AfIA

Association française
pour l'Intelligence Artificielle

RJCIA

Rencontres des Jeunes Chercheurs
en Intelligence Artificielle

PFIA 2024

Table des matières

Éditorial

Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Comme leur nom l'indique, les Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA) sont destinées aux jeunes chercheurs en Intelligence Artificielle (IA), doctorants ou titulaires d'un doctorat depuis moins d'un an. A ce titre, l'objectif de cette manifestation est double :

1. permettre aux jeunes chercheurs préparant une thèse en IA, ou l'ayant soutenue depuis peu, de se rencontrer et de présenter leurs travaux, et d'ainsi nouer des contacts avec d'autres jeunes chercheurs et d'élargir leurs perspectives en échangeant avec des spécialistes d'autres domaines de l'IA ;
2. former les jeunes chercheurs à la préparation d'un article, à sa révision pour tenir compte des observations du comité de programme, et à sa présentation devant un auditoire de spécialistes, leur permettant ainsi d'obtenir des retours de chercheurs de leur domaine ou de domaines connexes.

Toute contribution relevant de IA est la bienvenue ; l'IA s'invitant partout, traversant les disciplines et les défis en Recherche. Une liste non exhaustive de thématiques est donnée à titre indicatif :

— recherche heuristique et résolution de problèmes,
— incertitude et intelligence artificielle,
— logique, satisfiabilité et satisfaction de contraintes,
— apprentissage automatique,
— extraction, ingénierie, représentation et gestion des connaissances et raisonnement,
— planification, contrôle, aide à la décision,
— agents autonomes et systèmes multi-agents,
— reconnaissance des formes et vision par ordinateur,
— traitement automatique des langues naturelles,
— interaction avec l'humain,
— robotique,
— IA et web,
— environnements Informatiques d'Apprentissage Humain et apprentissage à distance,
— IA responsable, explicabilité, certification, éthique et IA
— ...

Portée par la Plate-Forme Intelligence Artificielle (PFIA), RJCIA est à sa vingt troisième édition. Cette année, près de 20 articles ont été soumis. Chaque article a été relu par trois membres du Comité de programme. Les auteurs des articles publiés dans les présents actes ont eu à prendre en compte les remarques qui leur ont été adressées dans le cadre de cette relecture.

Les présentations ont été réparties en six sessions, dont deux sont conjointes à CNIA. La diversité des thématiques abordées a donné lieu à diverses questions et de nombreux échanges.

Pour nourrir la réflexion collective et le débat, une table ronde sur le thème -très actuel- des LLMs a été organisée en collaboration avec CNIA et IC.

Le programme a été enrichi par la conférence de Samuel Tronçon, philosophe et chercheur à Résurgences R&D, spécialisé en informatique appliquée aux sciences sociales. Le thème de la conférence : *Le cyber-réductionnisme en question ; renoncer, succomber ou refonder ?* a permis de (re)poser la question des fondements et de la méthode dans le champ de l'informatique théorique et de l'IA, surnommé "Intelligence Computationnelle".

Enfin, j'éprouve une grande satisfaction pour avoir contribué, avec les membres du Comité de programme, à faire vivre cette Conférence dédiée à la Jeunesse.

Que l'ensemble des intervenants et intervenantes soit remercié pour sa précieuse contribution.

Nadia Abchiche-Mimouni

Comité de programme

Présidence

— Nadia Abchiche-Mimouni, I3S, CNRS/UNS - Université côte d'Azur.

Membres

— Amel Bouzeroub, professeur, Institut Polytechnique de Paris - Telecom SudParis
— Feng Chu, professeur IBISC, univ. Evry - université Paris-Saclay
— Victor David, Researcher, INRIA Sophia Antipolis
— Maxime Devanne, Maître de conférences, IRIMAS - Université de Haute-Alsace
— Catherine Faron, professeur, I3S - université Côte d'Azur
— Arnaud Ferre, Chargé de Recherche, MaIAGE, INRAE - Université Paris-Saclay
— Maxime FOLSCHETTE, Maître de conférences, CRIStAL - Centrale Lille Institut
— Fatima Ghedjati, Maitre de conférences, LICIIS - université Université de Reims Champagne-Ardenne
— Zahia Guessoum, Maître de conférences HDR, CReSTIC - université Université de Reims Champagne-Ardenne
— Camille Guinaudeau, Maitre de conférences, Japanese-French Laboratory for Informatics CNRS - Université Paris-Saclay
— Guillaume LOZENGUEZ, Maitre de conférences, IMT Nord-Europe
— Jean-Guy Mailly, professeur junior, IRIT, Université de Toulouse, UT Capitole
— Mohamed-Lamine MESSAI, Maître de conférences, ERIC - Université Lyon 2
— Pierre Monnin, Junior Fellow, Université Côte d'Azur, Inria, CNRS, I3S
— Charlotte Pelletier, Maître de conférences, IRISA - université Bretagne Sud
— Brian Ravenet, Maître de conférences, LISN - université Paris-Saclay
— Yasmina Sadi, Maître de conférences, IBISC - univ. Evry - université Paris-Saclay
— Nicolas Verstaevel, Maître de conférences, IRIT - université Toulouse Capitole
— Genane Youness, Maitre de conférences, CESI - LINEACT
— Farida Zehraoui, Maître de conférences, IBISC, univ. Evry - université Paris-Saclay

Session 1 : Agents et Systèmes multi-agents

A machine learning approach for cellular phenotyping using Raman spectral data

D. Kocak[1], J.L. Fleck[1], X. Xie[1], J. Marzi[2,3]

[1]Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, F - 42023 Saint-Etienne France
[2] Institute of Biomedical Engineering, Department for Biomedical Technologies & Regenerative Medicine, Eberhard Karls University Tübingen, Tübingen, Germany
[3] NMI Natural and Medical Sciences Institute at the University of Tübingen, Reutlingen, Germany

duru.kocak@emse.fr

Résumé

Le phénotypage cellulaire est le processus d'identification du phénotype d'une cellule. La microspectroscopie Raman est une méthode non invasive qui génère des données longitudinales à partir d'échantillons biologiques. Cependant, extraire des informations à partir de données Raman afin d'accomplir des tâches telles que le phénotypage cellulaire, traditionnellement réalisées au moyen de méthodes invasives, constitue une tâche difficile. Dans cet article, nous présentons une approche d'apprentissage automatique pour prédire le phénotype des cellules sanguines à l'aide de données Raman.

Mots-clés

Micro spectroscopie Raman spontanée, classification automatisée, regroupement automatique des signaux

Abstract

Cellular phenotyping is the process of identifying the phenotype of a cell. Raman Microspectroscopy is a noninvasive method that generates longitudinal data from biological samples. However, it is a challenging task to extract information from Raman data in order to accomplish tasks such as cellular phenotyping, which have been traditionally accomplished through invasive methods. In this paper, we present a machine learning approach to predict the phenotype of blood cells using Raman data.

Keywords

Spontaneous Raman Microspectroscopy, automated classification, PBMCs, automated signal clustering

1 Introduction

The phenotype of a cell is the collection of observable cellular characteristics that arise from its genetic constitution and environmental factors. Cell morphology describes the shape, structure, form and size of a cell, and is an important aspect of cellular phenotype. In response to cellular dynamics, the morphological signature of a cell, and hence its phenotype, may change over time. Cellular phenotyping is the process of identifying and characterizing the phenotype of a cell at a given point in time, and is an important step in understanding how cell activities are regulated.

Traditionally, fluorescence microscopy images were manually scored to define cellular phenotypes. Recent technological advances led to the development of machine learning and computer vision approaches for image-based cellular phenotyping [5], [3], [2]. Currently, additional approaches have been proposed to identify molecularly defined cell phenotypes using machine learning-based analyses of epigenomic [8] and single cell RNA-sequencing (scRNA-seq) data [4]. The aforementioned approaches share a common limitation in that they rely on invasive data generation methods that interfere with cellular function or kill the cell under investigation.

In this study, we address these limitations by developing an automatic pattern extraction pipeline that uses marker-independent imaging data from Raman microspectroscopy (RMS) for cellular phenotyping. RMS is a non-destructive, label-free technique that determines the molecular composition of samples in a variety of states. It uses laser light to discriminate between different cell and tissue types, and has shown great promise in in vivo diagnosis, with the potential to eliminate or reduce the need for biopsies [6]. Comparable to other spectroscopic techniques (e.g., mass spectrometry), individual bands in the Raman spectrum can be assigned to different molecular vibrations, enabling a molecular-sensitive characterization of a sample. In complex biological samples, the Raman spectral pattern is a holistic representation of the cell or tissue composition – at a similar specificity to a human fingerprint.

Here we demonstrate that accurate predictions of Raman-based cellular phenotypes can be achieved for human derived peripheral blood mononuclear cells (PBMCs).

2 Dataset Description

Our dataset consists of hyperspectral RMS images of PBMC-derived monocytes (MC) that were further

differentiated in vitro towards two types of immune effector cells: dendritic cells (DC) and macrophages (M0).

RMS images of single cells in suspension were obtained as described previously [7]. In brief, PBMCs were isolated from the blood of healthy volunteers after informed consent was obtained. Monocytes were separated from other PBMC populations via magnetic separation. Differentiation of MCs to DCs was induced by supplementation with granulocyte macrophage colony stimulating factor (GM-CSF, 500 U/ml) and interleukin 4 (IL-4, 500 U/ml). For M0 differentiation, macrophage colony stimulating factor (M-CSF, 50 ng/ml) was added to the culture medium of MCs. In preparation for RMS measurements, cells were detached from the culture plates and washed with saline aqueous solution. Single cell Raman images were obtained with a Witec 300R alpha Raman microspectrometer using a green laser (532 nm). Individual scans were focussing on one cell and obtained from an area of 30x30 µm at a special resolution of 1x1 µm and an overall integration time of 3 min per image. Each of the pixels is represented by a Raman spectrum containing the Raman shift (1/cm) on the x axis, with values ranging from -1540 to 3750 across 1024 points; and on the y axis, the intensity of the scattered light. The dataset used for this study is composed of 597 spectral images, originating from 12 different donors, with the following distribution: 104 spectral images of DCs, 182 spectral images of M0 and 97 spectral images of MCs.

3 Methods

In this study, the input dataset consisted of spectral images of single cells. We developed an automatic pattern extraction pipeline that classified each image according to its corresponding cellular phenotype (Fig 1).

As shown in Fig 1, the first step of our pipeline consists of preprocessing the input dataset. Preprocessing includes cosmic ray removal for each spectrum in the dataset, followed by normalization of the area under the curve of the spectra. In the second step, clustering is performed at the pixel level using the k-means algorithm [1]. Here the number of clusters (k) was chosen to be 7 and the Euclidean distance was used as distance metric. Additional configuration parameters include the choice of *kmeans++* as the initialization method and setting the number of initialisations to 1. Subsequently, outlier removal is performed by computing the number of images in which each cluster is present. If a cluster is present in less than 5 percent of the images, the images in which the cluster is present are considered outliers and are excluded from the dataset. As a fourth step, clustering at the pixel level is repeated using k-means with the same parameters as defined above. Using the clustering results, heat maps were generated for each image (Fig. 2) and validated by visual inspection. The last step in our pipeline consists of a quantitative machine learning analysis where the number of pixels belonging to each of the clusters is computed for each image. The format of the input data to the machine learning algorithms is shown in Table 1, where the target (output variable) is the cell phenotype.

Column Name	Format	Description
ID	[cell type]_[donor]_[sample number]_[day of sampling]	unique ID of each cell sample
Maturity	[cell_type]_[day of sampling]	DC_14, MC_0 and M0_7 are the kept maturity states in this dataset since we are focusing solely on mature cells.
count_Ci	count of pixels by Ci	Number of pixels attributed to the corresponding cluster in the cell sample
sum_clusters	sum of count_Ci (i is in [0:6])	Sum of counts of pixels attributed to their corresponding clusters. 900 for each image, this column is for data validation purposes only.
cell_type	[cell_type]	MC, DC or M0. This is the target variable.

Table 1 : Features, their formats and their descriptions

We used three established machine learning algorithms: logistic classification, Support Vector Machines (SVM), and decision trees. To avoid bias due to a dominant class, range normalization (MinMax normalization) was performed for all input variables. A total of 70% of the dataset was used for training and 30% for testing.

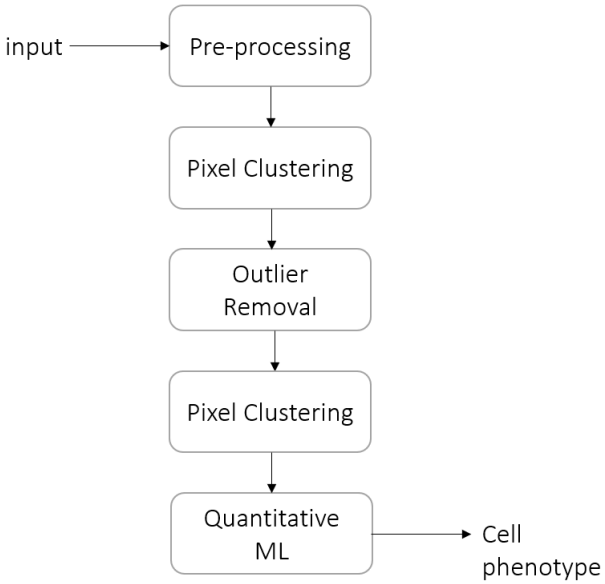


Fig 1 : The proposed pipeline for classifying cell phenotype

4 Results

Results of the k-means clustering are shown in Fig. 2. The curves depict the spectral signature of the centroid of each cluster, before and after outlier removal. The color scheme is identical to the ones used in the heatmaps (Fig. 3).

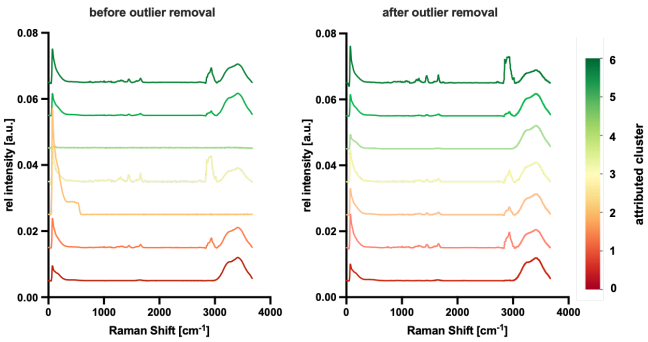


Fig 2 : Spectral signatures of the centroids obtained by the k-means method before (left plot) and after outlier removal (right plot)

From Fig 2, it can be seen that the spectral signatures before and after outlier removal are largely consistent for the majority of clusters. However, considerable differences are observed for clusters 2 and 4.

When the images containing outlying clusters were examined, it is observed that they do not contain the defined characteristics of a spectral cell image : a slightly round shape in the middle. The images that are significant usually contain a roundish shape such as demonstrated in Fig. 3, and some of them contain noise in the form of vertical lines.

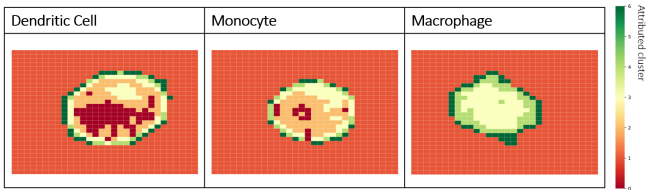


Fig 3 : Heat maps obtained by k-means clustering and the assignment of each pixel to a cluster

Results of the machine learning analysis are presented in Table 2. We have based the quality of the method's efficiency on the weighted average F1 score since the precision and recall values were similar among classes and the number of samples were uneven among classes. The decision tree method was the best performing, although all three algorithms exhibited similarly high performance.

Table 2: Results of the machine learning analysis

Method	Weighted Average F1 Score
Logistic Classifier	92
Support Vector Machines	91
Decision Tree	94

In sum, this proof-of-concept analysis confirms the feasibility of performing cellular phenotyping using Raman-based data. In the case of human derived PBMCs, we have shown that accurate predictions of monocyte, macrophage and dendritic cell phenotypes can be achieved. Our ongoing work includes the development of automated background removal methods to isolate pixels located within a cell based on their Raman spectral signatures. Our ultimate goal is to develop a machine learning pipeline where the pixels belonging to a given cell will be automatically detected from the background and their associated Raman spectra will be used to predict cellular phenotype.

5 References

[1] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," Information Sciences, vol. 622, pp. 178–210, Apr. 2023, doi: 10.1016/j.ins.2022.11.139.

[2] B. T. Grys et al., "Machine learning and computer vision approaches for phenotypic profiling," the Journal of Cell Biology, vol. 216, no. 1, pp. 65–71, Dec. 2016, doi: 10.1083/jcb.201610026.

[3] D. Chen et al., "Machine learning based methodology to identify cell shape phenotypes associated with microenvironmental cues," Biomaterials, vol. 104, pp. 104–118, Oct. 2016, doi: 10.1016/j.biomaterials.2016.06.040.

[4] J. Jin et al., "Robotic data acquisition with deep learning enables cell image–based prediction of transcriptomic phenotypes," Proceedings of the National Academy of Sciences of the United States of America, vol. 120, no. 1, Dec. 2022, doi: 10.1073/pnas.2210283120.

[5] J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon, and S. T. C. Wong, "Cellular phenotype Recognition for High-Content RNA Interference Genome-Wide Screening," SLAS Discovery, vol. 13, no. 1, pp. 29–39, Jan. 2008, doi: 10.1177/1087057107311223.

[6] K. J. I. Ember et al., "Raman spectroscopy and regenerative medicine: a review," Npj Regenerative Medicine, vol. 2, no. 1, May 2017, doi: 10.1038/s41536-017-0014-3.

[7] N. Feuerer et al., "Lipidome profiling with Raman microspectroscopy identifies macrophage response to surface topographies of implant materials," Proceedings of the National Academy of Sciences of the United States of America, vol. 118, no. 52, Dec. 2021, doi: 10.1073/pnas.2113694118.

[8] T. P. Wytock and A. E. Motter, "Distinguishing cell phenotype using cell epigenotype," Science Advances, vol. 6, no. 12, Mar. 2020, doi: 10.1126/sciadv.aax7798.

Introducing Interdependent Simple Temporal Networks under Uncertainty for Multi-agent Temporal Planning

Ajdin Sumic[1]

[1] Université Technologique de Tarbes, France

23 mai 2024

Résumé

Les réseaux temporels simples avec incertitude (STNU) constituent un formalisme largement utilisé pour représenter et raisonner sur des contraintes temporelles convexes en présence d'incertitude. Depuis leur introduction, ils ont été utilisés dans des applications de planification et d'ordonnancement pour modéliser des situations où l'agent d'ordonnancement ne contrôle pas certaines durées d'activité ou certains dates d'événements. Ce qu'il faut alors vérifier, c'est la contrôlabilité du réseau, c'est-à-dire qu'il existe une stratégie d'exécution valide quelles que soient les valeurs des contingents. Pour la première fois, cet article considère un modèle multi-agents, chacun ayant son propre STNU. Cependant, contrairement aux travaux précédents, le contrôle des durées d'activité est partagé entre les agents : ce que l'on appelle ici un contrat est une contrainte mutuelle contrôlable pour certains agents et incontrôlable pour d'autres. Nous proposerons une version modifiée des STNU qui sera composée d'un modèle Multi-agent Interdependent STNU.

Mots-clés

modèle formel, multi-agents, incertitude, planification temporelle.

Abstract

Simple Temporal Networks with Uncertainty are a powerful and widely used formalism for representing and reasoning over convex temporal constraints in the presence of uncertainty called contingent constraints. Since their introduction, they have been used in planning and scheduling applications to model situations where the scheduling agent does not control some activity durations or event timings. What needs to be checked is then the controllability *of the network, i.e., that there is a valid execution strategy whatever the values of the contingents. For the first time, this paper considers a multi-agent model, each having its own STNU. Still, as opposed to previous works here, the control over activity durations is shared among the agents : what is called here a contract is a mutual constraint controllable for some agents and uncontrollable for others. We will propose a modified version of STNUs that will be composed into a Multi-agent Interdependent STNUs model* [1].

Keywords

formal model, multi-agents, uncertainty, temporal planning.

1 Introduction

In many domains, such as planning and scheduling, systems diagnosis and control, business process management, etc, one needs to explicitly represent activities that may or must not overlap in time, last over some duration, and synchronize with timestamped expected events [4, 11, 10, 1]. The most commonly used model is Temporal Constraint Networks (TCN) [5] : nodes are time-points and edges express sets of possible durations relating them. A key issue is the ability to check the temporal satisfiability of the plan/system/process through the *consistency* of the TCN. The simplest class of TCN, called the Simple Temporal Network (STN), arises when they have only binary constraints with convex intervals of values. Consistency checking is made through polynomial-time propagation algorithms.

A well-known extension of STNs that handles uncertainties, called STNU (Simple Temporal Network with Uncertainty), has been proposed by [12]. An STNU contains uncertain (*contingent*) durations between time-points which means the effective duration is not under the control of the agent executing the plan, which is useful for addressing realistic dynamic and stochastic domains.

In STNUs, temporal consistency has been redefined in the form of *controllability* : an STNU is controllable if a strategy exists for executing the schedule, whatever the values are taken by the contingent durations. In [12], the authors introduce three levels of controllability (*Weak*, *Dynamic* and *Strong*) that are dependent on how and when the uncertainties are resolved, i.e., the actual durations are observed/known. Different approaches for checking them have then been proposed, widely discussed, and improved.

Considering several agents interacting in a common environment, each with its own set of temporal activities, has been studied but only for multiple STNs [6] or for a global

1. This work has been done in collaboration with Thierry Vidal (PhD supervisor), Andrea Micheli and Alessandro Cimatti, both researchers at the FBK research institute in Trento (Italy).

multi-agent STNU, all agents considering the same kind of contingent durations set by Nature [3] which hence cannot be modified in any way.

But there has been no work addressing the case where temporal uncertainty for one agent comes from decisions made by another agent, with which some synchronization is expected : the duration of a shared activity is then controllable (hence *flexible*) for say agent A_1 and contingent (only observed) for agent A_2. This paper's contribution is threefold : (1) first, it redefines the STNU model by clearly defining the execution and observation model of STNU ; (2) it proposes a new global model, called the *Multi-agent Interdependent STNUs model* (MISTNU), in which one can represent such activity durations whose status differs for distinct agents ; (3) it extend the three level of controllability into the MISTNU model.

The paper is organized as follows : first, some related work is exposed, and then we present revisited definitions of the STNU model according to execution and observation models and how they extend to our multi-agent problem. Then, we present the new model definition and revisit the controllability definitions. Then, we discuss a particular property of the model called *global controllability*. Finally, we conclude our contributions with a few prospects.

2 Related work

In the literature, some works have tackled the problem of multi-agent approaches for Simple Temporal Networks (STN) in a centralized manner by decoupling an STN into sub-networks or by distributing the control of a temporal network among a group of agents in real-time execution scenarios [6, 7]. A fully distributed approach with the notion of privacy between agents and MaSTN is given in [2]. However, this approach is incomplete in the sense that agents must agree in advance on some fixed durations, which prevents more dynamic solutions from being found. STNUs have received less attention in multi-agent settings. Still, a centralized approach that considers exogenous contingent constraints (a constraint that is contingent for everyone) is proposed in [3]. This work provides a decoupling algorithm for dynamic controllability (DC) that decouples a dynamically controllable STNU into sub-networks by ensuring all of them are DC using a MILP approach. A decentralized approach similar to the one in [2] for STNUs that allows communication delays is given in [13]. However, the proposed decoupling algorithm may still prune some solutions. Nevertheless, to the best of our knowledge, there exists no work in the literature that tackles the problem of durations that are controlled by another agent, as mentioned in the previous section.

In this paper, we formally define the multi-agent model with interdependencies in the form of partially controlled and negotiable durations. This model can be considered as an extension of the model presented in [3] as it also considers exogenous contingent constraints. In addition, we formally define its repair problem by tightening the negotiable contingent constraints and propose a series of encodings for the synthesis of valid repairs for the WC and SC. Differently from the literature reported above, we propose a model that doesn't require a strong decoupling algorithm that prunes executable solutions for the agents' solution space.

3 Definitions with Execution and Observation semantics

3.1 A revisited model for the single agent case

A Simple Temporal Network (*STN*)[5] is a pair, (V, E), where V is a set of time-points v_i representing event occurrence times, and E a set of temporal constraints between these time-points, in the form of convex intervals of possible durations. A reference time-point v_0 is usually added to V, which is the 'origin of time', depending on the application (might be, e.g., the current day at 0 :00). The goal is to assign values to time-points such that all constraints are satisfied, which is equivalent to assigning a value to each constraint in its interval domain.

An STN with Uncertainty (*STNU*) is an extension in which one distinguishes a subset of constraints whose values are parameters that cannot be assigned but will be observed [12].

As for the global planning/execution framework, we first recall that for a single agent, one usually reasons upon two phases : plan generation and execution. Considering specific constraints (resources, time, uncertainties) often requires an additional constraint satisfaction phase to validate the generated plan. Here, we focus on the problem of checking the satisfiability of a plan under temporal uncertainty. So we start the definition of our framework with a *planning*, a *validation*, and an *execution* phase.

Then, before defining the controllability levels and their associated semantics, we introduce the usual basic notations :

- minimal bounds $l_{ij} \in \mathbb{R} \cup \{-\infty\}$,
- maximal bounds $u_{ij} \in \mathbb{R} \cup \{+\infty\}$,
- $\preceq$ is the usual qualitative precedence relation between time-points : $v_i \preceq v_j \equiv l_{ij} \geqslant 0$.

Definition 1. ***(STNU)*** *An STNU is a tuple (V, E, C) such that :*

- *V is a set of time-points $\{v_0, v_1, \ldots, v_n\}$, partitioned into controllable (V_c) and uncontrollable (V_u) ;*
- *v_0 is the reference time-point : $\forall i, v_0 \preceq v_i$*
- *E is a set of requirement constraints $\{e_1, \ldots, e_{|E|}\}$, where each e_k is of the form $[l_{ij}, u_{ij}]$ with, $v_i, v_j \in V$.*
- *C is a set of contingent constraints $\{c_1, \ldots, c_{|C|}\}$, where each c_k is of the form $[l_{ij}, u_{ij}]$ with, $v_i \in V_c$, $v_j \in V_u$, and necessarily $v_i \preceq v_j$ [2]. We will note $end(c_k) = v_j$.*

Definition 2. ***(Schedule)*** *A* ***schedule*** *δ of an STNU $\mathcal{X} = (V, E, C)$ is a mapping δ from all the control-*

2. It is semantically not possible to have a contingent duration between two unordered time-points.

lable time-points to real values such that : $\delta = \{\delta(v_1), \ldots, \delta(v_{|V_c|})\}$ *with* $\forall i, v_i \in V_c,\ \delta : V_c \rightarrow \mathbb{R}$

Definition 3. ***(Situation and Projection)*** *Given an STNU* $\mathcal{X}$*, for all* $k = 1 \ldots |C|, c_k = [L_k, U_k], \Omega = [L_1, U_1] \times \ldots \times [L_{|C|}, U_{|C|}]$ *is the domain of all possible* ***situations*** *of* $\mathcal{X}$*.*
A tuple $\omega = \langle \omega_1 \in [L_1, U_1], \ldots, \omega_C \in [L_{|C|}, U_{|C|}] \rangle \in \Omega$ *is called a complete situation of* $\mathcal{X}$ *and* $\mathcal{X}_\omega$ *the* ***projection*** *of* $\mathcal{X}$*, is an STN where* $\mathcal{X}_\omega = (V, E \cup C')$ *with* $C' = \{[\omega_k, \omega_k]\} \mid c_k \in C$
Last, a schedule δ_ω *which satisfies all the constraints in* $\mathcal{X}_\omega$ *is called a* ***solution*** *of* $\mathcal{X}_\omega$*.*

From the previous definitions, one can notice that the existence of a solution schedule for a projection is simply equivalent to the STN consistency of that projection [5]. Then, in order to reach a semantically sound definition of the controllability properties, we need to express not only at which time a controllable time-point (resp. a contingent duration) is executed by the agent (resp. set by the owner), but also at which time that value is decided (resp. observed/known) by the execution controller in charge of the agent plan execution.

Definition 4. ***(Decisions and Observations)***
$\forall v_i \in V_c$*,* $dec(v_i)$ *is the time-point at which* $\delta(v_i)$ *is* ***decided*** *by the execution controller.*
$\forall \omega_k \in C$*,* $obs(\omega_k)$ *is the time-point at which* ω_k *is* ***observed*** *by the execution controller.*

Definition 5. ***(Weak Controllability (WC))*** *An STNU* $\mathcal{X}$ *is* ***weakly controllable*** *iff* $\forall \omega \in \Omega, \exists \delta$ *such that* δ *is a solution of* $\mathcal{X}_\omega$*.*
Execution semantics : $\forall \omega_k \in \omega$*,* $obs(\omega_k)$=v_0*, and the decision policy is free :* $\forall v_i \in V_c$*,* $dec(v_i) \preceq v_i$

In other words, WC assumes that values of contingent durations will be known **after** plan *validation*, but **before** the *execution* starts. Without any loss of generality, we will consider that all values are set at once exactly at the beginning of the execution : we call this process the *initialization* phase. Then, the schedule can be assigned at the beginning (fixed schedule) or during execution (flexible schedule).
For the definition of Dynamic controllability, we need an additional notation :

— $\omega^{\preceq v} = \{\omega_k \in \omega \text{ s.t. } end(c_k) \preceq v\}$ is the part of the situation ω which contingent constraints ending time-points precede v.

Definition 6. ***(Dynamic Controllability (DC))*** *An STNU* $\mathcal{X}$ *is* ***dynamically controllable*** *iff it is weakly controllable and* $\forall v_i \in V_c, \forall \omega, \omega' \in \Omega,\ \omega^{\preceq v_i} = \omega'^{\preceq v_i} \implies \delta(v_i) = \delta'(v_i)$
Execution semantics : $\forall \omega_k \in \omega$*,* $obs(\omega_k)$=$end(c_k)$*, and* $\forall v_i \in V_c$*,* $dec(v_i) = v_i$

In other words, DC assumes that values of contingent durations will be known **during** the *execution*, and exactly at the time of occurrence of the ending time-point of the contingent constraint. The schedule is also assigned during execution (flexible schedule), deciding the time of activation of some activity only when all preceding time-points have occurred. Hence, there is no *initialization* phase.

Definition 7. ***(Strong Controllability (SC))*** *An STNU* $\mathcal{X}$ *is* ***strongly controllable*** *iff* $\exists \delta$ *such that* $\forall \omega \in \Omega$*,* δ *is a solution of* $\mathcal{X}_\omega$*.*
Execution semantics : $\forall v_i \in V_c$*,* $dec(v_i) = v_0$*, and the observations are free : possibly no observation (*$\forall \omega_k \in \omega$*,* $obs(\omega_k) = \varnothing$*) or observations during execution that will just update the bounds of the constraints in the network.*

In other words, here, values of contingent durations may be known (or not) at any time since one demands a fixed schedule, which must be set before execution starts, for instance, because users or other agents need to know the precise timing in advance. So, that schedule must be *conformant* to any possible contingent values. Therefore, here, the *initialization* phase will be devoted to schedule assignment.

> **Example** : A medical vehicle must visit several villages to offer free COVID testing to the population. The number of people to show off and, hence, the duration of the stay in each village are uncertain. A valid flexible strategy needs to be designed and checked in advance anyway (*planning* and *validation*), knowing that the precise information will be known and sent to the agent by all the villages one hour before the route begins (*initialization*). So the agent will know exactly the durations its activities will take and can start *executing* the plan accordingly.
> Imagine the same example as before, but now the agent cannot know the number of people waiting in each village : the duration of their testing activities will hence be known only when the agent arrives in each village.
> Let's suppose now a rigid valid strategy is required, with village visiting times fixed in advance (hence at the *initialization* phase at the latest) and no prior knowledge of the number of people in each village. Then SC must be satisfied.

As a matter of conclusion for the single agent context, we have designed a general framework to deal with temporal uncertainty in planning, including 4 phases : *planning*, *validation*, *initialization*, and *execution*. There is only one possible backtrack : if the validation phase fails (controllability checking fails), the only thing to do is backtrack to the planning engine to find an alternative plan. We show these steps in Figure 1.

3.2 The case of multi-agent frameworks

For a single agent, the contingent duration assignment is exogenous ; hence, it is assumed that there is no way to influence them. Saying that 'Nature' will assign those values is a usual way to capture that.
However, in a multi-agent environment, a contingent duration may be decided by another agent. So even though the agent that depends on it cannot decide its value, it might be possible to communicate with the *owner* agent to influence it and change the possible values. Intuitively, that

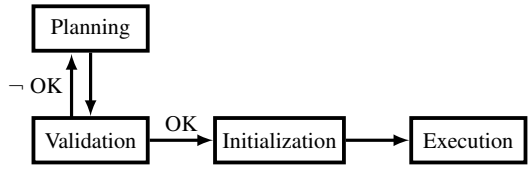


FIGURE 1 – The Figure shows the global framework for a single agent. It includes the different steps of the planning problem for a single agent. Please note that the initialization phase only exists for WC and SC as DC implies to decide during execution.

owner agent will decide the duration but commits to assign it within the lower and upper bounds. Some other *compliant* agents depend on that constraint, which is contingent on their network. The former agent (owner) communicates its commitment (lower and upper bound) to the compliant agents at some point before the agents check the controllability of their networks. We call this kind of commitment a *contract* between the owner and the compliant agents, where a compliant agent has the right to request new bounds as long as it guarantees the satisfaction of both agents, i.e. to ensure each is controllable.

First of all, to get a complete picture, we must recall that usually, in multi-agent planning, there might be a first phase of *task allocation* to distribute the goals to achieve to the agents. That phase is usually centralized or devoted to specialized agents. Then the *planning* may be centralized, the global plan being decoupled into separate agent plans or distributed, each agent building its own plan individually. In both cases anyway, dependencies and synchronizations must be considered, calling for some way to share activities controlled by one agent but which outcome is needed by another.

In the end, it is assumed that *execution* will be launched concurrently by all the agents. But before that, after the planning is completed, there is still the need to *validate* the individual plans through, in our case, temporal controllability checking algorithms. Once again, it can be done by a central agent having a view of all the plans or in a distributed way by each agent.

The way that can be done then depends on the observation and decision semantics that have been introduced in the previous section. First, if the application requires that all schedules must be fixed in advance, that means one needs to consider a common *initialization* phase to fix those schedules, which means all agents must ensure SC.

Second, if flexible schedules are allowed, then WC or DC apply. The difference depends on how and when the 'owner' agents set and communicate the values of activity durations on which other agents are dependent. If that is done before the execution, then they must consider a common *initialization* phase when all agents will decide and exchange the shared activities durations, which is consistent with the definition of WC. If such decisions are to be taken by each agent during the execution of the plans and communicated as soon as they occur and with no delay, then DC applies.

Example : In a hospital environment, a patient has to follow a path through several services that manage their timetabling separately. The path has to satisfy partially ordered constraints between the different services. Now consider that the durations of activities in each service depend, for instance, on the patient features that will only be assessed at the time each activity begins. Then, if all services require a rigid timetable where each operation has a unique starting time that is fixed in advance and appears in the calendar, then SC applies for all ; if flexibility is allowed in the sense that operations start times might be decided on the fly, then all services must account for some global DC. Now, consider that each service does not know in advance how many people will be working that day (due to last-minute staff allocation and potential sick leaves), which affects the duration of the patient care. In that case, a plan must be proven valid in advance without such information, but all services will know and exchange them through a common *initialization* phase when the day starts, which implies WC.

Of course, this framework is only relevant in homogeneous multi-agent problems when all agents have the same behavior, which we assume here. If not, the classical controllability checking algorithms will not be applicable, which is not our focus. It also assumes that the initialization phase is synchronous, i.e., all decisions must be taken by all agents at the same time, without communication nor hierarchy between them ; otherwise, the semantics of WC (or SC) will not be met, and what needs to be checked will also be something different that is out of the scope of our current study. Hence, extending the well-defined semantics of WC, SC, and somehow DC to a multi-agent setting aligns with specific and somehow restricted semantics of the behavior of the team of agents.

Then, going back to the *validation* phase, if at least one agent is not controllable, then it is still possible to backtrack to the planning phase to find an alternative. Still, it is possible to negotiate with other agents to change the values of some contracts they control. If the 'owner' agent agrees to change the bounds of the contract controls so that both agents are now controllable, the problem is solved without needing a more complex replanning stage. This new phase is the *repair* phase and may require controllability checking algorithms capable of diagnosing the source of uncontrollability [9].

