul le solveur BIGNUM est capable de les résoudre.

Le tableau 1 donne le nombre d'opérandes, le nombre total de lettres, et la longueur maximale des mots de chaque cryptarithme accompagnés du temps de résolution et du nombre de nœud du solveur. Les métriques du solveur indiquent que la difficulté des instances a considérablement augmenté. Chacune de ces additions en base 15 ou 16 est plus difficile que n'importe quel cryptarithme décimal de la collection.

b	$ W $	$\sum w $	$\max w $	Time (s)	Nodes
15	47	215	9	2185.619	549106
15	54	249	9	11716.522	1561266
16	33	154	10	2968.624	1573488

TABLE 1 – Résolution d'additions en base 15 et 16.

Les investigations sur le passage à l'échelle n'ont pas été poussées plus loin. D'abord, générer une collection en base non décimale est une tâche difficile. Puis, l'intérêt d'une telle collection est faible dans les cas d'utilisation pédagogiques, car le calcul devient compliqué ce qui n'est pas très amusant.

En conclusion, la réputation de facilité du problème reflète l'attention quasi-exclusive à la base 10.

5.3 Performances de la génération

Une méthode de génération examine des candidats générés à partir d'une liste de mots, et détermine s'ils ont une solution unique ou non. Une méthode doit filtrer les candidats pour être efficace, car la combinatoire des additions même sans répétition est considérable. Remarquez que la résolution des candidats est séquentielle, et non parallèle. Ici,

nous comparons les performances de notre approche CRYPTATOR et du solveur CRYPT pour générer toutes les additions avec un nombre d'opérandes fixé et un terme unique à droite. CRYPTATOR utilise le modèle SCALAR si aucun mot n'a plus de 8 lettres et BIGNUM sinon.

Pour ces expérimentations, nous utilisons deux listes de mots. La première est composée de 24 mots correspondant aux noms des lettres de l'alphabet grec (alpha, beta, ...). La seconde liste est composée de 143 mots correspondant à des couleurs. La table 2 donne quelques caractéristiques des listes de mots utilisées pour la génération de cryptarithmes.

	$ W $	$\min w $	$\text{med} w $	$\text{mean} w $	$\max w $
alphabet	24	2	4	4.167	7
couleurs	143	3	9	9.168	20

TABLE 2 – Caractéristiques des listes de mots.

Le tableau 3 récapitule les performances des deux méthodes avec les indicateurs suivants. Le nombre n indique le nombre de termes à gauche de l'addition. Le nombre s donne le nombre de cryptarithmes avec une solution unique. La partie haute donne les résultats pour l'alphabet grec, et la basse pour les couleurs. Pour chaque méthode, le tableau donne le temps t de génération avec résolution des candidats en secondes, et le nombre c de candidats. Pour CRYPTATOR, le tableau indique en plus le temps t_c de génération sans résolution. À droite, le nombre \bar{c} d'additions sans répétition est une borne supérieure sur le nombre de candidats.

n	s	CRYPTATOR			CRYPT		
		t_c	t	c	t	c	\bar{c}
2	4	0.8	13.0	1.4K	0.1	2.9K	6.1K
3	38	1.5	165.7	5.9K	1.9	17.0K	42.6K
4	128	1.7	631.1	11.4K	10.9	75.2K	215.2K
5	207	1.5	652.4	10.6K	26.1	261.8K	807.6K
6	184	1.0	540.4	4.7K	28.6	735.8K	2.4M
7	30	0.7	120.5	817	21.5	1.7M	5.9M
8	2	0.6	4.1	13	11.4	3.3M	11.8M
9	0	0.4	0.4	0	13.7	5.4M	19.6M
2	66	5.6	67.7	7.7K	1.8	558.9K	1.4M
3	315	6.7	307.0	17.2K	47.0	20.2M	66.8M
4	357	7.3	390.4	24.3K	1232.5	574.8M	2.3G
5	163	6.8	447.2	25.8K	31148.9	557.8M	64.1G
6	46	6.2	537.5	20.4K	–	–	1.5T

TABLE 3 – Génération avec l'alphabet grec (24 mots) en haut ou des couleurs (143 mots) en bas.

Premièrement, la croissance du nombre \bar{c} d'additions sans répétition est telle qu'une méthode de génération doit absolument en éliminer une large majorité pour passer à l'échelle. On observe aussi que le nombre s de cryptarithmes avec une solution unique est très petit en comparaison. Deuxièmement, la génération des candidats avec CRYPTATOR prend peu de temps t_c pour l'alphabet grec et un temps court par rapport au temps t de résolution des

candidats pour les couleurs. L'intérêt est évident puisque le nombre c de candidats de CRYPTATOR est inférieur d'un ou plusieurs de grandeurs à celui de CRYPT pour les couleurs et dès que n atteint 5 pour l'alphabet. Troisièmement, la comparaison des temps t de génération avec résolution est plus contrastée. Les meilleurs temps sont indiqués en gras. Pour l'alphabet, la réduction du nombre c de candidats par CRYPTATOR par rapport à CRYPT n'est pas suffisante pour compenser sa moindre rapidité de résolution pour les plus petites valeurs de n . Par contre, la comparaison s'inverse pour les plus grandes valeurs de n pour lesquelles le nombre de candidats devient très faible. Pour les couleurs, la liste est plus grande et l'explosion combinatoire bien plus rapide. La plus grande rapidité de CRYPT ne compense plus du tout le plus grand nombre c de candidats. Le temps de CRYPT à l'avant dernière ligne dépasse largement le temps cumulé de CRYPTATOR. Pour la dernière ligne, la résolution a été interrompue après 4 jours de calcul en ayant découvert seulement 2 cryptarithmes sur 46.

Ainsi, si la programmation par contraintes induit un surcoût pour la résolution par rapport à un solveur spécialisé comme CRYPT, elle reprend largement l'avantage pour la génération. La collection de la section 6 n'aurait pas pu être créée avec CRYPT.

Une perspective intéressante est d'hybrider la méthode de génération par la programmation par contraintes avec le solveur CRYPT pour la résolution.

6 Une collection remarquable

Nous présentons les résultats des expériences menées pour créer une collection de cryptarithmes complexes et remarquables. Le temps consacré à cette tâche est important, mais ne fait pas l'objet d'une analyse. Les résultats sont plutôt analysés à travers les caractéristiques et nouveautés de la collection.

6.1 Additions

Les additions sont la forme la plus fréquente de cryptarithmes. Par conséquent, elles constituent la majorité de la collection et se répartissent en trois groupes :

- des collections thématiques les plus complètes possibles ;
- des collections obtenues par échantillonnage du dictionnaire français présentant des caractéristiques variées et remarquables en termes de nombres d'opérandes ou de longueurs des mots ;
- des collections de cryptarithmes doublement vrai dans une quinzaine de langues différentes pour les sommes de 1 à 500.

Le tableau 4 présente quelques statistiques de ces additions, de gauche à droite, leur nombre d'opérandes, leur nombre total de lettres, et les longueurs moyenne, minimale, et maximale des mots. On remarque d'abord que la collection est assez variée en termes de nombre d'opérandes et de nombre total de lettres. Une majorité d'additions ont peu d'opérandes, mais un petit quart a plus de vingt opérandes. Même si le plus long cryptarithme connu est un ovni de 200 mots (avec répétitions), la plus longue addi-

tion atteint quand même 49 opérandes. Les trois quarts des cryptarithmes sont composés de mots courts entre trois et six lettres.

Indicateur	$ W $	$\sum w $	mean $ w $	min $ w $	max $ w $
Minimum	3.0	12.0	3.0	2.0	4.0
1er Quartile	5.0	27.0	4.5	3.0	6.0
Médiane	7.0	38.0	4.9	3.0	6.0
Moyenne	13.5	64.1	5.0	3.3	6.3
3ème Quartile	18.0	88.0	5.2	4.0	6.0
Maximum	49.0	216.0	12.8	12.0	20.0

TABLE 4 – Statistiques sur les 36821 additions.

La figure 2 présente une carte thermique dont l'abscisse représente le nombre d'opérandes et l'ordonnée la longueur du mot le plus court. Le gradient de couleur indique le logarithme décimal du nombre de cryptarithmes dans la collection. Le nombre de cryptarithmes par catégorie ($|W|$, min $|w|$, max $|w|$) est limité à mille pour la méthode par échantillonnage du dictionnaire. La zone claire, donc dense, en bas à gauche correspond aux collections thématiques. La zone foncée, donc peu dense, en haut à gauche correspond aux cryptarithmes avec de longs mots. La zone moins foncée, en bas à droite correspond aux cryptarithmes avec beaucoup d'opérandes.

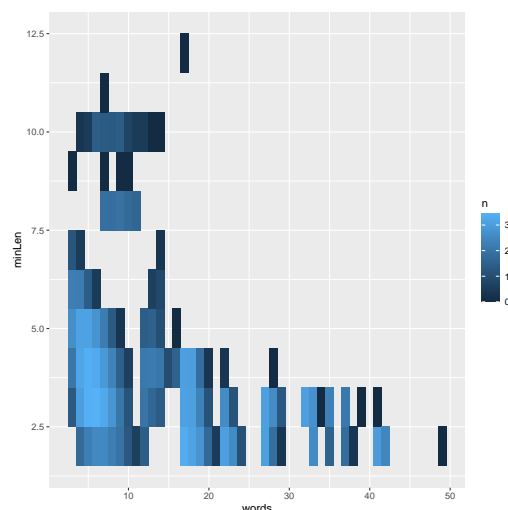


FIGURE 2 – Carte thermique des additions.

Certains des plus longs cryptarithmes, et le plus long en particulier, ne sont composés que de palindromes !

```
AA + ALLA + ANA + ANONA + ARA + ASA + AXA + ELLE + ERE + ERRE
+ESSE + ETE + ETETE + EUE + NANAN + NON + OXO + REER + ROTOR
+SALAS + SANAS + SAS + SASSAS + SELLES + SENES + SENNES+
SERES + SERRES + SES + SEXES + SOLOS + SONOS + SOS + STATS+
STOTS + STUUTS + SUS + TALAT + TALLAT + TANNAT + TARAT+
TASSAT + TATAT + TAXAT + TET + TNT + TOT + TUT = NAURUAN
```

Finalement, la découverte de cryptarithmes doublement vrai s'est avérée très fructueuse avec presque mille cryptarithmes dans dix langues différentes, dont plus de la moitié en Hindi !

SEIS + SETENTA + TRESIENTOSCATORCE = TRESCIENTOSNOVENTA
 ZERO + TRES + SEIS + CIENTOEDOIS = CENTOEONZE
 BES + ON + ONIKI + ELLIUC = SEKSEN

6.2 Multiplications

Cette fonctionnalité est plus récente et limitée par l'arithmétique simple précision, mais elle est également plus rare. Il a été assez rapide de générer quelques milliers de multiplications thématiques, ainsi que des doublement vrai. L'échantillonnage du dictionnaire est inutile, car l'arithmétique simple précision limite le nombre d'opérandes et la longueur des mots.