Once again, the repair phase might be centralized or distributed. It may also succeed or fail. If it fails (no solution exists), they will have no choice but to backtrack to the planning phase.

This new global framework is synthesized in Figure 2.

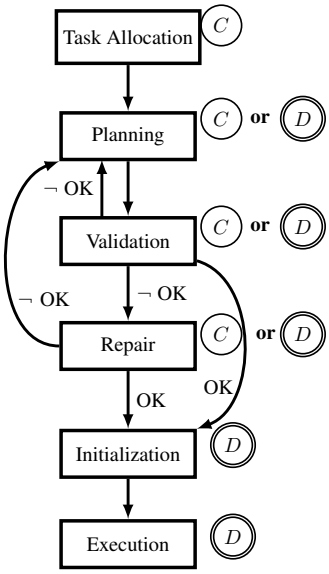


FIGURE 2 – The figure shows the different steps of the problem we are interested in. Node C and double circled node D refer to the possibility for that step to be either centralized (C) or distributed (D). Please note that the initialization phase only exists for WC and SC as DC implies to decide/observe during execution.

4 The MISTNU model

4.1 Definition

The concept of negotiable contingent constraints arises in a multi-agent context when such a constraint is not controlled by Nature but by one agent of the system. Hence we first need to slightly modify the definition of an STNU in the form of a *Contracting* STNU (cSTNU) by explicitly considering some constraints as *owned* by the agent and relating the contingent constraints to so-called *contracts*, the bounds and the owner of such contracts being now defined outside the model, to be shared by several agents [3].

Definition 8. ***(cSTNU)*** *A Contracting STNU (cSTNU) is an STNU where links representing contracts are labeled. A cSTNU is a tuple $\mathcal{S} = \langle V, R, W, E, C, O\rangle$ such that :*

- *V is a set of time points, partitioned into controllable (V_c) and uncontrollable V_u*
- *R and W are sets of contracts, such that $R \cap W = \varnothing$*
- *E is a set of requirement links of the form $v_i \xrightarrow{[l,u]} v_j$;*
- *C is a set of labeled contingent links of the form $v_i \overset{p}{\dashrightarrow} v_j$ where $p \in R$.*
- *O is a set of owned contract links of the form either $v_i \xrightarrow{p} v_j$ or $v_i \overset{p}{\dashrightarrow} v_j$, one for each contract $p \in W$.*

In addition, we require that $\forall v_j \in V_u$, there exists a unique labeled contingent link of the form $v_i \overset{p}{\dashrightarrow} v_j$ in $C \cup O$.

One can notice that an agent may consider its owned contracts contingent or requirement constraints, depending on its policy. Forcing that constraint to be contingent prevents the agent from shrinking it when running some local propagation algorithm to respect its commitment to others or to retain maximal flexibility on this contract at execution time. Hence, it refuses any form of reduction to allow other agents to regain control. That shall enable us to tune our global model accordingly in settings where agents are more or less selfish or cooperative.

Then, from Definition 8, we define the system of Multi-agent Interdependent Simple Temporal Networks under Uncertainty (MISTNU) as follows :

Definition 9. ***(MISTNU)*** *A MISTNU is a tuple $\mathcal{G} = \langle A, \Sigma, B\rangle$ such that :*

- *A is a set of agents $\{a_1, a_2, \ldots, a_n\}$;*
- *Σ is a set of cSTNUs $\mathcal{S}_a = \langle V_a, R_a, W_a, E_a, C_a, O_a\rangle$, one for every $a \in A$, such that*
  - *$\forall a \in A$, $v_z \in V_a$, where v_z is the reference time point ;*
  - *for every pair of agents $a, b \in A$, $W_a \cap W_b = \varnothing$*
- *B is a map from contracts to bounds B : $\bigcup_{a\in A}(R_a \cup W_a) \rightarrow \mathbb{R}^2$. For the sake of this paper, we write l and u for $\langle l_p, u_p\rangle = B(p)$.*

A MISTNU is a model $\mathcal{G}$ composed of a set of agents, where each agent has its own cSTNU that might own and read some contracts in which some can be negotiated. Then, $\mathcal{G}$ is also composed of a map of contracts to bounds B that indicates for every contract its lower/upper bound duration denoted as a tuple $\langle l, u\rangle$ with l and u respectively the lower and upper bound duration. This allows us to reduce a cSTNU into an STNU by applying B :

Definition 10. ***(cSTNU reduction)*** *Given a cSTNU $\mathcal{S} = \langle V, R, W, E, C, O\rangle$ and a map $B : W \cup R \rightarrow \mathbb{R}^2$ giving bounds to contracts, $\mathcal{S}$ can be reduced to an STNU $\mathcal{X} \doteq \langle V, E', C'\rangle$ with :*

- *$E' = E \cup \{v_i \xrightarrow{[l,u]} v_j \mid v_i \xrightarrow{p} v_j \in O,\ B(p) = \langle l, u\rangle\}$*
- *$C' = \{v_i \overset{[l,u]}{\dashrightarrow} v_j \mid v_i \overset{p}{\dashrightarrow} v_j \in C \cup O,\ B(p) = \langle l, u\rangle\}$*

As a cSTNU can be reduced to an STNU with Definition 10, the definitions of its situations and projections directly come from Definition 3. However, for the MISTNU model, it is different. We hence first provide further definitions :

- for any agent a, $P_a = R_a \cup W_a$ is the set of all its contracts
- $P = \bigcup_a P_a$ is the set of all contracts of all the agents
- for any cSTNU $\mathcal{S}$, $\sigma(\mathcal{S}, p) = v_i$ s.t. $\exists v_j, v_i \overset{p}{\dashrightarrow} v_j \in C \cup O$ or $v_i \xrightarrow{p} v_j \in O$.

Definition 11. ***(MISTNU situation)*** *Given a MISTNU $\mathcal{G} = \langle A, S, B\rangle$, the **situations** of $\mathcal{G}$ is a tuple of reals $\Omega_{\mathcal{G}}$ defined as :*

3. As already mentioned, an exogenous contingent constraint can still be modeled here, being related to a contract without any owner; which will be allowed in the global model.

$$\bigtimes_{p \in P} [l_p, u_p].$$

*A **situation** is an element ω of $\Omega_{\mathcal{G}}$ and we write $\omega(p)$ with $p \in P$ to indicate the element in ω associated with p in the cross product.*

Definition 12. *(**MISTNU projection**) Given a MISTNU $\mathcal{G} = \langle A, S, B \rangle$, and a situation ω, the **projection** $\mathcal{G}^\omega$ is a model $\langle A, \Sigma^\omega, B^\omega \rangle$ where :*

— *B^ω is a map from contracts to bounds $B^\omega : P \to \mathbb{R}^2$ such that :*

$$B^\omega = \{\langle \omega(p), \omega(p) \rangle \mid p \in B\}$$

— *Σ^ω is a set of STN $\mathcal{X}_a^\omega = \langle V_a, E'_a, \varnothing \rangle$, one per $a \in A$, such that for $\mathcal{S}_a = \langle V_a, R_a, W_a, E_a, C_a, O_a \rangle$ in Σ :*

$$E'_a = E_a \cup \{v_i \xrightarrow{B^\omega(p)} v_j \mid v_i \xrightarrow{p} v_j \in O_a\} \cup \{v_i \xrightarrow{B^\omega(p)} v_j \mid v_i \overset{p}{\dashrightarrow} v_j \in C_a \cup O_a\}$$

It is important to point out that as the system considers temporal networks created independently, the model must ensure that all the temporal networks in Σ are temporally well-formed. This means that for any contract of the form $v_i \xrightarrow{p} v_j$ or $v_i \overset{p}{\dashrightarrow} v_j$ shared among a set of agents, the date in time on which its execution starts (v_i) and finishes (v_j) must be the same in each of the temporal networks where the contract is involved. As the contract duration between v_i and v_j is guaranteed to be the same by B, we need to ensure that it is also the case for the start time-point v_i.

Definition 13. *(**Temporally well-formed**) A MISTNU $\mathcal{G} = \langle A, \Sigma, B \rangle$ is **temporally well-formed** if for every projection $\omega \in \Omega_{\mathcal{G}}$, for every pair of distinct agents a_1 and a_2 and for every contract $p \in P_1 \cap P_2$, all consistent schedules δ_1 of $\mathcal{X}_1^\omega$ and δ_2 of $\mathcal{X}_2^\omega$ are such that*

$$\delta_1(\sigma(\mathcal{S}_1, p)) = \delta_2(\sigma(\mathcal{S}_2, p)).$$

Theorem 1. *Let T be a map from a contract p to its unique predecessor p' or v_z, $T : P \to P \cup \{v_z\}$, such that $\forall p$, there is no sequence of the form $T(T(\dots T(p))) = p$.*
If for every agent $a \in A$, $\forall p \in (W_a \cup R_a)$:

— *$T(p) = v_z \iff v_z \xrightarrow{p} v_j \in O_a$ or $v_z \overset{p}{\dashrightarrow} v_j \in O_a \cup C_a$*

— *$T(p) = p' \iff (v_i \xrightarrow{p} v_j \in O_a$ or $v_i \overset{p}{\dashrightarrow} v_j \in O_a \cup C_a) \wedge (v_k \xrightarrow{p'} v_i \in O_a$ or $v_k \overset{p'}{\dashrightarrow} v_i \in O_a \cup C_a)$*

Proof. It is easy to see that if the contracts are chained to each other such that it can be represented as a directed graph of the form of a tree that starts from z, the common reference time-point, then a contract either starts at z or directly after the end of another contract. Then, ensuring that each agent reads all the needed contracts to reach z guarantees the networks to be temporally well-formed as map B will guarantee that each contract's start and end time-points are timestamped from z.
Let's suppose G is built such that a map T can be extracted and contains all the contracts, and G is not temporally consistent. This would mean that a contract exists and is shared with at least two agents, such that the set of possible start and end times of such a contract is different. However, this is not possible as each contract's start and end time-points are timestamped from z, and z is the common reference time-point for all the agents in G. Thus, the agents that share this contract also share the same set of possible start and end times for the contract.
However, Theorem 1 is not an absolute condition to follow to guarantee networks to be temporally well-formed. Another way would be to require the agent to agree on due dates for starting a contract or a fixed duration between two contracts.
Figure 3 shows an example of the MISTNU model with three agents (a, b, c) using the map T to ensure being Temporally well-formed between the contracts (p, a, r, s, t, u) with the contract u owned by Nature. This figure also shows the communication schema between the agents.

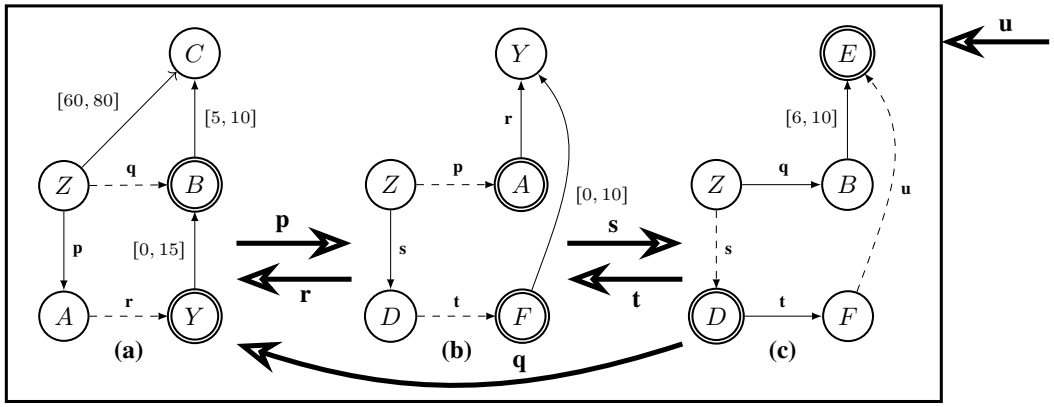
4.2 Controllability

In previous work, the controllability of MaSTN was defined as having all the network (STN) to be consistent [2]. In the same manner, the controllability of a MISTNU can be defined by considering $\forall S_a \in \Sigma$, each S_a to be controllable, i.e., to be dynamically controllable, the system would impose all the networks (S_a) to be dynamically controllable. Therefore, using Definition 10, it's easy to check the controllability of a cSTNU as this amounts to checking the controllability of an STNU, which has already been tackled in the literature. We called such controllability checking the *Local Controllability* as it implies checking the controllability of each network locally, and we defined it in Definition 14.

Definition 14. *(**Local Controllability**) Given a MISTNU $\mathcal{G} = \langle A, \Sigma, B \rangle$, and $\Omega_{\mathcal{G}}$ the situations of $\mathcal{G}$, we define the **local controllability** L_τ of $\mathcal{G}$ with $\tau = \{WC, DC, SC\}$ as :*

$$L_\tau \equiv \forall S_a \in \Sigma, S_a \text{ is } \tau - controllable.$$

However, this paper argues for the existence of another type of controllability property for MISTNU. Indeed, this model is the first to consider interdependent networks with related constraints that are shared and implied by one agent ; such constraints are controllable or not. Therefore, we believe that even though a network may not be dynamically controllable due to some contracts as contingents, the MISTNU could still be dynamically controllable due to the decision of the owners of these contracts that will guarantee the execution of the non-DC networks. We believe such cases happen due to private constraints not being visible to others, which forces the contract owner to indirectly satisfy the private constraints of others. In the following section, we



(a)

Contract	Bounds
p	[20, 25]
q	[60, 75]
r	[25, 35]
s	[30, 35]
t	[15, 25]
u	[15, 20]

(b).

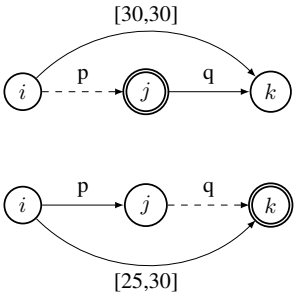
FIGURE 3 – Example of the MISTNU model $\mathcal{G}$ with three agents a, b, and c and their networks. Nodes represent time-points, doubly circled nodes are uncontrollable time-points, solid edges are requirement constraints, and dashed edges are labeled contingent constraints. $\mathcal{G}$ comprises six contracts : p is owned by A, r, and s by B, q, and t by C, and u by Nature. In addition, we show the communication schema between agents through the bigger edges (black), e.q., agent a communicates p to agent b. In Figure (b), we show the associated bounds of the contracts

will argue the existence of such property that we call *global controllability*.

5 Discussing Global Controllability

In this section, we argue the existence of local controllability and global controllability properties. Local controllability is the property of each agent being locally τ-controllable. Global controllability is more subtle as it represents the case where the system would be τ-controllable without requiring all networks to be locally τ-controllable. However, you can still ensure that execution will go smoothly due to agents' decisions over their controllable time points. We show a concrete example of global controllability in Figure 4a with two networks : the upper one belonging to agent a_1, which is DC, and the lower one belonging to agent a_2, which is not DC because you cannot decide the contract p such that you satisfy the private constraint $i \xrightarrow{[25,30]} k$ with the contract q. However, we can see that because of the private constraint $i \xrightarrow{[30,30]} k$ of agent a_1, during execution, it will decide a value for q according to its observation on p to satisfy its private constraint. Indirectly, its decision on q will always satisfy the private constraint $i \xrightarrow{[25,30]} k$ of a_2. Consequently, local controllability is too restrictive as in this example, agent a_2 would negotiate the bounds of q to guarantee its local controllability, but in reality, it's unnecessary.

The previous example focuses only on the problem of dynamic controllability, but obviously, such a situation should also exist for the problem of Weak and Strong Controllability. Therefore two questions arise from such property : How could global controllability be formally defined ? and how to verify/check such property. For the latter one, existing algorithms are insufficient to check global controllability as they would require a global network to check controllability and enable agents to distribute the execution among the agents by guaranteeing it will not fail. This is not possible with interdependent networks with shared constraints being either controllable or contingent. Thus, a fully distributed controllability-checking algorithm would be required to check such controllability as the one proposed in [8] for MaSTN. Another way is to share all the constraints linked to the contracts between the agents, but in the worst case, this would amount to sharing the whole network, which is a problem when privacy is required. Nonetheless, such questions are promising and pave the way for future work.



(a)

Contract	Bounds
p	[10, 20]
q	[10, 20]

(b).

FIGURE 4 – In Figure (a), we show two agent networks, the upper one belonging to agent a_1 and the lower one to agent a_2. In Figure (b), we show the associate table for the contracts and their bounds.

6 Conclusion

This paper presents a new multi-agent model for temporal problems under uncertainty called MISTNU that considers independent networks where, for an agent network, the execution of some tasks might be controlled by other agents. Hence, it's possible for an agent to negotiate the duration of such tasks. This paper formally defines the cSTNU model, which is an extension of the STNU model, the MISTNU model, and the problem of checking its controllability defi-

ned by the local controllability. In addition, the paper argues the existence of another controllability property called global controllability, which, contrary to local controllability, is a totally new property in which current algorithms are not enough to verify such property. Future work will focus on this new controllability property and try to formally define it with the current semantics of Temporal Networks under Uncertainty and work on an algorithm capable of checking such property for the three levels of controllability.

Références

[1] Arthur BIT-MONNOT et al. "FAPE : a Constraint-based Planner for Generative and Hierarchical Temporal Planning". In : *CoRR* abs/2010.13121 (2020).

[2] James C. BOERKOEL et Edmund H. DURFEE. "Distributed Reasoning for Multiagent Simple Temporal Problems". In : *J. Artif. Intell. Res.* 47 (2013), p. 95-156.

[3] Guillaume CASANOVA et al. "Solving dynamic controllability problem of multi-agent plans with uncertainty using mixed integer linear programming." In : Frontiers in Artificial Intelligence and Applications 285 (2016), p. 930-938.

[4] Amedeo CESTA, Angelo ODDI et Stephen F. SMITH. "A Constraint-Based Method for Project Scheduling with Time Windows". In : *J. Heuristics* 8.1 (2002), p. 109-136.

[5] Rina DECHTER, Itay MEIRI et Judea PEARL. "Temporal constraint networks". In : *Artificial intelligence* 49.1-3 (1991), p. 61-95.

[6] Luke HUNSBERGER. "Algorithms for a Temporal Decoupling Problem in Multi-Agent Planning". In : *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*. 2002, p. 468-475.

[7] Luke HUNSBERGER. "Distributing the control of a temporal network among multiple agents". In : *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*. 2003, p. 899-906.

[8] Shufeng KONG, Jae Hee LEE et Sanjiang LI. "Multiagent Simple Temporal Problem : The Arc-Consistency Approach". In : *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,* 2018.

[9] Josef LUBAS, Marco FRANCESCHETTI et Johann EDER. "Resolving conflicts in process models with temporal constraints". In : *Proceedings of the ER Forum and PhD Symposium*. 2022.

[10] Alessandro VALENTINI, Andrea MICHELI et Alessandro CIMATTI. "Temporal Planning with Intermediate Conditions and Effects". In : *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020,* 2020, p. 9975-9982.

[11] Gérard VERFAILLIE, Cédric PRALET et Michel LEMAITRE. "How to model planning and scheduling problems using constraint networks on timelines". In : *Knowl. Eng. Rev.* 25.3 (2010), p. 319-336.

[12] Thierry VIDAL et Hélène FARGIER. "Handling contingency in temporal constraint networks : from consistency to controllabilities". In : *Journal of Experimental & Theoretical Artificial Intelligence* 11.1 (1999), p. 23-45.

[13] Yuening ZHANG et Brian C. WILLIAMS. "Privacy-Preserving Algorithm for Decoupling of Multi-Agent Plans with Uncertainty". In : *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling, ICAPS 2021, Guangzhou, China (virtual), August 2-13, 2021*. AAAI Press, 2021, p. 426-434.

Session 2 : Apprentissage par renforcement

2.2

Multi-objective reinforcement learning: an ethical perspective

T. Deschamps[1], R. Chaput[1], L. Matignon[1]

[1] Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France

timon.deschamps@liris.cnrs.fr

Abstract

Reinforcement learning (RL) is becoming more prevalent in practical domains with human implications, raising ethical questions. Specifically, multi-objective RL has been argued to be an ideal framework for modeling real-world problems and developing human-aligned artificial intelligence. However, the ethical dimension remains underexplored in the field, and no survey covers this aspect. Hence, we propose a review of multi-objective RL from an ethical perspective, highlighting existing works, gaps in the literature, important considerations, and potential areas for future research.

Keywords

Reinforcement learning, multi-objective decision making, machine ethics.

Résumé

L'apprentissage par renforcement est de plus en plus employé pour des applications pratiques impactant l'humain, soulevant ainsi des questions éthiques. Spécifiquement, l'apprentissage par renforcement multi-objectif est considéré comme un cadre idéal pour la modélisation de problèmes concrets et le développement de systèmes d'intelligence artificielle alignés sur l'humain. Peu de travaux du domaine adoptent une perspective éthique, et les études existantes ne couvrent pas cet aspect. Ainsi, nous proposons une revue de l'apprentissage par renforcement multi-objectif d'un point de vue éthique, en détaillant les travaux existants, les lacunes de la littérature, les considérations importantes, et les potentielles pistes de recherche futures.

Mots-clés

Apprentissage par renforcement, prise de décision multi-objectifs, éthique computationnelle.

1 Introduction

The field of *reinforcement learning* (RL) has recently seen numerous breakthroughs, notably featuring artificial intelligence (AI) agents beating humans at a wide variety of games [51, 37, 8]. RL has also been applied to multiple real-world problems, with a potentially large impact on societies (e.g., language model alignment [38], nuclear fusion control [16], healthcare [70]). This calls for the study of the ethical issues that may arise from such uses, and the development of techniques to ensure that the agents have a behavior deemed *ethically-aligned* with human principles; so as to guarantee this technology will be beneficial to humanity. This is a complex endeavor, and a few works have started paving the way [67, 54].

In this paper, we focus on *multi-objective reinforcement learning* (MORL), a sub-field of RL in which multiple potentially conflicting goals are considered rather than a single one. Following the RL trend, MORL is being increasingly used in real world applications such as public bicycle dispatching [14] or energy management [18]. It has been argued that aligning AI with human goals is a multi-objective problem [60], making the study of MORL interesting in this regard. A few multi-objective decision making surveys have been published [24, 49], focusing on the theory and applications of multi-objective decision making algorithms. The goal of this work is to highlight the need for morally-aligned multi-objective methods and to conduct an analysis of MORL from an ethical standpoint. To do so, we start by discussing and categorizing existing MORL methods, before introducing important ethical considerations which we use to emphasize important gaps in the literature.

2 A motivating example

To illustrate the ethical concerns that can arise when AI agents are deployed in the real-world, we propose to study the case of self-driving vehicles. This sector has been increasingly interested in RL [27], which is viewed as a suitable paradigm: vehicles can be represented by agents taking actions such as steering and accelerating within an environment (road network).

RL agents typically optimize for a single objective (e.g., *speed*). However, when dealing with complex use-cases or when humans can be impacted, more flexibility is desirable to account for additional goals like *cost saving* and *comfort*.

MORL is ideal in such contexts, as it allows for representing and compromising between multiple objectives. This multi-objective aspect is essential when autonomous vehicles are deployed on real roads, as they will inevitably have to handle complex ethical dilemmas which require weighting between conflicting moral values (e.g., ensuring safety for both passengers and surrounding pedestrians).

This example motivates the study of MORL agents with an ethically-aligned behavior, and we will extend it throughout this paper to illustrate some of the notions discussed.

3 Background

3.1 Reinforcement learning

Reinforcement learning is a general framework to solve problems in which an agent alternatively takes *actions* and receives *observations* and *rewards* from an environment, and aims at maximizing the cumulative reward obtained. RL is usually modeled as a *Markov decision process* (MDP), defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$, where:

- $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively;
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, i.e., the probability of transitioning to a state s_{t+1} given that the action a_t was taken at time step t in state s_t;
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, which outputs a scalar reward for a given (s_t, a_t, s_{t+1}) tuple;
- $\gamma \in [0, 1)$ is a discount factor modulating the importance of long term rewards.

The agent acts according to a *stochastic policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which gives the probability of taking any action $a \in \mathcal{A}$ given the current state $s \in \mathcal{S}$. If in every state one of the actions is selected with probability 1, the policy becomes *deterministic*, denoted $\pi : \mathcal{S} \to \mathcal{A}$[1].
At any time step t, we can compute the sum of future rewards, or *return*, defined as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k. \quad (1)$$

The *value* of a state $V^\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$ is the expected return for an agent located in this state at time step t and following policy π. In turn, our goal is to find the optimal policy π^* which, when followed, maximizes the value for all states in $\mathcal{S}$.
We refer readers interested in diving deeper into reinforcement learning to the seminal book by Sutton and Barto [52]. To this day, RL remains a highly active discipline, with many emerging sub-fields such as multi-agent RL [22, 11], model-based RL [32] and multi-objective RL, the latter of which we discuss in the following section.

3.2 Multi-objective reinforcement learning

The field of multi-objective reinforcement learning (MORL) deals with *multi-objective Markov decision processes* (MOMDPs). MOMDPs differ from regular MDPs only in that the reward (and by extension the value) is vector-valued: $\mathbf{r} \in \mathbb{R}^m$ with m objectives[2]. This implies that finding a single optimal policy via a simple maximization process becomes impossible, as maximizing one of the component of the reward vector (called objective) could lead to a decrease in another one.
Utility functions, also referred to as scalarization functions, map the value vector $\mathbf{V}^\pi$ of a given policy π to a single scalar ($u : \mathbb{R}^m \to \mathbb{R}$). They provide a convenient way to formalize a decision maker's preferences and trade-offs over the objectives.
A common and simple class of utility functions are linear utilities, denoted as $u(\mathbf{V}^\pi) = \mathbf{w}^\top \mathbf{V}^\pi$, which combines a weight vector $\mathbf{w}$[3] and the value vector using a linear combination. Intuitively, each weight $w_o \in \mathbf{w}$ represent the importance of the associated objective $\mathbf{V}^\pi_o$.
If we have access to a linear utility function for the user, we can use it to simplify the problem back into the single-objective RL setting and solve it with classical methods. However, this is not an option when the utility function is not fully known in advance or is non-linear, which represents a large portion of real-life scenarios (see the motivating scenarios presented in [24]).
In these settings, we focus instead on a set of optimal policies: the *Pareto front* (PF). A policy $\pi \in \Pi$ belongs to the Pareto front PF(Π) if it is not Pareto-dominated by any other policy. The Pareto-dominance of a policy π over a policy π' is defined as:

$$\pi \succ_P \pi' := (\forall o : \mathbf{V}^\pi_o \geq \mathbf{V}^{\pi'}_o) \wedge (\exists o : \mathbf{V}^\pi_o > \mathbf{V}^{\pi'}_o). \quad (2)$$

In plain words, π's associated value vector is greater or equal to the one associated with π' for all objectives o, and strictly greater for at least one.
As the PF can have multiple policies with the same induced value function, we often refer to a *Pareto coverage set* (PCS), which simply retains a single policy for each non Pareto-dominated value function. Computing a PCS guarantees that we have access to all policies that are optimal under some monotonically increasing utility function. This allows to adapt to changes in the user's preferences while making minimal assumptions about u. In practice, however, PF and PCS can be prohibitively large to compute. Recent work [49, 24, 42] has argued for a utility-based approach, in which we use information we have about the utility function to guide our search in the space of policies. For example, when u is known to be linear, we can restrict our focus to subsets of the PF referred as *convex coverage sets* (CCS), which contain all maximal policies under this assumption.
To illustrate these concepts, let's take our example from section 2. Keeping only speed and comfort as objectives for ease of representation, we can visualize the PF and a CCS in figure 1. Each point represents a policy and its associated value vector, compromising between the two objectives. We can see that increasing speed usually leads to a decrease in comfort, but it is not always the case (for instance, faster speeds on very uneven roads could smooth out the cruise). Notice that points belonging to the represented CCS are also part of the Pareto front (in fact CCS(Π) $\in$ PF(Π)). Here, point b is not Pareto-dominated (see eq. 2) by either point a or c (nor by any other point). Furthermore, there is no $\mathbf{w}$ for which a linear scalarization would lead to b being maximal. Thus, we can conclude that b belongs to the PF but not to a CCS.

[1] Some work also use $\mu(s) = a$ specifically for deterministic policies.
[2] Note that we use the standard notation of boldface for vector variables.
[3] Note that by convention and without loss of generality, $\mathbf{w}$ is drawn from a m-simplex, i.e., $\forall w_i \in \mathbf{w}, w_i \geq 0$ and $\sum_{i=1}^m w_i = 1$.

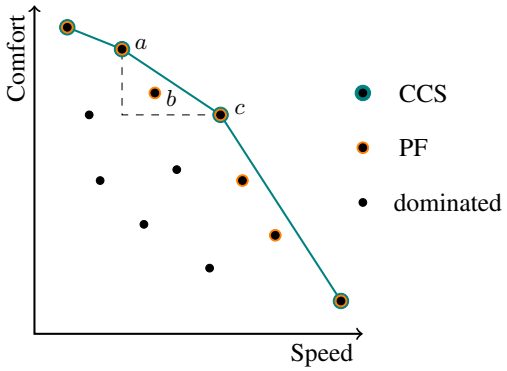


Figure 1: Visualization of the Pareto front and a convex coverage set for a 2-objective self-driving car example.

When using a scalarization function, two optimization criteria naturally arise: *scalarized expected returns* (SER) and *expected scalarized returns* (ESR). To optimize for SER, we scalarize an expectation over multiple runs of the vector-valued returns of a policy, whereas optimizing for ESR requires having a scalarized return for each run, and then computing an expectation over them. These two criteria have different properties and should be used in different scenarios. SER, the most studied one, is particularly suited when we aim to optimize over many policy executions, whereas using the ESR criterion is better to ensure that each execution is maximal over our utility function.
See [49, 24] for a detailed overview of the theory and methods of multi-objective reinforcement learning.

3.3 Machine ethics

As autonomous machines are increasingly integrated into domains with significant human implication, their impact, whether it be positive or negative, requires investigation. *Machine ethics* is concerned with how we can ensure that AI agents demonstrate ethically-aligned behaviors, i.e., behaviors whose outcomes are moral according to a chosen ethical framework [6]. In turn, we aim for them to be *explicit ethical agents* [34], i.e., agents whose inner workings allow for an explicit representation and computation of ethics. To evaluate the ethical alignment of these behaviors, we leverage insights from *normative ethics*. As it is concerned with the morality of actions, this field provides a suitable framework for such an analysis.
Normative ethics encompasses three main schools of thought: *consequentialism*, *virtue ethics* and *deontology*. According to consequentialism, only the outcome of actions are necessary to judge whether these actions are ethical or not. Consequentialist ethics are most known for utilitarianism, which argues that in every situation, the ethical action is the one that maximizes happiness and well-being for all. Virtue ethics shift the focus from the action to its motivation. In this view, an agent is ethical if it acts according to set values (e.g., confidence, honour, freedom). Deontology takes a rule-based approach, in which actions can either be right or wrong according to a list of principles. Kantian ethics is a prime example of deontological ethical theories. Refer to [56] for an extensive review of western moral philosophy.
As discussed in section 3.1, a defining feature of reinforcement learning agents is their ability to take actions in an environment, making normative ethics a natural framework for studying the ethical alignment of their behavior.
In fact, reinforcement learning has been characterized as an ideal framework to develop ethical agents [1], and recent work has surveyed RL-based moral learning agents [54]. Furthermore, we argue that the formulation of the reinforcement learning objective as the maximization of a future reward signal naturally aligns with a number of branches of consequentialism. Although some methods allow for the application of deontological ethics into RL [23, 5], none to our knowledge directly takes a moral perspective and is adapted to the multi-objective setting. Finally, it has been argued that MORL, on top of being ideal to model a number of real-world problems [24], is a particularly fitting framework to develop human-aligned artificial intelligence [60]. Moreover, we suggest that it is also suited for modeling virtue ethics, as each component of the vector-valued reward can encode a virtue to be followed. For a comprehensive overview of machine ethics implementations, refer to the survey of Tolmeijer et al. [57], which offers a well-organized taxonomy and extensive bibliography.

4 Classical MORL methods

The most commonly used taxonomy for multi-objective sequential decision making [49, 24] classifies methods depending on the type of policy and utility function they consider, resulting in a number of criteria:

- *single vs. multiple policies*: As mentioned in sec. 3.2, algorithms can either output a single solution (if the utility is fixed and known in advance) or a set of optimal policies. Multi-policy methods are **more costly**, but allow for **greater flexibility**: as fewer assumptions are made on the utility function, the user can adapt in the face of new data or changing contexts.

- *deterministic vs. stochastic policies*: While it was shown that **stochastic policies can outperform deterministic ones** in some environments [65, 58], their use can become ethically questionable or impossible in domains requiring strong guarantees (e.g., medical treatments).

- *linear vs. monotonically increasing u*: Using linear utility functions **simplifies the learning process**, allowing the MORL problem to be reduced to a single-objective one (for single-policy algorithms) or to restrict the policy search to a CCS (for multi-policy algorithms). Using monotonically increasing utility functions enables the expression of a much **richer relationship between the objectives**, at the cost of a **more complex learning process**, as the entire PF has to be considered.

	single policy (known u)		multiple policies (unknown u)	
	deterministic	stochastic	deterministic	stochastic
linear scalarization	one policy in Π_{DS}: DQN [31], REINFORCE [53]		CCS of policies in Π_{DS}: Envelope [69], PG-MORL [68], PD-MORL [9], CN [2]	
monotonically increasing scalarization	one policy in Π_{D}: EUPG [47], MOCAC [44], Q-steering [62]	mixture of policies in Π_{DS}: π-mix [58], S-rand [65]	PCS of policies in Π_{D}: PQL [33], PCN [43]	mixture of policies in Π_{DS}: CAPQL [28], π-mix [58], S-rand [65]

Table 1: Non-exhaustive classification of MORL algorithms, following the common utility-based taxonomy from [49, 24]. Here, Π_{D} and Π_{DS} denote the policy space restricted to deterministic and deterministic stationary policies, respectively.

For each combination of criteria, this taxonomy allows us to define a *solution set*, i.e., the type of policies that will constitute the solution to our given problem. In table 1, we categorize a non-exhaustive list of popular MORL methods according to said taxonomy. In this section, we present each class of solution set alongside its corresponding methods.

4.1 Linear scalarization

When the utility function is linear, Roijers et al. [49] show that **deterministic stationary**[4] **policies are optimal**. Furthermore, adding non-stationarity and stochasticity greatly increases the size of the policy space. Thus, MORL methods developped for linear utility functions tend to limit their search to deterministic stationary policies. In scenarios where u is known, only a single optimal policy is required. Conversely, when the utility is unknown or may change, we seek to retrieve a convex coverage set.

Note that by definition, the SER and ESR optimization criteria are equivalent under linear utility, and as such no distinction is made between them in this section.

4.1.1 One deterministic stationary policy

When a linear utility function is used, any single policy MORL problem can be cast into single-objective RL by scalarizing the reward vector. This setting can be solved with most of the existing RL methods (e.g., value-based methods, policy gradients).

For example, take the autonomous driving example discussed in section 2. Let's assume our user is budget-conscious, not in a hurry, and has recurrent back pain. They might then decide on a preference (weight) vector of $[0.1, 0.5, 0.4]$, meaning that they assign an importance factor of 0.1 to speed, 0.5 to cost saving, and 0.4 to comfort. If the car is driving towards a speed bump at step t, it can either brake or accelerate. The brake option yields a reward of $[-0.4, 0.4, 2.1]$ which gets scalarized to $0.1 \cdot -0.1 \cdot 0.4 + 0.5 \cdot 0.4 + 0.4 \cdot 2.1 = 1$. Accelerating gives $[5, -0.2, -1]$, resulting in a scalarized reward of $u([5, -0.2, -1]) = 0$. This indicates that braking is to be favored in this context. When the agent receives a reward vector from the environment, single-objective RL methods like REINFORCE [53] or DQN [31] can scalarize it as such before using the resulting value as an input.

[4] A policy π is stationary if the distribution of actions is constant in all states, i.e., it is not conditional on time step-dependent information.

4.1.2 CCS of deterministic stationary policies

As mentioned in section 3.2, using a linear utility function implies that all optimal policies lie on a convex coverage set. This means that a multi-policy algorithm able to recover a CCS has access to an optimal policy for any possible weight vector $\mathbf{w}$.

Most algorithms use some form of neural network conditioned on a weight vector in their architecture and train it with random values, allowing the model to produce robust outputs over any input $\mathbf{w}$. Conditioned Networks (CN) [2] popularized this approach by showing the potential of conditioned deep Q-networks to generalize across the weight space. Following work kept the same general structure, while focusing on efficient exploration and alignment of weight vectors. Envelope [69] propose to use multiple schemes such as homotopy optimization and Hindsight Experience Replay [7] and show that it allows them to consistently outperform CN. PG-MORL [68] was one of the first methods to tackle environments with large continuous action spaces. It features an evolutionary stage that allows it to efficiently search the space of policies and weights to best improve the CCS. PD-MORL [9] was able to beat Envelope and PG-MORL (on discrete and continuous action tasks respectively) by adding a preference guidance term to a double deep Q-network loss [64]. Note that some of these works use the terms Pareto coverage sets and convex coverage sets interchangeably, but their nature in fact strictly limit them to the retrieval of CCS.

4.2 Monotically increasing scalarization

When the utility function is non-linear, **deterministic stationary policies are not guaranteed to be optimal**. To retrieve policies from the Pareto front that do not lie on convex coverage sets, we need to introduce either non-stationarity or stochasticity.

Note that in this context of non-linear scalarization functions, the ESR and SER optimization criteria are distinct. Although not explicitly mentioned here, each method presented in this section optimizes for one of them.

4.2.1 Deterministic non-stationary policies

When the solution policies must be deterministic and the utility function is non-linear, White shows that non-stationary policies can dominate stationary ones [66]. Consequently, it is necessary to consider non-stationary policies to retrieve a PCS in this context.

Imagine an autonomous delivery company working for two large clients A and B. Its goal is to distribute as many items as possible, while avoiding to neglect either A or B as not to lose an important partnership. An autonomous truck receives a reward of $[1, 0]$ when customer A gets a successful delivery, and $[0, 1]$ for customer B. The utility function to use could then be $u(\mathbf{V}^\pi) = \min(V_A^\pi, V_B^\pi)$, effectively maximizing the total number of deliveries while ensuring no client is left out. Here, a deterministic non-stationary policy would be able to yield a satisfying utility while a stationary one would not. Indeed, instead of always acting the same in each state — which would be equivalent to always picking the same client and thus yielding a utility of 0 — the non-stationary policy could condition on the time-dependent past rewards. This allows the agent to make informed decisions about actions to take depending on whether A or B was most chosen until now.

The first and third cells in the second row of table 1 respectively represent the single and multi-policy (PCS) solution sets for deterministic non-stationary policies. Constructing such policies is often done by conditioning them on the current timestep t (EUPG [47], PCN[5][43]), or by splitting $\mathbf{G}$ (see eq. 1) into past (also known as accrued) and future returns (PQL [33], EUPG [47], MO-CAC [44]). For example, the EUPG algorithm employs a modified policy gradient loss including both accrued rewards and a t-conditioned policy. Q-steering [62] takes another approach, forming non-stationary combinations of deterministic stationary base policies. Q-steering is based on Q-learning, and as such is limited to discrete state and action spaces.

4.2.2 Deterministic stationary mixture policies

As previously mentioned, there are contexts in which having a predictable, deterministic policy is essential. Conversely, other applications can tolerate some degree of stochasticity. For example, when designing a fleet of autonomous cars, we might want to add randomness to the path-finding algorithm, such that not all agents converge to the same road, thus avoiding congested traffic and globally sub-optimal behaviors.

When allowed, stochastic policies should be considered as part of the solution, as they can dominate deterministic policies under non-linear utility function [49]. It was shown that in some cases, we can construct a Pareto front from a mixture (i.e., a stochastic combination) of deterministic stationary policies [58, 65]. This is ideal, as it means that recovering a CCS is sufficient to construct the entire PF, greatly reducing the amount of computation needed to find optimal policies.

For example, Vamplew et al. [58] introduce a new algorithm, which we refer to as π-mix, that randomly selects a deterministic policy at the start of each episode and for its entire duration. Although this method works as expected under SER, using one deterministic policy per episode is not suitable for learning under ESR. Following our autonomous delivery example from section 4.2.1, π-mix could learn to alternate between two policies, each favoring only client A or B. In expectation over multiple episodes, this would indeed result in a fair delivery between them. However, on a per-episode basis, one customer would not be supplied, and thus could end the contract.

The ESR case is more complex, as the choice of policy needs to happen at each state (instead of each episode), being effectively equivalent to a stochastic policy. Wakuta [65] introduces a such method in a simplified setting, which we designate as S-rand, where the probability of picking one of k policy is the same at each state.

However, Lu et al. [28] show that finding the correct weights of a stochastic policy to retrieve a specific value vector is in practice infeasible. They propose CAPQL which uses reward augmentation to recover otherwise unreachable value functions from the Pareto front, although the resulting policies are not stochastic.

4.3 Challenges and way forward

As seen throughout this section, the field of multi-objective reinforcement learning, despite its growing popularity, remains sparse and fragmented. The prevalent utility-based surveys [49, 24] have attempted to propose a unifying view, yet their taxonomy remains quite loose and do not cover some important details. The recent work of Hayes et al. [24] identifies a few understudied areas of MORL that requires further exploration: *complex multi-objective benchmarks*, dedicated *many-objectives methods*, specificities of *multi-agent settings* and the *dynamical identification and evolution of objectives*.

In particular, the study of many-objectives methods seems like an important future research area for MORL. Indeed, most MORL algorithms suffer from the *curse of dimensionality*, i.e., the exponential growth of the search space in the number of objectives makes retrieving satisfying policies highly complex. Note that the lack of MORL benchmarks has been partly addressed since the survey. Notably, the widely-used RL library *Gymnasium* was extended to the multi-objective case with *MO-Gymnasium* [20].

5 MORL and ethics

While it is important to take into account the normative ethics considerations mentioned in section 3.3, deploying MORL agents in society introduces additional concerns. Drawing from the machine ethics literature and considering potential issues caused by the use of naive MORL algorithms in real life scenarios, we identify four desirable features associated with ethical MORL agents.

They should have the ability to: (a) **follow user preferences**, (b) **adapt to an evolving society**, (c) **adhere to a set of norms**, and (d) **account for other agents**. Interestingly, the second and last properties, namely the evolution of objectives and the multi-agent aspect, are part of the list of open challenges for MORL research mentioned in section 4.3. Note that these properties are pointers for researchers wanting to consider the impact of their algorithms, and not an exhaustive list of required attributes to develop agents

[5] Pareto Conditioned Networks can be seen as a sort of deterministic non-stationary policy method, as the agent follows a policy trained using supervised learning that conditions on the "desired horizon".

with ethically-aligned behaviors. These features can even be contradictory in some cases, e.g., when a user's preferences are incompatible with the set of norms the agent ought to follow.

In this section, we discuss the place of each of the aforementioned properties in the MORL literature and highlight potential future work. A summarizing classification of existing methods according to these four principles is presented in table 2.

5.1 The user-centric approach

Etzioni and Etzioni [19] advocate for the *ethics bot*, an AI program that "extracts specific ethical preferences from a user and subsequently applies these preferences to the operations of the user's machine". This resonates with the example discussed in sec. 4.1.1, in which we want the agent to learn the passenger's preferences (e.g., prioritize speed if they are in a hurry or low costs if they want to save up) and adapt its driving profile accordingly.

Methods mentioned in section 4 are capable of producing one policy (or a set of them) that efficiently solves the input problem. However, most of them do not tackle how to find what utility function to use or which policy to pick from the Pareto front. Zintgraf et al. [72] noticed this gap in the literature and made a first step to address it by proposing and evaluating several preference elicitation strategies. Following this work, a number of papers have focused on making the human decision maker a bigger part of the multi-objective RL process.

With GUTS [48], Roijers et al. introduce an interactive approach for multi-armed bandits, where the agent learns simultaneously about the environment and the user's preferences. Contrary to previous methods, GUTS is able to learn non-linear utility functions, while querying the user a provably limited number of times.

MORAL [41] proposes a two-step method for aligning an agent's behavior with the preferences of a user. First, a set of reward functions is learned from expert demonstrations using adversarial inverse reinforcement learning [21]. The user is then faced with multiple queries, allowing the agent to find a preference vector between expert reward functions, while simultaneously optimizing a policy on this combination. Empirically, the authors show that an adversarial user would not be able to teach the agent behaviors actively avoided by the expert demonstrations, although no formal proof is given. DWPI [29] learns the user's preference vector from demonstrations of their behavior in the environment (in a way reminiscent of inverse RL [71]). Chaput et al. [13] argue for a more contextual and intelligible approach, and propose QSOM-MORL, which learns to identify and solve ethical dilemmas using contextual human preferences.

Although not discussed in this work, it is important to consider potential biases in the construction of the utility function when developing single-policy user-centric algorithms. For example, some work (notably in the economics literature) show that there can be a gap between observed and ground truth preferences [10]. As MORL algorithms get better, this discrepancy may become a bottleneck in user satisfaction, further emphasizing the need to take these factors into account.

5.2 Evolving norms and preferences

The methods for learning a user's preferences or utility function introduced in the previous section assume that this target is fixed and not subject to change. However, the owner of a self-driving vehicle, who usually favors comfort and savings over speed, may radically change their preferences in the case of an emergency. Similarly, the vehicle could be part of an autonomous taxi fleet, having to adapt to each customer profile. Therefore, detecting and adapting to changes in the user's inclinations can be desirable properties for autonomous agents to have.

There has been a few MORL methods developed to adapt to a user's evolving preferences. CN [2] and DMCRL [36] take similar approaches, using prior information from learned policies to adapt to changing preferences. Q-steering [62] includes an interactive mode, allowing the user to update the target during of after the learning phase.

Additionally, completely new objectives can be introduced, in which case we want our agent to adapt to them while retaining previously learned knowledge. As society evolves, the three values proposed in our example of section 2 could fail to address emerging considerations such as the environmental impact. Pavaloiu and Koose [40] emphasize that morality is subjective, varies across cultures, and continuously evolves. One naive way to approach this aspect could be to use a linear scalarization function, and take advantages of methods which support non-stationary reward functions (e.g., continual RL methods [26], Q(D)SOM [12]). Hayes et al. [24] identify the challenge of dynamic identification and addition of objectives as one of the main areas for future work in MORL, and to our knowledge the formulation of a variable sized vector-valued reward function has not been studied yet.

5.3 Lawful agents

Approaches for the ethical alignment of agents behavior can be categorized into 3 classes [4]:

- *Bottom-up* approaches do not enforce any obligatory or prohibited actions. Instead, the ethical behavior is learned through experience, and emerges from the definition of the agent and environment.
- *Top-down* approaches are rule-based, and incorporate a priori knowledge (such as deontological duties).
- Some work [54, 17] argue for *hybrid* methods which combine the top-down and bottom-up approaches.

When discussing their ethics bots, Etzioni and Etzioni [19] mention that they only address moral preferences, and disregard normative aspects (e.g., a legal framework). Thus, a MORL-based implementation of an ethics bot would only learn in a bottom-up fashion. Although some work [55] argues that top-down approaches are challenging and pose

MORL methods	user-centered	adaptable	normative	multi-agent
CN [2], DMCRL [36], Q-steering [62]	✓	✓		
MAEE [45]			✓	✓
GUTS [48], MORAL [41], DWPI [29], QSOM-MORL [13]	✓			
EE [46], TLO [61]			✓	
MO-MIX [25], PRBS/D [30], moral rewards [55]				✓

Table 2: Qualification of MORL methods with regards to ethical properties.

some risks, having a set of guarantees (via top-down or hybrid agents) can be crucial in some applications. Typically, we want to ensure that self-driving vehicles deployed on real roads act according to the locally enforced traffic regulations, so that their behavior is safe and predictable for human drivers. In fact, Pagallo [39] argues that values alone are not enough for the coordination of AI agents, and that rules are needed.

In MORL, Rodriguez-Soto et al. [46] take the perspective of the environment designer, allowing them to derive theoretical guarantees for the alignment of agents w.r.t. chosen ethical values. To do so, they start from a MOMDP whose reward functions are built upon a value system. Their proposed Multi-Valued Ethical Embedding (EE) algorithm then proceeds to compute a solution weight vector, resulting in a linearly scalarized MDP with the desired properties.

Using potential-based rewards, TLO [61] focuses on impact-minimizing agents, i.e., agents performing a primary task while aiming at disrupting the environment as little as possible. This approach is bottom-up by design, yet the authors demonstrate strong empirical results showing the ethical alignment of trained agents. These results are for now limited to discrete states and actions, although the algorithms proposed are theoretically extensible to the continuous cases.

For single-objective RL, there are a few works proposing a top-down or hybrid approaches. Shielding [5] uses temporal logic to enforce a set of properties on the resulting policy. AJAR [3] uses argumentation-based judges to compute the rewards based on a set of moral values. Extending such methods to the multi-objective case presents promising possibilities for future research.

5.4 Ethics as a multi-agent problem

Murukannaiah et al. [35] argue that the study of ethics intrinsically needs to be done in a multi-agent context, highlighting that research in AI ethics is to this day largely constituted of single-agent works and ignores the societal context. As a trained MORL agent is deployed in a real-life situation, it is likely to encounter other agents, and more importantly humans. The field of multi-objective multi-agent reinforcement learning (MOMARL) accounts by design for the interactions that can emerge in these cases. Being at the intersection of two sub-fields, MOMARL remains relatively understudied. Rădulescu et al. [42] have surveyed the field of multi-objective multi-agent decision making and concluded that many gaps still exist in the literature, particularly for RL-based methods. Although some MOMARL approaches have been proposed [25, 30], and there has been work on ethics in the multi-agent setting [15], very few MOMARL papers specifically take an ethical perspective. Rodriguez-Soto et al. [45] propose a method (MAEE) to construct environments in which agents are guaranteed to have an ethically-aligned behavior, while pursuing their individual goals. However, the multi-objective reward function they use is very simple, with only two component: an individual objective and an ethical objective (itself split between a normative and evaluative part). QSOM and QDSOM [12] are multi-agent algorithms based on self-organizing maps. Although not multi-objective, they were tested with a variety of reward functions combining ethical stakes, in a way analogous to ESR-optimized MORL. Tennant et al. [55] analyze the behavior of intrinsically-motivated RL agents rewarded according to moral theories when faced with moral dilemmas.

5.5 Benchmarking ethics

While some papers tackle the evaluation of MORL algorithms and the available benchmarks [59], few environments have become standard, and most of them are too simple for modern methods [24].

When trying to ensure the ethical alignment of an AI agent's behavior, the metric of success may be more complex than a simple sum of reward signals. Few MORL environments with an ethics-first approach have been proposed. The ethical gathering game by Rodriguez-Soto et al. [45] extends the regular gathering game, with the addition of beneficence as a moral value. Scheirlinck et al. [50] introduce the ethical smart grid, a complex environment with continuous actions and observations. They propose to use a number of (sometimes conflicting) moral values from the literature to evaluate the behavior of agents.

Additionally, there is a number of environments which are not created with ethics in mind but allow for the inclusion of one or more of the constraints previously mentioned. As such, any MORL environment (e.g., DST [63]) can be viewed through a user-centric lens by changing the setting or adding queries to a user to learn their preferences. Similarly, we can modify multi-agent multi-objective environments (e.g., MOBDP [30]) to shift the focus towards the alignment of agents with some specified ethical values.

6 Conclusion

As artificial intelligence agents are being increasingly deployed in society, there is a growing need to study ways of ensuring the ethical alignment of their behaviors. In this paper, we have focused on multi-objective reinforcement learning, a framework that has been deemed ideal for modeling the complexities of both ethics and real-world problems. First, we proposed a classification of existing multi-objective RL methods according to the prevalent taxonomy. Then, we explored the considerations required when one wishes to work in MORL while adopting an ethics-centered perspective. The literature at the intersection of MORL and ethics is still very limited, and a lot of work remains to be done, notably on methods explicitly implementing one or more of the four desirable properties for ethical agents highlighted in section 5: adherence to user preferences, adaptability to societal changes, compliance with norms and regulations, and considerations of other agents. We hope that this work can serve researchers at the intersection of MORL and ethics to visualize the state of current research and the still lacking areas deserving of further investigations.

Acknowledgements

This work was funded by ANR project ACCELER-AI (ANR-22-CE23-0028-01).

References

[1] David Abel, James MacGlashan, and Michael L Littman. Reinforcement learning as a framework for ethical decision making. In *AAAI Workshop: AI, Ethics, and Society*, 2016.

[2] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *ICML*, 2019.

[3] Benoît Alcaraz, Olivier Boissier, Rémy Chaput, and Christopher Leturc. Ajar: An argumentation-based judging agents framework for ethical reinforcement learning. In *AAMAS*, 2023.

[4] Colin Allen, Iva Smit, and Wendell Wallach. Artificial morality: Top-down, bottom-up, and hybrid approaches. *Ethics and information technology*, 2005.

[5] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, 2018.

[6] Michael Anderson and Susan Leigh Anderson. Machine ethics: Creating an ethical intelligent agent. *AI magazine*, 2007.

[7] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In *NeurIPS*, 2018.

[8] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *ICML*, 2020.

[9] Toygun Basaklar, Suat Gumussoy, and Umit Y. Ogras. PD-MORL: Preference-Driven Multi-Objective Reinforcement Learning Algorithm. In *ICLR*, 2023.

[10] John Beshears, James J Choi, David Laibson, and Brigitte C Madrian. How are preferences revealed? *Journal of public economics*, 92(8-9):1787–1794, 2008.

[11] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on SMC*, 2008.

[12] Rémy Chaput, Olivier Boissier, and Mathieu Guillermin. Adaptive reinforcement learning of multi-agent ethically-aligned behaviours: the QSOM and QDSOM algorithms. *arXiv e-prints*, 2023.

[13] Rémy Chaput, Laetitia Matignon, and Mathieu Guillermin. Learning to identify and settle dilemmas through contextual user preferences. In *ICTAI*, 2023.

[14] Jianguo Chen, Kenli Li, Keqin Li, Philip S Yu, and Zeng Zeng. Dynamic bicycle dispatching of dockless public bicycle-sharing systems using multi-objective reinforcement learning. *ACM TCPS*, 2021.

[15] Nicolas Cointe, Grégory Bonnet, and Olivier Boissier. Ethical Judgment of Agents' Behaviors in Multi-Agent Systems. In *AAMAS*, 2016.

[16] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 2022.

[17] Virginia Dignum. *Responsible artificial intelligence: how to develop and use AI in a responsible way*. 2019.

[18] Muhammad Diyan, Bhagya Nathali Silva, and Kijun Han. A multi-objective approach for optimal energy management in smart home using the reinforcement learning. *Sensors*, 2020.

[19] Amitai Etzioni and Oren Etzioni. Incorporating ethics into artificial intelligence. *The Journal of Ethics*, 2017.

[20] Florian Felten, Lucas Nunes Alegre, Ann Nowe, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and

Bruno Castro da Silva. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[21] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *ICLR*, 2018.

[22] Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: A survey. *Artificial Intelligence Review*, 2022.

[23] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022.

[24] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *AAMAS*, 2022.

[25] Tianmeng Hu, Biao Luo, Chunhua Yang, and Tingwen Huang. MO-MIX: Multi-Objective Multi-Agent Cooperative Decision-Making With Deep Reinforcement Learning. *IEEE PAMI*, 2023.

[26] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *JAIR*, 2022.

[27] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[28] Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-Objective Reinforcement Learning: Convexity, Stationarity and Pareto Optimality. In *ICLR*, 2022.

[29] Junlin Lu, Patrick Mannion, and Karl Mason. Inferring Preferences from Demonstrations in Multi-objective Reinforcement Learning: A Dynamic Weight-based Approach. In *ALA (AAMAS)*, 2023.

[30] Patrick Mannion, Sam Devlin, Jim Duggan, and Enda Howley. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review*, 2018.

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *NIPS*, 2013.

[32] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 2023.

[33] Kristof Van Moffaert and Ann Nowé. Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies. *JMLR*, 2014.

[34] James H Moor. The nature, importance, and difficulty of machine ethics. *IEEE intelligent systems*, 2006.

[35] Pradeep K Murukannaiah, Nirav Ajmeri, Catholijn M Jonker, and Munindar P Singh. New Foundations of Ethical Multiagent Systems. In *AAMAS*, 2020.

[36] Sriraam Natarajan and Prasad Tadepalli. Dynamic preferences in multi-criteria reinforcement learning. In *ICML*, 2005.

[37] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. 2019.

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

[39] Ugo Pagallo et al. Even angels need the rules: Ai, roboethics, and the law. In *ECAI*, 2016.

[40] Alice Pavaloiu and Utku Kose. Ethical artificial intelligence-an open question. *JOMUDE*, 2017.

[41] Markus Peschl, Arkady Zgonnikov, Frans A Oliehoek, and Luciano C Siebert. Moral: Aligning ai with human norms through multi-objective reinforced active learning. In *AAMAS*, 2022.

[42] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. Multi-objective multi-agent decision making: A utility-based analysis and survey. In *AAMAS*, 2020.

[43] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto Conditioned Networks. In *AAMAS*, 2022.

[44] Mathieu Reymond, Conor F. Hayes, Denis Steckelmacher, Diederik M. Roijers, and Ann Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. In *AAMAS*, 2023.

[45] Manel Rodriguez-Soto, Maite Lopez-Sanchez, and Juan A. Rodriguez-Aguilar. Multi-objective reinforcement learning for designing ethical multi-agent environments. *Neural Computing and Applications*, 2023.

[46] Manel Rodriguez-Soto, Roxana Rădulescu, Juan A Rodriguez-Aguilar, and Maite Lopez-Sanchez. Multi-objective reinforcement learning for guaranteeing alignment with multiple values. In *ALA (AAMAS)*, 2023.

[47] Diederik Roijers, Denis Steckelmacher, and Ann Nowe. Multi-objective Reinforcement Learning for the Expected Utility of the Return. In *ALA (AAMAS)*, 2018.

[48] Diederik M. Roijers, Luisa M. Zintgraf, Pieter Libin, and Ann Nowé. Interactive multi-objective reinforcement learning in multi-armed bandits for any utility function. In *ALA (AAMAS)*, 2020.

[49] Diederik Marijn Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A Survey of Multi-Objective Sequential Decision-Making. *JAIR*, 2013.

[50] Clément Scheirlinck, Rémy Chaput, and Salima Hassas. Ethical Smart Grid: A Gym environment for learning ethical behaviours. *JOSS*, 2023.

[51] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 2016.

[52] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[53] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1999.

[54] Elizaveta Tennant, Stephen Hailes, and Mirco Musolesi. Learning machine morality through experience and interaction. 2023.

[55] Elizaveta Tennant, Stephen Hailes, and Mirco Musolesi. Modeling moral choices in social dilemmas with multi-agent reinforcement learning. In *IJCAI*, 2023.

[56] Mark Timmons. *Moral theory: An introduction*. Rowman & Littlefield publishers, 2012.

[57] Suzanne Tolmeijer, Markus Kneer, Cristina Sarasua, Markus Christen, and Abraham Bernstein. Implementations in Machine Ethics: A Survey. *ACM Computing Surveys*, 2021.

[58] Peter Vamplew, Richard Dazeley, Ewan Barker, and Andrei Kelarev. Constructing Stochastic Mixture Policies for Episodic Multiobjective Reinforcement Learning Tasks. In *Advances in Artificial Intelligence*. 2009.

[59] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 2011.

[60] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology*, 2018.

[61] Peter Vamplew, Cameron Foale, Richard Dazeley, and Adam Bignold. Potential-based multiobjective reinforcement learning approaches to low-impact agents for ai safety. *Engineering Applications of Artificial Intelligence*, 2021.

[62] Peter Vamplew, Rustam Issabekov, Richard Dazeley, Cameron Foale, Adam Berry, Tim Moore, and Douglas Creighton. Steering approaches to pareto-optimal multiobjective reinforcement learning. *Neurocomputing*, 2017.

[63] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In *Advances in Artificial Intelligence*, 2008.

[64] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

[65] Kazuyoshi Wakuta. A note on the structure of value spaces in vector-valued Markov decision processes. *Mathematical Methods of Operations Research*, 1999.

[66] D. J White. Multi-objective infinite-horizon discounted Markov decision processes. *Journal of Mathematical Analysis and Applications*, 1982.

[67] Jess Whittlestone, Kai Arulkumaran, and Matthew Crosby. The societal implications of deep reinforcement learning. *JAIR*, 2021.

[68] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. *ICML*, 2020.

[69] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. In *NeurIPS*, 2019.

[70] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM CSUR*, 2021.

[71] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

[72] Luisa M Zintgraf, Diederik M Roijers, Sjoerd Linders, Catholijn M Jonker, and Ann Nowé. Ordered preference elicitation strategies for supporting multi-objective decision making. *AAMAS*, 2018.

Apprentissage par Renforcement Profond pour la Défense Aérienne

Valentin Colliard[1,2], Alain Pérès[1], Vincent Corruble[2]

[1] Thales LAS France, Rungis, France
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Résumé

Dans cet article, nous proposons une nouvelle approche à base d'apprentissage par renforcement pour traiter le problème de la défense aérienne. TAADA (Transferable Agent for Air Defense Application) est une solution pour réaliser à la fois l'évaluation de la menace et l'affectation des armes dans le contexte de la défense antiaérienne. En utilisant Starcraft II comme environnement de simulation, nous mettons en place des scénarios d'attaque où l'agent doit apprendre à défendre ses effecteurs et ses points d'intérêt. En estimant une valeur pour chaque paire effecteur-menace, l'agent montre une capacité à être robuste dans plusieurs scénarios pour distinguer et prioriser les menaces afin d'accomplir sa mission.

Mots-clés

Apprentissage par renforcement profond, Simulation, Décision, Défense aérienne, TEWA

Abstract

In this paper, we propose a new approach based on reinforcement learning in an air defense context. TAADA (Transferable Agent for Air Defense Application) is a solution for performing both threat assessment and weapon assignment in the context of air defense. By using Starcraft II as a game environment to create attack scenarios where the agent must learn to defend its effectors and points of interest. By estimating a value for each effector-threat pair, our agent shows an ability to be robust in multiple scenarios to truly distinguish and prioritize threats in order to achieve the mission.

Keywords

Deep Reinforcement Learning, TEWA, Wargames, Decision, Air defense

1 Introduction

Au fil des années, l'apprentissage par renforcement (RL) a démontré une réelle capacité à apprendre dans des environnements complexes. Pendant une longue période, l'apprentissage par renforcement est resté principalement confiné à des jeux simples, avec des espaces d'action et d'observation relativement petits. Les jeux de plateau s'y prêtent particulièrement bien. L'environnement a des règles et des objectifs explicites. Il est ainsi plus simple d'établir une fonction de récompense et de s'entraîner avec un agent apprenant. Alimentés par la puissance de calcul croissante et les progrès de l'apprentissage profond, les algorithmes de RL sont aujourd'hui capables d'accomplir des tâches extrêmement complexes, comme en témoignent les récents résultats obtenus avec le jeu Starcraft II [26]. Mais le RL n'est plus confiné à ce domaine, et s'exporte dans de nombreuses applications du monde réel telles que la robotique, la finance et dans le domaine de la santé par exemple, avec des résultats prometteurs[27]. Un nouveau domaine dans lequel le RL peut s'avérer utile est celui de la défense aérienne. Avec les systèmes de défense au sol, l'objectif est de protéger des points ou des sites d'intérêt contre les attaques aériennes. L'émergence de l'utilisation de drones et de nouvelles formes d'attaques et de menaces pose des défis inédits. Les futurs outils d'aide à la décision qui aideront les opérateurs à prendre les bonnes décisions dans des situations critiques devront être très robustes aux différentes situations. Il faut donc un agent capable d'analyser la situation et de déterminer le niveau de menace de chaque unité ennemie. Ensuite, il doit suggérer à un décideur humain la meilleure action défensive possible. Ce processus s'appelle la TEWA (Threat Evaluation & Weapon Assignment). Elle est généralement divisée en deux sous-problèmes : le problème de l'évaluation de la menace (TE pour *Threat Evaluation*) et le problème de l'allocation des armes (WTA pour *Weapon-Target Assignment*). Le RL constitue une approche convaincante pour résoudre ces problèmes. Sa capacité d'adaptation et d'apprentissage dans des environnements complexes et dynamiques en fait un outil particulièrement adapté aux scénarios caractérisés par l'évolution des menaces et des conditions incertaines. En déployant des agents RL pour analyser la situation et réaliser la TEWA, les applications de défense aérienne peuvent améliorer leur efficacité et leur réactivité, protégeant ainsi plus efficacement les actifs défensifs contre les menaces émergentes. Dans cet article, après avoir réalisé un état de l'art des méthodes utilisées pour résoudre la TEWA et des techniques d'apprentissage par renforcement. Nous allons démontrer que cette approche atteint dans un premier temps les performances de script à base de règles. En outre, nous illustrerons comment cette approche surpasse ces méthodes dans des scénarios plus complexes. Les deux principales contributions de cet article résident dans l'architecture du réseau, qui permet un transfert de connaissances d'un scénario à l'autre et de rendre l'agent insensible aux variations de taille des données d'entrée dû à un nombre de menaces variable.

2 Etat de l'art

2.1 Évaluation de la menace & Assignation des armes

Tout d'abord, le problème de la TE vise à déterminer le niveau de menace de chaque entité présente sur le terrain. Une *menace* est une entité qui est perçue comme ayant l'intention d'infliger des dommages, de blesser ou d'endommager des points d'intérêt pour le défenseur. Il existe une grande variété de critères, dont la plupart sont sujets à interprétation, même si ces critères sont objectifs, comme la vitesse ou l'altitude, par exemple. De plus, dans le domaine de la défense, la plupart des recherches ne sont pas publiées. L'un des moyens les plus efficaces pour savoir si l'évaluation de la menace était de qualité est de réaliser la WTA et ainsi de voir les conséquences de la TE. Une bonne évaluation de la menace engendre une bonne allocation des armes. La plupart du temps, TE et WTA sont donc réalisées ensemble. Il existe 4 familles de critères permettant d'évaluer le niveau de menace. Tout d'abord, la *capacité* d'une menace à infliger des dommages. Par exemple, le type de menace est crucial. Le critère de *proximité* est parfois inclus dans le critère précédent. Comme son nom l'indique, il se base sur des informations géographiques. Par exemple, la distance entre la cible et le POI (Point Of Interest) que l'on souhaite protéger. Ensuite, le critère *d'intentionnalité* est utilisé pour déterminer si la cible a l'intention ou non d'attaquer et d'infliger des dommages. Ce critère est particulièrement compliqué à interpréter, d'autant plus que la cible n'a aucun intérêt à se montrer volontairement agressive. Pour le déterminer, il est possible de se baser, par exemple, sur les données cinématiques. L'étude du comportement de menace est donc essentielle, mais évidemment non triviale à traiter. Enfin, le dernier critère utilisé est *l'opportunité*, qui vise à déterminer la fenêtre temporelle dans laquelle la menace pourrait causer des dommages. Les solutions de base au problème de la TE sont des algorithmes basés sur des règles, avec l'aide d'experts humains. Ces algorithmes peuvent être performants, mais en raison de leur manque d'adaptabilité, de la complexité de l'ingénierie et des dépendances humaines, ils sont beaucoup trop limités et ne fonctionnent que sur des modélisations simples ou de petite taille [6]. Une fois les règles établies, des moteurs d'inférence sont utilisés pour obtenir une solution. Dans [11], Johansson utilise les réseaux bayésiens dans le processus d'évaluation des menaces. Il compare ensuite le réseau bayésien avec la logique floue, qui présentent chacun des avantages et des inconvénients [12]. Il existe également d'autres méthodes d'Intélligence Artificielle (IA) pour résoudre le problème de la TE. Paradis a utilisé FAN (Functional Abstraction Network) [19]. En 1999, Dong et Quing ont utilisé des réseaux de neurones (NN) pour résoudre le problème TE [3] et Azak et Bayrak ont étendu l'utilisation des NN au problème TEWA [2]. L'une des meilleures façons d'évaluer la qualité de l'évaluation de la menace est de réaliser la WTA, c'est pourquoi la plupart des articles traitent des deux en même temps.

Passons maintenant à la deuxième partie du problème TEWA, le problème de l'affectation des cibles aux armes. La recherche dans ce domaine a commencé il y a plusieurs décennies, dans les années 1950 [16], avec l'apparition des missiles balistiques pendant la guerre froide. Elle n'a jamais été abandonnée depuis. La recherche est beaucoup plus riche et plus avancée que celle sur l'évaluation des menaces. Le problème de la WTA consiste à associer des moyens défensifs à des cibles menaçant des systèmes défensifs, des zones à défendre ou des points d'intérêt. Ce problème est NP Complet [15]. Il existe deux versions différentes de ce problème, une version statique [7] et une version dynamique [8]. La principale différence est que la version dynamique prend en compte la notion de temps. Cela signifie qu'il ne s'agit pas d'une allocation unique. Les menaces sont engagées et font évoluer la situation pour réaliser à nouveau la WTA. Cette version est plus complexe que la simple affectation de tous les moyens défensifs à toutes les menaces en une seule fois. Il existe également deux approches différentes de la WTA. En cherchant à minimiser la valeur totale des menaces, cela conduit à se concentrer sur la destruction des menaces. Cette approche est appelée *target-based*. La seconde méthode vise à maximiser la capacité de survie des sites à défendre, appelée *asset-based*. Les deux méthodes sont étroitement liées en principe, mais présentent des différences dans la modélisation mathématique des problèmes [7, 8]. Étant NP-complet, les méthodes exactes deviennent rapidement inefficaces, mais restent très utiles sur une petite instance du problème. Lorsque ce n'est pas possible, des heuristiques ou d'autres méthodes approximatives sont souvent utilisées. Johansson a publié un état de l'art détaillé de la TE et de la WTA [10], Huaiping et al. se concentrent particulièrement sur la DTWA (Dynamic WTA) [9]. Naseem et al. ont considéré le problème TEWA comme un algorithme de mariage stable tridimensionnel [18].

2.2 Apprentissage par renforcement

La principale limite des méthodes mentionnées auparavant est leur manque d'adaptabilité. La plupart des modèles utilisés sont fondées sur l'expertise humaine, ce qui entraîne une subjectivité dans la création et l'interprétation des règles. La modification de ces modèles est un véritable défi. Avec l'émergence de nouvelles menaces et l'augmentation du nombre et des types de données à gérer, une nouvelle méthode semble nécessaire pour réaliser la TEWA. L'apprentissage par renforcement (RL) est un sous-domaine de l'apprentissage automatique qui consiste à placer un agent dans un environnement et à le laisser agir, dans le but de maximiser une récompense. L'agent apprend une politique par essai-erreur : il sélectionne et exécute une action, et l'environnement lui renvoie une récompense. Au fil du temps, l'agent apprend à sélectionner les actions qui lui procurent la meilleure récompense cumulative [24]. L'apprentissage par renforcement profond (DRL) est particulièrement utile lorsque l'on ne peut pas stocker toutes les informations relatives à l'expérience de l'agent, ou estimer correctement une fonction de récompense. Le DRL

combine l'apprentissage par renforcement et l'apprentissage profond. Il permet aux agents d'apprendre des correspondances complexes entre les observations et les actions, en particulier avec des espaces d'entrée à haute dimension. Le DRL a permis de réaliser des progrès considérables dans les domaines de la vision par ordinateur, des jeux complexes, des tâches de contrôle robotique et des systèmes de prise de décision autonomes [1]. Les jeux vidéo sont des environnements utiles pour l'évaluation des algorithmes DRL. L'une des plus grandes avancées dans le domaine du DRL et des jeux a été le DQN (Deep Q-Network) utilisé pour jouer à plusieurs jeux Atari [17]. Au cours des années suivantes, le DRL a été utilisé pour jouer à des jeux de plus en plus complexes. Les dames, les échecs et le go sont des jeux célèbres qui ont été étudiés depuis le début de l'IA [21, 23, 22]. Ces articles ont montré le pouvoir du DRL à surpasser les performances humaines. Après le jeu de Go, DeepMind a décidé de s'attaquer à un jeu beaucoup plus complexe : Starcraft II (SC2). Starcraft II est un jeu de stratégie en temps réel avec un espace d'action estimé à 10^{26} actions possibles. Ils ont combiné diverses méthodes de ML et DRL pour battre les joueurs professionnels[25]. Ils ont également fourni un framework open source nommé *PySC2* pour utiliser SC2 comme environnement RL [26]. C'est celui que a été utilisé comme simulateur pour réaliser les expériences. SC2 a déjà été utilisé dans un cadre militaire pour recréer une véritable opération militaire ayant eu lieu dans le passé et utiliser le DRL pour la réaliser [5]. L'utilisation du DRL dans le cas spécifique de TEWA est récente, en illustre divers articles de Qiang Fu et al. [4] et Jiayi Liu et al.[14], utilisant un simulateur spécifique conçu pour créer et jouer des scénarios de défense aérienne, ils ont utilisé le DRL pour agir en tant que décideur et réaliser la TEWA. Ces travaux présentent certaines limites importantes. D'une part, ils manquent d'informations détaillées sur l'environnement utilisé, ce qui réduit la reproductibilité des résultats. De plus, les entraînements sont effectués sur un seul scénario en termes de nombre d'unités, ce qui limite la généralisation.