Nous n'avons réussi à générer pour le moment qu'une dizaine de multiplications longues dont les résultats ont au plus cinq lettres. Cependant, nous retrouvons rapidement celles-ci lorsque la liste de mots en contient.

6.3 Mots croisés

La collection contient une centaine de mots croisés composés de mots de trois lettres ou moins. À notre connaissance, c'est la première fois qu'ils sont créés dans une langue naturelle.

7 Diffusion scientifique

La bibliothèque CRYPTATOR est développée en Java et utilise Maven pour la compilation et le déploiement automatique du projet. Ses dépendances principales sont le solveur Choco [5], le générateur de parseur antlr4 [16], le parseur de ligne de commandes args4j [11], et le framework de tests unitaire junit 5 [2].

Plus précisément, Choco est utilisé pour la modélisation, la résolution, et la génération de cryptarithmes sous la forme de modèles en programmation par contraintes. Antlr4 est utilisé pour définir la grammaire, lire l'équation du problème, et la génération de l'arbre syntaxique correspondant à cette équation. Args4j est utilisé pour définir les options et lire les arguments de la ligne de commandes. Ces options permettent de changer les niveaux de verbosité dans la console, de configurer les modèles, de choisir les variantes, de relâcher les contraintes. La librairie junit 5 sert à tester profondément CRYPTATOR. Avec une couverture du code dépassant les 85%, les tests vérifient le bon fonctionnement de la résolution et de la génération. Entre outre, de nombreux résultats ont été doublement validés avec le solveur CRYPT ou les collections de cryptarithmes en ligne.

Une application mobile² est à un stade précoce de développement. Ses fonctionnalités basées sur la bibliothèque Java intègrons la résolution et la génération de cryptarithmes, mais aussi un jeu pédagogique autour de leur résolution. Le support pour la création de l'application est le framework *React Native* [15]. Ce framework permet la création d'applications utilisables autant sur les systèmes Android que sur les systèmes iOS ou sur le web. Un objectif principal est de faire connaître au grand public la programmation par contraintes par le vecteur des cryptarithmes. L'application

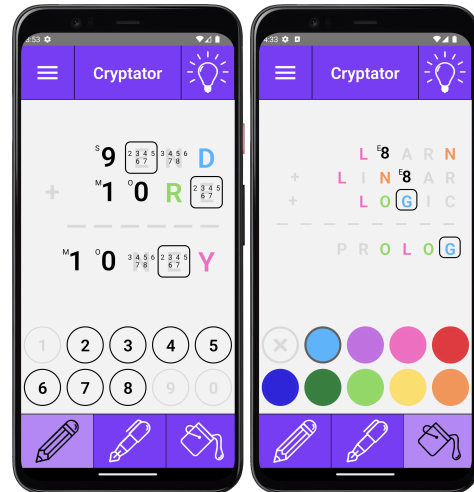


FIGURE 3 – L'application mobile du CRYPTATOR

pourra s'utiliser en autonomie ou dans un cadre pédagogique.

L'application doit permettre de saisir et visualiser un cryptarithme, une solution, ou une liste de mots. Le jeu doit aussi permettre d'annoter des cryptarithmes, de construire une solution même partielle, ou d'afficher des indices. Le jeu devra aussi proposer différents niveaux de difficulté pour s'adapter à tous les publics.

Le mode de résolution des cryptarithmes permet d'affecter à chaque lettre une valeur numérique. Lorsqu'une lettre est affectée à une valeur, cette valeur est supprimée des domaines des autres lettres. Lorsque toutes les lettres sont affectées à une valeur, le cryptarithme est résolu. Lorsque l'utilisateur affecte une valeur à une occurrence d'une lettre, cette valeur est affectée à toutes les autres occurrences de cette lettre. En sélectionnant une lettre, on peut ainsi voir les valeurs possibles pour cette lettre. Pour aider à la résolution, il est possible d'annoter les lettres par des valeurs encore disponibles dans leurs domaines. Il est donc possible d'attribuer plusieurs valeurs provisoires à une lettre.

La figure 3 présente une capture d'application d'un prototype du jeu. Le concept pédagogique est de représenter visuellement le domaine des lettres et des chiffres. L'utilisateur résout le problème en réduisant les domaines par des décisions prises graphiquement avec l'outil stylo. Il peut aussi annoter ces domaines avec l'outil pinceau et colorier les lettres et chiffres avec l'outil seau de peinture. Le moteur de jeu peut indiquer quand une décision invalide est prise ou afficher des indices avec l'outil ampoule lumineuse.

8 Conclusion

Nous proposons une approche générale, efficace, et facile d'utilisation pour la résolution et la génération de cryptarithmes en programmation par contraintes. Ce travail n'a jamais été réalisé à notre connaissance malgré la popularité du problème. L'approche est très générale en simple précision, c'est-à-dire quand les mots sont courts, mais restreinte aux additions en multiprécision. La résolution est ef-

2. <https://github.com/FissoreD/CryptatorApp>

ficace, mais reste largement dominé par un algorithme spécialisé moins général. La génération est sans conteste un point fort comme en atteste la variété de la collection de cryptarithmes. L'approche est disponible actuellement sous la forme d'une bibliothèque Java et d'une application en ligne de commande.

On peut dégager trois perspectives principales. L'ajout de la multiplication pour généraliser le modèle multiprécision. L'hybridation avec l'algorithme spécialisé améliorerait l'efficacité de la génération. La finalisation de l'application mobile avec les améliorations et les ajouts nécessaires et son adaptation à une page web pour faciliter l'utilisation et la diffusion.

Références

- [1] CSPLib : A problem library for constraints. <http://www.csplib.org>, 1999.
- [2] junit 5. <https://junit.org/junit5/>, 2023.
- [3] Maxey Brooke. *150 Puzzles in Crypt-Arithmetic*. Dover Publications, Inc., 1963. URL <http://cryptarithms.awardspace.us/150-puzzles-in-crypt-arithmetic.pdf>.
- [4] V. Chandra Prakash, V. Kantharao, JKR Sastry, and V. Bala Chandrika. Expert system for building cognitive model of a student using crypt arithmetic game and for career assessment. *International Journal of Recent Technology and Engineering*, 7 :684–689, 03 2019.
- [5] Choco Team. Choco : an Open-Source Java Constraint Programming Library. www.choco-solver.org, 2023.
- [6] Truman Collins. Alphametic puzzles. <http://www.tkcs-collins.com/truman/alphamet/index.shtml>, 2023.
- [7] F. Rea Cyrus. *KLOOTO Games Cryptodigits*. CreateSpace Independent Publishing Platform, 2015. ISBN 151681973X.
- [8] Henry Dudeney. The strand magazine vol. 68, juillet 1924, p. 97 et 214. 1924.
- [9] D Epstein. On the NP-completeness of cryptarithms. *SIGACT News*, 18(3) :38–40, apr 1987. ISSN 0163-5700. doi : 10.1145/24658.24662. URL <https://doi.org/10.1145/24658.24662>.
- [10] Steven Kahan. *At last!! Encoded totals second addition*. Baywood Publishing Co., 1994.
- [11] Kohsuke Kawaguchi. args4j : Java command line arguments parser. <https://args4j.kohsuke.org/>, 2023.
- [12] Mike Keith. An alphametic page. <http://www.cadaeic.net/alphas.htm>, 2023.
- [13] Joseph S. Madachy. *Madachy's Mathematical Recreations*. Dover Pubns, 1979.
- [14] Mahmoha. *Alphametic Puzzle - Numbers Behind Letters . Special Edition : Names of Countries with Colorful Flags*. 2022. ISBN 979-8360917267.
- [15] Meta Open Source. React native. <https://reactnative.dev/>, 2022.
- [16] Terence Parr. antlr4. <https://www.antlr.org/>, 2023.
- [17] Jean-Charles Régin. Generalized arc consistency for global cardinality constraint. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, page 209–215. AAAI Press, 1996. ISBN 026251091X.
- [18] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, USA, 2006. ISBN 0444527265.
- [19] Torsten Sillke. Alphametics. <https://www.math.uni-bielefeld.de/~sillke/PUZZLES/ALPHAMETIC/>, 2023.
- [20] Jorge A. C. B. Soares. Cryptarithms online. <http://cryptarithms.awardspace.us/index.html>, 2002.
- [21] Naoyuki Tamura. Cryptarithmic puzzle solver. <https://tamura70.gitlab.io/web-puzzle/cryptarithm/>, 2023.
- [22] S Widodo, U Najati, and P Rahayu. A study of cryptarithmic problem-solving in elementary school. *Journal of Physics : Conference Series*, 1318(1) : 012120, oct 2019. doi : 10.1088/1742-6596/1318/1/012120. URL <https://dx.doi.org/10.1088/1742-6596/1318/1/012120>.

Une contrainte globale pour l'ordonnancement des transferts de données dans les missions spatiales

J. Rouzot^{1,2}, C. Artigues¹, P. Garnier², E. Hebrard¹, P. Lopez¹

¹ LAAS-CNRS, Université de Toulouse, CNRS, Toulouse

² IRAP, Université de Toulouse, CNRS, UT3, CNES, Toulouse

Résumé

Dans le cadre des missions spatiales où les ressources à bord sont limitées, il est essentiel de transférer les données acquises de manière efficace afin de limiter le risque de saturation des mémoires. Nous nous intéressons ici en particulier à l'ordonnancement des transferts de données de la mission Rosetta de l'ESA (l'agence spatiale européenne). Ce problème a été résolu par des méthodes heuristiques qui ont donné de bons résultats, voire des résultats optimaux dans certains cas. Dans la lignée d'un travail précédent, nous proposons dans cet article un modèle de programmation par contraintes pour résoudre exactement ce problème, nous proposons une nouvelle contrainte globale pour le partage de bande passante en fonction de priorités, ainsi qu'une contrainte globale pour les classements de type Dense Ranking.

Des premiers résultats montrent la pertinence et l'efficacité d'une telle approche exacte pour la résolution de ce problème. Un des buts dans la conception de ces contraintes est leur généralité, afin qu'elles puissent être appliquées à toute mission spatiale nécessitant des transferts de données, ainsi qu'à tout problème nécessitant un classement Dense Ranking.

Mots-clés

Ordonnancement, Programmation par contraintes, Contrainte globale, Mission d'exploration spatiale, Dense Ranking

1 Introduction

La mission Rosetta a été lancée par l'Agence spatiale européenne (ESA) en 2004 dans le but d'analyser la comète 67P/Tchourioumov-Gerassimenko, et a abouti en 2014. Cette mission est représentative de nombreuses missions soutenues par le "Deep Space Network", et l'étudier permet donc de concevoir des systèmes de planification efficaces pour les missions futures.