3 Approche proposée

3.1 Algorithme et représentation

Comme indiqué précédemment, le DRL peut gérer certaines limitations des approches classiques utilisées pour résoudre le problème TEWA, en particulier leur manque d'adaptabilité et leur complexité de conception. Les méthodes acteur-critique permettent d'utiliser simultanément un acteur qui prend des décisions et une critique qui évalue ces décisions. Cela a plus de sens dans des environnements complexes comme celui de la défense aérienne, offrant une plus grande stabilité et fonctionnant très bien avec les réseaux neuronaux profonds. Dans cette classe d'algorithme, PPO (Proximal Policy Optimization) [20] semble être la meilleure option. L'idée derrière PPO est de limiter l'amplitude de la mise à jour de la politique afin d'éviter des changements trop drastiques. Cela conduit à une meilleure stabilité pendant l'entraînement et augmente les performances. Nous décidons d'utiliser une méthode acteur-critique pour traiter le problème TEWA, et plus spécifiquement PPO.

Algorithm 1 Fonction de clippage pour PPO

Données : $\pi_{\theta_{old}}$ (ancienne politique), π_θ (nouvelle politique), A (avantage), ϵ (hyperparamètre)
Sortie : Mise à jour de la politique
for chaque pas d'optimisation **do**
  for chaque échantillon s, a dans le batch **do**
    $r(\theta) \leftarrow \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$
    $L^{CPI}(\theta) \leftarrow r(\theta) \cdot A(s,a)$
    $L^{clip}(\theta) \leftarrow \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon) \cdot A(s,a)$
    $L(\theta) \leftarrow \min(L^{CPI}(\theta), L^{clip}(\theta))$
  end for
  Mettre à jour θ en maximisant $L(\theta)$
end for

L'algorithme 1 montre le fonctionnement du clippage dans PPO. Ce processus garantit que la politique actuelle ne s'éloigne pas trop de la politique précédente. Cela risque de ralentir l'apprentissage, mais le rend plus stable. À présent, il faut définir l'espace d'observation et d'action.
Chaque effecteur ou menace est considéré comme une unité dans un ensemble qui partagent les mêmes attributs de description.

- Type de plateforme
- Affiliation (Allié ou Ennemi)
- Point de vie
- Position en coordonnées polaires (Par rapport au centre de la carte)
- Orientation (Cosinus et Sinus)
- Prêt à tirer ?

En plus de ces informations, trois tableaux sont fournis, montrant la vitesse radiale, la vitesse angulaire et la distance de chaque menace par rapport à chaque effecteur. Et enfin, un masque, indiquant sur quelle menace chaque tourelle a la possibilité à tirer.
De manière plus formelle, le problème peut être représenté de manière simple avec les trois entités présentes sur le terrain.

- M menaces
- T effecteurs
- P points d'intérêt

L'espace d'action est donné par la combinaison de menaces et d'effecteurs, c'est-à-dire $M \times T$.
Le reward, ou récompense, peut être calculé comme suit :

$$\text{Récompense} = (-M + P) + T$$

La récompense se base donc sur la destruction des unités. L'agent est pénalisé s'il perd un effecteur ou un point d'intérêt mais est récompensé en détruisant une unité ennemie. Elle est simpliste mais permet une bonne représentation du problème et des objectifs. Des pondérations sur les unités, afin de tenir compte de leur importance relative n'est pas utilisée pour le moment. Beaucoup d'autres informations peuvent être intégrées à cette récompense, comme par exemple des malus lors d'un tir dans l'optique d'économiser des munitions. En considérant une approche classique,

deux problèmes principaux sont rencontrés. Premièrement, changer le nombre de menaces change le problème. L'agent peut fonctionner dans un scénario unique, et la diminution du nombre de menaces peut entraîner une perte de performances alors que le problème est plus simple. Ensuite, dans le monde réel, seules les capacités défensives sont connues, et il n'est pas possible de connaître à l'avance les capacités de l'ennemi. Il ne faut donc pas baser l'approche proposée sur une architecture dépendante de la taille. Au-delà du problème de la taille des entrées, un problème latent du RL émerge : le problème de la généralisation. Lors de l'entraînement d'un agent dans un environnement, il n'est pas rare qu'une simple modification de ce dernier, qui peut paraître insignifiante pour un être humain, se traduise par une forte baisse de performance. Ces problèmes peuvent être dus à des problèmes de représentation, ce qui signifie que l'agent apprend à résoudre une instance spécifique plutôt que le problème dans sa généralité [13]. Une solution à ce problème est l'apprentissage par transfert (Transfer Learning). L'apprentissage par transfert est une technique d'apprentissage automatique utilisée pour permettre aux connaissances acquises lors de l'apprentissage sur une tâche source d'être transférée vers une tâche cible similaire ou différente, améliorant ainsi les performances des modèles d'apprentissage automatique, en particulier dans les cas où les données sont rares ou coûteuses à collecter [28]. Avoir un agent performant qui ne peut pas s'adapter aux différentes situations n'est pas une option dans cette situation étant donné qu'il n'est pas possible de savoir clairement ce que prépare l'ennemi, ni quelles sont ses capacités et ses intentions. L'objectif est de résoudre la TEWA de manière générale. En s'inspirant du concept d'apprentissage par transfert, le but est de développer un algorithme capable de bien fonctionner sur différents scénarios sans avoir besoin de refaire un apprentissage. En acquérant des connaissances générales sur le domaine du problème plutôt que sur des cas spécifiques, l'objectif est de créer une solution polyvalente capable de s'adapter et de fonctionner efficacement dans divers scénarios. Dans ce cas, une représentation basée sur des paires effecteur-cible est proposée. L'idée est d'estimer l'intérêt de choisir un effecteur pour tirer sur une menace à ce moment précis. Considérant les attributs de chaque unité et d'autres informations récupérées, qui sont traitées comme un contexte global. Cette valeur peut être comparée à une *Valeur de menace*. Ce processus est appliqué à chaque couple possible. Au final, n valeurs sont obtenues pour les n couples et, en appliquant une fonction softmax, l'agent décide quel effecteur utiliser sur quelle cible. Ensuite, l'agent apprend, à partir d'informations pour estimer l'utilité de cette action, que celle-ci ne dépend pas du nombre d'unités, il étudie tous les couples de la même manière. En effectuant cela, le problème de la taille d'entrée des observations est ainsi résolu et le modèle se concentre simplement à résoudre la TEWA. Cela ne dépend d'aucune structure ou environnement particulier. Disposant de données riches et abondantes, il pourra, quelle que soit la situation, choisir la bonne action sans même effectuer d'apprentissage par transfert. Concrètement, le réseau dispose de deux extracteurs d'informations, un pour les menaces et l'autre pour les effecteurs. Ils sont utilisés pour augmenter les dimensions et extraire des informations sur l'unité. Les sorties de ces derniers sont ensuite concaténées avec un vecteur de contexte. Ce vecteur créé résume les connaissances sur l'effecteur et la menace, en tenant compte d'autres informations plus générales, telles que les vitesses radiales et angulaires de la menace par rapport à l'effecteur. Il passe par un réseau de neurones qui produit une valeur d'utilité.

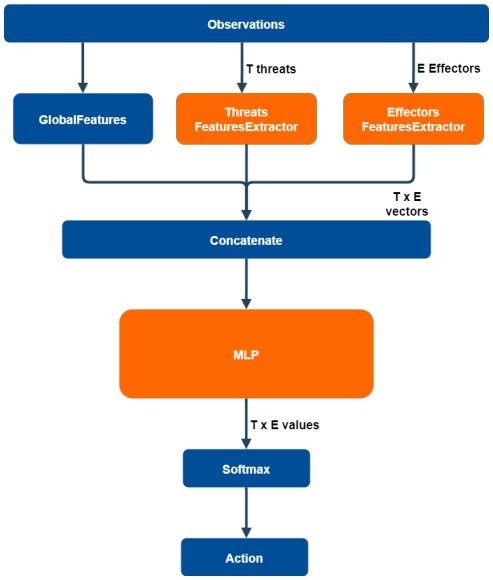


FIGURE 1 – Architecture de TAADA

La figure 1 illustre l'architecture décrite ci-dessus. L'acteur et la critique sont deux réseaux distincts. La critique repose sur la même architecture que l'acteur, à la différence qu'à la place d'un softmax permettant de choisir l'action, une couche supplémentaire est ajoutée. En prenant la valeur minimale, maximale et la moyenne de toutes les valeurs en sortie du MLP, cette couche permet d'obtenir une valeur finale pour la critique. L'acteur et le critique ne partagent pas le même réseau, ce qui permet au réseau de l'acteur de se concentrer sur l'apprentissage de la politique et au réseau du critique de se focaliser sur l'évaluation précise des actions. Cette séparation améliore ainsi la stabilité et l'efficacité de l'apprentissage, bien qu'elle augmente la complexité computationnelle. Une version avec un réseau partagé a été testée, mais elle n'offrait pas le même niveau de performance. Les entraînements et les tests sont réalisés avec un processeur i7-12700H, une carte graphique RTX 3070 et 32 Go de RAM.

3.2 Environnement

Tout d'abord, nous avons besoin d'un simulateur pour créer des scénarios, car il est presque impossible d'obtenir des données réelles en raison de la nature des données. Ces données auraient été trop complexes à obtenir et à traiter, nous avons donc recherché un environnement répondant à certains critères. Le simulateur avait besoin de fonctionnalités spécifiques. Premièrement, nous avons besoin d'un en-

vironnement bac à sable, dans lequel il est possible de créer des scénarios à partir de zéro, afin de pouvoir contrôler la crédibilité, le réalisme et la difficulté des scénarios créés. Il faut également pouvoir exécuter ces scénarios très rapidement, pour accélérer l'apprentissage, idéalement avec plusieurs environnements fonctionnant en parallèle. Le simulateur doit également être capable de générer de nombreuses variantes d'un scénario donné. Idéalement, le simulateur devrait offrir un bon équilibre entre le réalisme des scénarios d'un point de vue militaire et leur complexité, connaissant tous les paramètres pouvant être pris en compte dans la problématique. Nous avons besoin d'un large éventail d'unités, depuis les unités terrestres représentant les systèmes de défense jusqu'aux unités aériennes jouant le rôle de menaces. La diversité donne lieu à des scénarios plus variés, rendant l'apprentissage plus compliqué, mais plus robuste. Un critère clé dans le choix du simulateur était sa capacité à intégrer un algorithme Python pour agir à la place de l'intelligence artificielle du jeu. La plupart des simulateurs de jeux de guerre militaires ont été éliminés sur ce critère. De ce fait, une multitude de simulateurs ont été étudiés, et nous avons décidé d'utiliser Starcraft 2 [26].

Avant tout, l'environnement de jeu python de Starcraft II fournit une interface (PySC2) permettant aux agents RL d'interagir avec le jeu, de recevoir des observations et d'envoyer des actions. Bien que totalement hors du champ de la défense aérienne, cet environnement spécifiquement conçu pour RL offre une modularité intéressante, que ce soit pour créer des cartes personnalisées ou modifier les attributs des unités. Le but de cet environnement est de créer une représentation simplifiée d'une situation de défense aérienne. Le défenseur dispose d'un ou plusieurs effecteurs défensifs, et de plusieurs bâtiments à protéger. Du côté des attaques, plusieurs ennemis volants sont chargés de détruire les défenses et les bâtiments. Les nombres et les types peuvent varier.

4 Résultats et comparaisons

4.1 Détails sur les scénarios

Premièrement, SC2 n'est pas conçu pour des scénarios de défense aérienne réalistes. En effet, plusieurs aspects du jeu ont dû être modifiés pour créer les scénarios. Les *Terran* sont une des espèces disponibles dans le jeu, elle représente des unités terriennes futuristes. C'est donc cette espèce qui a été choisie pour rendre l'apparence des scénarios plus proche d'une modélisation réelle. Plusieurs unités aériennes et terrestres ont été adaptées pour composer le scénario. Voici un tableau montrant certains attributs important de chaque unité.

Type	Equivalent	Santé	Dégats
Viking	Avion	2	3
Banshee	Drone	1	1
Phoenix	Leurre	1	$\emptyset$
SiegeTank	Effecteur	3	1

TABLE 1 – Attributs des unités

Les *Viking*, *Banshee*, et *Phoenix* sont des unités ennemies aériennes, tandis que les *SiegeTank* composent la défense au sol. Pour les premiers tests, nous créons une carte générique avec n effecteurs au centre de la carte, et m menaces apparaissant autour d'eux et les attaquent. Aucun radar ou POI ne sont ajoutés, l'objectif principal de cette carte est de vérifier si l'agent peut apprendre une stratégie défensive. La défense fait face à deux types de menaces, les Vikings et les Banshees. Les Vikings sont beaucoup plus dangereux que les Banshees et l'objectif est de voir comment l'agent gère cette différence.

FIGURE 2 – Vu depuis le jeu Starcraft II

Sur la figure 2, en rouge sont représentées les menaces (Viking et Banshee), en bleu les effecteurs et en vert les unités ennemies n'infligeant pas de dommages. Sur la minicarte, en bas à droite, sont visibles les autres unités qui arrivent. Sur base de ce scénario, différentes compositions ou nombres d'unités ont été testées, de 3 à 22 menaces.

4.2 Résultats sur les cartes simples

Dans ce scénario, TAADA doit apprendre à se défendre, en apprenant une stratégie défensive. La défense est composée de n effecteurs, et elle fait face à m menaces. Les menaces sont les Vikings et les Banshees. Comme les Vikings sont plus rapides et plus forts que les Banshees, l'une des meilleures stratégies pour gagner la partie serait de se concentrer d'abord sur les Vikings, qui peuvent détruire les effecteurs en un seul coup. Ensuite, il faut se concentrer sur les banshees. Les entraînements et tests ont été réalisés sur différentes compositions d'unités, mais le déroulement du scénario n'a pas changé. Tous les effecteurs sont situés au centre de la carte, et les menaces apparaissent de manière aléatoire autour d'eux.

Carte	# Effecteurs	# Banshees	# Vikings
Small	2	2	1
Large	4	4	3

TABLE 2 – Compositions d'unités des cartes

La table 2 résume les différentes compositions sur les cartes testées.
En ce qui concerne les paramètres d'apprentissage utilisés avec l'algorithme PPO, le learning rate est de : α= 0.0001, avec 20 minibatchs de taille=90, 20 mises à jour par époque

et pendant 100.000 pas. À partir des valeurs de référence les plus couramment utilisées avec PPO, de nombreux tests ont été effectués sur ces paramètres afin d'obtenir les valeurs optimales maximisant les performances.
Un simple script adapté à ce problème a été implémenté pour servir de référence. Les vikings sont l'unité la plus dangereuse, et ils arrivent le plus souvent en premier dans le champ de tir des effecteurs (car plus rapide que les banshees). Le script sélectionne aléatoirement un effecteur et tire sur la menace la plus proche de ce dernier. Un agent aléatoire est également utilisé comme référence supplémentaire.

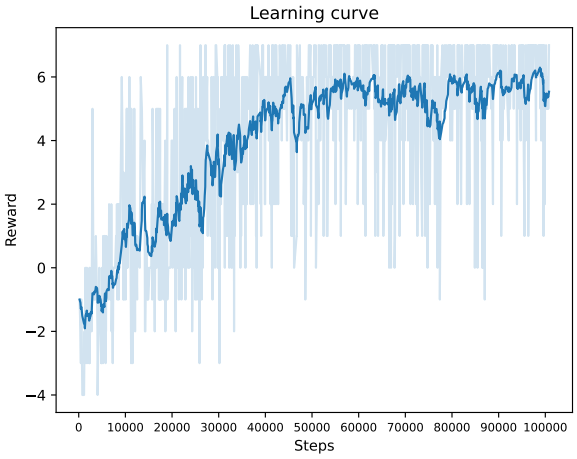


FIGURE 3 – Courbe d'apprentissage de TAADA sur la carte Large

La figure 3 montre l'évolution de la récompense au fil du temps sur la carte Large. La récompense est comprise entre -4 et +7. L'agent démarre autour de -1 et atteint rapidement une récompense supérieure à 6 après 50K itérations. Il se stabilise ensuite autour de cette valeur. Également, l'agent est testé sur les autres cartes afin d'observer son comportement. En plus de l'agent scripté et de l'agent aléatoire, une comparaison est faite avec un agent entraîné sur la carte.

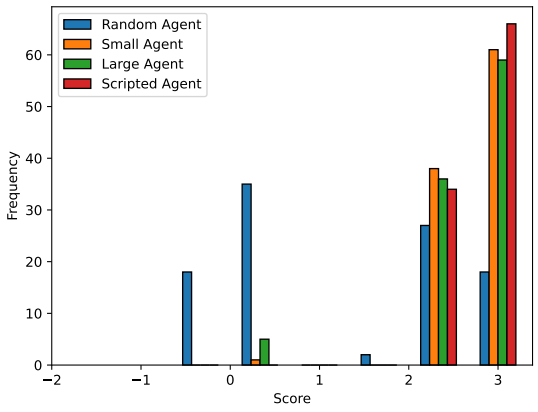


FIGURE 4 – Performance des agents sur la carte Small

Agent	Moy	σ	Var	%Win
Random	0.92	1.43	2.05	45
Small	2.59	0.55	0.30	99
Large	2.49	0.74	0.55	95
Script	2.66	0.47	0.22	100

TABLE 3 – Resultat sur la carte Small

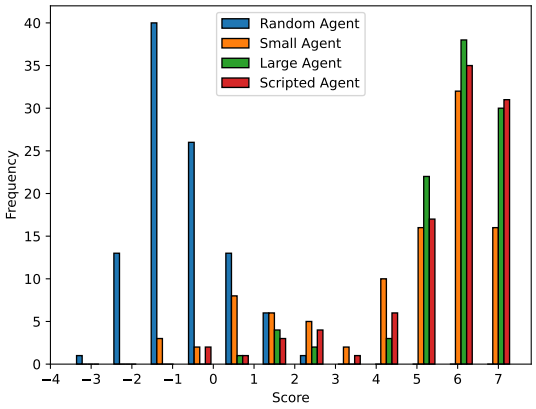


FIGURE 5 – Performance des agents sur la carte Large

Agent	Moy	σ	Var	%Win
Random	-1.41	1.14	1.30	0
Small	4.38	2.52	6.36	74
Large	5.68	1.49	2.22	93
Script	5.48	1.81	3.27	89

TABLE 4 – Resultats sur la carte Large

Sur les figures 4 et 5, sont présentés les résultats des tests sur les cartes Small et Large. D'une part, sur le premier histogramme (figure 4), l'agent entraîné sur la carte Large, le Large_Agent, est au même niveau de performance que l'agent scripté et le Small_Agent, entraîné sur la carte Small. D'autre part, dans le second histogramme (figure 5), le Small_Agent est moins performant que l'agent scripté ou l'agent Large, mais reste bien meilleur que l'agent aléatoire. En considérant que le problème est beaucoup plus complexe dans ce sens et que l'agent ne s'est pas du tout entraîné sur cette carte, la performance est encourageante.

4.3 Ajout d'un troisième type de menace

Pour rendre les scénarios plus complexes, la décision a été prise d'ajouter un troisième type de menace qui sera perçu comme des "leurres". L'idée était de surcharger l'espace aérien avec des unités qui n'infligent en réalité aucun dégât. Les leurres sont considérés comme des menaces même s'ils n'infligent pas de dégâts. Cela se justifie par le fait qu'ils appartiennent au camp ennemi et que, n'ayant aucune connaissance à priori d'eux, ils sont potentiellement dangereux par leur seule présence. Dans ce cas, le besoin d'une bonne évaluation de la situation tactique et ainsi se concentrer d'abord sur les véritables menaces, et ensuite,

gérer les leurres. En se basant sur la carte Large, 3 leurres ont été ajoutés. Leur destruction entraîne une récompense de +1, mais l'agent doit comprendre qu'en s'en occupant tôt, il subira une perte plus importante par la suite.

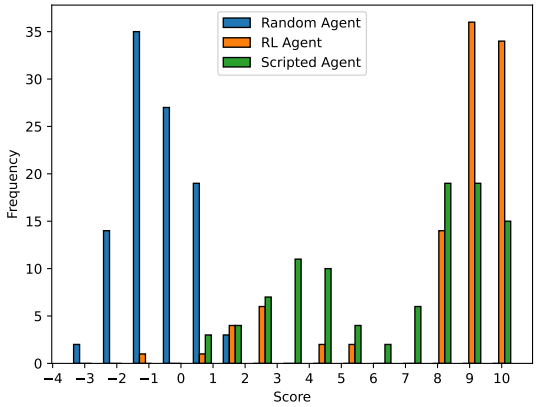


FIGURE 6 – Performance des agents sur la carte Large avec leurres

Agent	Moy	σ	Var	%Win
Random	-1.44	1.10	1.20	0
RL	8.08	2.82	7.97	84
Script	6.38	3.07	9.42	59

TABLE 5 – Resultats sur la carte Large avec leurres

La figure 6 montre la comparaison entre l'agent aléatoire, l'agent scripté et TAADA (RL Agent). Ce scénario montre la supériorité de l'agent sur l'agent scripté. Dans les résultats précédents, les deux agents avaient des résultats assez similaires, mais dans ce cas, il y a beaucoup plus de scores 9 et 10 de la part de TAADA. Il est évident que la stratégie de tir au plus proche n'est pas la plus appropriée, mais il est intéressant de noter que TAADA adapte sa stratégie à la situation.

4.4 Résultats sur scénarios réalistes

Afin de valider ces premiers résultats intéressants, la prochaine phase est de complexifier les scénarios en les rendant réalistes. La première étape a été d'ajouter une probabilité de réussite des tirs, la *Pkill*. En effet, en application réelle, lorsqu'il est décidé d'utiliser un effecteur, une réflexion se fait pour déterminer le meilleur moment pour engager une cible afin de maximiser la *Pkill*. Dans un premier temps, cette valeur sera propre à chaque type d'effecteur. Cette valeur est donc fixée en amont, mais est destinée à devenir dynamique. En association avec la *Pkill*, est également instaurée la notion de stock. Les munitions ne sont plus infinies, et chaque tir n'est pas forcément réussi. Cela donne une nouvelle dimension à la stratégie, puisqu'il faut prendre en compte de nouveaux paramètres avant de choisir la bonne action. En plus de ces ajouts, un nouveau type d'effecteur est ajouté, avec des caractéristiques très différentes de l'unique effecteur utilisé jusqu'à présent. Des paramètres tels que la portée, la cadence de tir et les dégâts des unités ont également été modifiés.

Unité	Dégâts	Vie	Portée	Pkill	Stock
Viking	1	5	2-15	1	∞
Banshee	1	1	1	1	∞
Canon	1	1	2-40	0.4	20
SiegeTank	5	1	40-80	0.8	5

TABLE 6 – Attributs des unités

La table 6 décrit certains attributs des unités présentes dans le scénario. En supplément, il faut prendre en compte que le temps de rechargement du SiegeTank est 3.5 fois plus important que celui du canon. Le déroulement du scénario est le suivant : une attaque vient de l'Est de la carte, les troupes sont composées de 3 vikings et 7 banshees. La défense dispose d'un lanceur et de trois canons. Leur objectif est de défendre une usine de cette attaque. Les banshees se déplacent bien plus rapidement que les vikings. La stratégie jugée optimale est de conserver les munitions du lanceur contre les vikings, car si le tir est réussi, cela detruit l'unité, alors que le canon aurait besoin de 5 tirs réussis. En prenant en compte la probabilité de réussite, les stocks, et le temps de rechargement des armes, cette stratégie semble la plus appropriée. Dans ce sens, en plus du script tirant sur la cible la plus proche de l'effecteur sélectionné, un script reproduisant la stratégie décrite a été implémenté afin d'avoir une baseline plus pertinente. Ces scripts seront respectivement nommés *ScriptBasique* et *ScriptAmélioré*.
Le même algorithme a été utilisé avec les mêmes paramètres, mais entraîné sur 250.000 itérations.

Agent	Moy	σ	Var	%Win
ScriptAléatoire	-1.21	1.46	2.13	0
ScriptBasique	3.43	3.52	12.41	23
ScriptAmélioré	7.04	3.75	14.07	72
RL Agent	8.27	2.54	6.43	85

TABLE 7 – Résultats sur le scénario réaliste

La table 7 permet de comparer les performances des différents agents sur le scénario réaliste. La stratégie du tir au plus proche n'est plus du tout efficace, puisqu'elle n'engendre une victoire que 23% du temps contre 77% pour la stratégie adaptée au scénario. Cette dernière obtient une récompense moyenne plus de deux fois supérieure à la stratégie basique. En outre, l'agent entraîné avec TAADA est meilleur que le ScriptAmélioré. Bien que la stratégie semble optimale d'un point de vue théorique, son application ne l'est pas forcément. En raison de la composante aléatoire dans la sélection de l'effecteur et l'apparition des menaces, il y a naturellement une perte d'optimalité, ce qui peut potentiellement modifier la stratégie optimale. C'est dans ce cas précis que le RL est plus intéressant qu'une méthode scriptée puisque l'algorithme peut faire preuve d'une certaine adaptabilité. En entrant plus dans les détails du scénario, l'objectif n'est pas seulement de détruire toutes les menaces, mais principalement de protéger l'usine. Le

ScriptAmélioré parvient 46% du temps à la protéger tandis que l'agent le fait 64% du temps. En sachant qu'il n'y a pas de distinction dans l'environnement, la perte de l'usine ou d'un effecteur engendre la même pénalité. Il est intéressant de noter une hausse de 39% des cas de sauvetages de l'usine par l'agent RL par rapport au script.

5 Discussions

Tout d'abord, abordons quelques points concernant le simulateur. Malgré ses aspects positifs, ce simulateur présente un certain nombre de lacunes, car il n'a pas été conçu à l'origine pour cette application. Nous avons également décidé de faire des choix de simplification. L'utilisation d'un simulateur de type militaire permettrait de valider ces résultats, avec des comportements et des spécifications techniques des unités plus proches de la réalité. Ensuite, d'une manière générale, le DRL présente certaines limites. L'une des principales limites de l'approche proposée est le manque d'interprétabilité des résultats. Bien que nous soyons en mesure de récupérer les valeurs de chaque paire, qui peuvent être interprétées comme leur niveau de menace. Mais le processus par lequel nous les obtenons reste difficile à expliquer en raison de la nature de boîte noire des réseaux neuronaux. En pratique, il est inimaginable de prendre de décisions critiques sans pouvoir les justifier. En ce sens, l'utilisation du mécanisme d'attention pourrait nous aider à comprendre les éléments les plus importants dans le processus de prise de décision. Avec ces résultats, l'agent a réussi à égaler, voire à surpasser, un script dans les différents scénarios. Ces résultats préliminaires sont particulièrement intéressants si l'on considère que plus la difficulté des scénarios augmente, plus le RL se montre à son avantage. Il est envisageable qu'avec des scénarios encore plus complexes, prenant en compte des unités plus nombreuses et plus diversifiées, TAADA devrait être encore plus performant par rapport aux scripts. Tandis que l'agent RL sera capable de s'adapter à différentes situations, il faut en parallèle concevoir des scripts de plus en plus complexes. Cela implique également que, pour une utilisation réelle, il est requis que quelqu'un puisse développer un tel algorithme, lequel doit prendre en considération toutes les situations. Ce transfert de connaissance reste limité, car bien que le nombre d'unités ennemies et les compositions changent, la disposition de la défense reste globalement la même. Il serait intéressant d'observer comment se comporte l'agent si le dispositif défensif est beaucoup plus aléatoire. Pour l'instant, la fonction de récompense reste basée sur la perte et la destruction d'unités ou de bâtiments. En conséquence, la modification de la récompense sera cruciale, car ce ne sera plus seulement la destruction d'unités qui sera prise en compte. L'enjeu sera donc de réussir à modéliser une récompense qui prenne en compte tous les aspects de la défense. L'ajout du brouillard de guerre sera également une évolution importante pour la crédibilité des résultats.

6 Conclusion

Avec cette approche, nous avons montré que dans des cas simples, le RL peut égaler ou même surpasser un script basé sur des règles. En basant l'approche sur l'évaluation des paires effecteur-menace, nous disposons d'une approche plus explicable et redondante pour différents scénarios. Elle est très robuste et transférable d'un scénario à l'autre. Ne pas dépendre du nombre et du type de menaces qui composent le scénario est crucial dans les applications réelles. Dans nos travaux futurs, nous visons à rendre l'architecture plus efficace, en prenant en compte le contexte global à travers les cellules récurrentes et le mécanisme d'attention. Mais également en utilisant de l'adversarial learning, car nous estimons qu'entraîner un attaquant intelligent permettra de rendre la défense plus efficace. Nous obtiendrons ainsi un agent capable de mener à bien la TEWA dans une multitude de scénarios en créant des scénarios de plus en plus complexes. Cela nous permettra de voir si le RL reste une bonne alternative par rapport aux algorithmes basés sur des règles.

Références

[1] Kai Arulkumaran, Marc Deisenroth, Miles Brundage, and Anil Bharath. A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine*, 34, 08 2017.

[2] Mustafa Azak and Ahmet Bayrak. A new approach for threat evaluation and weapon assignment problem, hybrid learning with multi-agent coordination. pages 1–6, 01 2008.

[3] An Xiao Dong and Liu Gou Qing. Application of neural network in the field of target threat evaluation. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, volume 6, pages 4237–4240 vol.6, 1999.

[4] Qiang Fu, Cheng-Li Fan, Yafei Song, and Xiang-Ke Guo. Alpha c2–an intelligent air defense commander independent of human decision-making. *IEEE Access*, 8 :87504–87516, 2020.

[5] Vinicius G. Goecks, Nicholas R. Waytowich, Derrik E. Asher, Song Jun Park, Mark R. Mittrick, John T. Richardson, Manuel Vindiola, Anne Logie, Mark Dennison, Theron Trout, Priya Narayanan, and Alexander Kott. On games and simulators as a platform for development of artificial intelligence for command and control. *CoRR*, abs/2110.11305, 2021.

[6] Mehmet Fatih Hocaoğlu. Rule based target evaluation and fire doctrine. In *Summer Simulation Multiconference*, 2019.

[7] Patrick A Hosein and Michael Athans. Preferential defense strategies. part i : The static case. 1990.

[8] Patrick Ahamad Hosein, Michael Athans, et al. Preferential defense strategies. part ii : The dynamic case. 1990.

[9] Cai Huaiping, Liu Jingxu, Chen Yingwu, and Wang Hao. Survey of the research on dynamic weapon-target assignment problem. *Journal of Systems Engineering and Electronics*, 17(3) :559–565, 2006.

[10] Fredrik Johansson. Evaluating the performance of tewa systems. 01 2010.

[11] Fredrik Johansson and Göran Falkman. A bayesian network approach to threat evaluation with application to an air defense scenario. pages 1 – 7, 08 2008.

[12] Fredrik Johansson and Göran Falkman. A comparison between two approaches to threat evaluation in an air defense scenario. pages 110–121, 10 2008.

[13] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *CoRR*, abs/2111.09794, 2021.

[14] Jiayi Liu, Gang Wang, Xiangke Guo, Siyuan Wang, and Qiang Fu. Deep reinforcement learning task assignment based on domain knowledge. *IEEE Access*, 10 :114402–114413, 2022.

[15] Stuart P Lloyd and Hans S Witsenhausen. Weapons allocation is np-complete. In *1986 summer computer simulation conference*, pages 1054–1058, 1986.

[16] Alan Manne. A target-assignment problem. *Operations Research*, 6(3) :346–351, 1958.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[18] Afshan Naseem, Shoab Khan, and Asad Malik. A real-time man-in-loop threat evaluation and resource assignment in defense. *Journal of the Operational Research Society*, 68 :1–14, 12 2016.

[19] S. Paradis, Abder Benaskeur, M. Oxenham, and P. Cutler. Threat evaluation and weapons allocation in network-centric warfare. volume 2, page 8 pp., 08 2005.

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[21] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529 :484–489, 01 2016.

[22] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017.

[23] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550 :354–359, 10 2017.

[24] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[25] Oriol Vinyals, Igor Babuschkin, Wojciech Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John Agapiou, Max Jaderberg, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575, 11 2019.

[26] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado Van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, and Rodney Tsing. Starcraft ii : A new challenge for reinforcement learning. 08 2017.

[27] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare : A survey. *ACM Comput. Surv.*, 55(1), nov 2021.

[28] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019.

Emulation of Zonal Controllers for the Power System Transport Problem

Eva Boguslawski[1,2], Alessandro Leite[1], Matthieu Dussartre[2], Benjamin Donnot[2], and Marc Schoenauer[1]

[1]TAU/LISN/Inria
[2]RTE Réseau de Transport d'Electricité

Résumé

Dans la phase de transition énergétique, les gestionnaires de réseaux de transport (GRT) développent des contrôleurs de zone pour surveiller le réseau électrique à l'aide d'algorithmes d'optimisation. Ces contrôleurs peuvent recevoir des consignes de la part d'opérateurs humains. Pour aider les opérateurs, les GRT ont besoin d'un assistant d'aide à la décision qui suggère des consignes. L'apprentissage par renforcement (RL) constitue une solution prometteuse, bien qu'il nécessite généralement de nombreuses itérations d'entraînement qui prennent beaucoup de temps. Notre recherche vise à émuler un contrôleur à l'aide de l'apprentissage par renforcement optimisé pour des itérations d'entraînement rapides et des temps de calcul gérables.

Mots-clés

Exploitation du réseau électrique, Apprentissage par renforcement, Trajectoire de consigne, Controleurs zonaux

Abstract

In the energy transition phase, Transmission System Operators (TSOs) are developing zonal controllers to monitor the power grid using optimization algorithms. These controllers can receive target plans from human operators. To assist operators, TSOs require a decision-support assistant that suggests smart target plans. Reinforcement learning (RL) is a candidate solution, though it typically demands numerous time-intensive training iterations. Our research aims to emulate a controller using reinforcement learning optimized for rapid training iterations and manageable computation times.

Keywords

Power grid operation, Reinforcement Learning, Target plan, Zonal controllers

1 Introduction

The massive arrival of renewable energies modifies the flows on the power grid, which are less predictable. To limit the need to adapt and reinforce the grid and to operate the grid as close as possible to its limits, the Transmission System Operators are developing zonal controllers. Each controller monitors a given zone of the grid. Thanks to predictive control models [1], one can handle line flow issues with a look-ahead window of a few minutes. Hence, a controller can limit some renewable productions through curtailment, control storage units, and take topological actions.

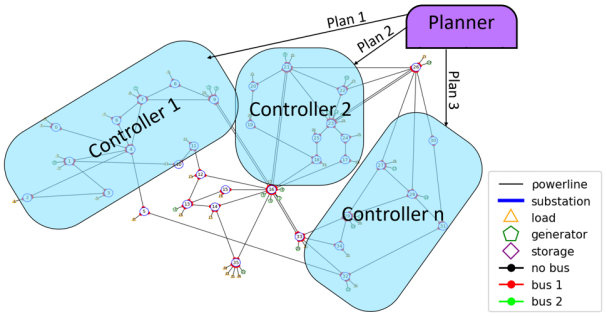


Figure 1: Representing three zonal controllers with a unique planner in a power grid with 36 nodes

To take into account information about other areas and problems that might occur after the few minutes window, each controller can receive a target plan from human operators to decide the appropriate actions to execute. Figure 1 provides an example of such scenario. In this case, each controller is responsible for implementing a plan set by a planner for its zone. The planner includes high-level planning tasks, which are handled nowadays by human operators. Examples of a target plan include (un)desirable configurations in the next hours in any zone. As a result, a controller tries to follow the target plan as closely as possible unless real-time issue forces it to take correction actions.

Operators have to manage more and more tasks in real-time. To make their work easier, TSOs needs a decision-support assistant to recommend relevant target plans. RL is a promising method to elaborate such target plans. However, to achieve good performances, RL requires a high number of training iterations which

is time-consuming. In this work, our goal is to design an emulator of a zonal controller that is fast enough to obtain a sufficient number of iterations and reasonable computation times during training. Indeed, in RL, a less realistic but faster emulator can be more efficient than a very accurate but slow one [2]. Accordingly, a controller must avoid a blackout at all costs and follow the target plan whenever possible. Consequently, we face a multi-objective and high-dimensional optimization problem.