Nous nous intéressons ici au problème du vidage des données de l'orbiteur Rosetta, défini par Chien, Rabideau et al. [2, 5], similaire au problème de vidage des données de l'atterrisseur Philae [7]. Dans les deux cas, un certain nombre d'activités, qui produisent des données temporairement stockées dans la mémoire de leur instrument, doivent être ordonnancées. De plus, un ordre de priorité doit être

établi pour chacune des fenêtres de visibilité entre la sonde et la Terre. Cet ordre de priorité dicte le partage de bande passante entre les différents instruments lors du transfert de données pendant cette fenêtre. L'objectif est de minimiser le pic d'utilisation maximal des mémoires (i.e. maximiser la marge de sécurité, où la marge de sécurité est le ratio entre la capacité de la mémoire et son remplissage maximal au cours de la mission).

Dans le cas de Philae, l'ordre de priorité est considéré fixe, et les variables de décision correspondent à l'ordonnancement des activités. Dans le cas de Rosetta, à l'inverse, l'ordonnancement des activités est une entrée, et les variables correspondent à l'ordre de priorité pour le transfert des données.

Il s'agit d'un ordre partiel, plus précisément un *classement* : les mémoires de même priorité se partagent la bande passante grâce à un algorithme de type Round-Robin (file d'attente circulaire). Ainsi, les mémoires avec la plus forte priorité peuvent utiliser toute la bande passante initiale, puis la bande passante restante est redistribuée jusqu'à ce qu'il n'en reste plus, ou que les mémoires soient toutes vides. On considère ici que le remplissage des mémoires est complètement déterminé par la planification des expériences scientifiques et donc connu à l'avance. Les priorités ne peuvent être changées qu'en début de visibilité, mais sont déterminées en amont, même si elles peuvent être amenées à être recalculées au cours de la mission.

Comme la relation entre l'utilisation des mémoires et les priorités est complexe, utiliser une modélisation de programmation par contraintes classique est très peu efficace. Afin d'accélérer la résolution, nous avons donc développé les contraintes globales suivantes :

- La contrainte `PRIORITY TRANSFER` permet de déterminer le pic maximal d'utilisation des mémoires au cours d'une fenêtre de vidage et l'utilisation des mémoires à la fin de cette fenêtre en fonction d'une affectation de priorités. Cette contrainte permet également d'inférer de nouvelles bornes sur les variables de priorité en fonction de la borne supérieure du pic d'utilisation maximal courant. Cette contrainte utilise les algorithmes décrits par Hebrard et al. [3].
- La contrainte `DENSE RANKING` permet de filtrer efficacement les domaines des variables de priorité pour obtenir un classement sans symétries.

Nous décrivons dans cet article les contraintes globales précédentes, puis un algorithme de calcul rapide de bornes inférieures sur les mémoires et les pics d'utilisation pour chaque fenêtre de vidage. Nous concluons sur l'intérêt de ces contraintes globales ainsi que sur les perspectives pour améliorer le modèle.

2 Modélisation du problème

Dans le problème du vidage des données de Rosetta, ou *overlapping Memory Dumping Problem (oMDP)* [6], on considère m fenêtres de vidage aux dates de début et fin $[s_j, e_j]_{\{j=1..m\}}$ pendant lesquelles la bande passante à partager entre les mémoires est une constante δ_j . On a n mémoires concurrentes, avec une capacité limitée. On note $r_{i,j}$ l'usage maximum de la mémoire i par rapport à sa capacité au cours de la fenêtre j . Pour ce problème, nous considérons que la fonction de remplissage de chaque mémoire au cours du temps est une fonction constante par morceaux. Cela implique que l'évolution de l'utilisation des mémoires au cours du temps suit une fonction linéaire par morceaux, puisque le transfert des données est aussi constant par morceaux.

Nous définissons les priorités de la façon suivante :

- Des groupes de priorité sont définis, pouvant prendre une valeur de 1 à n . Le groupe 1 est le plus prioritaire, tandis que le groupe n est le moins prioritaire.
- Les mémoires sont affectées à un groupe de priorité pour chaque fenêtre de vidage ; plusieurs mémoires peuvent être affectées au même groupe.
- Les mémoires au sein d'un même groupe de priorité partagent la bande passante disponible grâce à un algorithme de type Round-Robin.

Nous proposons un modèle de programmation par contraintes pour résoudre les instances du oMDP. Dans notre modèle, nous séparons les instances par fenêtre de visibilité j et nous introduisons les variables suivantes :

- $p_{i,j}$ représente le groupe de priorité de la mémoire i au cours de la fenêtre j .
- $mem_{i,j}^s$ et $mem_{i,j}^e$ représentent respectivement l'utilisation de la mémoire i au début et à la fin de la fenêtre j .
- $r_{i,j}$ représente le pic d'utilisation maximum de la mémoire i au cours de la fenêtre j .
- $rmax$ représente le pic d'utilisation maximum des mémoires sur l'instance globale.

L'objectif est de minimiser l'usage maximal des mémoires sur l'instance : $\min rmax$.

La figure 1 montre l'évolution de l'utilisation des mémoires au cours du temps pour une solution particulière (affectation de priorités pour chaque fenêtre de vidage). Les zones vertes représentent les fenêtres de visibilité, pendant lesquelles les données des mémoires peuvent être transférées sur Terre à un taux δ_j , partagé entre les mémoires en fonction de leur priorité. Pendant ces fenêtres de visibilité, les mémoires ont une priorité fixe, représentée sur la figure 1 en rouge. Pour chaque mémoire, on représente le remplissage

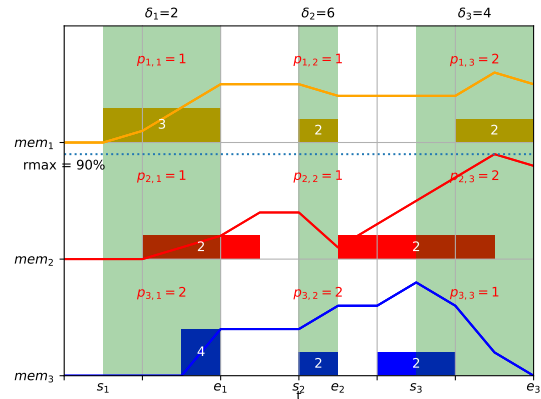


FIGURE 1 – Utilisation des mémoires au cours du temps et pic maximum pour une affectation de priorités.

au cours du temps par des rectangles de couleur où la hauteur représente le taux (constant) de remplissage. L'évolution de l'utilisation de chaque mémoire est représentée par une fonction linéaire par morceaux. Dans cet exemple, la capacité de chaque mémoire est de 10 unités ; le pic maximum est donc atteint pour la mémoire 2, pour une valeur de 9 unités, soit $rmax = r_{2,3} = \frac{9}{10} = 90\%$.

3 Contraintes globales

La principale difficulté pour oMDP est d'exprimer efficacement la relation entre une affectation de priorité et l'évolution de l'utilisation des mémoires résultant. Cette difficulté a motivé le développement de la contrainte globale PRIORITY TRANSFER qui, pour chaque fenêtre j , lie les variables $p_{i,j}, mem_{i,j}^e, mem_{i,j}^s$ et $r_{i,j}$.

De plus, la modélisation des variables de priorité a l'avantage d'être directe par rapport à des priorités relatives entre chaque mémoire, mais comporte de nombreuses symétries qui doivent être cassées pour accélérer la résolution. Nous présentons la contrainte globale DENSE RANKING qui permet de briser ces symétries efficacement.

3.1 Contrainte Priority Transfer

Une fois que les variables $p_{i,j}$ et $mem_{i,j}^s$ sont fixées, on peut calculer l'évolution des mémoires au cours du temps dans la fenêtre j . On utilise l'algorithme SIMULATION [3] pour déterminer les valeurs des variables $mem_{i,j}^e$ et $r_{i,j}$ au cours de la résolution. Cependant, cet algorithme ne permet pas de faire de propagation sur les priorités.

Comme l'objectif de notre modèle est de minimiser le pic d'utilisation maximum de la mémoire, il est possible d'éliminer certaines valeurs dans les domaines des variables de priorité, en fonction de la valeur de la borne supérieure de la variable $rmax$ au cours de la résolution. En effet, on peut montrer que certaines affectations de priorité mènent nécessairement à un dépassement du pic maximum courant, grâce à l'algorithme SINGLE WINDOW [3].

Plus précisément, cet algorithme construit une affectation

de priorités partielles pour un pic d'utilisation maximum donné. L'affectation est partielle car elle ne donne pas une valeur stricte pour les variables de priorité, mais un ordre de priorité relatif entre les mémoires.

Puisque cet ordre partiel doit être respecté pour satisfaire le pic maximal courant sur la fenêtre, on peut inférer de nouvelles bornes sur les variables de priorité sur la fenêtre. L'ordre partiel nous donne un ensemble de relations $p_{i,j} < p_{i',j}$ qui permettent de filtrer les bornes des variables de priorité.

La complexité de l'algorithme `SINGLE WINDOW` étant en $O(n^3 \log(n))$, la contrainte `PRIORITY TRANSFER` permet donc un filtrage des domaines des variables de priorité aussi en $O(n^3 \log(n))$ puisque la mise à jour des domaines, après avoir construit l'ordre partiel entre les mémoires, se fait en $O(n^2)$.

3.2 Contrainte Dense Ranking

Il existe plusieurs façon d'effectuer des classements (ex. classement standard "1224", classement modifié "1334", etc.) dont certains ont déjà été étudiés en programmation par contraintes, comme le classement standard [1].

Dans notre modèle, les priorités sont définies pour chaque mémoire par l'appartenance à un groupe de priorité et on autorise plusieurs mémoires à avoir une priorité égale. Afin d'éviter les symétries dans les solutions réalisables, nous avons décidé d'adopter un classement *Dense Ranking*.

Les règles suivantes doivent être respectées pour qu'une solution soit considérée valide à l'égard d'un *Dense Ranking* :

- Le groupe de priorité 1 doit être non vide.
- Si le groupe de priorité $i + 1$ est non vide, alors le groupe de priorité i doit être non vide.

Pour éliminer les solutions qui ne respectent pas ces règles, nous présentons la contrainte globale `DENSE RANKING` qui permet un filtrage efficace de type cohérence de borne des priorités. L'algorithme de filtrage des domaines est découpé en deux parties :

- L'algorithme `PRIORITY SAT` vérifie si les domaines des variables de priorité permettent de construire une solution qui satisfait la contrainte `DENSE RANKING`.
- L'algorithme `SYMMETRY FILTERING` permet de réaliser la cohérence de bornes en utilisant l'algorithme précédent comme oracle.