2 Related works

Recent years have seen an increasing interest in developing RL policies to support the operation of power grids. It is mainly due to the advancement of deep reinforcement learning (DRL) methods [3]. For instance, the results of the Learning to Run a Power Network (L2RPN) [4] competition showed that the use of continuous actions (e.g., dispatching, curtailment, and actions on storage units) helps achieving good performance [5]. Nevertheless, the proposed approaches mainly rely on optimization or sampling techniques to choose an action without any exploration phase. In this work, we combine an RL policy with expert rules to learn to operate a power grid while adhering to a plan. We successfully use RL to handle continuous actions such as curtailment and storage power changes. A major challenge comprises maximizing the survival time and minimizing the deviations from the target plan.

3 Combining an RL policy and a heuristic

3.1 Modeling through Markov Decision Process (MDP)

Reinforcement learning (RL) [6] is a subset of machine learning (ML) where an agent learns from interaction with its environment through a system of rewards and punishments. Instead of relying on pre-existing data, the agent adopts a trial-and-error approach.

In order to apply the RL methods, we model our problem through a MDP shown in Figure 2. It can be formally defined as tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, T, \gamma)$ with $\mathcal{S}$ the set of states, $\mathcal{A}$ the set of actions, $\mathcal{T}$ the transition probability function, R the immediate reward function, T the length of an episode and γ the discount factor. Over time, the RL algorithm aims to find a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the discounted return $\sum_{k=t}^{T-1} \gamma^{k-t} r_{k+1}$, prompting the agent to refine its strategy.

Environment. We illustrated the environment in Figure 2. We use a simulated but realistic power grid with 118 substations. A time-step lasts 5 minutes. Each scenario lasts a week (2017 time-steps) and contains simulated productions, consumption and time series that respect a temporal consistency. The concrete development of the planner is not part of this work, so we decided to sample plans even if it ends with a stupid plan. If a plan endangers the power grid, the agent must be able to ignore it if necessary.

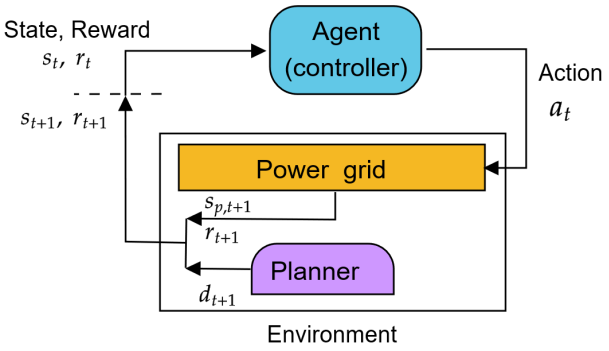


Figure 2: MDP to solve.

State space $\mathcal{S}$. At time t, the state is s_t containing:

1. The complete state of the power grid $s_{p,t}$. All information over power nodes (electricity produced and consumed), line flows, storage powers, curtailment limits.
2. The target plan sent by the planner d_t. In this work, the plan contains :
   - desirable storage charges setpoint for the next time step. The agent can control the storage powers but not directly their charges.
   - desirable curtailments limits. In our case, the desirable limits are always 100% of the maximum possible productions. In other words, we curtail as little as possible.

Action space $\mathcal{A}$. Three types of actions are allowed:

1. Line connection or disconnection.
2. Curtailment on renewable generators (impose maximum limits on each renewable production).
3. Storage powers setpoint change.

End condition. End of the scenario (the agent survived until the end of the week) or electrical blackout (game over).

We want to draw attention to the fact that there will be two levels of tasks using RL: (a) the planner level (not discussed here, in purple in Figure 1) aiming at coordinating controllers through target plans and (b) the lower level (subject of this work, in blue in Figure 1) aiming at emulating controllers and facilitating the future training of the planner.

3.2 Using heuristics

Previous works showed that RL agents can reach better performances when associated with a heuristic [7, 8] as shown in Figure 3. For example, doing nothing is a very wise action when the grid is safe (most time steps). It can also reduce the action space.

In our work, we design a heuristic dedicated to safe situations and topological actions to allow an RL policy (a neural network with 3 layers of 300 neurons)

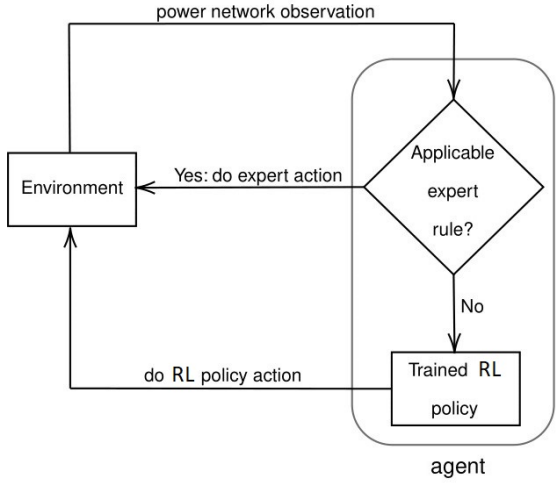


Figure 3: Embedding expert rules into an RL policy

Table 1: Agent's performance compared with the baselines

Agent	Survival time (avg. ± std)	Curtailment limits (avg.±std)	Storage setpoint difference (avg.±std)
DoNothing	549	100%	−−
RecoPowerline	471	100%	−−
Survivor agent	**1404±66**	32%±3%	30%±2%
Final agent	1013 ± 61	**71%±2%**	**2%±0.1%**

to focus its learning on risky situations and continuous actions. We use the Proximal Policy Optimization (PPO) algorithm [9] to train the policy.

4 Experimental results

Table 1 describes our agent's performance compared with three baselines: (a) an agent that always does nothing, (b) an agent reconnecting disconnected lines as soon as it can, and (c) an agent trained only to survive independent of the plan. Compared to the "survivor" agent, one can see that our heuristic considerably increases the average curtailment limits and gives an almost zero average distance to the storage setpoint. The drawback is a drop in the average survival time. Consequently, our heuristic might be overly strong when removing curtailment limits in safe situations. However our final agent still manages to survive for a long time compared with the "DoNothing" and "RecoPowerline" agents.

5 Discussion

These results are similar to the ones observed in the literature [4]. We observe that although adding a second objective reduces the survival time, more experiments are still necessary to confirm such results in more complex settings.

6 Conclusion

We propose a reinforcement learning (RL) policy to emulate a zonal controller in a multi-objective and high-dimensional problem. Our policy relies on a heuristic to enable the agent to handle unsafe situations and continuous actions. Experimental results show that our policy results in a good trade-off between survival and following the target plan. Future works includes replacing heuristic by an agent trained to handle curtailment optimally. Furthermore, we also plan to consider multiple zonal controllers and multiple target plans.

References

[1] C. Straub, S. Olaru, J. Maeght, and P. Panciatici, "Zonal congestion management mixing large battery storage systems and generation curtailment," in *IEEE Conference on Control Technology and Applications*, 2018, pp. 988–995.

[2] J. Truong, M. Rudolph, N. Yokoyama, S. Chernova, D. Batra, and A. Rai, "Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation," 2022.

[3] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[4] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O'Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Panciatici, and C. Romero, "Learning to run a power network challenge: a retrospective analysis," in *NeurIPS 2020 Competition and Demonstration Track*, H. J. Escalante and K. Hofmann, Eds., vol. 133, 2020, pp. 112–132.

[5] M. Dorfer, A. R. Fuxjäger, K. Kozak, P. M. Blies, and M. Wasserer, "Power grid congestion management via topology optimization with alphazero," 2022.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[7] D. Yoon, S. Hong, B.-J. Lee, and K.-E. Kim, "Winning the l2rpn challenge: Power grid management via semi-markov afterstate actor-critic," in *International Conference on Learning Representations*, 2021.

[8] E. van der Sar, A. Zocca, and S. Bhulai, "Multi-agent reinforcement learning for power grid topology optimization," *arxiv:2310.02605*, 2023.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

Emulation of Zonal Controllers for the Power System Transport Problem

Eva Boguslawski[1,2], Alessandro Leite[1], Matthieu Dussartre[2], Benjamin Donnot[2], and Marc Schoenauer[1]

[1]TAU/LISN/Inria
[2]RTE Réseau de Transport d'Electricité

Résumé

Dans la phase de transition énergétique, les gestionnaires de réseaux de transport (GRT) développent des contrôleurs de zone pour surveiller le réseau électrique à l'aide d'algorithmes d'optimisation. Ces contrôleurs peuvent recevoir des consignes de la part d'opérateurs humains. Pour aider les opérateurs, les GRT ont besoin d'un assistant d'aide à la décision qui suggère des consignes. L'apprentissage par renforcement (RL) constitue une solution prometteuse, bien qu'il nécessite généralement de nombreuses itérations d'entraînement qui prennent beaucoup de temps. Notre recherche vise à émuler un contrôleur à l'aide de l'apprentissage par renforcement optimisé pour des itérations d'entraînement rapides et des temps de calcul gérables.

Mots-clés

Exploitation du réseau électrique, Apprentissage par renforcement, Trajectoire de consigne, Controleurs zonaux

Abstract

In the energy transition phase, Transmission System Operators (TSOs) are developing zonal controllers to monitor the power grid using optimization algorithms. These controllers can receive target plans from human operators. To assist operators, TSOs require a decision-support assistant that suggests smart target plans. Reinforcement learning (RL) is a candidate solution, though it typically demands numerous time-intensive training iterations. Our research aims to emulate a controller using reinforcement learning optimized for rapid training iterations and manageable computation times.

Keywords

Power grid operation, Reinforcement Learning, Target plan, Zonal controllers

1 Introduction

The massive arrival of renewable energies modifies the flows on the power grid, which are less predictable. To limit the need to adapt and reinforce the grid and to operate the grid as close as possible to its limits, the Transmission System Operators are developing zonal controllers. Each controller monitors a given zone of the grid. Thanks to predictive control models [1], one can handle line flow issues with a look-ahead window of a few minutes. Hence, a controller can limit some renewable productions through curtailment, control storage units, and take topological actions.

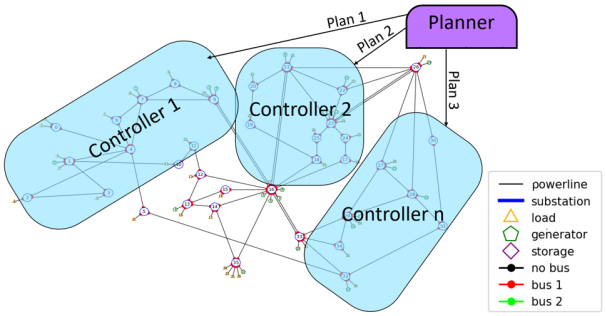


Figure 1: Representing three zonal controllers with a unique planner in a power grid with 36 nodes

To take into account information about other areas and problems that might occur after the few minutes window, each controller can receive a target plan from human operators to decide the appropriate actions to execute. Figure 1 provides an example of such scenario. In this case, each controller is responsible for implementing a plan set by a planner for its zone. The planner includes high-level planning tasks, which are handled nowadays by human operators. Examples of a target plan include (un)desirable configurations in the next hours in any zone. As a result, a controller tries to follow the target plan as closely as possible unless real-time issue forces it to take correction actions.

Operators have to manage more and more tasks in real-time. To make their work easier, TSOs needs a decision-support assistant to recommend relevant target plans. RL is a promising method to elaborate such target plans. However, to achieve good performances, RL requires a high number of training iterations which

is time-consuming. In this work, our goal is to design an emulator of a zonal controller that is fast enough to obtain a sufficient number of iterations and reasonable computation times during training. Indeed, in RL, a less realistic but faster emulator can be more efficient than a very accurate but slow one [2]. Accordingly, a controller must avoid a blackout at all costs and follow the target plan whenever possible. Consequently, we face a multi-objective and high-dimensional optimization problem.

2 Related works

Recent years have seen an increasing interest in developing RL policies to support the operation of power grids. It is mainly due to the advancement of deep reinforcement learning (DRL) methods [3]. For instance, the results of the Learning to Run a Power Network (L2RPN) [4] competition showed that the use of continuous actions (e.g., dispatching, curtailment, and actions on storage units) helps achieving good performance [5]. Nevertheless, the proposed approaches mainly rely on optimization or sampling techniques to choose an action without any exploration phase. In this work, we combine an RL policy with expert rules to learn to operate a power grid while adhering to a plan. We successfully use RL to handle continuous actions such as curtailment and storage power changes. A major challenge comprises maximizing the survival time and minimizing the deviations from the target plan.

3 Combining an RL policy and a heuristic

3.1 Modeling through Markov Decision Process (MDP)

Reinforcement learning (RL) [6] is a subset of machine learning (ML) where an agent learns from interaction with its environment through a system of rewards and punishments. Instead of relying on pre-existing data, the agent adopts a trial-and-error approach.

In order to apply the RL methods, we model our problem through a MDP shown in Figure 2. It can be formally defined as tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, T, \gamma)$ with $\mathcal{S}$ the set of states, $\mathcal{A}$ the set of actions, $\mathcal{T}$ the transition probability function, R the immediate reward function, T the length of an episode and γ the discount factor.

Over time, the RL algorithm aims to find a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the discounted return $\sum_{k=t}^{T-1} \gamma^{k-t} r_{k+1}$, prompting the agent to refine its strategy.

Environment. We illustrated the environment in Figure 2. We use a simulated but realistic power grid with 118 substations. A time-step lasts 5 minutes. Each scenario lasts a week (2017 time-steps) and contains simulated productions, consumption and time series that respect a temporal consistency. The concrete development of the planner is not part of this work, so we decided to sample plans even if it ends with a stupid plan. If a plan endangers the power grid, the agent must be able to ignore it if necessary.

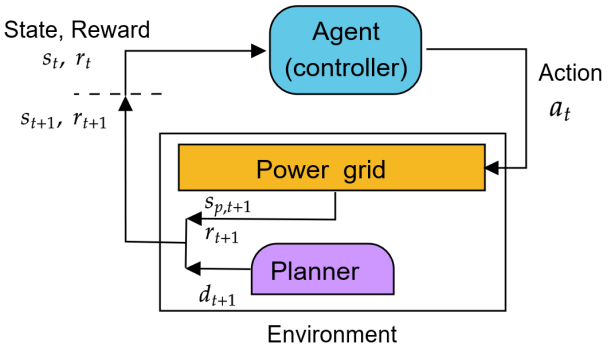


Figure 2: MDP to solve.

State space $\mathcal{S}$. At time t, the state is s_t containing:

1. The complete state of the power grid $s_{p,t}$. All information over power nodes (electricity produced and consumed), line flows, storage powers, curtailment limits.
2. The target plan sent by the planner d_t. In this work, the plan contains :
   - desirable storage charges setpoint for the next time step. The agent can control the storage powers but not directly their charges.
   - desirable curtailments limits. In our case, the desirable limits are always 100% of the maximum possible productions. In other words, we curtail as little as possible.

Action space $\mathcal{A}$. Three types of actions are allowed:

1. Line connection or disconnection.
2. Curtailment on renewable generators (impose maximum limits on each renewable production).
3. Storage powers setpoint change.

End condition. End of the scenario (the agent survived until the end of the week) or electrical blackout (game over).

We want to draw attention to the fact that there will be two levels of tasks using RL: (a) the planner level (not discussed here, in purple in Figure 1) aiming at coordinating controllers through target plans and (b) the lower level (subject of this work, in blue in Figure 1) aiming at emulating controllers and facilitating the future training of the planner.

3.2 Using heuristics

Previous works showed that RL agents can reach better performances when associated with a heuristic [7, 8] as shown in Figure 3. For example, doing nothing is a very wise action when the grid is safe (most time steps). It can also reduce the action space.

In our work, we design a heuristic dedicated to safe situations and topological actions to allow an RL policy (a neural network with 3 layers of 300 neurons)

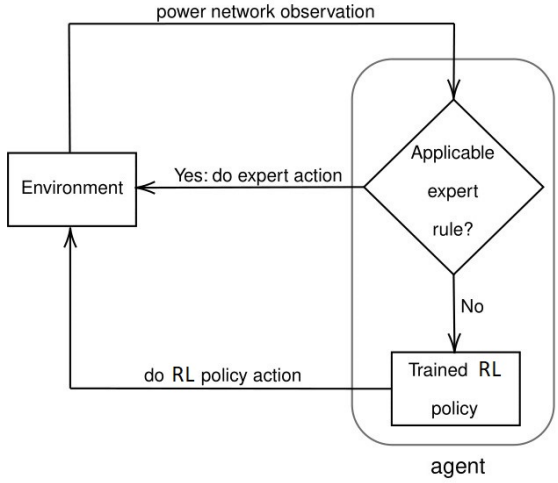


Figure 3: Embedding expert rules into an RL policy

Table 1: Agent's performance compared with the baselines

Agent	**Survival time** (avg. ± std)	**Curtailment limits** (avg.±std)	**Storage setpoint difference** (avg.±std)
DoNothing	549	100%	−−
RecoPowerline	471	100%	−−
Survivor agent	**1404±66**	32%±3%	30%±2%
Final agent	1013 ± 61	**71%±2%**	**2%±0.1%**

to focus its learning on risky situations and continuous actions. We use the Proximal Policy Optimization (PPO) algorithm [9] to train the policy.

4 Experimental results

Table 1 describes our agent's performance compared with three baselines: (a) an agent that always does nothing, (b) an agent reconnecting disconnected lines as soon as it can, and (c) an agent trained only to survive independent of the plan. Compared to the "survivor" agent, one can see that our heuristic considerably increases the average curtailment limits and gives an almost zero average distance to the storage setpoint. The drawback is a drop in the average survival time. Consequently, our heuristic might be overly strong when removing curtailment limits in safe situations. However our final agent still manages to survive for a long time compared with the "DoNothing" and "RecoPowerline" agents.

5 Discussion

These results are similar to the ones observed in the literature [4]. We observe that although adding a second objective reduces the survival time, more experiments are still necessary to confirm such results in more complex settings.

6 Conclusion

We propose a reinforcement learning (RL) policy to emulate a zonal controller in a multi-objective and high-dimensional problem. Our policy relies on a heuristic to enable the agent to handle unsafe situations and continuous actions. Experimental results show that our policy results in a good trade-off between survival and following the target plan. Future works includes replacing heuristic by an agent trained to handle curtailment optimally. Furthermore, we also plan to consider multiple zonal controllers and multiple target plans.

References

[1] C. Straub, S. Olaru, J. Maeght, and P. Panciatici, "Zonal congestion management mixing large battery storage systems and generation curtailment," in *IEEE Conference on Control Technology and Applications*, 2018, pp. 988–995.

[2] J. Truong, M. Rudolph, N. Yokoyama, S. Chernova, D. Batra, and A. Rai, "Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation," 2022.

[3] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[4] A. Marot, B. Donnot, G. Dulac-Arnold, A. Kelly, A. O'Sullivan, J. Viebahn, M. Awad, I. Guyon, P. Panciatici, and C. Romero, "Learning to run a power network challenge: a retrospective analysis," in *NeurIPS 2020 Competition and Demonstration Track*, H. J. Escalante and K. Hofmann, Eds., vol. 133, 2020, pp. 112–132.

[5] M. Dorfer, A. R. Fuxjäger, K. Kozak, P. M. Blies, and M. Wasserer, "Power grid congestion management via topology optimization with alphazero," 2022.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[7] D. Yoon, S. Hong, B.-J. Lee, and K.-E. Kim, "Winning the l2rpn challenge: Power grid management via semi-markov afterstate actor-critic," in *International Conference on Learning Representations*, 2021.

[8] E. van der Sar, A. Zocca, and S. Bhulai, "Multi-agent reinforcement learning for power grid topology optimization," *arxiv:2310.02605*, 2023.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

Session 3 : Apprentissage Automatique (Commune avec CNIA)

Apprentissage multijoueurs supervisé

M. Kazi Aoual[1,3], H. Soldano[2,3], C. Rouveirol[1], V. Ventos[3]

[1] UMR CNRS 7030 Institut Galilée - Université Sorbonne Paris-Nord, LIPN
[2] UMR CNRS 7205 Museum National d'Histoire Naturelle, ISYEB
[3] NukkAI, Paris

mkazi[at]nukk.ai, henry.soldano[at]mnhn.fr, rouveirol[at]lipn.univ-paris13.fr, vventos[at]nukk.ai

Résumé

Dans le cadre de l'apprentissage supervisé, si les données sont étiquetées par différents annotateurs qui ont des avis différents, il peut être difficile d'établir un modèle compact incluant tous les avis. Une solution à ce problème est d'inclure l'identité de l'annotateur dans le langage des hypothèses, mais cette approche n'est pas adaptée à un grand nombre d'annotateurs. Dans nos travaux, nous présentons une approche utilisant l'identité de l'annotateur dans l'apprentissage sans l'inclure dans le langage des hypothèses, en l'illustrant sur une situation de Bridge.

Mots-clés

Multijoueurs, apprentissage supervisé, plusieurs annotateurs.

Abstract

In the context of supervised learning, if the data is labeled by different annotators who exhibit different behaviors, it can be challenging to establish a general model. One solution to this problem is to include the identity of the annotator in the hypothesis language, but this approach is not suitable for a large number of annotators. In our work, we present an approach that allows the use of the annotator's identity in learning without including it in the hypothesis language, illustrating it in a Bridge situation.

Keywords

Multiplayers, supervised learning, multiple annotators.

1 Introduction

Dans le cadre de l'apprentissage supervisé, il est courant que les données soient étiquettées par de multiple sources. Ces sources peuvent être différentes personnes, différents capteurs, différentes machines, etc. Apprendre un modèle général à tout le monde devient difficile lorsque les différentes sources se comportent de façon différente face à une même situation. Dans cet article, nous traitons d'un cas particulier de l'apprentissage supervisé dans lequel l'univers des objets, qui contient l'ensemble d'apprentissage, est constitué d'objets provenant de différentes sources et où la valeur prédite par le modèle dépend, dans une certaine mesure, de la source de l'objet.

Notre cas d'étude dans cet article est l'entame dans le jeu du Bridge, qui est la première carte déposée par un joueur durant la phase du jeu de la carte. La particulartié de cette situation tient au fait qu'elle fait suite à une suite d'enchères bien connue du monde du Bridge, et les joueurs peuvent avoir des décisions différentes lors de cette entame. Le but est de prédire, étant donné une situation et une main, si un joueur va jouer une carte d'une couleur majeure (♠, ♡) ou mineure (♢, ♣).

Ce cas d'étude se situe dans le cadre que nous avons appelé *Apprentissage multijoueurs supervisé*, dont l'objectif est d'apprendre un modèle compact qui permet de prédire la décision d'un joueur face à une nouvelle situation. Ce cadre d'apprentissage peut s'appliquer à n'importe quelle situation multiannotateurs telles que celle mentionnée ci-dessus.

Ce problème peut se ramener à un problème d'apprentissage supervisé classique, dans lequel l'identité du joueur est utilisée en tant que caractéristique présente dans l'espace de recherche. Mais nous montrerons qu'une augmen-

tation du nombre de joueurs rend rapidement l'apprentissage très long, et les modèles très gros, ce qui limite les cas d'usages dans lesquels on voudrait apprendre des stratégies sur des centaines de joueurs, ou plus généralement de sources. Nous proposons ainsi dans cette contribution une approche dite *parcimonieuse*, qui n'inclut pas explicitement l'identité du joueur dans l'espace de recherche, mais qui l'utilise tout de même pour l'apprentissage et la prédiction.

2 Préliminaires

2.1 Présentation du problème

Le Bridge est un jeu qui se joue à quatre joueurs (*N*,*S*,*E*,*W*), répartis en deux équipes, le *déclarant* (*N*,*S*) et le *défenseur* (*E*,*W*). Le jeu se déroule en deux phases, les *enchères*, durant lesquelles les joueurs déterminent le nombre de pli qu'ils pensent pouvoir réaliser, et le *jeu de la carte*, durant lequel les joueurs jouent les cartes de leur main. Dans notre cas d'étude, on considère que les enchères sont terminées. L'expérience qui nous intéresse est l'entame, qui est la première carte jouée de la phase du jeu de la carte et est jouée par *West*. Cette phase de jeu survient juste après les enchères, qui donnent de nombreuses informations à tous les joueurs sur les mains des autres joueurs. Dans l'expérience, on propose à différents joueurs experts de Bridge de jouer l'entame dans une situation particulière où la distribution des cartes dans la main de *West* dans l'ordre des couleurs ($\spadesuit, \heartsuit, \diamondsuit, \clubsuit$) est $(3,3,4,3)$ ou $(3,3,3,4)$. La décision binaire que l'on veut modéliser est *"est-ce que West va jouer une carte d'une couleur majeure ou jouer la carte de la couleur mineure contenant 4 cartes ?"*. Dans cette expérience, le joueur est tiraillé entre entamer avec sa couleur la plus longue (une mineure) ou entamer dans une majeure.

2.2 Apprentissage supervisé

Nous nous intéressons ici à la version la plus simple de l'apprentissage automatique, l'apprentissage supervisé de concept [1]. Le problème se formule de la manière suivante : Soit un concept-cible c, un langage d'hypothèse L, un univers d'objets O et un sous-ensemble $E \subseteq O$ pour les objets x desquels on sait si c est vrai. On écrit aussi $E^+ \subseteq E$ le sous-ensemble d'objets pour lesquels c est vrai et E^- le sous-ensemble $E \setminus E^+$ des objets pour lesquels c est faux. Les éléments de E sont appelés *exemples*, divisés en exemples *positifs* dans E^+ et *négatifs* dans E^-.

On pose que pour tout élément h de L et pour tout objet x de O on peut savoir si h est vrai pour x, on dira alors que h *couvre* x. On note $\mathrm{ext_X}(h)$ la *couverture* de $X \subseteq O$ formée des éléments de X pour lesquelles h est vraie. Le problème posé est de trouver une hypothèse h dans L telle que $\mathrm{ext_O}(h) = O^+$, c'est-à-dire une hypothèse de L vraie pour les mêmes objets de O que le concept-cible c. Comme on sait si c est vrai ou non seulement sur les objets de E, on cherchera une solution h telle que :

$$\mathrm{ext_E}(h) = E^+ \tag{1}$$

et on supposera alors que $\mathrm{ext_O}(h) = O^+$.

Une hypothèse h est ici un ensemble de règles concluant sur c. On aura $h = \{r_1, \ldots, r_n\}$ où une règle est de la forme $t \rightarrow c$ où t est une conjonction d'atomes pris dans un ensembles d'atomes A. Un objet x est lui-même décrit comme le sous-ensemble d_x des atomes de A qui sont vrais pour x, et représente donc d'un point de vue logique une interprétation. La conjonction t est alors vraie pour x si tous les atomes de t sont dans d_x. Il en résulte que h est vraie pour x si et seulement s'il existe une règle de h, $r_i = t_i \rightarrow c$, telle que t_i vrai pour x.

Une hypothèse h est alors une solution de l'équation 1 si et seulement si :

1. Toute règle r_i de h est *valide* sur E :

$$\forall x \in E^-, \forall r_i \in h, t_i \text{ faux pour } x \tag{2}$$

   et ne conclut donc jamais c pour les objets de E^-, ce qu'on peut aussi écrire $\mathrm{ext_{E^-}}(r) = \emptyset$ et, cela étant vrai pour toutes les règles de h, on a $\mathrm{ext_{E^-}}(h) = \emptyset$

2. Pour tout exemple x de E^+ il existe une règle r_i dans h qui conclut sur c

$$\forall x \in E^+, \exists r_i \in h \text{ telle que } t_i \text{ vrai pour } x \tag{3}$$

   et on a donc $\mathrm{ext}_{E^+}(h) = E^+$.

Enfin nous ne considérerons que les solutions constituées de règles *porteuses* c'est-à-dire qui permettent de déduire c pour au moins un exemple de E^+, formellement :

$$\forall r_i \in h, \exists x \in E^+, \text{ tel que } t_i \text{ vrai pour } x \tag{4}$$

L'ensemble des règles valides et porteuses sur E constitue l'ensemble des règles *candidates* : toute solution est

constituée de règles candidates. Parmi toutes les solutions on cherchera de plus une solution h de complexité la plus faible possible, mesurée en nombre de règles.

Considérons un objet x non étiqueté (on ne sait pas si c est vrai pour x), si il existe r_i dans h telle que t_i est vrai pour x, on peut déduire c de h et de d_x et on décide alors que c est vrai pour x, sinon on décide que c est faux pour x. Cette règle ne commet par définition pas d'erreur sur l'ensemble E.

Plus généralement, l'apprentissage peut ne pas être *réalisable*, c'est-à-dire qu'il n'existe pas de solution, ou bien, même si il est réalisable, on peut préférer une hypothèse h ne satisfaisant pas les équations 2 et 3 mais représentant un compromis entre sa justesse (proportion d'objets de E pour lesquels la prédiction est correcte) et sa complexité, en utilisant par exemple une fonction de pénalisation de l'hypothèse croissante avec sa complexité. Dans tous les cas, on évaluera a posteriori la qualité de la solution retenue h sur des exemples qui n'ont pas été utilisés en apprentissage. Pour cela on utilise une fonction d'évaluation, comme l'estimation de sa justesse en dehors de E.

3 Apprentissage multijoueur

Le problème considéré est le suivant. On dispose d'un ensemble d'états S étiquetées par un ensemble de joueurs J. Soit $j \in J$ et $x \in S$. L'objectif est d'apprendre un modèle qui décide pour une paire (j, x) si c est vrai. On se place dans le cadre de l'apprentissage supervisé décrit en section 2.2, et l'apprentissage multijoueur supervisé en est un cas particulier. Un exemple positif est une paire (j, x) telle que c est vrai, un exemple négatif est une paire (j, x) telle que c est faux. E^+ est l'ensemble des exemples positifs et E^- l'ensemble des exemples négatifs. Chaque état est décrit par un ensemble d'atomes.

Une solution sera dans ce contexte $h = r_1, \ldots, r_n$ où chaque règle r_i est de la forme $1_C \wedge t \rightarrow c$ où C est un sous-ensemble de joueurs et 1_C est vrai pour (j, x) si $j \in C$ et t est une conjonction d'atomes inclus dans A. Pour cet article, on omettra le connecteur $\wedge$ pour les conjonctions d'atomes pour décrire l'état.

Exemple 1 *Soit* $J = \{j_1, j_2, j_3\}$, $A = \{a, b, d, e\}$. $1_{\{j_1,j_2\}} \wedge bd$ *est vrai pour la paire* (j_1, bde) *car* $j_1 \in \{j_1, j_2\}$ *et* $bd \subseteq bde$.

3.1 Approche directe

L'approche directe (*Dir*) pour la résolution de ce type de problème, consiste à prendre en compte l'identité du joueur dans l'espace de recherche. La composante 1_C d'une règle sera représentée par une conjonction d'atomes. On représente les $2^{|J|}$ sous-ensembles possibles de joueurs par $|J|$ atomes, avec le codage suivant :
Soit $1_{\overline{k}}$ l'atome prenant sa valeur vrai pour (j, x) si $j \neq k$. 1_C est équivalent à la conjonction des atomes $1_{\overline{k}}$ tels que $k \notin C$.

Exemple 2 *Soit* $J = \{j_1, j_2, j_3\}$. $1_{\{j_1,j_2\}}$ *s'écrit* $1_{\overline{j_3}}$ *et* $1_{\{j_2\}}$ *s'écrit* $1_{\overline{j_1}} \wedge 1_{\overline{j_3}}$.

3.2 Approche parcimonieuse

Dans l'approche que nous proposons, on ne cherchera que les solutions dont les règles r sont *parcimonieuses*, c'est-à-dire telles que C est le sous ensemble C_t de tous les joueurs j tels que $t \rightarrow c$ est *valide* et *porteuse* pour le joueur j au sens suivant :

Définition 1 *Soit* E_j *le sous-ensemble de* E *des paires* (j, x) *du joueur* j,

1. $t \rightarrow c$ *est valide sur* E_j *si et seulement si*

   Pour toute paire $(j, x) \in E_j^-$, *telle que* t *est faux pour* x (5)

2. $t \rightarrow c$ *est porteuse sur* E_j, *si et seulement si*

   Il existe une paire $(j, x) \in E_j^+$, *telle que* t *est vrai pour* x (6)

3. $1_C \wedge t \rightarrow c$ *est parcimonieuse si et seulement si*

   $C = C_t$ *où* $C_t = \{j \in J \mid t \rightarrow c$ *valide et porteuse pour* $j\}$ (7)

Exemple 3 *On a 9 paires (joueur, état) concernant 3 joueurs* j_1, j_2, j_3 *et 4 états* $x_1 \ldots x_4$. *Un état est ici une situation de jeu, et le concept-cible correspond au choix d'entamer d'une couleur majeure* $(majeure)$ *ou mineure* $(\neg majeure)$ *lors de la phase du jeu de la carte. Chaque état est décrit par les valeurs de vérités prises par les atomes* a, b, d, e. *La table 1 présente* E. *Chaque paire* p_k *=(joueur* j, *état* x*) est associée à l'étiquette* $+$ $(majeure)$ *ou* $-$ $(\neg majeure)$.
On constate ainsi dans la table que les joueurs j_2 *et* j_3 *jouent de la même manière dans l'état* x_1 $(majeure)$ *alors*

Joueurs/Objets	$x_1 = bde$	$x_2 = ade$	$x_3 = abe$	$x_4 = abd$
j_1	p_1^-	p_2^+	p_3^+	
j_2	p_4^+	p_5^-	p_6^+	
j_3	p_7^+		p_8^+	p_9^-

TABLE 1 – Paires (joueur,objet) dans E pour l'Exemple 3. La première ligne montre la représentation des états et la première colonne montre les joueurs. Dans la ligne l et la colonne c on trouve la paire $p_k = (j_l, x_c)$ avec son étiquette $+$ ou $-$.

que j_1 joue différemment d'eux ($\neg majeure$). Considérons les conjonctions t et t' suivantes :

- *$t = ae$ est vrai pour les états x_2 et x_3 et donc pour les paires $\{p_2^+, p_3^+\} \subseteq E_1$, $\{p_5^-, p_6^+\} \subseteq E_2$ et pour la paire $\{p_8^+\} \subseteq E_3$. En conséquence la règle $r = ae \rightarrow c$ est valide et porteuse pour les joueurs j_1 et j_3, et donc la règle $r = 1_{C_{ae}}, ae \rightarrow c$ est parcimonieuse avec $C_{ae} = \{j_1, j_3\}$ et permet de décider majeure pour les paires $\{p_2^+, p_3^+, p_8^+\}$.*
- *$t' = be$ est vrai pour les états x_1 et x_3 et donc pour les paires $\{p_1^-, p_3^+\} \subseteq E_1$, $\{p_4^+, p_6^+\} \subseteq E_2$ et $\{p_7^+, p_8^+\} \subseteq E_3$ est donc valide et porteuse pour j_2 et j_3. La règle $r = 1_{\{j_2, j_3\}} \wedge bd$ est parcimonieuse et permet de décider majeure pour $\{p_4^+, p_6^+, p_7^+, p_8^+\}$.*

On en conclut que l'hypothèse $h = \{r, r'\}$ décide majeure pour les paires $\{p_2^+, p_3^+, p_4^+, p_6^+, p_7^+, p_8^+\}$ c'est-à-dire E^+ et ne décide majeure pour aucune paire de E^-, elle est donc solution du problème d'apprentissage multijoueur posé.

La restriction de la recherche des solutions aux solutions parcimonieuses est justifiée par le résultat suivant :

Proposition 1 *Soit h une solution de l'approche directe, et h' l'hypothèse parcimonieuse obtenue en remplaçant chaque règle $r = c \leftarrow 1_C \wedge t$ de h par la règle parcimonieuse $r' = c \leftarrow 1_{C_t} \wedge t$, alors h' est aussi une solution avec $|h'| = |h|$.*

Démonstration 1 *La règle parcimonieuse associée r' est telle que $C_t = V_t \cap P_t$ où V_t est le sous-ensemble de joueurs pour lesquels $t \rightarrow c$ est valide et P_t le sous-ensemble de joueurs pour lesquels $t \rightarrow c$ est porteuse. Comme r appartient à h, elle est valide, donc en particulier $t \rightarrow c$ est valide pour tous les joueurs de C. En conséquence on a (i) $C \subseteq V_t$. De même si $t \rightarrow c$ n'est pas porteuse pour un joueur j de C alors ce joueur ne participe pas à la couverture de r, c'est-à-dire $\mathrm{ext}_{E_j^+}(r) = \emptyset$. Soit alors $C'' = C \cap P_t$ et $r'' = 1_{C''} \wedge t \rightarrow c$, on a (ii) $\mathrm{ext}_{E^+}(r) = \mathrm{ext}_{E^+}(r'')$. De (i) et (ii) on déduit que $\mathrm{ext}_E(r) = \mathrm{ext}_E(r'')$ et comme $C'' \subseteq C_t$ on a aussi $\mathrm{ext}_E(r') \supseteq \mathrm{ext}_E(r'')$ et en conséquence $\mathrm{ext}_{E^+}(r') \supseteq \mathrm{ext}_{E^+}(r)$. Comme cela est vrai pour toute les règles r de h on a donc $\mathrm{ext}_E(h') \supseteq \mathrm{ext}_E(h) = E^+$ et les règles de h' étant valides on a $\mathrm{ext}_E(h') = E^+$. On en conclut que h' est aussi solution.*

Corollaire 1 *Soit h une solution de taille minimale alors la solution parcimonieuse associée h' est aussi de taille minimale.*

Pour trouver une solution de taille minimale il suffit donc de parcourir les hypothèses parcimonieuses. L'espace R' des règles parcimonieuses est de taille $|R'| = 2^{|A|}$, exponentiellement plus petit (en nombre de joueurs) que l'espace R de toutes les règles dans l'approche directe et dont la taille est $|R| = 2^{|J|} * 2^{|A|}$.
S'il existe r_i telle que $j \in C_{t_i}$ et t_i vrai pour x, on peut déduire c. La règle de décision est définie dans la Définition 2 et ne commet pas d'erreur sur E

Définition 2 (Règle de décision multijoueur) *Soit une paire (j, x). S'il existe r dans h telle que $j \in C_{t_i}$ et t vrai pour x, on décide c pour (j, x) sinon on décide $\neg c$ pour (j, x)*

4 Un algorithme d'apprentissage multijoueur

L'apprentissage multijoueur étant un cas particulier d'apprentissage supervisé à base de règles, on peut utiliser pour l'approche directe un algorithme de parcours des règles descendant standard en parcourant l'espace des conjonctions de la forme $1_C \wedge t$ qui décrit également la règle associée $1_C \wedge t \rightarrow c$. Le même schéma algorithmique peut-être utilisé dans l'approche parcimonieuse en parcourant l'espace des conjonctions t portant sur les états et donc l'ensemble des règles parcimonieuses $1_{C_t} \wedge t \rightarrow c$. Nous présentons ci-dessous l'algorithme général avec les notations multijoueur : un objet est une paire (j, x) et une règle est notée (t, C).

4.1 Algorithme d'apprentissage glouton

Dans l'algorithme glouton par couverture de l'apprentissage de concept à base de règles, on ramène la recherche d'une meilleure solution $h = r_1 \dots r_m$ à la recherche itérative d'une meilleure règle qui soit candidate, c'est-à-dire valide et porteuse. L'algorithme s'arrête lorsqu'on a trouvé une solution $h = r_1 \dots r_m$ qui ne peut-être améliorée. La couverture $\text{ext}_{E^+}(h)$ de la solution partielle courante h augmente à chaque itération, et on peut alors réduire à l'étape suivante l'ensemble des exemples positifs à ceux qui ne sont pas couverts par cette solution partielle. Dans la variante présentée ci-dessous on utilise à chaque itération un exemple positif particulier, non encore couvert, appelé "graine". On n'explore alors à chaque itération que les conjonctions couvrant cette graine. Dans cette variante, très utilisée en particulier en Programmation Logique Inductive [2], l'espace des conjonctions à explorer est réduit. Pour la recherche d'une meilleure règle candidate nous utilisons une recherche par faisceau de largeur k.

Algorithm 1 multiPlayer learning

Require: Ensemble d'apprentissage E, largeur du faisceau k, profondeur maximale de la recherche d
Ensure: h : solution

```
while X ≠ ∅ do
    Sélectionner une graine p0 = (j0, x0) dans X
    (t, C) ← bestRule(p0, E−, X, k, d)
    if t = faux then
        X ← X \ {p0}
    else
        X ← X \ covered((t, C), X)
        h ← h + (t, C)
    end if
end while
```

— X est l'ensemble des exemples positifs non couverts par la solution partielle h et pouvant encore l'être : sont exclus de X les exemples pour lesquels aucune règle valide et porteuse n'a été trouvée lorsqu'ils jouaient le rôle de graine lors d'une itération précédente.

— $bestRule(X, C, k)$ renvoie la meilleur règle (t, C) candidate couvrant la graine p_0 pour l'ensemble X d'exemples positifs courant et l'ensemble d'exemples négatifs E^-. La meilleure règle est la règle candidate (donc valide et porteuse) maximisant le nombre d'exemples positifs de X. (t, C) est ajoutée à la solution partielle courante h. Si la recherche échoue (t= faux) la solution partielle n'est pas modifiée et la graine est retirée de X.

— $covered((t, C), X)$ renvoie l'ensemble des exemples positifs de X couverts par la règle sélectionnée (t, C) et incluent la graine p_0 de l'itération courante. Ils sont retirés de X.

Remarquons que si l'apprentissage est réalisable (s'il existe une solution) la recherche d'une meilleure règle candidate n'échoue jamais et h couvre E^+. Cependant, d'une part, on peut ajouter des contraintes, par exemple un budget b en nombre d'itérations : on aura alors au plus b règles dans h, ou encore un nombre d'atomes maximum $\#a$ dans une règle, ce qui conduit parfois à l'échec de la recherche d'une meilleure règle et à renvoyer une solution partielle ne couvrant pas tout E^+. D'autre part, même sans contraintes supplémentaires, l'apprentissage peut ne pas être réalisable, par exemple si pour un même joueur j et une même situation x on trouve deux exemples avec les étiquettes c et $\neg c$. L'algorithme décrit ci-dessus renvoie éventuellement une solution partielle, valide mais ne couvrant qu'une partie de E^+.
Dans le cas général *bestRule* est un algorithme descendant qui parcourt un espace de conjonctions à partir de $\top = \text{vrai}$, le spécialise en développant un arbre, et renvoie la meilleure règle candidate trouvée, au sens d'une fonction s. Pour des raisons de complexité, la recherche n'explore pas tout l'espace des conjonctions, c'est une recherche en *faisceau* (ou *beam search*) de *largeur* k qui utilise une fonction de *guidage* g.

Algorithm 2 bestRule

Require: Graine p_0, exemples négatifs E^-, Ensemble des exemple positifs non couverts encore X, largeur du faisceau k, profondeur maximale de la recherche d
Ensure: r : meilleure règle

```
L ← {vrai}
r ← r⊥
while L ≠ ∅ do
    L ← ∪l∈L minimalSpecializations(l, p0)
    S ← candidate(L)
    r ← bestCandidate(s, r, S)
    L ← L \ S
    L ← kMostPromisingTerms(k, g, L)
end while
```

A une itération les éléments de la liste courante L sont spécialisés pour former une nouvelle liste L, on retire de L l'ensemble S des conjonctions associées à des règles candidates, et si la meilleure règle issue de S, au sens de la fonction s, est meilleure que la meilleure règle candidate courante r elle la remplace. g renvoie un score mesurant l'intérêt de spécialiser une conjonction et seules les k meilleures

conjonctions de L sont gardées pour l'itération suivante.

4.2 Approche parcimonieuse versus approche directe

Les deux approches se différencient d'abord par l'espace de recherche exploré, plus petit dans l'approche parcimonieuse et par la fonction de guidage g et la fonction d'évaluation s à maximiser.

Dans le cas direct une conjonction est constituée d'atomes portant sur le joueur, formant la composante 1_C et d'atomes portant sur l'état formant la composante t.

La fonction $candidate(L)$ renvoie la liste S des conjonctions dont la règle associée (C, t) est candidate, $bestCandidate(s, r, S)$ retourne la meilleure règle au sens de s, c'est-à-dire soit r soit une une règle issue de S meilleure que r.

Dans le cas parcimonieux un élément de L est constitué uniquement d'atomes portant sur l'état x. $candidate(L)$ renvoie l'ensemble des conjonctions t associés à des règles parcimonieuses (t, C_t), où C_t est l'ensemble des joueurs pour lesquels $t \rightarrow c$ est valide et porteuse, et $bestCandidate(s, r, S)$ met à jour la meilleure règle parcimonieuse courante. Dans les deux cas la fonction d'évaluation s des solutions candidates est la couverture : la meilleure règle candidate est celle couvrant le plus d'exemples de X (comme elle est candidate on a aussi $\text{ext}_{E^-}(r) = \emptyset$) :

$$s(r) = |\text{ext}_X(r)| \tag{8}$$

Dans l'approche directe, nous n'utilisons pas explicitement l'information sur le joueur dans la fonction de guidage g. Celle-ci est une altération de la précision $P/P + N$ telle que à précision égale la règle ayant la plus petite fréquence d'erreurs de déclenchement sera préférée :

$$g(r) = \frac{P - N}{P + N} \tag{9}$$

où $P = \text{ext}_X(r)$ et $N = \text{ext}_{\text{E}^-}(r)$

Dans l'approche parcimonieuse nous avons utilisé pour la fonction de guidage la séparation de E en sous-ensembles $E_1, \ldots E_{|J|}$ de même joueur :.

$$g_P(r) = \text{Max}_{j \in J}\ g_j(t) \tag{10}$$

où $P_j = \text{ext}_{X_j}(t \rightarrow c)$, $N_j = \text{ext}_{\text{E}_\text{j}^-}(t \rightarrow c)$ et $g_j(t) = \frac{P_j - N_j}{P_j + N_j}$.

Remarquons que g_P n'utilise que t, sous la forme d'une règle $t \rightarrow c$, et considère tous les joueurs de J. En effet, l'intérêt de la règle $r = (t, C_t)$ se mesure aussi en fonction de joueurs pour lesquels la règle n'est pas encore valide mais peut le devenir au cours d'une spécialisation. En revanche, concernant l'évaluation des règles candidate (t, C_t), on utilise bien la règle $1_{C_t} \wedge t \rightarrow c$ comme dans l'approche directe.

5 Expérimentations

On dispose d'un jeu de données contenant 7264 exemples, étiquetés par 10 joueurs différents. Chaque exemple correspond à un état de jeu et les caractéristiques associées à chaque exemple sont des descriptions de la main face à laquelle le joueur doit prendre la décision d'entamer avec une majeure ou une mineure. Les enchères sont implicites et ne sont pas décrites dans les exemples. La figure 1, indique la mesure d'accord entre les différents joueurs sur cette tâche. L'histogramme mesure le pourcentage d'exemples pour lequel les joueurs prennent la même décision.

Pour effectuer cette mesure, nous avons entraîné un modèle *parcimonieux* (décrit en section 3.2), qui apprend une solution composée de règles d'entame spécifiques à des sous-ensembles de joueurs. Ainsi, pour chaque exemple (j, x, c_x^j) dans l'ensemble de test, on construit pour l'état x les paires (j', x) pour chaque joueur j' différent de j, et à chaque paire (j', x) on associe $c_x^{j'}$ l'étiquette prédite par le modèle. On obtient donc les exemples étiquetés $(1, x, c_x^1), (2, x, c_x^2), \ldots, (j - 1, x, c_x^{j-1}), (j, x, c_x^j), \ldots, (10, x, c_x^{10})$ qui sont ensuite présentés au modèle dont on récolte les décisions pour chaque joueur. L'histogramme mesure donc un degré d'accord relatif à une solution h, et non un degré d'accord absolu entre les joueurs. Le modèle est entraîné sur 1016 exemples, et on mesure le désaccord sur 2904 exemples de test.

On remarque un très faible nombre d'exemples (0.4%) sur lesquels tous les joueurs en accord. On a une majorité (92.5%) des exemples sur lesquels on a entre 50% et 80% des joueurs qui sont en accord. Cette mesure indique que les joueurs ont des comportements très différents en fonction des exemples et justifie l'apprentissage d'un modèle multijoueur.

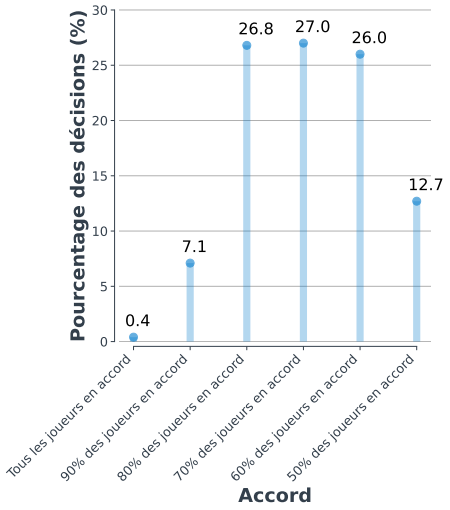


FIGURE 1 – Mesure d'accord entre les joueurs. En abscisse on a le nombre de joueurs dont les exemples ont la même étiquette pour le même état.

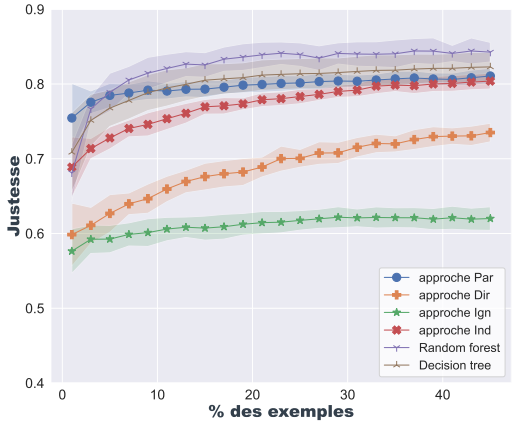


FIGURE 2 – Justesses des différentes approches en fonction du nombre d'exemples d'apprentissage

5.1 Comparaison des approches en fonction du nombre d'exemples d'apprentissage

Nous considérons pour des ensembles d'apprentissage de taille $|E| = n$ croissante, la justesse des prédictions de plusieurs modèles multiclasse c_1 =*Majeure*, c_2 = *Mineure*. Chaque justesse est calculée sur N essais, chacun associé à une permutation différente de l'ensemble de tous les exemples. On prend pour chaque essai les n premiers éléments de la permutation pour former E et les k derniers pour former l'ensemble test sur lequel la justesse est évaluée. On utilise le même algorithme présenté ci-dessus selon plusieurs approches :

— **Approche directe** *Dir*

— **Approche parcimonieuse** *Par*

— **Approche directe individuelle** *Ind* où le contexte ne peut être que de la forme $C = \{j\}$ ou $C = J$, c'est-à-dire qu'on apprend des règles propres à chaque joueur et des règles indépendantes du joueur.

— **Approche ignorante** *Ign* où on décrit la paire (j, x) uniquement selon l'état x

Enfin, nous ajoutons le résultat de deux méthodes de référence *Random forest* et *Decision tree*, dans une version utilisant l'identité du joueur. La figure 2 représente les justesses obtenues avec les différentes approches.

On observe que les approches *Ign* ont une justesse n'excédant pas 0.63 lorsque le pourcentage des exemples de E d'exemples sur lesquels le modèle est construit augmente, montrant le désaccord entre joueurs représenté Figure 1. On constate aussi que l'approche *Dir* a une mauvaise performance par rapport aux approches *Ind* et *Par*, alors qu'elle utilise pleinement l'identité du joueur. De son coté, l'approche *Ind*, dont les règles sont bien moins expressives concernant l'identité du joueur, que celles l'approche *Dir*, a une bien meilleure justesse (≈ 0.76). Sa justesse est cependant moins bonne que celle de l'approche *Par* (≈ 0.78). La forêt aléatoire obtient un résultat supérieur (≈ 0.84). Enfin, malgré une justesse supérieure de l'arbre de décision en augmentant le nombre d'exemples d'apprentissage, l'approche *Par* présente une meilleure justesse lorsque le nombre d'exemples est inférieur à 10%. On remarque :

1. Le bon résultat de la forêt aléatoire dans sa version informée doit être relié à sa nature ensembliste : la forêt est constituée d'une centaine d'arbres, rendant l'interprétation du modèle difficile.
2. Le résultat de l'arbre de décision est légèrement meilleur que l'approche *Par*, mais la justesse moyenne de l'approche *Par* reste dans l'écart-type de l'arbre de décision. De plus, l'écart-type de l'approche *Par* est plus faible que celui de l'arbre de décision, suggérant que sa performance est moins sujette à des variations importantes de l'ensemble d'apprentissage.
3. Le mauvais résultat de l'approche *Ind*, aussi expressive que l'approche *Par*, est lié à la très grande taille de l'espace de règles, la recherche d'une meilleure règle allant alors vers de mauvais optimums locaux. Dans l'approche *Dir* la contrainte sur le joueur ne peut être que nulle ou l'identité exacte du joueur, et l'algorithme de recherche d'une meilleure règle explore un espace

plus petit et se dirige mieux.

Complexité syntaxique des solutions La Figure 3 montre la complexité mesurée des solutions produites par les différentes approches à base de règles en fonction du nombre d'exemples pour 10 joueurs, de 5% des exemples ($n = 363$) à 45% des exemples ($n = 3268$). La complexité des solutions des différentes approches est mesurée en nombre de règles du modèle appris, on précise que dans ce cas, on considère qu'un exemple non-couvert par la solution est une règle.
Dans le cas des forêts aléatoires, sa complexité est mesurée en nombre de feuilles moyen par arbre de la forêt (équivalent à un nombre de règles). La forêt étant constituée de 100 arbres, le modèle est cependant bien plus complexe, avec en moyenne 1848 feuilles par arbre. Pour l'arbre de décision, le nombre de feuilles est égal au nombre de règles des solutions *Par* (si l'on ne considère pas les exemples non couverts comme des règles), mais la taille moyenne des règles des solutions *Par* est bien inférieure à la taille moyenne des branches de l'arbre de décision. Pour 11% des exemples en apprentissage, la taille moyenne des règles *Par* est de 4.2 atomes tandis que la taille moyenne des branches est de 13.4 tests.
On remarque que les approches *Dir* et *Par* fournissent des solution plus simples que l'approche *Ind*, avec une pente de la courbe de la complexité supérieure pour l'approche *Ind* que les deux autres approches. Cela vient du fait que les règles de la solution *Ind* ne peuvent concerner qu'un seul joueur, faisant ainsi augmenter le nombre de règles redondantes dont la seule différence est le joueur associé.

5.2 Comparaison des approches en fonction du nombre de joueurs

Une des motivations de l'apprentissage multijoueurs parcimonieux est le traitement de problèmes dans lesquels on a un grand nombre de joueurs. Pour étudier le comportement des approches *Dir* et *Par* en fonction du nombre de joueurs nous avons artificiellement altéré le jeu de données de la manière suivante : nous définissons pour chaque joueur j du jeu de données original un nouvel ensemble de k joueurs et chaque exemple de E_j, son étiquette $+$ ou $-$ incluse, est attribué à tous ces ces nouveaux joueurs. Ainsi, on associe à chaque joueur original un groupe de joueurs se comportant de la même manière que le joueur original. Pour $k = 2$ on a ainsi un nouveau jeu de données où les exemples sont

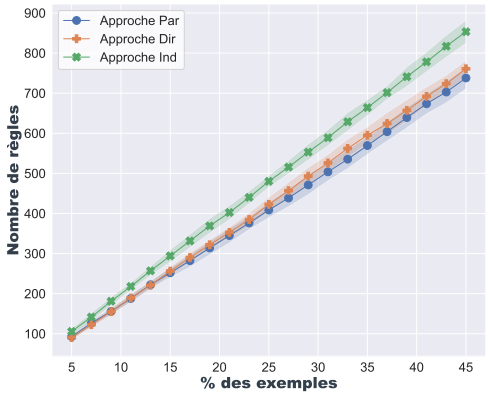


FIGURE 3 – Nombre de règles dans les solutions fournies par les différentes approches en fonction du nombre d'exemples d'apprentissage.

les exemples originaux dupliqués, doublant ainsi la taille de l'ensemble des exemples.

Justesse et temps CPU. Pour un ensemble d'apprentissage original de taille $n = 1380$ (19%) nous avons comparé les approches *Dir* et *Par* et montré les résultats dans la table 2. Ce nombre d'exemples a été choisi pour que le temps CPU de l'approche *Dir* reste raisonnable.
L'approche *Par* montre une décroissance négligeable de la justesse en fonction du nombre de joueurs et pour un nombre croissant d'exemples d'apprentissage. C'est ce à quoi on peut s'attendre en raison du fait que les exemples sont dupliqués, et qu'une règle valide et porteuse pour un joueur artificiel j' issu d'un joueur original j le sera également pour tous les joueurs artificiels issus de j. On remarque en revanche une baisse plus substantielle de la justesse de l'approche *Dir*, en particulier lorsqu'on introduit uniquement un joueur artificiel (ligne 2 de la table 2), avec une baisse de plus de 4 points en pourcentage.
Concernant le temps CPU, comme attendu, on remarque une augmentation du temps d'apprentissage pour l'approche *Dir* supérieure à celle de l'approche *Par* en raison de l'augmentation de l'espace de recherche, contrairement à l'approche *Par* où le nombre de joueurs dans le jeu de données n'influe pas sur la taille de l'espace de recherche.

Complexité syntaxique. La figure 4 représente les complexités syntaxiques en nombre de règles des solutions obtenues avec les approches *Par* et *Ind* en fonction du nombre de joueurs, pour 19% des exemples ($n = 1380$). Pour l'approche *Ind*, on remarque une augmentation lineaire du nombre de règles dans la solution en fonction du nombre de

# de joueurs	Justesse Par	CPU Par (min)	Justesse Dir	CPU Dir (min)	N
10	0.80	15	0.68	125	1380
20	0.795	46	0.646	659	2760
30	0.79	116	0.652	2707	4140
40	0.799	236	0.667	8836	5520

TABLE 2 – Justesses moyennes et temps CPU des approches *Dir* et *Par* pour un nombre croissant de joueurs et pour 19% ($n = 1380$) exemples originaux

FIGURE 4 – Nombre de règles des solutions fournies par les approches *Par* et *Ind* en fonction du nombre de joueurs, pour 19% des exemples ($n = 1380$).

joueurs. Cela est dû au fait que chaque règle d'une solution de l'approche *Ind* ne peut concerner qu'un joueur unique ou bien tous les joueurs. Ainsi, chaque règle est dupliquée du nombre de joueurs artificiels créés rendant l'approche linéaire en fonction du nombre de joueurs.

L'approche *Par* montre son efficacité en terme de complexité des solutions ici, avec une baisse de la pente en fonction du nombre de joueurs. Chaque règle d'une solution de l'approche *Par* pouvant être valide et porteuse pour un sous ensemble de joueurs, on s'attend effectivement à une complexité moindre de cette approche par rapport à l'approche *Ind* en augmentant le nombre de joueurs.

5.3 Utilisation des résultats de l'apprentissage multijoueurs pour l'analyse des groupes de joueurs

Nous avons utilisé les résultats de l'approche *Par* afin de construire des groupes de joueurs se comportant de la même manière. Pour cela, nous avons utilisé l'algorithme de clustering K-means sur les matrices joueur-règle obtenues par l'approche *Par*. Pour construire cette matrice, nous avons récupéré l'ensemble des règles de la solution obtenue par l'approche *Par* et pour chaque joueur, nous avons construit un vecteur binaire indiquant si la règle est valide et porteuse pour le joueur ou non. Nous avons ensuite appliqué un K-means sur la matrice obtenue. Nous avons fait varier les nombres de clusters de 2 à 9. Nous pouvons alors analyser les groupes de joueurs obtenus pour affirmer la ressemblance des joueurs entre eux. Afin d'étudier l'effet du clustering, pour chaque nombre de clusters analysé, nous avons effectué un nouvel apprentissage avec les données où l'identité des joueurs a été remplacée par leur cluster, pour chaque nombre de clusters. Cette approche est nommée *ParClust*.

Afin de comparer l'approche *ParClust* avec une approche « témoin », nous avons réalisé un K-Means sur les données initiales, sans apprentissage multijoueurs. Pour cela, nous avons représenté chaque joueur par un vecteur de carcactéristiques correspondant à la concaténation du vecteur de la distribution sur les caractéristiques des exemples qu'il a étiqueté positivement et du vecteur de la distribution sur les caractéristiques des exemples qu'il a étiqueté négativement. Nous avons ensuite appliqué un K-means sur la matrice obtenue. La figure 5 montre la justesse des modèles appris avec remplacement de l'identité des joueurs par leur cluster dans les données avec 45% des exemples. La ligne rouge indique la justesse de l'approche *Par* avec 10 clusters correspondant aux joueurs originaux. On remarque que la justesse de l'approche *ParClust* est la même que l'approche *Par* pour un nombre de clusters supérieur ou égal à 4. On peut alors considérer que si l'on souhaite réduire le nombre de joueurs, ou établir des clusters de joueurs se comportant de la même manière sans perte de justesse, alors le nombre de cluster donnant un regroupement optimal des joueurs est 4. On remarque également que la justesse de l'approche *ParClust* est supérieure à celle de l'approche témoin pour un nombre de clusters inférieur à 8. Cela indique que le clustering à partir des résultats de l'apprentissage multijoueurs est plus pertinent que le clustering à partir des données initiales.

FIGURE 5 – Justesse des modèles appris avec remplacement de l'identité des joueurs par leur cluster dans les données pour 45% des exemples ($n = 3268$).

6 Travaux associés

Certains travaux ont abordé le problème de l'annotation par différents experts lorsqu'il y a un problème de désaccord entre les étiquettes attribuées par chacun des experts. Dans [5], les auteurs présentent un cadre d'apprentissage où le désaccord entre les experts est modélisé et utilisé de manière à régulariser le classifieur, dans [6] les auteurs présente une mesure de l'expertise des différents annotateurs de manière à utiliser cette information pour déterminer la vraie étiquette d'une observation. Les auteurs de [3] présentent une étude comparative des différentes méthodes de calcul d'un consensus sur l'annotation des données par différents experts. Mais ces méthodes s'appuient sur le fait que les annotateurs étiquettent les mêmes données, et construisent un modèle général où l'identité de l'annotateur n'est pas prise en compte dans la décision face à une nouvelle observation. A notre connaissance, aucune étude sur l'utilisation implicite de l'identité de l'annotateur dans le cadre simple de l'apprentissage supervisé de concept n'a été effectuée. Enfin, l'idée d'utiliser séparément l'identité de l'annotateur et l'objet annoté était déjà présente des des travaux sur l'apprentissage abstrait [4].

7 Conclusion

Dans cet article, nous avons proposé une approche dans le cadre d'apprentissage supervisé de concept dans lequel les données peuvent être étiquetées de différentes manières par différents experts. Nous avons proposé ici l'approche parcimonieuse, qui prend en compte l'identité de l'annotateur dans la recherche d'hypothèses sans qu'elle soit incluse dans l'espace de recherche. L'approche parcimonieuse construit des règles qui sont valides et porteuses pour un sous-ensemble des annotateurs, et cette information est utilisée par la suite dans la prédiction sur de nouvelles données. Pour illustrer notre approche, nous nous sommes intéressés à un problème d'ouverture au Bridge dans lequel les décision sur les situations peuvent varier fortement chez les experts du jeu. Nous avons montré que notre approche avait de meilleures performances que l'approche classique (*Dir*) grâce à la réduction de l'espace que notre approche explore et produisait des solutions moins complexes. Nous avons ensuite montré la pertinence des groupes de joueurs formés par les règles apprises par notre approche en appliquant un clustering entre les joueurs à partir des règles apprises, puis en remplaçant l'identité du joueur par le cluster auquel il appartient dans les données. Nous avons remarqué une réduction significative du nombre de joueurs sans perte de performance.

Références

[1] Mitchell, T.M. : Generalization as search. Artif. Intell. **18** (1982)

[2] Muggleton, S.H. : Inverse entailment and progol. New Gener. Comput. **13** (1995)

[3] Sheshadri, A., Lease, M. : SQUARE : A benchmark for research on computing crowd consensus. In : Hartman, B., Horvitz, E. (eds.) Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2013, November 7-9, 2013, Palm Springs, CA, USA. AAAI (2013)

[4] Soldano, H. : A modal view on abstract learning and reasoning. In : Genesereth, M.R., Revesz, P.Z. (eds.) Proceedings of the Ninth Symposium on Abstraction, Reformulation, and Approximation, SARA 2011. AAAI (2011)

[5] Wang, C., Gao, Y., Fan, C., Hu, J., Lam, T.L., Lane, N.D., Bianchi-Berthouze, N. : Learn2agree : Fitting with multiple annotators without objective ground truth. In : Chen, H., Luo, L. (eds.) Trustworthy Machine Learning for Healthcare - First International Workshop, TML4H 2023, Proceedings. Lecture Notes in Computer Science, vol. 13932. Springer (2023)

[6] Yan, Y., Rosales, R., Fung, G., Ramanathan, S., Dy, J.G. : Learning from multiple annotators with varying expertise. Mach. Learn. **95** (2014)

Session 4 : Adoption de l'apprentissage automatique par les usagers (commune avec CNIA)

Test à base de scénarios de programmes apprenant en ligne

Maxence Demougeot[1], Sylvie Trouilhet[1], Jean-Paul Arcangeli[1], Françoise Adreit[2]

[1] IRIT, Université de Toulouse, UT3, Toulouse, France
[2] IRIT, Université de Toulouse, UT2J, Toulouse, France

{Prénom.Nom}@irit.fr

Résumé

En apprentissage automatique, un modèle est un programme qui fait des prédictions ou prend des décisions. Il ne résulte pas d'une activité de programmation traditionnelle mais d'une construction automatique par un programme apprenant alimenté par des exemples. Comme tout programme, les programmes apprenants doivent être vérifiés et validés. Le test est un moyen d'y parvenir et d'assurer une certaine confiance dans une solution d'apprentissage automatique. Dans cet article, nous proposons une analyse du problème du test logiciel dans le contexte de l'apprentissage automatique, et nous approfondissons l'état de l'art en nous concentrant sur le programme apprenant et en mettant l'accent sur l'apprentissage en ligne et interactif. Face au non-déterminisme possible, à la dynamique de l'apprentissage en ligne et à la présence de l'utilisateur dans l'apprentissage interactif, nous proposons une approche à base de scénarios pour le test de programmes apprenants. En outre, nous présentons deux outils prototypes qui permettent l'implantation et l'exécution de scénarios pour évaluer un programme apprenant en interaction avec l'utilisateur.

Mots-clés

Ingénierie des systèmes logiciels à base d'apprentissage automatique, évaluation de programmes apprenants, test, scénario, apprentissage en ligne, apprentissage interactif.

Abstract

A machine learning (ML) model serves as a program for making predictions or decisions. It is not built by traditional programming but automatically by a learning program fed by examples. As any program, learning programs need to be verified and validated. Testing is a way of doing this and providing trust in ML. In this paper, we provide an analysis of the testing problem in ML context, and go deeper into the state-of-the-art focusing on the learning program with particular emphasis on online and interactive ML (IML). In front of the possible non-determinism, the dynamics of online learning, and the presence of the user in IML, we propose a scenario-based approach to test learning programs. In addition, we present a prototype tooling that supports scenario implementation and running for evaluation purposes of a learning program that interacts with the user.

Keywords

Engineering Software Systems based on Machine Learning, Evaluation of Learning Programs, Testing, Scenario, Online Learning, Interactive Learning.

1 Introduction

Pour qu'une machine effectue une tâche, son comportement doit être programmé. Cependant, il n'est pas toujours possible ou envisageable d'écrire un programme, par exemple si l'algorithme est inconnu ou trop complexe à mettre en œuvre. Dans de tels cas, les équipes de développement peuvent opter pour des logiciels basés sur l'apprentissage automatique (*machine learning* ou ML) [17] qui visent à déduire et généraliser le comportement attendu de la machine à partir d'exemples. Comme les logiciels « traditionnels » (ceux qui ne sont pas basés sur l'apprentissage automatique), la qualité de ce type de logiciel doit être évaluée. Cette évaluation peut se faire par le test, mais ceci reste un défi majeur. La recherche dans le domaine est récente. Les travaux ciblent majoritairement l'apprentissage supervisé hors ligne et n'englobent pas tous les paradigmes d'apprentissage automatique, en particulier l'apprentissage en ligne [24] ou l'apprentissage interactif [9].

Dans cet article, nous prenons un point de vue « ingénierie logicielle » : l'objectif de notre travail est d'**accompagner le développement de solutions à base d'apprentissage automatique** au moyen de méthodes et d'outils de test qui participent à la réalisation de produits fiables qui répondent aux besoins. Le processus de développement et de production des logiciels basés sur l'apprentissage automatique est d'abord comparé à celui des logiciels traditionnels afin de mettre en évidence les défis posés par le test, en particulier lorsque l'apprentissage est fait en ligne ou lorsque l'utilisateur humain est au centre du processus. **Nous ciblons ensuite le programme apprenant**, et nous analysons la problématique du test dans le cadre de l'apprentissage en ligne et de l'apprentissage interactif. Notre approche pour la **conception**, l'**implantation** et l'**exécution** des cas de test est basée sur la notion de scénario. Nous avons conçu deux outils pour implanter et exécuter des scénarios afin d'évaluer un programme apprenant qui interagit avec l'utilisateur pour construire des applications « ambiantes ».

Ces travaux font partie du projet de recherche OppoCompo dont l'objectif est de développer une solution à base d'ap-

prentissage automatique qui construit à la volée des applications en environnement ambiant (Internet des Objets, Ville Intelligente…), avec l'utilisateur dans la boucle. Nous développons un prototype de « moteur de composition opportuniste » appelé OCE qui apprend en ligne par renforcement [27] et de manière interactive [9], les besoins et les préférences de l'utilisateur, afin de proposer des applications pertinentes en fonction du contexte [30]. La pertinence de ses résultats, l'adaptation à l'environnement ambiant et à l'utilisateur sont des qualités que nous voulons évaluer afin de vérifier que notre solution apprenante est fonctionnelle et digne de confiance.

Cet article est organisé de la manière suivante. La section 2 compare les processus de développement des logiciels traditionnels et des logiciels à base d'apprentissage automatique. La section 3 différencie le test de modèles et le test de programmes apprenants. Puis elle présente les principales difficultés du test d'un programme apprenant en ligne et avec l'utilisateur dans la boucle, en s'appuyant sur les principaux travaux du domaine. La section 4 propose le concept de scénario de test, puis présente deux outils pour implanter (Maker) et exécuter (Runner) des scénarios à des fins d'évaluation ainsi que leur intégration dans un système logiciel qui apprend en ligne en interaction avec l'utilisateur. La section 5 présente comment d'autres travaux utilisent la notion de scénario pour tester des logiciels basés sur l'apprentissage automatique et fait état de quelques outils de développement de solutions d'apprentissage par renforcement. Enfin, la section 6 résume notre contribution et ses limites, et discute quelques perspectives.

2 Développement de logiciels à base d'apprentissage automatique

Nous analysons la manière dont les logiciels basés sur l'apprentissage automatique sont développés et mis en production par rapport aux logiciels traditionnels, en se focalisant sur l'apprentissage en ligne et l'apprentissage interactif.

La figure 1 présente une vue synthétique du processus classique de développement et de production de logiciels traditionnels : à partir des exigences, l'équipe de développement met en œuvre plus ou moins manuellement une solution sous la forme d'un programme, puis le programme est exécuté et produit des résultats à partir des entrées.

La figure 2 présente une vue synthétique du processus dans le contexte de l'apprentissage hors ligne. À l'équipe de développement s'ajoutent les experts en apprentissage automatique, les experts du domaine et potentiellement les futurs utilisateurs ou leurs représentants qui collaborent pour concevoir ou sélectionner puis paramétrer le programme apprenant et définir les données d'entraînement. À l'instar du développement logiciel traditionnel, il s'agit de construire un programme, appelé **modèle**. Contrairement aux logiciels traditionnels, le modèle est construit automatiquement au cours d'une phase d'apprentissage (également appelée phase d'entraînement) à l'aide d'un **programme apprenant** alimenté par des données d'apprentissage. Les données d'apprentissage sont généralement des couples composés d'une entrée et de la sortie associée, l'ensemble des données devant être complet, cohérent et représentatif de ce que le logiciel doit apprendre [26]. Comme pour les logiciels traditionnels, une fois construit et validé, le modèle est mis en production.

2.1 Apprentissage en ligne

Il est parfois difficile d'anticiper les données que le modèle rencontrera en production, et le modèle doit s'adapter au fil du temps en fonction des données reçues. L'apprentissage automatique en ligne cible cette question. Selon S. Russel et P. Norvig [24], cette approche repose sur des comparaisons répétées entre les résultats produits et les résultats attendus, avec, éventuellement, une phase préalable d'apprentissage : lorsque le modèle produit une sortie pour une entrée donnée (décision), un expert du domaine lui fournit la bonne réponse (la sortie attendue) afin de déclencher une nouvelle phase d'apprentissage, comme le montre la figure 3. Ainsi, il y a une alternance continue entre phase d'apprentissage et phase de production ; le programme apprenant met à jour le modèle de manière itérative tout au long de son exécution.