Dans `oMDP`, on a n variables de priorité avec des domaines $[1, \dots, n]$. Les domaines des variables de priorité sont valides si et seulement si l'on peut construire une solution en respectant les règles de *Dense Ranking*. L'algorithme `PRIORITY SAT` (algorithme 1) permet de vérifier l'existence d'une solution à partir des domaines des variables de priorité en $O(n \log(n))$. Les variables de priorité p sont d'abord triées par borne inférieure croissante (ligne 2) puis, pour chaque rang de priorité, on ajoute dans un tas h , ordonnées par borne supérieure croissante, toutes les variables de priorité dont la borne inférieure est égale au rang courant (lignes 5-8). S'il est possible d'affecter au moins une variable au rang courant à chaque étape (lignes 9-10), jusqu'à ce que toutes les variables soient affectées, il existe une so-

lution réalisable pour les domaines courants. Si l'on ne peut pas traiter un des rangs (lignes 12-13), et qu'il reste des variables qui n'ont pas été affectées, alors il n'existe pas de solution réalisable pour les domaines courants. Pour chaque rang, il faut retirer du tas les variables qui n'ont pas été affectées, mais dont la borne supérieure est égale au rang courant, ce qui revient à affecter la même priorité à plusieurs variables (ligne 13-14).

En effectuant une recherche dichotomique sur les domaines des variables de priorité, on peut donc faire le filtrage des bornes en $O(n^2 \log(n)^2)$.

La contrainte globale `Dense Ranking` peut s'exprimer de manière équivalente avec la contrainte `AtLeastNValues(N, X)`, basée sur `SoftAllDiff(X)`, qui assure que les variables de l'ensemble X prennent au moins N valeurs distinctes [4]. En effet, `Dense Ranking(X)` est équivalent à `AtLeastNValues(Max(X), X)`. Cependant, l'algorithme pour trouver un support pour la contrainte `AtLeastNValues` se base sur le couplage de cardinalité maximum dans un graphe biparti. Avec $n = |X|$ le nombre de variables et $m = \sum |D(X)|$, l'algorithme est en $O(m n^{\frac{1}{2}})$, soit en $O(n^{\frac{5}{2}})$ pour nos domaines. Le filtrage complet des domaines se fait aussi en $O(n^{\frac{5}{2}})$, ce qui est moins efficace qu'avec notre méthode. De plus, le test de satisfaisabilité est bien plus efficace avec la contrainte globale `Dense Ranking`.

Algorithm 1 PRIORITY SAT

```

1: Input :  $p$ 
2: Trier  $p$  par borne inférieure
3:  $i \leftarrow 0$ ,  $current \leftarrow -1$ 
4: while  $i < n$  do
5:    $current \leftarrow current + 1$ 
6:   while  $p_i.Lb() = current$  do
7:      $h.Add(p_i)$ 
8:      $i \leftarrow i + 1$ 
9:   if not  $h.Empty()$  then
10:     $h.RemoveMin()$ 
11:   else
12:     return False
13:   while  $h.Min().Ub() = current$  do
14:      $h.RemoveMin()$ 
15: return True

```

4 Borne inférieure

Une borne inférieure sur les variables $mem_{i,j}^s$, $mem_{i,j}^e$ et $r_{i,j}$ est facile à déterminer en utilisant l'algorithme `SIMULATION` [3]. Du point de vue d'une mémoire i , le meilleur scénario possible est d'être plus prioritaire que les autres mémoires à chaque fenêtre de vidage. Ainsi, pour chaque mémoire i , en affectant la priorité $p_{i,j} = lb(p_{i,j})$ et $p_{i',j} = ub(p_{i',j}) \forall i' \neq i$, et en supposant que $mem_{i,j}^s$ est égale à sa borne inférieure on peut déterminer un meilleur cas d'utilisation de mémoire i en simulant cette affectation particulière de priorités, et donc une borne inférieure sur $mem_{i,j}^e$ et $r_{i,j}$. L'algorithme `SIMULATION` est de complexité $O(n \log(n))$, ce qui est relativement peu coûteux.

Instances	Dense Ranking	Temps moyen	Temps médian	% solutions optimales
2 mémoires, 2 fenêtres (2m/2f)	Non	351 μs	341 μs	100%
4 mémoires, 4 fenêtres (4m/4f)	Non	170 ms	169 ms	40%
6 mémoires, 6 fenêtres (6m/6f)	Non	6 s	448 ms	30%
2 mémoires, 2 fenêtres (2m/2f)	Oui	650 μ s	572 μ s	100%
4 mémoires, 4 fenêtres (4m/4f)	Oui	2 s	12 ms	90 %
6 mémoires, 6 fenêtres (6m/6f)	Oui	28 s	6 s	50 %

TABLE 1 – Temps et % des instances résolues à l’optimal avec et sans la contrainte Dense Ranking.

On peut donc déterminer une borne inférieure sur chaque variable $mem_{i,j}^s$, $mem_{i,j}^e$ et $r_{i,j}$ en $O(n \log(n))$.

5 Premiers résultats expérimentaux

Pour étudier l’utilité de la contrainte DENSE RANKING, nous avons généré 30 instances aléatoires de taille variable (2 mémoires et 2 fenêtres 2m/2f, 4 mémoires et 4 fenêtres 4m/4f, 6 mémoires et 6 fenêtres 6m/6f). Nous résolvons ces instances avec et sans la contrainte globale DENSE RANKING avec la configuration suivante :

- Solveur : OR-Tools Original CP solver¹.
- Temps limite : 100 secondes.
- Configuration matérielle : 12th Gen Intel Core i7-1265U 100 MHz, 32 GB de RAM.

Les résultats expérimentaux montrent une forte augmentation du taux d’instances résolues à l’optimum. On observe cependant une augmentation globale du temps de résolution avec la contrainte DENSE RANKING.

6 Conclusion

Nous avons présenté deux nouvelles contraintes globales pour l’ordonnancement des transferts de données par priorités, et nous montrons l’efficacité de la contrainte DENSE RANKING pour améliorer la résolution. La contrainte PRIORITY TRANSFER permet un filtrage efficace des variables de priorité, mais nécessite de trouver des bornes supérieures sur le pic d’utilisation maximum pour être vraiment utile. Trouver des solutions de qualité est donc un point crucial pour l’amélioration du modèle. Nous dirigeons aujourd’hui notre recherche vers l’intégration d’heuristiques efficaces dans notre modèle de programmation par contraintes, en particulier celles décrites dans la référence [3], ainsi que sur le calcul de bornes de meilleure qualité. Ces améliorations feront l’objet d’une étude des performances du modèle sur des instances plus grandes. La construction d’une procédure plus efficace que celle présentée dans cet article pour faire le filtrage de bornes des variables de priorité pour la contrainte DENSE RANKING est aussi en cours d’étude. La contrainte sera testée sur des problèmes variés et comparée avec d’autres méthodes pour démontrer son efficacité.

Références

[1] Christian Bessiere, Emmanuel Hebrard, George Katsirelos, Zeynep Kiziltan, and Toby Walsh. Ranking

constraints. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 705–711. IJCAI/AAAI Press, 2016.

- [2] Steve Chien, Gregg Rabideau, Daniel Tran, Martina Troesch, Joshua Doubleday, Federico Nespoli, Miguel Perez Ayucar, Marc Costa Sitja, Claire Vallat, Bernhard Geiger, et al. Activity-based scheduling of science campaigns for the Rosetta orbiter. In *Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-15*, Buenos Aires, Argentina, July 2015.
- [3] Emmanuel Hebrard, Christian Artigues, Pierre Lopez, Arnaud Lussion, Steve Chien, Adrien Maillard, and Gregg Rabideau. An efficient approach to data transfer scheduling for long range space exploration. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4635–4641. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.
- [4] Thierry Petit, Jean-Charles Régin, and Christian Bessiere. Specific filtering algorithms for over-constrained problems. In *Principles and Practice of Constraint Programming—CP 2001 : 7th International Conference, CP 2001 Paphos, Cyprus, November 26–December 1, 2001 Proceedings 7*, pages 451–463. Springer, 2001.
- [5] Gregg Rabideau, Steve Chien, M. Galer, Federico Nespoli, and Manuel Costa. Managing spacecraft memory buffers with concurrent data collection and downlink. *Journal of Aerospace Information Systems*, 14(12) :637–651, 2017.
- [6] Gregg Rabideau, Steve Chien, Federico Nespoli, and Manuel Costa. Managing spacecraft memory buffers with overlapping store and dump operations. In *Workshop on Scheduling and Planning Applications, International Conference on Automated Planning and Scheduling (SPARK, ICAPS 2016)*, London, UK, June 2016.
- [7] Gilles Simonin, Christian Artigues, Emmanuel Hebrard, and Pierre Lopez. Scheduling scientific experiments for comet exploration. *Constraints An Int. J.*, 20(1) :77–99, 2015.

1. https://developers.google.com/optimization/cp/original_cp_solver

Étude de transférabilité des clés pour le liage de données entre graphes de connaissances

Thibaut Soulard¹, Fatiha Saïs¹, Joe Raad¹, Gianluca Quercini¹

¹ LISN, CNRS (UMR 9015), Université Paris Saclay, France

Résumé

Le liage de données dans des graphes de connaissances est un problème crucial et de longue date ; il consiste à déterminer des liens entre les descriptions des entités de ces graphes désignant une même entité du monde réel. Les clés, qui sont des sous-ensembles de propriétés permettant d'identifier chaque instance d'un graphe, sont des éléments importants pour la découverte de ces liens d'identité. L'approche classique de liage de données fondée sur les clés consiste à découvrir un ensemble de clés dans chaque graphe, et ensuite appliquer une procédure de fusion (e.g., le produit cartésien des clés). Mais cette approche peut être très coûteuse en temps et peut parfois conduire à très peu de clés communes entre les deux graphes. Dans ce travail, afin de réduire le temps de calcul de la découverte de clés et par conséquent de la tâche de liage de données, nous étudions la question de transférabilité des clés découvertes dans un graphe vers un autre graphe. Plus précisément, nous avons conduit des expérimentations sur DBpedia et Wikidata afin d'évaluer l'impact en performance de la transférabilité des clés à la fois en termes de temps de calcul et de qualité des clés transférées.

Mots-clés

Liage de données, découverte de clés, graphes de connaissances, transférabilité.

Abstract

Data linking in knowledge graphs is a crucial and long-standing problem ; it involves finding links between descriptions of entities in these graphs that refer to the same real-world entity. Keys, which are subsets of properties that identify each entity, are important elements in finding these identity links. The classical key-based approach if data linking is based on the discovery of a set of keys in each graph, and then apply a merge (e.g. intersection) of these keys. But this approach can be very time consuming and can sometimes lead to very few common keys between the two graphs. In this work, in order to reduce the computational complexity of the data linking task using keys, we study the issue of transferability of keys discovered in one graph to another graph. More precisely, we conducted experiments on DBpedia and Wikidata in order to evaluate the performance impact of key transferability both in terms of computation time and quality of transferred keys.

Keywords

Data linking, key discovery, knowledge graphs, transferability.

1 Introduction

Aujourd'hui, nous assistons à une production sans précédent de ressources, publiées sous forme de données ouvertes liées. Cela conduit à la création de graphes de connaissances (KG) contenant des milliards de triplets RDF (Resource Description Framework), comme DBpedia, YAGO et Wikidata du côté académique, et le Google Knowledge Graph ou eBay Knowledge Graph du côté commercial. Ces KG contiennent des millions d'entités (telles que des personnes, des protéines ou des livres) et des millions de faits les concernant. Ces KG sont soit indépendant du domaine, comme Yago, DBpedia ou Wikidata, ou spécifiques à un domaine, comme la géographie avec Geonames¹ ou la biologie avec Bioportal². En 2020, le web de données contenait plus de 650 milles graphes de données reliés entre eux. Ces derniers contiennent des connaissances qui sont généralement exprimées en RDF³, comme des faits de la forme `<subject, propriété, object>` tels que `<Macron, presidentDe, France>`. En proposant RDF comme standard, les chercheurs de la communauté du Web sémantique ont promu la représentation des connaissances et des données sous la forme de graphes. Dans de tels graphes, les nœuds représentent des entités (par exemple, Paris) qui peuvent avoir des types représentés par des classes (par exemple, Paris est une ville), les arcs représentent des relations entre les entités (par exemple, hasMayor). Parfois, les différents types et relations sont représentés dans une ontologie OWL2 (Web Ontology Language)⁴, qui définit leurs interrelations et des axiomes tels que la subsomption, la disjonction et la fonctionnalité des propriétés.

Pour pouvoir exploiter toute la richesse des données et des connaissances contenues dans ces graphes, il est important d'établir des liens sémantiques entre leurs entités. Les liens d'identité sont parmi les liens les plus importants pour l'amélioration de la complétude des KGs, puisqu'ils permettent de relier des descriptions qui réfèrent la même en-

1. <https://en.wikipedia.org/wiki/GeoNames>

2. <https://bioportal.bioontology.org/>

3. <https://www.w3.org/RDF/>

4. <https://www.w3.org/OWL/>

tité du monde réel, ce qui permet de propager les valeurs des propriétés d'une description d'entité à une autre. Différents types d'approches ont été développées pour la détection des liens d'identité dans les graphes de connaissances (voir [9, 4] pour une revue récente de la littérature).

En effet, détecter ces liens entre les entités des deux graphes G_1 et G_2 est un problème complexe, d'une part du fait de la nécessité de comparer les entités deux-à-deux ce qui donne un espace de comparaison de taille $n_1 \times n_2$ (avec n_1 et n_2 le nombre d'instances dans G_1 et G_2 respectivement).

Dans cet article, nous nous appuyons sur les travaux de liage de données fondés sur les clés de liage [11, 1] qui sont des sous-ensembles minimaux de propriétés permettant d'identifier chaque entité (e.g., l'ensemble {nom, prénom, e-mail} peut être une clé pour identifier les personnes). L'intérêt d'utiliser les clés pour le liage de donnée est double : (i) il permet de distinguer des sous-ensembles de propriétés ayant un fort impact sur le liage et qui pourront donc potentiellement déduire des liens d'identité avec un bon taux de précision, et (ii) il permet également de réduire le nombre de couples propriétés valeurs considérés lors de la comparaison des descriptions d'entités. Pour ce dernier point, cela peut s'avérer très efficace dans le cas où les graphes contiennent des entités qui sont décrites par un grand nombre de propriétés. En revanche, l'acquisition des clés est une tâche complexe, puisque soit celles-ci sont spécifiées par un expert du domaine ou bien découvertes automatiquement par des outils dédiés comme [15, 16, 3, 13]. Mais lorsque les graphes sont volumineux ou contiennent un grand nombre de propriétés, le coût en termes de temps de calcul de la découverte de clés, qui peut atteindre quelques jours, nécessite alors d'être réduit. C'est la raison pour laquelle, nous étudions dans ce travail s'il est possible d'économiser ce temps de calcul en appliquant un transfert de clés découvertes dans un graphe vers un autre graphe.

L'article est organisé comme suit : en section 2, nous présentons les travaux de l'état de l'art. Ensuite, en section 3 nous donnons les définitions préliminaires des notions manipulées dans ce travail. En section 4, nous décrivons la méthodologie que nous avons mise en place pour étudier la transférabilité des clés d'un graphe source à un graphe cible. Ensuite, la section 5, présente l'évaluation expérimentale, une analyse et une discussion des résultats. Enfin, en section 6 nous concluons l'article et donnons quelques perspectives.

2 État de l'art

Dès lors que nous souhaitons combiner ou exploiter des données provenant de différentes sources, il devient alors important de comparer les données et détecter les descriptions qui réfèrent la même entité du monde réel (par exemple, la même personne, le même livre, le même gène). C'est ce que nous nommons liage de données.

Diverses approches ont été proposées dans la littérature, certaines appliquent des techniques de similarité [8], d'autres utilisent l'apprentissage automatique [7, 6] ou en-

core des connaissances du domaine déclarées dans une ontologie [11, 1].

Une des difficultés du liage de données réside dans l'hétérogénéité des données qui rend le calcul de similarité entre descriptions d'entité plus difficile. L'autre difficulté à laquelle les outils du liage de données doivent faire face est le nombre de combinaisons de descriptions d'entités et le coût de leur comparaison. En effet, le coût de la comparaison des descriptions d'entités peut avoir un impact important sur les performances en terme de temps de calcul de la tâche de liage d'entités entre graphes de connaissances. En effet, pour décider si deux descriptions réfèrent la même entité du monde réel, une tâche de comparaison deux-à-deux des couples propriétés valeurs décrivant les entités dans G_1 et G_2 se rend nécessaire. Cette tâche est d'autant plus difficile si les propriétés sont multi-valuées.

Les *clés* sont des éléments fondamentaux pour la découverte de liens entre entités. La spécification manuelle des clés est généralement irréalisable à l'échelle du Web, en raison du volume des ensembles de données et de l'hétérogénéité de leurs entités. C'est pourquoi plusieurs approches de découverte de clés ont émergé au fil des ans, avec des applications communes dans les bases de données relationnelles [12] et les graphes de connaissances [15, 16, 3].

Pour utiliser les clés pour lier les entités représentées dans deux graphes G_1 et G_2 , une approche idéale serait de découvrir les clés dans G_1 et dans G_2 et de calculer une intersection, ou d'appliquer une procédure de fusion (voir [10] pour un exemple). Mais cette approche peut être très coûteuse en temps et peut parfois conduire à très peu de clés communes entre les deux graphes. Une autre approche [3] exploite simultanément les deux graphes et l'alignement de leur propriétés pour découvrir des Linkkey qui sont valides dans les deux graphes. Tout comme les approches classiques de découverte de clés, cette approche a besoin d'exploiter les données des deux graphes.

Afin de réduire la quantité de données exploitées par la découverte de clés et par conséquent le temps de calcul de cette tâche, dans ce travail, nous étudions la question de *transférabilité* des clés découverte dans un graphe G_1 vers un autre graphe G_2 . Plus précisément, est-ce qu'il est possible de déterminer l'ensemble de clés découvertes, et donc valides dans un graphes G_1 , qui peuvent aussi être valides dans un graphe G_2 . Ainsi, il serait possible de s'affranchir de l'application de la découverte de clés dans G_2 et donc de l'étape de fusion de clés. À notre connaissance, le problème de transférabilité des clés n'a pas encore fait l'objet d'études.

3 Définitions et notions

Dans cette section nous introduisons les notions importantes utilisées dans notre approche de transfert de clés.

3.1 Graphes de connaissances

Définition 1. (Graphe de connaissances RDF). Nous considérons un graphe de connaissances défini par un couple $(\mathcal{O}, \mathcal{G})$, où :

- $\mathcal{O} = (\mathcal{C}, \mathcal{P})$ est une ontologie représentée en OWL et

composée d'un ensemble de classes \mathcal{C} et de propriétés \mathcal{P} pouvant être soit de type `owl:objectProperty`, dont le domaine et le co-domaine sont des classes, ou de type `owl:dataTypeProperty`, dont le domaine est une classe et le co-domaine est un type de données atomique (e.g date, string, integer).

– \mathcal{G} est un ensemble de triplets RDF décrivant des instances de classes de \mathcal{O} formant un graphe de données RDF.

Définition 2. (Graphe de données RDF). Un graphe de données RDF \mathcal{G} est un ensemble de faits représenté par des triplets de la forme $\{(sujet, prédicat, objet) \mid sujet \in \mathcal{I}, propriété \in \mathcal{P}, objet \in \mathcal{I} \cup \mathcal{L}\}$, où \mathcal{I} est l'ensemble des entités désignés des IRIs, \mathcal{P} est l'ensemble des propriétés, et \mathcal{L} est l'ensemble des littéraux (tels que les nombres et les chaînes de caractères).

Définition 3. (Description RDF d'une entité). Considérons une entité d'un graphe RDF \mathcal{G} représentée par un IRI $i \in \mathcal{I}$, sa description RDF est l'ensemble $D(i)$ défini comme suit : $D(i) = \{(p, v) \mid (i, p, v) \in \mathcal{G} \text{ ou } (v, p, i) \in \mathcal{G} \text{ (pour les propriétés } p \text{ de type objet)}\}$. On notera $P(i)$ l'ensemble de propriétés p tel qu'il existe une valeur v avec $(p, v) \in D(i)$. On notera $V(i, p)$ l'ensemble de valeurs v d'une propriété p apparaissant dans $D(i)$.

Définition 4. (Alignement de propriétés). Un alignement entre deux propriétés p_1 et p_2 de \mathcal{G}_1 et \mathcal{G}_2 (resp.) est une relation de mise en correspondance exprimant une relation d'équivalence sémantique entre p_1 et p_2 que nous notons par : $p_1 \equiv p_2$.

3.2 Clés

Une clé est un ensemble de propriétés permettant d'identifier de façon unique chaque instance (entité) d'une classe. Dans les graphes de connaissances, les clés peuvent être exploitées pour la détection de liens d'identité `owl:sameAs` entre descriptions d'entités dans les graphes de données RDF.

En général, si un ensemble de propriétés est déclaré comme étant une clé pour une classe, la non-satisfaction de la clé dans un graphe de données peut être due à des erreurs dans les valeurs des propriétés ou à des doublons inconnus. Alors que lorsque des paires d'instances provenant de différents graphes RDF ne satisfont pas la clé, ces paires d'instances peuvent être considérées comme des candidates à l'établissement de liens d'identité. En effet, chaque paire d'instances qui partagent les mêmes valeurs pour toutes les propriétés d'une clé peuvent être considérées comme candidates au liage d'entités.

Différentes sémantiques de clés ont été proposées dans le domaine du web sémantique (voir [2] pour une comparaison théorique et expérimentale). Elles diffèrent selon la stratégie appliquée pour gérer les propriétés multi-valuées et les valeurs non renseignées des propriétés. Dans cet article nous considérons la sémantique S -clé qui est celle du constructeur `owl:hasKey`⁵ qui est formalisée dans la dé-

finition suivante.