Contrairement au cas de l'apprentissage hors ligne, le programme apprenant opère donc quand le modèle est en production pour l'adapter et le faire évoluer. Dans un contexte ouvert et possiblement non prévisible, il doit traiter des données d'apprentissage qui arrivent au fur et à mesure de l'exécution, qui peuvent ne pas être de bonne qualité mais dont les éventuels défauts ne peuvent pas être corrigés.

2.2 Apprentissage interactif

L'apprentissage automatique interactif [9] place l'utilisateur humain au centre du processus d'apprentissage [2]. Il est fréquemment combiné avec l'apprentissage en ligne. Dans ce cas, les experts en apprentissage automatique ne sont pas partie prenante dans le processus. Ce sont les utilisateurs finaux qui alimentent le programme apprenant afin de personnaliser le modèle créé pour leur propre usage, même s'ils n'ont généralement pas de compétences en apprentissage automatique. En pratique, les utilisateurs peuvent fournir un retour après chaque décision ou prédiction du modèle, ce qui permet à ce dernier de s'adapter à leurs besoins et à leurs préférences au fur et à mesure de son exécution, afin d'améliorer les prédictions futures. En contrepartie, l'apprentissage automatique interactif est sujet à la versatilité de l'utilisateur humain.

3 Test de logiciels à base d'apprentissage automatique

3.1 Principes de base du test de logiciels

Le test est une activité au sein du processus de développement des logiciels. Il permet principalement de : *(1)* mettre en évidence des situations où le programme ne se comporte pas comme attendu, afin de pouvoir apporter une correction, *(2)* montrer que le programme répond aux exigences, pour convaincre les parties prenantes que le programme fonctionne, *(3)* mettre au point le paramétrage, et *(4)* comparer différentes versions d'un même programme [3]. Les tests

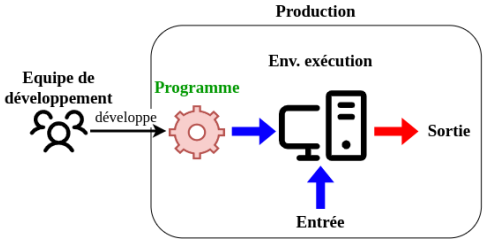


FIGURE 1 – Processus traditionnel de développement et de production de logiciels

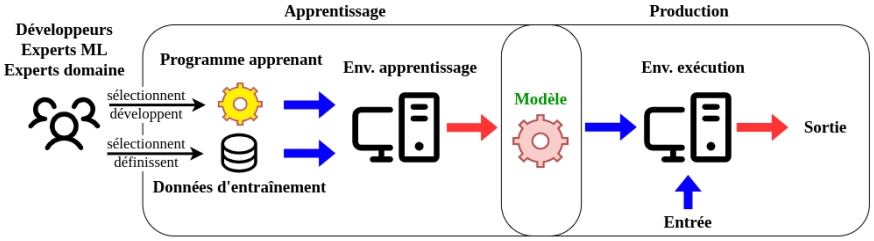


FIGURE 2 – Développement et production dans des environnements d'apprentissage hors ligne

consistent généralement à exécuter le programme dans un environnement proche de l'environnement de production, puis à analyser les résultats. Dans une équipe de développement, « l'oracle » évalue et interprète ces résultats. Si ces résultats montrent que le programme ne se comporte pas comme attendu, l'équipe de développement recherche la présence d'un défaut, reprend le développement pour le corriger, puis effectue un nouvelle série de tests. Un défaut (ou bug) est défini comme une imperfection ou une déficience dans un produit logiciel qui ne répond pas aux exigences ou aux spécifications [11]. La notion de défaut et le problème de leur détection sont analysés dans [3].

3.2 Analyse de la problématique

Contrairement aux travaux menés dans le domaine du test de logiciels traditionnels, la recherche sur le test de logiciels basés sur l'apprentissage automatique est récente. Plusieurs articles traitent du problème général du développement de logiciels basés sur l'apprentissage automatique avec un point de vue ingénierie logicielle, mais ne fournissent que de brèves explications sur les questions du test [10, 16].

Cependant, quelques articles importants ont été publiés ces dernières années. Zhang et al. [31] ont proposé une étude complète du test de logiciels basés sur l'apprentissage automatique, en mettant l'accent sur l'évaluation du modèle et de propriétés telles que l'exactitude, la robustesse, l'équité. Riccio et al. [23] ont analysé la littérature de manière systématique et mis en évidence les principaux défis liés au test, notamment la spécification des cas de test, les critères d'adéquation, le coût et le problème de l'oracle.

Pour analyser la problématique du test de logiciel à base d'apprentissage automatique, nous soulignons la différence entre le test de modèles et le test de programmes apprenants. Puis nous nous concentrons sur le test de l'apprentissage en ligne et de l'apprentissage interactif.

3.2.1 Test de modèles *vs* test de programmes apprenants

L'apprentissage automatique recouvre des activités d'apprentissage (construction de modèle) et de décision (exploitation du modèle), que ces activités soient entrelacées ou non. En matière de test, il faut donc distinguer deux volets : le test du modèle et le test du programme apprenant.

Tester le modèle consiste à l'exécuter pour répondre à la question suivante : le modèle est-il un « bon » modèle ? En d'autres termes, la machine a-t-elle « bien » appris ? Le test de modèles a les mêmes objectifs que le test de logiciels traditionnels mais en mettant l'accent sur des propriétés de qualité du modèle telles que l'exactitude, la pertinence ou la robustesse [31]. Cela pose plusieurs problèmes que nous examinons ci-dessous.

Comme les modèles résultent de l'exécution d'un programme apprenant alimenté par des exemples, les défauts peuvent provenir du programme apprenant, des données d'apprentissage ou d'une inadéquation entre les deux (lorsqu'un programme apprend mal sur certaines données) [31]. En pratique, mettre l'accent sur des propriétés peut aider à localiser les sources des défauts : par exemple, tester la pertinence du modèle (comme le sur-apprentissage [12]) permet de trouver des défauts dans les données d'apprentissage.

Il peut être difficile de remonter à la source d'un défaut afin de le corriger. En effet, les modèles n'ont pas la même matérialité que les logiciels traditionnels : ils ne consistent pas en un code source mais fonctionnent le plus souvent en « boîte noire » et sont composés de divers éléments plus ou moins tangibles (code, paramètres, données).

D'autre part, comme c'est aussi traditionnellement le cas, il est essentiel pour l'évaluation des modèles de sélectionner de manière appropriée les données utilisées pour le test.

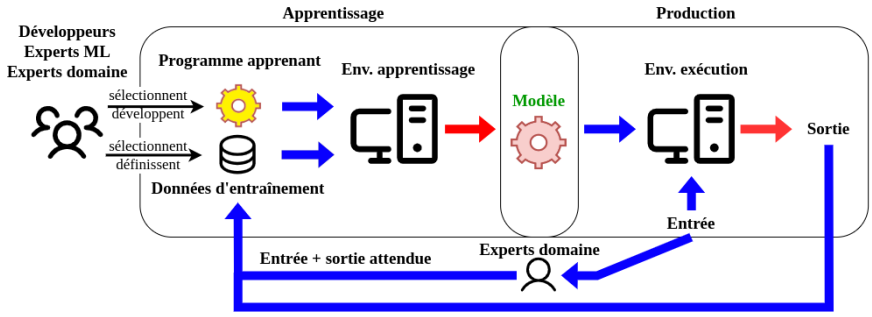


FIGURE 3 – Développement et production dans des environnements d'apprentissage en ligne

Mais, ici, l'espace est potentiellement complexe et infini.
Enfin, l'apprentissage automatique est parfois utilisé par les équipes de développement dans des situations où les résultats attendus ne sont pas connus à l'avance [18]. Dans ce cas, prédire et interpréter les résultats des tests est un défi supplémentaire, et il peut être difficile de déterminer si un test passe ou non. Les logiciels présentant ce problème, appelé le problème de l'oracle, sont souvent considérés comme non testables [29] en raison de l'absence d'oracle ou de la difficulté d'en concevoir [19].

Tester le programme apprenant n'est pas la même chose que tester un modèle, bien que les deux problèmes soient étroitement liés. La question est la suivante : le programme apprenant apprend-il « bien » dans l'ensemble du champ d'application pour lequel il a été conçu ? En d'autres termes, le programme apprenant construit-il de « bons » modèles relativement aux données qui lui sont fournies ? Pour répondre à cette question, les testeurs doivent faire en sorte que le programme apprenant construise différents modèles pour différents cas d'utilisation, dans le but de tester ensuite ces modèles. La sélection des données d'apprentissage pour construire des modèles est essentielle pour la qualité de l'ensemble des tests. En outre, comme il n'est pas possible d'exprimer les modèles attendus pour faire des comparaisons (le problème de l'oracle à nouveau), chaque modèle construit doit être exécuté pour évaluer la qualité du programme apprenant. Le test de programmes apprenants est par conséquent coûteux ; il demande une forte expertise et, autant que possible, une automatisation.
C'est le problème du test de programmes apprenants que nous ciblons ici.

Test de programmes apprenants *vs* test de compilateurs. Programmes apprenants et compilateurs sont des programmes dont l'exécution produit des programmes. Ainsi, le test de programmes apprenants présente des similitudes avec le test de compilateurs. Nous développons ici cette analogie pour aider à mieux distinguer test du modèle et test du programme apprenant.
La figure 1 cache une étape importante du développement de logiciels : la compilation. Dans le cas général, le programme mis en production est produit par un compilateur à partir d'un programme source, comme un modèle est produit à partir d'un programme apprenant.
Dans [7], les auteurs passent en revue les travaux portant sur le test de compilateurs, en mettant l'accent sur la propriété de correction, c'est-à-dire la conformité sémantique entre le programme source et le programme exécutable produit par le compilateur. Comme les programmes apprenants, les compilateurs sont des logiciels complexes dotés de multiples fonctions et paramètres. Ils manquent de spécifications précises, ce qui rend leur vérification difficile. Entre autres, le problème de l'oracle est aussi présent car il est tout à fait impossible pour les testeurs de fournir le résultat attendu (programme exécutable) à comparer avec le résultat obtenu pendant le test. Par conséquent, comme pour un programme apprenant, un compilateur ne peut être évalué qu'indirectement en testant ce qu'il produit (le programme exécutable).
De plus, pour vérifier qu'il produit systématiquement le bon exécutable, l'ensemble des entrées utilisées pour les tests (les programmes sources) doit couvrir le plus possible les différents types de programmes qui seront à compiler.

3.2.2 Test de programmes apprenant en ligne

Dans le cas de l'apprentissage en ligne, le test pose des problèmes particuliers que nous examinons ici.
De manière générale, la qualité d'un modèle dépend de la qualité des données d'apprentissage. Comme celle-ci ne peut pas être contrôlée dans le cadre de l'apprentissage en ligne, la question du test ne porte plus vraiment sur la présence d'un défaut dans les données d'apprentissage. Il s'agit plutôt de vérifier la cohérence entre les données d'apprentissage et le modèle construit. On peut aussi chercher à vérifier la robustesse, c'est-à-dire la capacité de la solution à opérer convenablement lorsque les données d'apprentissage sont de mauvaise qualité.
Pour tester, il faut imaginer un ensemble de cas que la solution pourra rencontrer en production. Or, il n'est pas toujours possible de savoir à l'avance à quelles données le programme apprenant et le modèle seront confrontés. Il est donc difficile de définir *a priori* des cas de tests représentatifs des futures données réelles et, pour le testeur, d'anticiper certains cas qui pourraient se présenter.
D'autre part, les cas de test doivent non seulement permettre l'évaluation à proprement parler mais aussi construire préalablement le modèle, ce qui complexifie à la fois la conception et l'exécution du test.
De plus, les modèles construits par apprentissage en ligne

sont soumis de manière continue à des changements (si l'apprentissage ne converge pas ou si l'environnement n'est pas stationnaire). Autrement dit, en production, ils évoluent. Les tester à un certain stade peut n'avoir que peu voire pas de sens, car ils peuvent devenir rapidement obsolètes. Parce que le processus d'apprentissage en ligne est itératif, le choix du « bon » moment pour tester se pose, et il est difficile de décider quand un modèle est suffisamment mature pour être testé.

Un autre problème réside dans la nature possiblement non-déterministe de l'apprentissage automatique [15, 26], notamment (mais pas seulement) dans le cas de l'apprentissage en ligne. La part d'aléatoire présente dans les mécanismes d'apprentissage et de décision conduit à des résultats variables d'une exécution à l'autre, avec la même configuration et les mêmes entrées. Ainsi, il peut arriver que le modèle ne produise pas les résultats attendus lors des tests alors que le mécanisme d'apprentissage fonctionne correctement [14]. Par exemple, dans l'apprentissage par renforcement (généralement effectué en ligne), il est normal que la machine choisisse parfois, à des fins d'exploration, une solution qui n'est ni la meilleure ni celle logiquement attendue. Il est donc difficile de déterminer si une sortie inattendue résulte d'un facteur aléatoire ou d'un défaut dans le mécanisme d'apprentissage.

3.2.3 Test de programmes apprenants en interaction avec l'humain

La présence d'un utilisateur humain dans la boucle complexifie la conception et la réalisation des tests.

D'une part, les résultats produits par un modèle peuvent convenir à un utilisateur mais pas à un autre. D'autre part, les besoins, les attentes ou les préférences des utilisateurs peuvent varier dans le temps et en fonction de leur situation. Il est donc difficile de créer des cas de test qui englobent un large éventail de profils d'utilisateurs tout en anticipant leur dynamique ou leurs possibles incohérences.

Un autre problème réside dans la qualité des données fournies par l'utilisateur lorsqu'il interagit avec le modèle. Par exemple, il peut mal comprendre les résultats du modèle et donner un retour non pertinent. Dans ce cas, le modèle risque de produire des résultats de test incorrects alors que le mécanisme d'apprentissage fonctionne correctement. Par conséquent, il est difficile de déterminer si la machine n'a pas réussi à apprendre ou si le problème provient des interactions entre l'utilisateur et la machine apprenante.

3.2.4 Synthèse

Nous formulons ci-dessous les questions de recherche que nous avons identifiées concernant le test de programmes apprenant de manière automatique. Les 3 premières sont relatives à la conception et à l'implantation des cas de tests, les deux autres à leur exécution et à l'analyse :

- **QR1.1** : Comment concevoir un cas de test incluant des temps d'apprentissage et des temps d'évaluation, en intégrant l'interaction avec l'humain ?
- **QR1.2** : Comment concevoir un ensemble de cas de test suffisamment couvrant du champ d'application du programme apprenant tout en contrôlant la taille de cet ensemble ?
- **QR1.3** : Comment implanter un cas de test dans le but d'automatiser son exécution ?
- **QR2.1** : Comment automatiser l'exécution des tests en prenant en compte le non-déterminisme ?
- **QR2.2** : Comment mesurer la qualité des résultats obtenus ?

Notre contribution est centrée algorithmique au sens de [4] plutôt qu'humain : ce n'est pas l'utilisabilité et l'expérience utilisateur que nous cherchons à évaluer mais la qualité des modèles produits en fonction des données d'apprentissage. Nous ciblons dans cet article les questions **QR1.1**, **QR1.3**, **QR2.1** et **QR2.2**, à savoir la conception, l'implantation et l'exécution des cas de test. D'autres approches apportent des réponses à nos questions ; par exemple, le test métamorphique [8] s'intéresse à la vérification du comportement du modèle en l'absence d'oracle et cible la question **QR2.2**.

3.3 Test d'OCE, un programme apprenant en ligne avec l'utilisateur dans la boucle

Dans le cadre du projet de recherche OppoCompo, nous développons un « moteur de composition opportuniste » appelé OCE, qui apprend en ligne par renforcement à partir des retours de l'utilisateur. Dans le contexte des systèmes ambiants qui sont ouverts et fortement dynamiques par nature, par exemple une ville ou un bâtiment intelligent, OCE détecte les composants logiciels [25] qui peuplent l'environnement ambiant de l'utilisateur. Il les assemble automatiquement pour faire émerger une application de l'environnement. Comme les applications ne sont pas spécifiées ou demandées au préalable, il n'est pas possible de les connaître à l'avance. Pour prendre une décision, OCE s'appuie sur un modèle construit par apprentissage automatique lors des étapes qui précèdent l'étape de décision : lors d'une étape (appelée cycle), OCE propose une application à l'utilisateur humain qui peut l'accepter, la modifier ou la rejeter. OCE apprend de ce retour et met à jour le modèle afin d'améliorer les propositions des cycles suivants [30].

OCE est donc le programme apprenant qu'il nous faut tester. Pour cela, il faut évaluer les différents modèles qu'il construit et qu'il met à jour d'un cycle à un autre : ces modèles sont-ils capables de proposer des applications pertinentes pour l'utilisateur, c'est-à-dire conformes à ses préférences habituelles ? Comme un modèle ne se prête pas à la comparaison avec un modèle attendu ou à l'analyse de code, il doit être exécuté pour vérifier les applications qu'il fait émerger. Notons qu'il ne s'agit pas d'évaluer la qualité brute de ces applications (fonctionnalités, performance, sécurité, etc.), mais le fait qu'elles satisfont les attentes de l'utilisateur.

Grâce au test, nous cherchons à trouver et à corriger des défauts, à comparer différentes versions ou paramétrages, et finalement à instaurer de la confiance dans OCE. Mais, à travers le cas d'étude qu'est OCE, nous cherchons plus généralement à faire progresser l'état de l'art en matière de test de solutions apprenantes en ligne en interaction avec l'utilisateur humain.

Dans les sections suivantes, nous présentons une solution

pour concevoir, implanter et exécuter des cas de test.

4 Une approche basée sur les scénarios pour évaluer OCE

Cette section examine d'abord ce que sont les cas de test et leur structure. Elle présente ensuite deux outils, Maker et Runner, qui permettent de les implanter et de les exécuter automatiquement en interaction avec OCE.

4.1 Scénarios de test

Dans un contexte itératif tel que l'apprentissage en ligne, la conception d'un cas de test demande la définition d'une séquence d'interactions entre le programme apprenant et son environnement d'apprentissage (l'environnement ambiant et l'utilisateur dans notre cas). L'apprentissage en ligne nécessite un certain nombre d'interactions pour apprendre, donc pour dériver un modèle. Pour vérifier que le modèle se comporte comme attendu, d'autres interactions sont nécessaires. Nous appelons « scénario de test » cette séquence d'interactions dédiées à l'apprentissage et à l'évaluation [1]. Les interactions liées à l'apprentissage et à l'évaluation peuvent être imbriquées. Toutefois, dans cet article et à des fins de simplification, nous ne considérons qu'une phase d'apprentissage et une phase d'évaluation exécutées l'une après l'autre.

La **phase d'apprentissage** implique une séquence de cycles avec, pour chacun, une proposition d'OCE, un retour de l'utilisateur puis un apprentissage. Pour définir la phase d'apprentissage, chaque cycle doit être spécifié par : *(i)* la liste des composants logiciels peuplant l'environnement ambiant (variable d'un cycle à l'autre) avec leurs interfaces pour les assembler, *(ii)* pour éviter que le testeur n'ait à interagir en permanence avec OCE lors des tests, ce qui serait trop fastidieux et coûteux, l'« assemblage idéal » spécifie le résultat (la sortie) que l'utilisateur attendrait dans ce cas. À partir de l'assemblage des composants proposé par OCE et de l'assemblage idéal, OCE génère un retour et apprend, tout comme il le fait dans un fonctionnement normal. Au terme de cette phase, OCE a construit un modèle adapté aux configurations de l'environnement (y compris les retours de l'utilisateur). Le modèle est ensuite à évaluer. Pour simplifier les tests et l'analyse des résultats, la **phase d'évaluation** peut être réduite à un seul cycle, mais ce n'est pas obligatoire. Un cycle d'évaluation est défini par : *(i)* l'environnement ambiant comme dans la phase d'apprentissage et *(ii)* la sortie attendue appelée « assemblage attendu ». Ainsi, en donnant la sortie attendue, le concepteur du test se comporte comme un oracle. Des mesures de distance entre la sortie proposée par OCE et la sortie attendue sont calculées pour évaluer la pertinence (exactitude au sens de Zhang *et al.* [31]) de la décision d'OCE.

Cette approche à base de scénarios répond à la question de recherche **QR1.1**. De cette manière, il est possible de définir et de tester différents environnements ambiants et leur dynamique, ainsi que différents profils d'utilisateurs. Une fois implantés, ces scénarios sont destinés à être exécutés automatiquement. Il convient de noter que les problèmes liés à l'identification de scénarios de test représentatifs et à l'implication de l'utilisateur dans leur définition dépassent le cadre de cet article.

4.1.1 Exemple

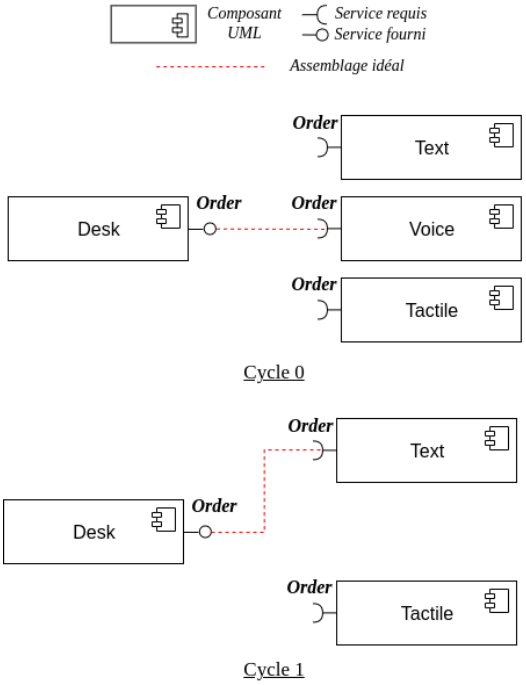


FIGURE 4 – Représentation UML [21] des cycles d'apprentissage du scénario exemple

Pour illustrer ce qu'est un scénario, prenons un exemple, tiré de [30] et réduit à des fins de simplification. Mary est au travail. Dans son environnement ambiant, il y a un composant logiciel [21] **Desk** qui fournit un service de réservation de salle appelé *Order* et trois composants **Text**, **Voice** et **Tactile** qui permettent de faire une demande de réservation (avec différents modes d'interaction pour l'utilisateur) et requièrent le service *Order*. Via le service *Order*, ces trois composants peuvent être assemblés avec **Desk**. Trois applications sont donc possibles (**Text-Desk**, **Voice-Desk**, **Tactile-Desk**) permettant à Mary de réserver une salle de réunion. En tant que testeurs, nous souhaitons par exemple vérifier que si Mary exprime une préférence pour **Voice-Desk**, alors, lorsqu'une situation similaire se présentera, OCE proposera à nouveau **Voice-Desk**.

Imaginons un scénario simple à trois cycles dans lequel **Voice** disparaît puis réapparaît. Les cycles 0 et 1 (figure 4) définissent la phase d'apprentissage. Le cycle 0 est défini par la liste des 4 composants et l'assemblage idéal **Voice-Desk** (l'expression de la préférence de Mary). Pour le cycle 1, les composants sont **Desk**, **Text** et **Tactile**. L'assemblage idéal est alors **Text-Desk**. Pour la phase d'évaluation, le

1. Dans le domaine du test de logiciels, le terme « scénario de test » désigne habituellement l'organisation et planification du processus de test dans le cadre d'un projet de développement. Notre définition du terme scénario de test est différente : il s'agit d'un scénario d'utilisation destiné à être testé, constituant ainsi un cas de test.

cycle 2 est défini par la liste composée de **Desk**, **Text** et **Voice**, et l'assemblage attendu est **Voice-Desk**. Dans ce qui suit, nous présentons deux outils qui supportent la définition et l'exécution d'un tel scénario.

4.2 Un outillage pour le test d'OCE

Les outils que nous avons développés pour tester le programme apprenant OCE sont présentés dans cette section. Le code source est disponible [2], ainsi qu'une courte vidéo [3] complétant cette section et démontrant leur utilisation dans le scénario présenté ci-dessus.

4.2.1 Maker

Il s'agit d'un outil interactif pour implanter des scénarios qui répond à la question **QR1.3**. Pour un cycle donné, le testeur peut réutiliser ou définir des composants fictifs (figure 5, panneau de gauche), les glisser (*drag and drop*) dans un cadre qui définit l'environnement et lier les services afin de définir l'assemblage idéal ou attendu (figure 5, panneau de droite). Des fonctionnalités telles que la possibilité de dupliquer un cycle réduisent la charge de travail du testeur. À partir d'une séquence de cycles, Maker génère un fichier JSON implantant le scénario.

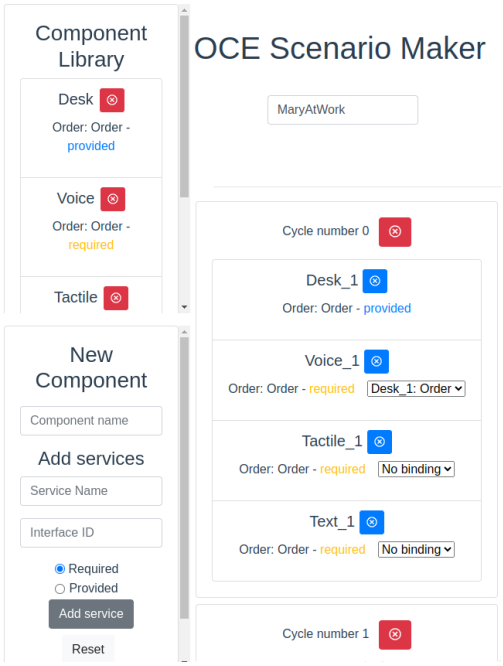


FIGURE 5 – Interface de Maker

4.2.2 Runner

Il s'agit d'une application Java qui, couplé avec OCE, permet l'exécution de scénarios au format JSON, comme ceux générés par Maker. Pour réduire l'impact de la nature non-déterministe de l'apprentissage automatique, l'exécution d'un scénario peut être répétée plusieurs fois et des valeurs moyennes des résultats mesurés sont calculées. Plusieurs paramètres doivent être définis, tels que les valeurs des paramètres d'apprentissage (par exemple, le taux d'exploration dans le cas de l'apprentissage par renforcement), la version d'OCE et le nombre de répétitions. Runner prend en paramètre une indication des cycles dédiés à l'apprentissage et des cycles dédiés à l'évaluation, cette indication définissant le moment des tests. Runner permet ainsi de répondre à la question **QR2.1**. Une fois les valeurs des paramètres définies, il fonctionne sans aucune autre intervention du testeur.

En réponse à la question **QR2.2**, pour évaluer la pertinence des applications proposées lors de l'exécution des cycles d'évaluation, Runner compare l'assemblage proposé par OCE et l'assemblage attendu en calculant un indice de similarité de Jaccard [4]. Il fournit un indice moyen sur l'ensemble des cycles d'évaluation d'un scénario. Cette mesure indique si les modèles construits ont fait des propositions pertinentes en fonction de ce qu'OCE a appris, c'est-à-dire en fonction des préférences de l'utilisateur dans différents environnements ambiants.

4.2.3 Architecture

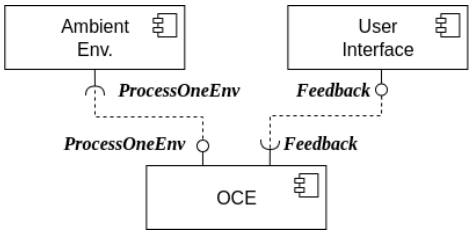


FIGURE 6 – Vue architecturale d'OCE en production

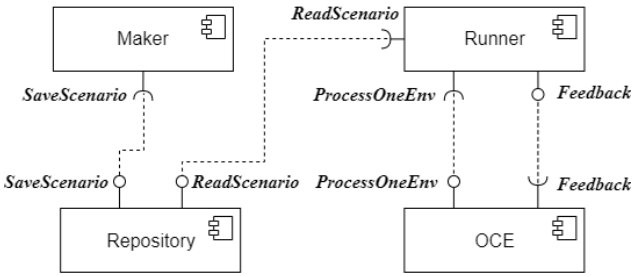


FIGURE 7 – Vue architecturale d'OCE en test

Les diagrammes de composants UML [21] (figures 6 et 7) montrent comment Maker et Runner ont été intégrés dans l'architecture du système logiciel OCE.

Dans la configuration de production (figure 6), lorsqu'il y a une variation de l'environnement ambiant, OCE reçoit du composant **Ambient Env.** la liste des composants présents dans l'environnement ambiant (service ***ProcessOneEnv***. L'assemblage construit par OCE est proposé à l'utilisateur qui fournit un retour sous la forme d'un assemblage (service ***Feedback***) via une interface graphique.

2. https://www.irit.fr/OppoCompo/resources/
3. https://www.irit.fr/OppoCompo/makerrunnerusecase2024/
4. https://en.wikipedia.org/wiki/Jaccard_index

L'architecture modulaire d'OCE permet de remplacer l'environnement ambiant et l'interface utilisateur par Runner pour effectuer les tests. Dans la configuration de test (figure 7), Runner exécute un scénario issu d'un répertoire de scénarios (service ***ReadScenario***). Le répertoire est alimenté par les scénarios implantés avec Maker (service ***SaveScenario***). Pour chaque cycle du scénario, Runner sollicite OCE pour obtenir une proposition d'assemblage (service ***ProcessOneEnv***). Suite à la proposition, qu'il évalue dans le cas des cycles d'évaluation, Runner fournit à OCE l'assemblage idéal ou attendu (service **Feedback**), que ce dernier traite comme un retour de l'utilisateur.

4.2.4 Expérimentation

Les outils Maker et Runner ont été utilisés lors d'une première campagne de test du comportement d'OCE. Nous avons défini une dizaine de scénarios visant à tester des cas de base, comme l'apprentissage d'un composant préféré, celui d'un composant à écarter ou la sensibilité à la nouveauté ; pour un même cas, nous avons fait varier la dynamique de l'environnement. Runner a été testé avec les scénarios créés avec Maker. Aucune défaillance de ces deux outils n'a été relevée. Ils ont simplifié le travail du testeur et accéléré la campagne en évitant de lancer manuellement OCE des centaines de fois, en permettant de déclencher des changements dans l'environnement et ainsi de simuler sa dynamique, et en fournissant une mesure de pertinence sans analyse laborieuse des connaissances d'OCE.
Ces tests ont permis de déceler certains défauts dans les prises de décision d'OCE. Ils ont également permis une évaluation initiale de l'apprentissage, d'affiner le paramétrage et de corriger des défauts dans le traitement du retour de l'utilisateur.

5 Travaux connexes

Cette section présente des travaux qui portent sur l'utilisation de scénarios pour tester des logiciels basés sur l'apprentissage automatique, et positionne notre proposition dans ce contexte. Des outils dédiés au développement et à l'expérimentation de solutions à base d'apprentissage par renforcement sont ensuite brièvement introduits et analysés par rapport à la question du test et de l'usage de scénarios.

5.1 Des scénarios pour tester

5.1.1 Le concept de scénario

C. Kaner [6] propose une définition du concept de scénario : un scénario est une histoire d'une personne qui essaie de faire quelque chose avec le produit testé. Hussain et al. [13] proposent une autre définition proche de la précédente, mais qui ne fait pas directement référence au test : un scénario est une description informelle d'une utilisation spécifique d'un logiciel ou d'une partie d'un logiciel par un utilisateur. Ici, les scénarios sont définis à partir des cas d'utilisation (des besoins de l'utilisateur) et sont utilisés pour dériver des cas de test.
Un scénario permet de décrire le déroulement complet d'une utilisation, donc de tester le logiciel de bout en bout dans sa globalité (test de niveau « système »). Il est décrit sous la forme d'une séquence d'interactions entre le logiciel et un utilisateur. C'est également le cas des scénarios de test d'OCE (cf. section 4), qui décrivent une séquence d'environnements ambiants (un environnement étant représenté par une liste de composants logiciels) mais aussi les assemblages idéaux qui eux modélisent les interactions entre un utilisateur et OCE.

5.1.2 Le test de modèles pour les véhicules autonomes

Nous retrouvons la notion de scénario dans le cadre de l'évaluation de comportements de véhicules autonomes basés sur l'apprentissage automatique [20]. Ulbrich *et al.* [28] décrivent un scénario comme un déroulement temporel d'une séquence de scènes, où chaque scène représente une configuration de l'environnement physique dans lequel un ou des véhicules autonomes doivent opérer. La description d'une scène comprend la position et la dynamique des autres entités présentes telles que des véhicules, des piétons, l'infrastructure routière et même les conditions météorologiques. La description d'un scénario comprend aussi les transitions entre les différentes scènes décrites par des actions ou des événements dans les scénarios, ainsi que les critères d'évaluation. Par exemple, dans un contexte de simulation, un scénario de test peut permettre de vérifier qu'un véhicule autonome peut circuler sans encombre d'un point A à un point B en présence d'autres véhicules, de piétons traversant la route et dans des conditions météorologiques défavorables.
Dans ce cadre, la notion de scénario ne correspond plus à une séquence d'interactions entre un utilisateur et un logiciel, mais plutôt à une séquence d'environnements. On peut noter que, dans le cadre de véhicules autonomes, les environnements sont hétérogènes et plus délicats à modéliser. D'autre part, dans nos scénarios de test, les actions et les événements entre les différents cycles ne sont pas explicitement spécifiés : les transitions entre les cycles sont représentées par des variations dans l'environnement ambiant, telles que l'apparition, la disparition ou la réapparition de composants, mais ces transitions restent implicites dans la description des scénarios. En revanche, les objectifs sont aussi décrits dans un scénario de test d'OCE grâce aux assemblages attendus dans les cycles d'évaluation, qui permettent de mesurer la pertinence des propositions d'OCE.

5.1.3 Analyse

Le principal avantage de l'utilisation de scénarios pour le test réside dans la capacité à évaluer des logiciels de bout en bout dans des situations particulières. Cette approche se révèle particulièrement adaptée pour évaluer des logiciels à base d'apprentissage automatique, en raison de leur nature « boîte noire ».
Cependant, nos objectifs de test à base de scénarios diffèrent de ceux des approches décrites dans la section 5.1.2. Alors que les scénarios de test pour les véhicules autonomes visent à évaluer les décisions prises par les véhicules, ce qui revient à tester les modèles construits antérieurement par apprentissage automatique, les scénarios de test d'OCE sont conçus pour évaluer le programme apprenant qu'est OCE, en testant sa capacité à produire de « bons »

modèles. Nos scénarios intègrent ainsi non seulement des phases d'évaluation mais également des phases (préalables) d'apprentissage.

5.2 Quelques outils

La littérature propose des outils pour le développement de solutions d'apprentissage par renforcement ; nous en avons sélectionné trois et nous étudions si et comment ils considèrent la question du test.

Gymnasium [5] est une librairie Python qui propose des environnements d'apprentissage par renforcement standardisés. L'évaluation d'une solution d'apprentissage se fait en la comparant à d'autres solutions, en les exécutant dans un ou plusieurs environnements proposés. Gymnasium ne propose aucun moyen pour réaliser cette comparaison, qui reste à la charge de l'utilisateur.

DotRL [22] permet l'évaluation d'algorithmes d'apprentissage standards (SARSA, QLearning…) ou personnalisés dans des environnements prédéfinis. Pour cela, le testeur choisit l'algorithme, l'environnement et ce qu'il veut mesurer, par exemple la récompense moyenne. Ce mode de fonctionnement se rapproche ainsi de Runner. En revanche aucun travail d'analyse n'est effectué à partir des valeurs mesurées, qui sont affichées sous leur forme brute. De plus, lors de l'exécution, il n'y a pas de distinction entre phase d'apprentissage et phase d'évaluation.

Cogment [1] prend en compte l'humain dans la boucle. Cette plate-forme facilite le développement de solutions dans lesquelles l'humain peut intervenir pour accélérer l'apprentissage. Cogment cible l'apprentissage interactif par renforcement mais son objectif n'est pour le moment que la mise en œuvre de solutions et il n'y a pas d'outil spécifique au test.

Les outils cités ci-dessus sont destinés au développement et à l'expérimentation de solutions d'apprentissage par renforcement. Ils ne permettent que de comparer des algorithmes dans des environnements prédéfinis. Seul DotRL propose des moyens d'évaluation, en restituant les valeurs mesurées pendant l'exécution de l'expérimentation.

La notion de scénario de test est absente dans ces outils. De plus, les environnements proposés sont pour la plupart stationnaires, alors qu'OCE opère dans des environnements qui sont, par nature, dynamiques et ouverts. Les scénarios de test d'OCE permettent d'intégrer ces caractéristiques, en donnant la possibilité au testeur de modéliser une séquence d'environnements avec sa dynamique (apparition d'un composant, disparition d'un composant, réapparition d'un composant…).