Définition 5. (Sémantique d'une S -clé) [2]. La sémantique d'une S -clé $\{p_1, \dots, p_n\}$ pour une classe⁶ C est donnée dans la règle en logique du premier ordre suivante :

$$\forall x \forall y \forall z_1 \dots z_n (C(x) \wedge C(y) \wedge \bigwedge_{i=1}^n (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow x = y)$$

Déclarer que l'ensemble $\{p_1, \dots, p_n\}$ est une S -clé pour une classe C est noté par S -clé($C, (p_1, \dots, p_n)$). Nous notons également $prop(k)$ l'ensemble de propriétés formant la clé k .

Définition 6. (Instantiation d'une clé). Soit k une S -clé($C, (p_1, \dots, p_n)$) pour une classe C dans \mathcal{G} . L'instanciation de la clé k pour une instance i de la classe C dans \mathcal{G} est un n -uplet de valeurs de propriétés $\pi(k, i)$ défini comme suit :

$$\pi(k, i) = \{(v_1, \dots, v_n) \mid \{(i, p_1, v_1), \dots, (i, p_n, v_n)\} \subseteq \mathcal{G}\}$$

4 Méthodologie de transfert de clés

Pour lier les instances d'un graphe G_1 aux instances d'un graphe G_2 , l'approche classique de liage de données fondée sur les clés consiste, tout à d'abord, à découvrir un premier ensemble de clés K_1 dans le graphe G_1 et un autre ensemble de clés K_2 dans le graphe G_2 , et d'appliquer une procédure de fusion de clés. La procédure de fusion peut simplement revenir au calcul de l'intersection des clés découvertes dans les deux graphes à la réécriture de propriétés équivalentes prés. Elle peut également être calculée, tel que proposé dans [10], par des produits cartésiens des ensembles de propriétés apparaissant dans K_1 et celles apparaissant dans K_2 et de conserver uniquement les minimales (une clé k_1 pour qui, il n'existe pas de clé k_2 tel que $prop(k_2) \subset prop(k_1)$). Il est important de noter que lorsque les graphes de données RDF sont décrits selon deux ontologies différentes, des alignements de propriétés (cf. définition 4) sont alors nécessaires pour réécrire les clés d'un graphe selon les propriétés alignées dans l'autre graphe (voir 4.1 la procédure d'alignement de propriétés que nous avons utilisée et la définition de la réécriture d'une clé).

Les clés obtenues de l'étape de fusion de clés sont alors considérées comme étant valides dans les deux graphes et peuvent par conséquent être utilisées pour lier les instances des deux graphes. Néanmoins, lorsque les graphes sont volumineux ou contiennent un grand nombre de propriétés le coût en termes de temps de calcul de la découverte de clés, qui peut atteindre quelques jours, nécessite alors d'être réduit. C'est la raison pour laquelle, nous avons étudié de manière expérimentale s'il est possible d'économiser ce temps de calcul en appliquant un transfert de clés découvertes dans un graphe vers un autre graphe. Pour ce faire, nous

6. Cela pourrait être également une expression de classe définie en OWL2 https://www.w3.org/TR/owl2-direct-semantics/#Class_Expressions

5. <https://w3.org/TR/owl2-direct-semantics/>

nous appuyons sur des mesures de qualité des clés qui sont le support et la discriminabilité que nous définissons, ci-après. Ces mesures permettent d'évaluer à quel point les clés lorsqu'elles sont transférées conservent leur propriétés d'unicité et leur capacité potentielle à produire des liens d'identité.

4.1 Alignement de propriétés

Pour calculer les alignements des propriétés décrivant les instances dans deux graphes RDF, nous considérons les ontologies auxquelles sont conformes ces graphes et avons utilisé une nouvelle approche présentée dans [14]. Cette méthode exploite les informations terminologiques et conceptuelles (i.e., définition des domaines et co-domaines des propriétés) directement disponibles dans les ontologies comme les labels, les descriptions, les types, les domaines de valeurs des propriétés ainsi que les labels et les descriptions des classes. Ensuite, à l'aide de techniques de type *transformer* telles que BERT [5] issues du traitement automatique de la langue, nous calculons des scores de similarité entre les propriétés des ontologies décrivant les deux graphes.

Cette méthode permet d'avoir un alignement $m - n$ de propriétés qui permet de capturer, notamment, des cas où les ontologies contiennent des propriétés alternatives implicites et lorsque elles diffèrent au niveau de la structure. Un exemple d'un alignement $1 - n$ judicieux pourrait être l'alignement de la propriété `wk:P625` (coordinate location) de Wikidata avec les propriétés `geo:lat` (latitude) & `geo:long` (longitude) ou bien l'alignement de `wk:P35` (head of state) avec les relations `db:monarch` & `db:leaderName` de DBpedia.

Cet alignement de propriétés peut ensuite être exploité pour réécrire les clés (voir définition 7) découvertes dans un graphe \mathcal{G} en exploitant les propriétés alignées d'un graphe \mathcal{G}' .

Définition 7. (Réécriture d'une clé). Soient deux graphes \mathcal{G} et \mathcal{G}' et $\mathcal{M} = \{(p_1 \equiv p'_1), \dots, (p_m \equiv p'_m)\}$ l'ensemble de m alignements de propriétés de \mathcal{G} et \mathcal{G}' .

Soit k une S -clé($C, (p_1, \dots, p_n)$) pour une classe C dans \mathcal{G} . Une réécriture $\rho(k, \mathcal{M})$ de la clé k pour le graphe \mathcal{G}' , selon l'ensemble des alignements de propriétés \mathcal{M} , est un ensemble de S -clés de la forme S -clé($C', (p'_1, \dots, p'_n)$) tel que il existe un alignement de propriétés $m = \{(p_1 \equiv p'_1), \dots, (p_n \equiv p'_n)\} \subseteq \mathcal{M}$ et que nous avons $C \equiv C'$.

4.2 Mesures de qualité des clés

Afin d'évaluer la qualité d'une clé, nous définissons ci-dessous le support d'une clé, son nombre d'exceptions et son taux de discriminabilité.

Définition 8. (Support d'une clé). Soit k une S -clé($C, (p_1, \dots, p_n)$) pour une classe C dans \mathcal{G} . Le support de k dans \mathcal{G} est le nombre d'instances de C ayant au moins une valeur pour chacune des propriétés $prop(k)$. Plus for-

mellement :

$$support(k) = |\{x \mid \forall p \in prop(k), \exists y, (x, p, y) \in G\}|$$

Le support d'une clé relativement au nombre d'instances de la classe est formellement défini comme suit :

$$support^R(k) = \frac{support(k)}{|\{x \mid \forall x, C(x) \in G\}|}$$

Le support permet de mesurer la couverture d'une clé en termes de nombre d'instances pour qui la clé pourrait générer un lien d'identité. Un autre critère de qualité d'une clé est son degré de discriminabilité qui mesure à quel point la clé permet de distinguer une instance parmi toutes les autres instances du graphe. Comme dans [3, 13], pour mesurer ce degré de discriminabilité on s'appuie sur le nombre de partitions d'instances partageant les mêmes valeurs pour les propriétés de la clé et qui sont réduites à une seule instance.

Définition 9. (Partition de l'ensemble d'instances d'une clé). Étant donnée une clé $k = S$ -clé($C, (p_1, \dots, p_n)$) dans un graphe de données RDF G , la partition de l'ensemble d'instances de C pouvant être formée par k est définie par l'ensemble : $\Delta(k, G) = \{\delta_1, \delta_2, \dots, \delta_m\}$. C'est l'ensemble de partitions d'instances pouvant être formées en regroupant dans chaque partition $\delta_i \in \Delta(k, G)$ le sous-ensemble d'instances partageant les mêmes valeurs pour $prop(k)$.

Définition 10. (Taux de discriminabilité d'une clé). Le taux de discriminabilité d'une clé $k = S$ -clé($C, (p_1, \dots, p_n)$) est le nombre de partitions $\delta_i \in \Delta(k, G)$ réduites à une seule instance relativement au nombre total de partitions de $\Delta(k, G)$. Plus formellement,

$$discr(k, G) = \frac{|\{\delta_i \mid \delta_i \in \Delta(k, G), |\delta_i| = 1\}|}{|\Delta(k, G)|}$$

Lorsque les graphes de connaissances contiennent des redondances (i.e. l'hypothèse du nom unique non vérifiée) ou contiennent des propriétés dont les valeurs sont erronées, la découverte de clés strictes devient impossible. C'est alors pour cela que des approches comme [15] ont introduit la notion de clés tolérant quelques exceptions dont le nombre est fixé par un seuil.

Ci-dessous nous donnons la définition du nombre d'exceptions d'une clé inspirée de [15].

Définition 11. (Nombre d'exceptions d'une clé). Le nombre d'exceptions d'une clé k de la forme S -clé($C, (p_1, \dots, p_n)$) est le nombre d'instances de C qui ne satisfont pas la clé. Plus formellement :

$$ex(k) = |\{x \mid \forall p \in prop(k), \exists y, y \neq x, V(x, p) \cap V(y, p) \neq \emptyset\}|$$

La définition 11 permet de définir des clés avec des exceptions dont le nombre est calculé par rapport au

nombre d’instances de la classe dans le graphe. Cependant, cette définition n’est plus pertinente dans le cas où les valeurs des propriétés ne sont pas toutes renseignées, ce qui est souvent le cas dans les graphes de connaissances (e.g., la propriété `fax-number` (P2900) de la classe `human` (Q5) n’est renseignée que pour 31 instances dans le graphe Wikidata). Dans cet article nous proposons une nouvelle définition du nombre d’exceptions qui tient compte du support des propriétés de la clé.

Définition 12. (Nombre d’exceptions d’une clé relatif à son support). Le nombre d’exceptions d’une clé k de la forme S -clé($C, (p_1, \dots, p_n)$) relativement au nombre d’instances supportant k est formellement défini comme suit :

$$ex^R(k) = \frac{ex(k)}{support(k)}$$

4.3 Approche de transfert de clés

Afin d’étudier la transférabilité des clés d’un graphe G_1 vers un graphe G_2 , nous avons procédé selon les étapes suivantes.

(1) Découverte de clés dans le graphe d’origine. Pour chaque classe d’instances du graphe G_1 , pour lesquelles il existe un alignement avec une classe du graphe G_2 , appliquer un outil de découverte de clés pour générer un ensemble de clés \mathcal{K}_1 . Pour cette première étape, nous avons utilisé l’outil SAKey [15] qui découvre des S -clés en considérant un nombre d’exceptions passé en paramètre. Il est à noter, comme indiqué dans la section 2, SAKey étant basée sur le calcul de non-clés d’abord il ne permet pas de fournir des mesures quantitatives des clés telles que le support et la discriminabilité.

(2) Sélection des clés en fonction de leurs scores de qualité dans le graphe d’origine. Pour chaque clé découverte, nous évaluons sa qualité en termes de nombre d’exceptions générées et son support. Ces deux mesures nous permettent ensuite d’obtenir le taux d’exceptions relatif $ex^R(k)$ et de sélectionner seulement les clés qui respectent le taux d’exceptions relatif maximum ex_{max}^R passé en paramètres.

(3) Réécriture des clés sélectionnées en exploitant l’alignement de propriétés entre les deux graphes. Cette étape consiste à identifier les clés alignées, i.e., le sous-ensemble de clés $K_1 \subseteq \mathcal{K}_1$ découvertes sur G_1 pour lesquelles des alignements de propriétés existent avec le graphe G_2 .

(4) Évaluation de la qualité des clés réécrites dans le graphe cible. Enfin, pour chaque réécriture de clé retenue nous évaluons sur le graphe cible G_2 sa qualité en termes de taux d’exceptions relatif $ex^R(k)$ en considérant le même seuil qu’en étape (2). Cette évaluation nous permet de détecter les clés transférées qui auraient dégénéré vers une non-clé, i.e. un ensemble de propriétés ayant un taux d’exceptions $ex^R(k) > ex_{max}^R$, et de les écarter ensuite pour les étapes de liage d’entités.

5 Évaluation expérimentale

L’évaluation expérimentale dans ce travail a pour objectif d’évaluer si le gain en performance d’une approche de liage d’entités à base de clés ayant appliqué une procédure de transfert de clés du graphe source vers un graphe cible, impacte-t-il la qualité des clés dans le graphe cible. En d’autres termes, il s’agit d’étudier comment la qualité des clés évolue quand elles sont transférées à un autre graphe.

Pour répondre à cette question nous avons fait varier le taux d’exception relatif toléré pour chaque clé découverte dans le graphe source. Ceci permet de limiter l’impact de la présence d’erreurs ou de doublons dans les graphes considérés sur les résultats. Plus précisément, pour chaque taux d’exceptions maximal ex_{max}^R (avec $ex_{max}^R \in [0, 5]$), nous procédons comme suit :

1. découverte de l’ensemble de clés K_1 de C_1 dans G_1 avec chaque clé ayant un taux d’exception $< ex_{max}^R$
2. alignement des propriétés dans K_1 avec les propriétés dans G_2
3. utilisation T_x , un sous-ensemble des clés de K_1 dont laquelle chaque clé dans T_x a toutes ses propriétés alignées au moins à une propriété dans G_2
4. calcul du support, du taux d’exception relatif et de la discriminabilité de chaque clé dans T_x dans G_1
5. calcul du support, du taux d’exception relatif et de la discriminabilité de chaque clé dans T_x dans G_2
6. comparaison des résultats des étapes 4 et 5

5.1 Jeux de données

Les jeux de données utilisés pour l’évaluation expérimentale proviennent de DBpedia⁷ et de Wikidata⁸.

Pour le premier jeu de données, nous avons extrait la partie du graphe RDF représentant les instances de la classe `Person` (i.e., ayant comme `rdf:type Person`) extrait en décembre 2022. Ce premier jeu de données est ainsi composé de 1,863,013 entités et 18,960 propriétés. Cependant, il est à noter que pour un nombre non négligeable de ces propriétés correspondent à un IRI mal encodé ou erroné (IRI qui ne correspond pas à un IRI d’une des ontologies qui décrit ce graphe) comme la présence de `http://dbpedia.org/ontology/award21n,2` au lieu de `http://dbpedia.org/ontology/award`. Ces propriétés sont alors supprimées du graphe.

Le deuxième jeu de données a été extrait du graphe Wikidata du 3 mars 2021 disponible sur rdfhdt⁹. Cette extraction a été réalisée pour les IRIs typés par le triplet `<?IRI, wdt:P31, wd:Q5>` (wd:Q5 représente la classe "Human" dans Wikidata). Ce jeu de données est composé de 3,020,916 et 3,506 propriétés.

À ces jeux de données nous avons également considéré les ontologies décrivant les instances de personnes qui contiennent respectivement 4,604 et 10,472 propriétés.

7. <https://www.dbpedia.org/>

8. <https://www.wikidata.org/>

9. <https://www.rdfhdt.org/datasets/>

Nous avons appliqué un algorithme d’alignement de propriétés que nous avons développé (décrit dans [14]) et qui applique une mesure de similarité fondée sur les plongements sur la description textuelle associée aux propriétés et aux classes de ces deux ontologies.

Suppression des propriétés non-pertinentes. Après l’extraction des données, une étape de pré-traitement a été appliquée pour réduire le nombre de propriétés inutilisables. Cette étape est motivée par le temps d’exécution de l’algorithme de découverte de clés qui est dépendant du nombre de propriétés dans le graphe. Pour ce faire, nous avons seulement gardé les propriétés utilisées dans les deux graphes où nous avons au moins un alignement de propriétés disponible. Nous avons gardé seulement les propriétés du type `owl:DatatypeProperty`, c’est à dire dont les valeurs sont des littéraux. Nous avons appliqué cette stratégie car une propriété non transférable ou commune aux deux graphes n’est pas utilisable pour une réécriture de clé. De plus, une clé qui est composée de `owl:ObjectProperty` n’est utilisable que par les outils exploitant la propagation de scores de similarité entre paires d’instances comme [1, 11]. Après ce pré-traitement, nous obtenons respectivement 239 et 135 propriétés pour DBpedia et Wikidata.

5.2 Résultats en termes de nombre de clés découvertes

Dans le tableau 1, nous présentons le nombre de Clés-Sources (C-S) pour chaque graphe de connaissances ainsi que le nombre de Clés-Réécrites (C-R) durant la vérification dans le graphe cible. Cette vérification est divisée en deux étapes, la première filtre les réécritures n’ayant aucune instantiation (cf. définition 6) (i.e. ayant un support égal à 0). Dans un second temps, nous vérifions aussi que ces réécritures respectent le taux d’exception maximal autorisé.

Le principal facteur de ce filtrage est directement liée au manque d’instanciation des réécritures dans le graphe cible. Ce manque peut être expliqué par une différence dans les données décrivant les entités, ainsi un graphe pourrait représenter les entités personnes d’un point de vue social et un autre d’un point de vue professionnel. Il est aussi explicable par un alignement $n - m$ de certaines propriétés trop lâches donnant ainsi une réécriture inutile car ces propriétés ne sont pas utilisées pour décrire un `humain` par exemple. Enfin, lors de la vérification du taux d’exceptions nous pouvons déjà avoir une première observation sur la dégénérescence des clés réécrites vers des Non-Clés Réécrites (i.e. ayant un $ex^R(k) > ex^R(k)_{max}$). Un exemple de ces Clés Réécrites et vérifiées est :

```
{(wk:P2561≡db:name),
(wk:P1477≡db:originalName) }
```

5.3 Résultats en termes d’ensemble de clés

Dans le tableau 2, nous pouvons observer l’évolution de la discriminabilité de l’ensemble des clés en fonction du taux d’exception relatif maximal. A partir de 0.5%, nous

		ex_{max}^R	0%	0.5 %	2%	5%
DB	C-S	835	642	642	643	
	C-R brut	1 572	1 199	1 199	1 200	
	C-R (sup≠ 0)	69	64	64	65	
	C-R vérifiées	49	52	54	56	
WK	C-S	357	237	238	240	
	C-R brut	475	327	328	330	
	C-R (sup≠ 0)	82	82	82	83	
	C-R vérifiées	41	48	49	55	

TABLE 1 – Résultat de la découverte de clés dans DBpedia et Wikidata sur les instances représentant des personnes

atteignons un plateau qui ne permet pas de discriminer un nombre plus important d’entités. Ce plateau est d’autant plus intéressant que pour des taux d’exceptions maximaux plus haut nous introduisons aussi plus de bruit (i.e. des exceptions) rendant la tâche final de liage d’entités plus difficile. De plus, ces données montrent également une certaine limitation de la méthode de découverte de clés avec un pourcentage de discriminabilité très faible pour les clés découvertes sur Wikidata appliquées sur ce même graphe, i.e. évaluation des mesures de qualité des clés sur le même graphe.

		ex_{max}^R	0%	0.5 %	2%	5%
DB	DB	0.32%	80.72%	80.72%	80.72%	
	WK	0.90%	28.88%	28.89%	28.89%	
WK	WK	0.31%	0.88%	0.88%	1.03%	
	DB	0.10%	80.73%	80.73%	80.73%	

TABLE 2 – Pourcentage de discriminabilité sur l’ensemble des Clés-Réécrites découvertes dans X puis appliquées dans X & Y

5.4 Évolution du pourcentage d’exceptions après le transfert de clés

Dans la figure 1, nous montrons l’évolution du pourcentage d’exceptions en trois échelles : une réduction, une stabilité ou une augmentation du taux exceptions dans le graphe destination. Dans le cas du transfert des clés DBpedia vers Wikidata nous pouvons observer que les Clés-Réécrites ont tendance à garder leurs taux d’exceptions dans les deux graphes et donc à être relativement stable lors du transfert. Par exemple, pour les 65 clés découvertes dans DBpedia avec un taux maximale de 5%, 46 clés (70%) gardent le même taux d’exceptions quand elles sont transférées à Wikidata, 18 clés (27%) voient leur taux d’exceptions augmenter et une seule clé voit son taux d’exception diminuer. Néanmoins, dans le sens inverse nous avons un comportement bien différent avec une majorité de clés dégénérent en des Non-Clés-Réécrites. Ce comportement, impose donc à ce que les Clés-Réécrites soient bien évaluées dans le graphe cible pour s’assurer de la qualité des liens d’identité

potentiels. Enfin pour ces deux graphes le cas où la Clé-Réécrite réduit son taux d'exceptions est relativement rare.

5.5 Évolution de la discriminabilité après le transfert de clés

Dans la figure 2, nous montrons l'évolution du taux de discriminabilité d'une Clé-Réécrite en trois échelles, comme pour le pourcentage d'exceptions cas précédent : une réduction, une stabilité ou une augmentation du taux discriminabilité dans le graphe destination.

Sur ce deuxième indicateur nous observons un comportement plus régulier pour le transfert dans les deux sens, avec une majorité de Réécritures gardant un taux de discriminabilité stable.

5.6 Temps d'exécution pour chaque étape des stratégies et scénarios

Dans le tableau 3, nous présentons les différentes valeurs du temps d'exécution en fonction de la stratégie et du scénario suivi. Toutes les expériences ont été réalisées sur un CPU "Intel® Xeon® E5-2630 v4" avec 10 coeurs et 128 GB de RAM. Le point le plus important est le temps d'exécution de la découverte des Clés-Sources de DBpedia. Cette étape est le goulot d'étranglement pour la stratégie classique appliquant la fusion de clés découvertes dans les deux graphes. Par conséquent, en partant du graphe ayant le temps d'exécution le plus rapide pour la découverte de C-S, c'est à dire celui ayant le moins de propriété, (ici le graphe source serait Wikidata) nous pouvons drastiquement réduire le temps d'exécution total. Néanmoins, en partant de DBpedia nous perdons les bénéfices de cette nouvelle stratégie tout en restant tout de même dans des temps similaires à la méthode classique appliquant la fusion de clés.

5.7 Discussion

Au travers de cette évaluation expérimentale, nous avons pu analyser le comportement des clés réécrites du côté de la source et de la cible. Malgré la nécessité de vérification des clés réécrites dans le graphe cible nous avons pu montrer qu'une majorité de ces clés restent stables et ne dégénèrent pas en non clés. Néanmoins, avec la définition des exceptions relatives appliquée sur les clés trouvée par SAKey, impacte la monotonie des clés et par conséquent la complétude de l'ensemble des clés transférées. En effet, dans le tableau 4 la clé K_1 composée de $\{P_1, P_2\}$ respecte un taux maximal d'exceptions de 25% mais la clé K_2 composée de $\{P_1, P_2, P_3\}$ ne la respecte plus avec un support plus faible tout en ayant un même nombre d'exceptions donnant ainsi un taux d'exceptions de 100% alors que la clé $K_3 = \{P_1, P_2, P_3, P_4\}$ respecte ce taux.

La complétude est d'autant plus importante que des clés plus précises (i.e. ayant un support plus faible) peuvent différencier des entités que des clés plus générales (i.e. un support plus élevé) ne le peuvent pas. Un exemple de ce phénomène est illustré dans le même tableau 4, avec la clé K_1 qui

ne peut pas différencier e_2 de e_3 là où K_3 le permet. Ainsi, nous pouvons espérer augmenter le taux de discriminabilité avec l'ajout de clé réécrites plus précises différenciant des cas particuliers. Cependant, cet ajout n'est pas trivial de part la perte de la monotonie des S-Clés avec le taux d'exception relatif. Mais bien-sûr cet ajout de clés engendrera une augmentation du temps d'exécution pour la découverte de clés ainsi que pour le liage d'entités, car plus de valeurs devront être comparées. Cette problématique est néanmoins commune à l'approche classique et celle appliquant le transfert de clés, ainsi elle n'a pas d'impact direct sur cet étude mais pourrait améliorer les résultats des approches basés sur les clés.

6 Conclusion et perspectives

Dans cet article, nous avons pu décrire une stratégie de transfert de clés pour éviter le recours à la découverte de clés systématique dans tous les graphes considérés lors du liage de données. Cette découverte de clés qui peut se révéler parfois très coûteuse en temps d'exécution. Nous avons aussi pu observer l'importance de l'évaluation des Clés-Réécrites dans le graphe cible pour éviter l'utilisation de Non-Clés Réécrites. Enfin, nous avons aussi pu montrer une limitation de l'utilisation des clés pour l'alignement d'entités avec un taux de discriminabilité qui peut se révéler très faible pour certain graphes.

Dans de futurs travaux, à très court termes nous envisageons d'évaluer la qualité des clés transférées en évaluant la qualité des liens d'identité (i.e. rappel, précision et F-mesure) qu'elles permettent de générer par le biais d'un outil de liage de données. Nous comptons également comparer les résultats de cette approche avec des approches qui découvrent des clés en prenant deux graphes simultanément tel que Linkkeys[3]. Enfin, nous souhaiterons développer une approche hybride pour le liage d'entités en explorant la possibilité d'utiliser les clés pour générer automatiquement un premier ensemble réduit de lien `schema:sameAs`. Ces liens pourront ensuite être utilisés pour l'initialisation des techniques d'alignement d'entités utilisant les plongements de graphes qui requiert un ensemble de liens entre les deux graphes. Nous voudrions aussi explorer la création d'un outil de découverte de clés prenant en compte les paramètres d'exception relatifs et permettant d'avoir toutes les clés respectant le taux défini.

Références

- [1] Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset. Inferring same-as facts from linked data : An iterative import-by-query approach. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 9–15. AAAI Press, 2015.
- [2] Manuel Atencia, Michel Chein, Madalina Croitoru, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatih Saïs, François Scharffe, and Danai Symeonidou. Defining key semantics for the RDF datasets : Expe-

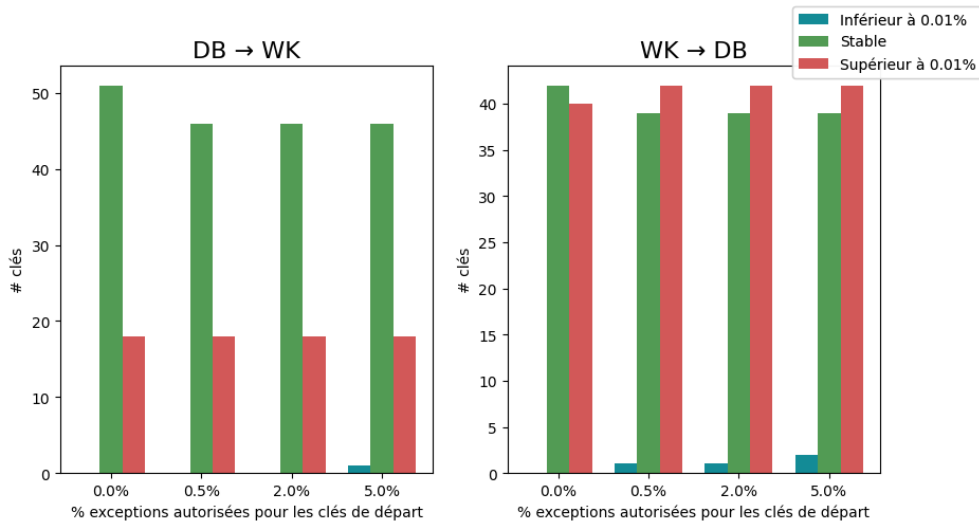


FIGURE 1 – Évolution du taux d’exceptions suite au transfert des clés découvertes

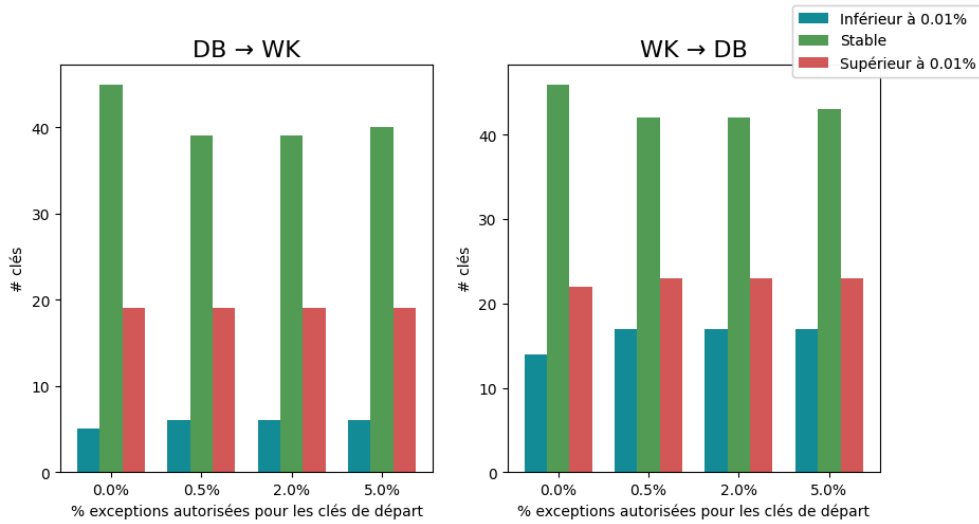


FIGURE 2 – Évolution du taux de discriminabilité suite au transfert des clés découvertes

		ex_{max}^R	0%	0.5 %	2%	5%
Classique	DB	Découverte C-S DB	279.3	276.0	270.8	278.0
	DB ∩ WK	Découverte C-S WK	5.1	5.0	5.1	5.0
		Évaluation C-S DB	33.7	25.3	25.4	25.4
		Évaluation C-S WK	3.4	1.5	1.5	1.5
		Total	321.7	308.0	302.9	310.1
Transfert	Départ DB	Découverte C-S	279.3	276.0	270.8	278.0
	Départ WK	Évaluation C-S	33.7	25.3	25.4	25.4
		Évaluation. C-R	11.7	11.6	11.7	11.9
		Total	324.8	313.0	308.0	315.4
Transfert	Départ WK	Découverte C-S	5.1	5.0	5.1	5.0
	Départ DB	Évaluation C-S	3.4	1.5	1.5	1.5
		Évaluation C-R	5.9	5.8	5.9	6.0
		Total	14.5	12.4	12.6	12.6

TABLE 3 – Temps d’exécution en minutes pour chaque scénario et étape.

	P_1	P_2	P_3	P_4
e_1	A1	A2	-	-
e_2	B1	B2	A3	B4
e_3	B1	B2	A3	C4
e_4	D1	D2	-	-
e_5	E1	E2	-	-

TABLE 4 – Exemple de non monotonie de clés.

- riments and evaluations. In *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iași, Romania, July 27-30, 2014, Proceedings*, pages 65–78, 2014.
- [3] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, and Jérémy Vizzini. Link key candidate extraction with relational concept analysis. *Discret. Appl. Math.*, 273 :2–20, 2020.
- [4] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, 53(6), dec 2020.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert : Pre-training of deep bi-directional transformers for language understanding. 2018.
- [6] Nikolaos Fanourakis, Vasilis Efthymiou, Dimitris Kotzinos, and Vassilis Christophides. Knowledge graph embedding methods for entity alignment : An experimental review, 2022.
- [7] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, An-Hai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching : A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18*, page 19–34, New York, NY, USA, 2018. Association for Computing Machinery.
- [8] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes—a time-efficient approach for large-scale link discovery on the web of data. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [9] Natasha Noy, Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of current link discovery frameworks. *Semant. Web*, 8(3) :419–436, jan 2017.
- [10] Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. An automatic key discovery approach for data linking. *Journal of Web Semantics*, 23 :16–30, 2013.
- [11] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. In *Journal on Data Semantics XII*, pages 66–94. Springer, 2009.
- [12] Yannis Sismanis, Paul Brown, Peter J Haas, and Berthold Reinwald. Gordian : efficient and scalable discovery of composite keys. In *Proceedings of the 32nd international conference on Very large data bases*, pages 691–702, 2006.
- [13] Tommaso Soru, Edgard Marx, and Axel-Cyrille Ngonga Ngomo. Rocker : A refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, page 1025–1033, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.
- [14] Thibaut Soulard. Knowledge-based Entity Linking in Heterogeneous Knowledge Graphs at Web-Scale. Technical report, Université Paris Saclay, September 2022. Master thesis report.
- [15] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. Sakey : Scalable almost key discovery in rdf data. In *International Semantic Web Conference*, pages 33–49. Springer, 2014.
- [16] Danai Symeonidou, Luis Galárraga, Nathalie Pernelle, Fatiha Saïs, and Fabian Suchanek. Vickey : Mining conditional keys on knowledge bases. In *International Semantic Web Conference*, pages 661–677. Springer, 2017.



En partenariat avec



L'AVENIR EST AUX VALEURS SÛRES