6 Bilan et perspectives

Dans ce papier, nous avons d'abord analysé la problématique du test dans le contexte de l'apprentissage automatique, en mettant particulièrement l'accent sur l'apprentissage en ligne et interactif. Nous nous sommes concentrés sur le test de programmes apprenants (par opposition au test de modèles).

Pour répondre à nos questions de recherche concernant la conception, l'implantation et l'exécution des tests d'un programme apprenant en interaction avec l'humain, nous avons proposé une approche à base de scénarios composés de phases d'apprentissage et de phases d'évaluation. Pour mettre en pratique cette approche et tester OCE, nous avons proposé deux outils : Maker et Runner. Ceux-ci sont opérationnels mais en cours d'évolution : plusieurs fonctionnalités sont actuellement développées ou à développer, telles que donner aux testeurs la possibilité de définir leurs propres mesures de qualité (par exemple précision, rappel…), ou de suivre l'exécution des scénarios cycle par cycle. Cette solution à base de scénarios a été conçue pour répondre spécifiquement au besoin d'évaluation d'OCE ; néanmoins, les principes que nous proposons pourraient s'appliquer au test d'autres solutions d'apprentissage interactif en ligne.

Notre proposition laisse encore un certain nombre de questions ouvertes. À ce stade, notre solution ne propose pas d'accompagnement pour la conception des scénarios : le testeur définit « à la main » tous les scénarios de test avec leurs objectifs, et en fixe le nombre. L'exécution d'un test est automatisée mais la conduite d'une campagne de test ne l'est pas. On peut ajouter que, dans le contexte dynamique et ouvert de l'apprentissage en ligne, le programme apprenant peut rencontrer de multiples situations qu'il peut être difficile d'anticiper, ou encore, pour ce qui concerne OCE, des environnements contenant un grand nombre de composants et de services. Un problème est donc de couvrir au mieux le champ d'application du programme apprenant et de tester des cas que le testeur ou l'utilisateur n'auraient peut-être pas envisagés. Pour cela, nous travaillons sur la génération et la sélection automatique de scénarios de test basées sur des modèles d'environnement avec leur dynamique.

Notre approche soulève également des questions relatives à l'implication de l'utilisateur dans le processus de test. Dans l'état actuel, les testeurs ont besoin non seulement de compétences en matière de test de logiciels et de compétences en apprentissage automatique, mais aussi de connaissances sur le domaine métier. Les utilisateurs finaux pourraient contribuer à la fois à la conception de scénarios et à l'interprétation des résultats. Une piste consisterait à permettre aux utilisateurs finaux de définir des scénarios dans un langage facile d'utilisation, qui seraient automatiquement traduits au format JSON à l'aide de techniques inspirées de l'ingénierie dirigée par les modèles.

Références

[1] AI Redefined, S. K. Gottipati, S. Kurandwad, C. Mars, G. Szriftgiser, and F. Chabot. Cogment : Open Source Framework For Distributed Multi-actor Training, Deployment & Operations. *CoRR*, abs/2106.11345, 2021.

[2] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people : The role of humans in interactive machine learning. *Ai Magazine*, 35(4) :105–120, 2014.

[3] P. Ammann and J. Offutt. *Introduction to software testing*. Cambridge University Press, 2016.

[4] N. Boukhelifa, A. Bezerianos, and E. Lutton. Evaluation of Interactive Machine Learning Systems. In *Human and Machine Learning : Visible, Explainable, Trustworthy and Transparent*, pages 341–360. Springer, 2018.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *CoRR*, 2016.

[6] JD Cem Kaner. An introduction to scenario testing. *Florida Institute of Technology, Melbourne*, pages 1–13, 2013.

[7] J. Chen, J. Patra, M. Pradel, Y. Xiong, H. Zhang, D. Hao, and L. Zhang. A survey of compiler testing. *ACM Comput. Surv.*, 53(1), 2020.

[8] T. Y. Chen, S. C. Cheung, and S. M. Yiu. Metamorphic testing : a new approach for generating next test cases. *arXiv preprint arXiv :2002.12543*, 2020.

[9] J. A. Fails and D. R. Olsen Jr. Interactive machine learning. In *Proceedings of the 8th Int. Conf. on Intelligent User Interfaces*, pages 39–45, 2003.

[10] G. Giray. A software engineering perspective on engineering machine learning systems : State of the art and challenges. *J. of Systems and Software*, 180 :111031, 2021.

[11] D. Graham, R. Black, and E. van Veenendaal. *Foundations of Software Testing : ISTQB Certification*. Cengage, 4 edition, 2020.

[12] D. M. Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1) :1–12, 2004.

[13] A. Hussain, A. Nadeem, and M. T. Ikram. Review on formalizing use cases and scenarios : Scenario based testing. In *2015 Int. Conf. on Emerging Technologies (ICET)*, pages 1–6. IEEE, 2015.

[14] F. Khomh, B. Adams, J. Cheng, M. Fokaefs, and G. Antoniol. Software engineering for machine-learning applications : The road ahead. *IEEE Software*, 35(5) :81–84, 2018.

[15] D. Marijan, A. Gotlieb, and M. K. Ahuja. Challenges of Testing Machine Learning Based Systems. In *Proc. of the 1st IEEE Artificial Intelligence Testing Conf.*, San Francisco, CA, USA, 2019. IEEE.

[16] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A.-M. Vollmer, and S. Wagner. Software engineering for AI-based systems : a survey. *ACM Trans. on Software Engineering and Methodology (TOSEM)*, 31(2) :1–59, 2022.

[17] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[18] C. Murphy, G. E. Kaiser, and M. Arias. An approach to software testing of machine learning applications. In *Int. Conf. on Software Engineering and Knowledge Engineering*, 2007.

[19] S. Nakajima. Generalized Oracle for Testing Machine Learning Computer Programs. In *Software Engineering and Formal Methods - SEFM 2017*, volume 10729 of *LNCS*, pages 174–179. Springer, 2017.

[20] D. Nalic, T. Mihalj, M. Bäumler, M. Lehmann, A. Eichberger, and S. Bernsteiner. Scenario based testing of automated driving systems : A literature survey. In *FISITA web Congress*, volume 10, 2020.

[21] OMG. *Unified Modeling Language*, chapter 11.6. 2017. `https://www.omg.org/spec/UML/2.5.1/PDF`.

[22] B. Papis and P. Wawrzyński. dotRL : A platform for rapid Reinforcement Learning methods development and validation. In *2013 Fed. Conf. on Computer Science and Information Systems (FEDCSIS)*, pages 129–136. IEEE, 2013.

[23] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella. Testing machine learning based systems : a systematic mapping. *Empirical Software Engineering*, 25 :5193–5254, 2020.

[24] S. J. Russell and P. Norvig. *Artificial intelligence : A Modern Approach*. Pearson Education, Inc., 2010.

[25] I. Sommerville. Component-based software engineering. In *Software Engineering*, pages 464–489. Pearson Education, 10^{th} edition, 2016.

[26] K. Sugali. Software testing : Issues and challenges of artificial intelligence & machine learning. *Int. J. of Artificial Intelligence & Applications*, 12(1) :101–112, 2021.

[27] R. Sutton and A. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 2nd edition, 2018.

[28] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. Defining and substantiating the terms scene, situation, and scenario for automated driving. In *IEEE 18th Int. Conf. on intelligent transportation systems*, pages 982–988. IEEE, 2015.

[29] E. J. Weyuker. On testing non-testable programs. *The Computer Journal*, 25(4) :465–470, 1982.

[30] W. Younes, S. Trouilhet, F. Adreit, and J.-P. Arcangeli. Agent-mediated application emergence through reinforcement learning from user feedback. In *29th IEEE Int. Conf. on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE)*, pages 3–8. IEEE Press, 2020.

[31] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing : Survey, landscapes and horizons. *IEEE Trans. on Software Engineering*, 48(1) :1–36, 2020.

Performances et explicabilité de ViT et d'architectures CNN : une étude empirique utilisant LIME, SHAP et GradCam

M. Colin[1*] I. Chraibi Kaadoud[2,3]
[1] Cali Intelligences
[2] IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest, France
[3] Centre INRIA de l'Université de Bordeaux, France

{melissa.colin0}@proton.me,{ikram.chraibi-kaadoud}@inria.fr

Résumé

Ces dernières années, l'IA explicable a été mise en avant comme la solution à plébisciter pour instaurer la confiance entre les utilisateurs et les systèmes d'IA. Pour étudier cette hypothèse, nous proposons une étude empirique sur le lien entre la performance et l'explicabilité de quatre algorithmes de vision par ordinateur : ViT, ResNet50, VGG16 et InceptionV3. Notre étude utilise trois méthodes d'explicabilité locale : LIME, SHAP et GradCam. Nous montrons que si l'IA explicable peut être un outil permettant de questionner la représentation artificielle d'un algorithme et son comportement, elle peut aussi présenter des problèmes de robustesse ou d'informations contradictoires susceptibles de miner la confiance. Les résultats de notre étude montrent que multiplier les outils d'explicabilité permet de vérifier la fiabilité des explications et des informations extraites.

Mots-clés

Explicabilité locale, Vision par ordinateur, Réseaux de neurones convolutifs, Vision Transformers, LIME, SHAP, GradCam

Abstract

In recent years, explainable AI has been presented as the main solution for building trust between users and AI systems. To investigate this hypothesis, we propose an empirical study on the link between the performance and explainability of four computer vision algorithms : ViT, ResNet50, VGG16 and InceptionV3. Our study uses three local explainability methods : LIME, SHAP and GradCam. We show that, while explainable AI can be a tool for challenging the artificial representation of an algorithm and its behavior, it can also present robustness problems or contradictory information that undermines trust. Our results show that by multiplying the use of explainable AI algorithms to explain one prediction, it is possible to verify the reliability of the explanations and extracted information.

Keywords

Local explainability, Computer vision, Convolutional neural networks, Vision Transformers, LIME, SHAP, GradCam

*Contact author

1 Introduction

1.1 Contexte

La transparence d'un système d'Intelligence Artificielle (IA) concerne, selon l'Union européenne, les données, le système en lui-même et les modèles économiques associés [1]. À ce titre, l'IA explicable, que nous noterons XAI [1] pour *explainable AI*, est devenue une exigence technique nécessaire pour établir la confiance entre les utilisateurs et les systèmes d'IA et ainsi atteindre une IA digne de confiance [1, 2]. Plus précisément, l'explicabilité et l'intelligibilité dans l'IA sont devenues un aspect important de la conception de systèmes d'IA acceptables [3], et ont été reconnues comme beaucoup plus importantes que la performance pure dans les systèmes d'IA [4, 5].

Or, depuis peu, la génération d'explication est l'objet d'étude, car : (i) Bien que diverses techniques d'XAI existent, il n'existe pas de solution universelle et le choix de l'approche d'XAI dépend de facteurs tels que la complexité du modèle, les données disponibles, le public cible et le domaine métier étudié [6], (ii) chaque algorithme d'explicabilité possède lui-même au niveau technique ses propres avantages et limites [7], (iii) les explications générées résultent elles-mêmes d'un compromis "complétude vs intelligibilité" qui peut introduire des biais d'interprétation ou de la désinformation [8].

Dans le domaine de la vision par ordinateur, les algorithmes d'explicabilité peuvent être utilisés pour : (i) exposer le fonctionnement interne du modèle expliqué au moyen de visualisations ou d'explications graphiques, ce qui peut aider les utilisateurs à comprendre comment le modèle traite les données et prend des décisions, (ii) identifier les caractéristiques d'une image qui contribuent le plus aux prédictions d'un modèle, en évaluant l'impact de chaque caractéristique sur les performances du modèle, (iii) mettre en évidence des caractéristiques spécifiques (pixels ou régions de l'image) qui ont conduit à une décision particulière afin d'expliquer des prédictions individuelles (répondre à la question "comment le modèle est-il parvenu à ce résul-

1. Nous utiliserons le sigle XAI dans la suite de cet article, car largement utilisé dans les communautés francophone et anglophone associées à ce domaine.

tat ?") [9]. Il s'agit alors d'explicabilité locale qui consiste à fournir une explication pour un résultat précis, i.e. pour une décision en particulier sur une échelle très réduite [10].

1.2 Positionnement et motivation

Nos travaux s'inscrivent dans ce dernier axe. Nous proposons ici de réaliser une étude comparative empirique sur les explications locales visuelles générées par 4 algorithmes différents de classifications d'image. Nos principales hypothèses sont les suivantes : (i) les explications visuelles issues de plusieurs algorithmes de classification d'image seront similaires pour une même classification ; (ii) les explications visuelles explicitent la représentation du monde de l'algorithme expliqué ; (iii) il est possible de générer des explications résultant de la fusion de plusieurs algorithmes d'XAI.

Nous focalisons notre étude sur deux familles d'algorithmes de classification d'images : les réseaux de neurones convolutifs (CNN pour *Convolutional Neural Networks*) et les *Vision Transformers* (ViT) [2]. Les expériences menées ont pour objectif de comparer les explications fournies par 3 algorithmes d'XAI appliqués à ces deux familles d'algorithmes de vision par ordinateur pour expliquer une prédiction, de tester les limites des différentes méthodes d'explicabilité, de comparer les performances et les ressources nécessaires pour l'ensemble des algorithmes, et de questionner l'utilisabilité des algorithmes d'XAI pour comprendre les différences entre les architectures convolutives et Transformers dans leurs façons de classer une image. Précisons ici que nos travaux se focalisent sur les explications des résultats des modèles et non les composants des modèles et leur rôle dans le comportement ou l'explicabilité de ces modèles.

1.3 Structure de l'article

Cet article est structuré de la manière suivante : La section 2 décrit la méthodologie utilisée. Les résultats expérimentaux sont présentés et discutés dans la section 3 au niveau des performances et de l'explicabilité, et une conclusion est donnée dans la section 4.

2 Méthodologie

L'objectif de notre étude est de comparer empiriquement les résultats d'algorithmes d'IA explicable locale sur deux familles d'algorithmes de vision par ordinateur, et non d'obtenir des résultats de pointe.

Par conséquent, nos expériences sont conçues pour que la configuration soit simple et qu'elles permettent une analyse comparative entre les résultats obtenus. Nous décrivons ci-dessous les données, les architectures et les algorithmes d'XAI utilisés pour permettre la reproductibilité des travaux présentés.

L'ensemble des expérimentations ont été effectuées sur la plateforme de calcul Google Colab [3] avec la configuration technique par défaut utilisant un CPU.

2. Nous utiliserons dans la suite de l'article les sigles CNN et ViT, car largement utilisés dans les communautés IA francophones et anglophones.
3. https ://colab.research.google.com/?hl=fr

2.1 Dataset

Le dataset utilisé dans notre étude est **un sous-ensemble de Asirra** (*Animal Species Image Recognition for Restricting Access*)[11] une base de données d'images de chiens et de chats annotées, soit deux classes d'objets [12] [4].

2.2 Algorithmes de classification d'images

Nous focalisons notre étude sur deux familles d'algorithmes populaires pour le traitement d'images : les CNN, qui sont historiquement les réseaux de neurones connus pour leur fiabilité en vision par ordinateur [13, 14], et les ViT, architectures plus récentes qui ont su démontrer de nombreuses fois de hautes performances [15].

2.2.1 Réseau de neurones convolutionnels

Les CNN reposent sur un empilement de couches de traitement [13]. Nous décrivons ici le fonctionnement des principales : la couche de convolution et la couche de pooling [5]

La couche de convolution repose sur le principe de convolution : un procédé de traitement d'image consistant en une opération de multiplication de deux matrices de tailles différentes mais de même dimensionnalité, correspondant respectivement à l'image en entrée et au *kernel* (le filtre ou fenêtre de convolution), afin d'en produire une nouvelle matrice, i.e. l'image filtrée, également de même dimensionnalité. Ce principe permet un traitement non coûteux pour la machine car il s'agit d'opérations simples (addition et multiplication).

Après chaque opération de convolution, une couche de pooling est généralement appliquée pour compresser l'information en réduisant la taille de l'image filtrée obtenue. Le pooling permet par ce procédé de regrouper les informations au sein de fenêtres de taille inférieure. Ce processus de convolution et de pooling est répété plusieurs fois en fonction de l'architecture du CNN, ce qui permet d'extraire les caractéristiques (*features*) importantes des images. Les données de sortie sont aplaties (*flattened*) pour être traitées par une couche de neurones entièrement connectés (*fully connected layer*). Traditionnellement, la sortie de cette couche est passée à une fonction softmax pour obtenir des probabilités de classe.

Différentes architectures reposant sur ces principes de convolution et de pooling existent. Dans cette étude, nous nous sommes intéressés particulièrement aux modèles suivants : ResNet50 [16], VGG16 [17], InceptionV3 [18]. Le lecteur pourra se référer à [19] pour une étude comparative des détails techniques de ces architectures.

2.2.2 Vision Transformers

L'architecture des ViT [15], schématisée dans la figure 1, repose sur le même principe que les Transformers utilisés dans le traitement du langage naturel [20].

Ce principe peut être décrit comme suit :

(i) Dans un premier temps, l'image est segmentée en plusieurs parties de taille fixe, nommées patchs, qui sont linéarisés (figure 1.a). Cette opération linéaire attribue un poids à chaque pixel du patch, représentant de façon

4. https ://zenodo.org/records/5226945 (Accès le 05/04/2024)
5. Pour plus de détails techniques, le lecteur pourra se référer à [13].

numérique chaque portion de l'image sous forme de jetons, ou *tokens*. Pour conserver l'information sur la localisation des patchs dans l'image, des embeddings de position sont ajoutés à chaque jeton, fournissant une indication précise de leur emplacement (figure 1.b).
(ii) Ensuite, le Transformer composé de n blocs (au nombre de 6 dans le modèle original) agit en tant qu'encodeur au travers de la répétition des 2 étapes suivantes [20] au sein de chaque bloc :
(ii.1) Utilisation d'un mécanisme d'attention multi-tête (*Multi-Head Attention, MHA*)[6] pour calculer le score d'attention entre chaque jeton par rapport aux autres (figure 1.d). Cela permet de déterminer l'importance relative de chaque jeton par rapport aux autres dans la représentation latente de l'image. Les jetons ayant les scores d'attention les plus élevés auront une contribution plus importante dans la génération de la représentation latente globale de l'image. Cette étape permet de mettre en évidence les parties de l'image les plus importantes pour la tâche en renforçant l'attention du modèle sur ces zones ;
(ii.2) Utilisation d'un réseau de neurones à propagation avant (*feedforward*) appliqué à la sortie du mécanisme d'attention multi-tête (figure 1.e). Ce réseau est constitué de transformations linéaires et non linéaires qui agissent sur les représentations des patchs. Cela permet au modèle de capturer des relations plus complexes entre les différentes régions de l'image et extraire des caractéristiques discriminantes des images, telles que les contours, les textures et les motifs significatifs.
(iii) Les sorties du Transformer sont finalement envoyées à un perceptron multicouche qui renvoie les probabilités finales pour chaque classe (figure 1.f).

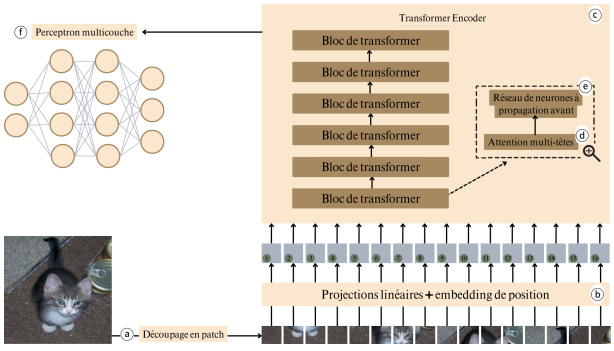


FIGURE 1 – Schéma de l'architecture ViT

2.3 Explicabilités d'XAI

Pour notre étude comparative, nous avons implémenté plusieurs méthodes d'explicabilité locale.

6. Le mécanisme d'attention Multi-tête est une extension du mécanisme d'auto-attention (*Self-Attention*), où au lieu de calculer une seule fonction d'attention, plusieurs fonctions d'attention sont calculées en parallèle, chacune avec sa propre matrice de poids.

2.3.1 LIME

LIME, pour *Local Interpretable Model-agnostic Explanations*, appartient à la famille des modèles de substitution locaux qui sont formés pour approximer les prédictions individuelles du modèle que l'on cherche à expliquer [21]. LIME peut être appliquée aux données tabulaires, aux textes et aux images. Pour ces dernières, LIME segmente l'image associée à la prédiction à expliquer en "superpixels" et active ou désactive ces derniers afin de créer des variations de cette image. Ensuite, LIME prédit la classe de chacun des points de données artificiels qui ont été générés à l'aide de notre modèle entraîné. Par superpixels, on entend des pixels interconnectés avec des couleurs similaires, qui peuvent être désactivés en remplaçant chaque pixel par une couleur définie par l'utilisateur, telle que le gris. L'utilisateur peut également spécifier une probabilité de désactivation d'un superpixel dans chaque permutation [22].
LIME permet d'utiliser différentes méthodes de segmentation telles que SLIC, Quickshift et Felzenszwalb [23] afin de diviser l'image à expliquer en superpixels ce qui facilite un traitement plus rapide des images.

2.3.2 GradCam

GradCam, pour *Gradient-weighted Class Activation Mapping*, est une technique permettant de produire des "cartes de chaleur" (*heatmaps*). Cet algorithme a été initialement créé pour les modèles CNN en utilisant les informations de gradient spécifiques à la classe qui circulent dans la dernière couche convolutionnelle du modèle [24]. Cependant, cette méthode a démontré des résultats tout aussi prometteurs sur les ViT en se focalisant sur le jeton de la dernière couche d'attention [25]. Quel que soit l'architecture sur laquelle GradCam est appliquée, cette méthode utilise les gradients pour peser l'importance de chaque neurone dans la couche en question, ce qui permet d'identifier les régions dans l'image d'entrée qui ont le plus contribué à la prédiction de la classe.

2.3.3 SHAP

SHAP pour *SHapley Additive exPlanations* permet d'expliquer les prédictions d'un modèle en attribuant à chaque caractéristique une valeur représentant son impact sur la prédiction [26]. Dans le contexte d'une image, une caractéristique peut être une valeur de pixel individuel, une texture, une forme, une couleur, une orientation, etc. Cette méthode est basée sur la théorie des jeux coopératifs et est capable de fournir des explications locales et globales pour les prédictions d'un modèle. Pour chaque classification, SHAP génère deux images qui correspondent aux masques expliquant l'implication des superpixels de l'image d'entrée dans les classes considérées pour la classification. Ces masques se présentent sous forme de superpixels allant de bleu à rouge, pour représenter respectivement ceux impactant négativement et ceux impactant positivement la classification. L'échelle de valeurs apporte un poids objectif à chaque contribution. Notons que pour les modèles complexes tels que les ViT, l'utilisation de SHAP peut nécessiter une grande quantité de mémoire RAM (au-delà de 13

Go) et de temps de calcul en raison du grand nombre de caractéristiques à traiter. Dans certains cas, cela peut rendre l'exécution de SHAP difficile ou même impossible.

2.4 Protocole expérimental

Notre travail se concentre sur la comparaison des perceptions des algorithmes d'IA appliquées au domaine de la vision par ordinateur au travers des explications générées. Pour ce faire, notre protocole expérimental est le suivant :
Étape (1) Préparation du dataset Asirra pour le fine-tuning : Cet ensemble de données a été segmenté en 348 images pour le dataset de test, 753 images pour le dataset d'entraînement et 273 images dans le dataset de validation avec 50% d'images de chats et 50% d'images de chien, soit un total de 1374 images utilisées pour notre étude. L'ensemble des images ont été redimensionnées avec la taille suivante : 224×224
Étape (2) Chargement des algorithmes de vision par ordinateur pré-entraînés : les 4 algorithmes pré-entrainés sur le dataset ImageNet ont été téléchargés et intégrés dans l'environnement d'expérimentation.
Étape (3) Fine-tuning et évaluation des algorithmes : Chaque algorithme est fine-tuné, i.e. son apprentissage est ajusté selon le nouveau dataset, pendant 10 epochs sur un batch de 8. Les performances des algorithmes sur les dataset de validation lors de l'entraînement sont sauvegardées pour connaître l'évolution de celui-ci. Et le modèle final est évalué à travers les données de tests pour en ressortir les matrices de confusion et les valeurs des *accuracy*.
Étape (4) Génération d'explications locales pour chaque algorithme : Pour chaque classification à expliquer :
• Pour LIME, nous avons créé une instance d'explication qui contient des détails sur les pixels qui ont contribué à la prédiction en générant : (i) une image, (ii) un masque et (iii) une combinaison de l'image et du masque qui peut être tracée pour voir quels pixels ont contribué à la prédiction. Parmi les différentes méthodes de segmentation, nous avons utilisé celle par défaut, i.e. Quickshift, et nous avons également calculé l'intersection entre les masques générés avec un coefficient de similarité de Jaccard.
• Pour SHAP, nous avons généré une explication avec (i) l'image, (ii) les pixels ayant contribué à la classification dans la classe "chien" et (iii) les pixels ayant contribué à la classification dans la classe "chat".
• Pour GradCam, nous avons généré la carte de chaleur qui explique la classification réalisée par l'algorithme de vision par ordinateur.
Notons ici que SHAP, par défaut, permet d'expliquer le rôle de chaque bloc de pixels dans la classification dans chacune des 2 classes considérées, alors que GradCam et LIME fournissent une explication liée à une classification réalisée au travers respectivement d'une heatmap et d'un masque. Précisons qu'un pixel qui contribue positivement à une classification implique qu'il joue un rôle positif dans le comportement de l'algorithme à classer la donnée en entrée dans la classe prédite. Inversement, un pixel qui contribue négativement à une classification "pousse" l'algorithme à ne pas classer la donnée en entrée dans la classe prédite. Tous nos algorithmes d'explicabilité locale offrent la possibilité d'expliquer visuellement les pixels qui contribuent positivement ou négativement à chaque classification.

3 Résultats et discussions

Dans cette section, nous présentons et comparons les performances des 4 algorithmes étudiés et les explications locales générées pour une classification en particulier, celle d'une image de chat présentée en figure 4, avant de les discuter en détail.

3.1 Comparaison des performances des algorithmes

Les performances des différents algorithmes ont été évaluées sur le dataset de test. Les mesures d'évaluation comprennent l'exactitude des prédictions, i.e. *accuracy*, le temps d'inférence, la durée du fine-tuning et le nombre d'epoch nécessaires pour atteindre l'exactitude maximale sur les données de validation avant 10 epoch.

La figure 2 représente les performances des algorithmes selon la métrique *accuracy*, le temps d'inférence en seconde nécessaire à réaliser une classification et le temps de Fine-Tuning en minute.

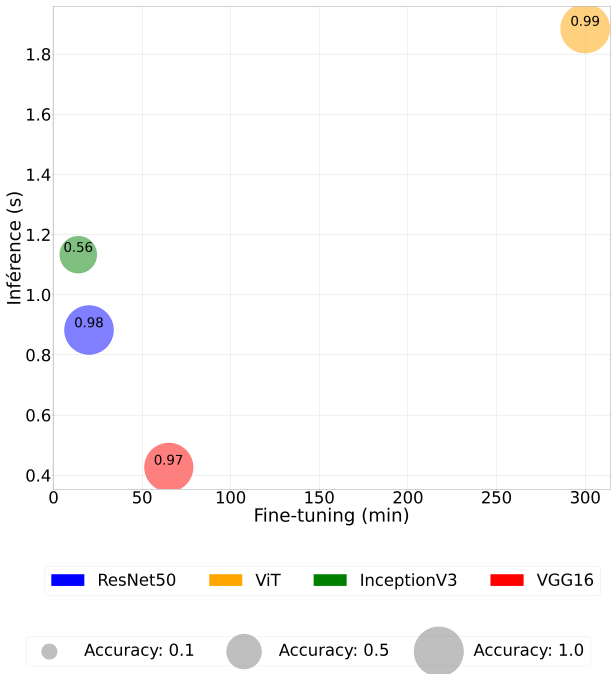


FIGURE 2 – Représentation des performances des algorithmes à travers la valeur de *accuracy*, selon le temps d'inférence en secondes nécessaire à réaliser une classification et le temps de Fine-Tuning en minute. Les valeurs des *accuracy* sont affichées dans les cercles.

De l'analyse des résultats présentés en figure 2, il ressort que :
• le ViT est l'architecture la plus performante avec une *accuracy* à 0,99, même si relativement proche du ResNet50

qui est à 0,98. Il reste important de noter que la différence de performance entre le ResNet50 et le ViT peut sembler minimale dans ce contexte, étant donné qu'ils tendent tous les deux vers 1.
• le ViT se démarque par son temps de fine-tuning de 300 minutes et d'inférence de 1,89 secondes qui sont plus importants que ceux des autres algorithmes qui sont tous fine-tunés en moins de 100 minutes et qui infèrent en moins de 1,2 secondes.
• le VGG16 se démarque des autres algorithmes en termes de temps d'inférence avec 0,43 secondes, cependant il présente une valeur *accuracy* similaire à celle du ResNet50 et au ViT avec une différence respective de 0,02 et 0,01.

Algorithme	**Temps mesuré pour le Fine-tuning sur 10 epoch en minutes**	**Temps estimé pour atteindre *accuracy* maximal en minutes**	**Epoch où *accuracy* est maximal**
ViT	300	45	1,5
ResNet50	20	10	5
InceptionV3	14	11,2	8
VGG16	60	30	5

TABLE 1 – Temps mesuré et estimé pour respectivement le fine-tuning des algorithmes, et l'atteinte de *accuracy* maximale lors de ce fine tuning avec les données de validation.

Le tableau 1 affiche le temps mesuré lors du fine-tuning sur 10 epoch, le nombre d'epoch nécessaire pour que chaque algorithme atteigne son *accuracy* maximale et le temps estimé pour atteindre cet epoch. Ce dernier temps est calculé à partir des deux précédentes informations. L'analyse de ces différents temps montre que : le ViT met le plus de temps à s'ajuster aux données lors de la phase de fine-tuning (300 min mesurée), alors qu'il est celui qui atteint son *accuracy* maximal avec un nombre d'epoch minimal (1,5 epoch) en seulement 45 min, ce qui représente 15% du temps total. Les autres architectures quant à elles nécessitent 50% à 80% du temps de fine-tuning pour atteindre leur *accuracy* maximale.

Les matrices de confusion de ViT, ResNet50, VGG16 et InceptionV3, figurant respectivement dans la figure 3.(a), (b), (c) et (d), permettent de comprendre plus en détail les différences de performances entre ces différents algorithmes de classifications. Elles permettent d'observer que ViT, ResNet50 et VGG16 rencontrent des difficultés sur la même classe : des images de "chat" sont classées en "chien" (4 images de chats sont mal classées pour ResNet50, contre 1 image pour ViT et 9 pour VGG16), ce qui suggère des points communs dans les éléments qui impactent positivement et négativement ces algorithmes. La matrice de confusion de InceptionV3 illustre bien que l'algorithme est globalement moins performant que les autres, mais également qu'il est plus performant sur la classe "chien" que "chat".

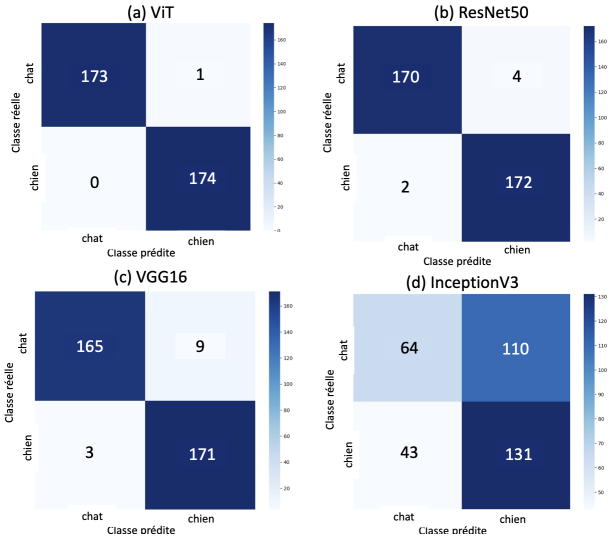


FIGURE 3 – Matrice de confusion : (a) ViT, (b) ResNet50, (c) VGG16 et (d) InceptionV3

3.2 Comparaison des résultats d'XAI

Nous présentons ici les explications locales générées sur les algorithmes étudiés. À des fins d'intelligibilité, nous présentons les résultats appliqués à une même donnée d'entrée (l'image de chat présentée en figure 4) afin de pouvoir comparer les explications générées par les différents algorithmes. Notons que :
(i) les algorithmes d'XAI LIME et GradCam ont pu être implémentés sur l'ensemble des 4 algorithmes, alors que SHAP n'a pu l'être que sur 3 (pas d'explicabilité SHAP pour ViT) à cause de limites techniques
(ii) une comparaison des explications a été réalisé quand la classification est correcte (i.e. l'image de chat est correctement classée dans la catégorie chat) pour les algorithmes ViT, ResNet50, et VGG16, et lorsque la classification est erronée avec l'algorithme InceptionV3.
(iii) un coefficient de similarité de Jaccard J a été calculé entre les masques générés par LIME par pour chacun des trois algorithmes ayant correctement classé l'image de chat (figure 4) : ViT, ResNet50 et VGG16. Il s'agit d'un coefficient utilisé pour comparer la similarité entre deux ensembles. Il est défini comme étant le rapport de la taille de l'intersection de deux ensembles et de la taille de l'union de ces mêmes ensembles.

FIGURE 4 – Image de chat utilisée dans notre étude et pour l'ensemble des explications générées.

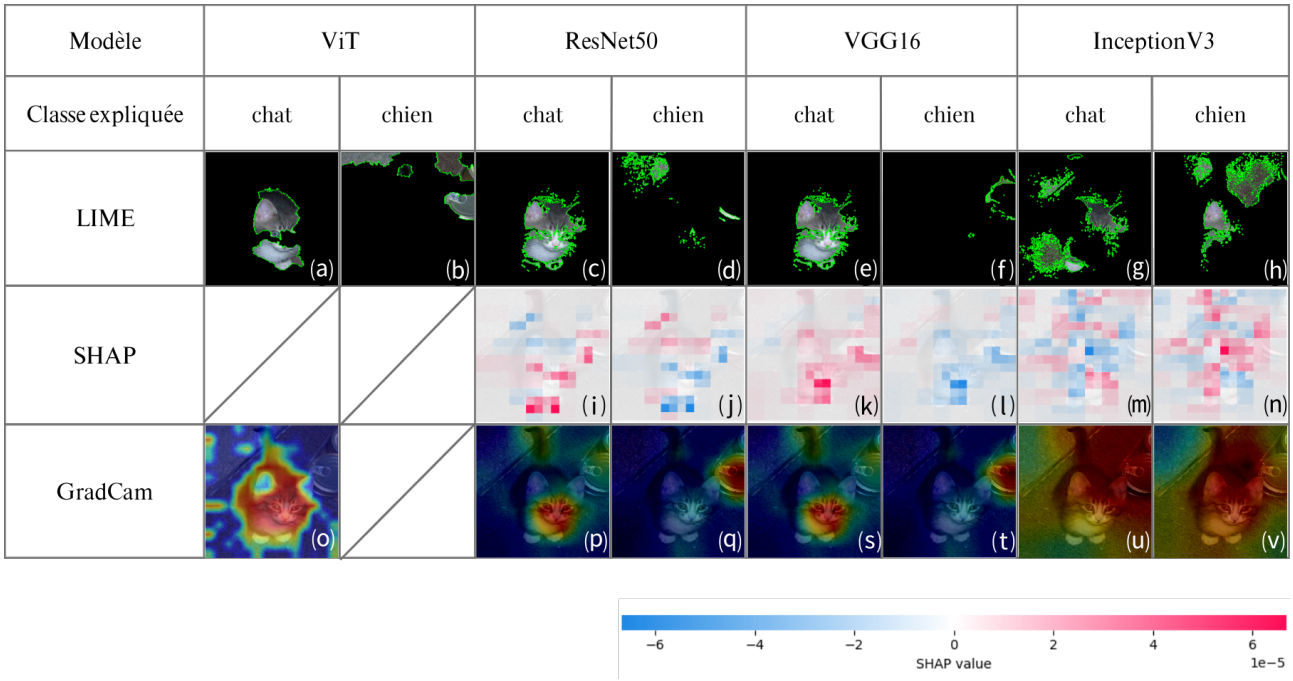


FIGURE 5 – Résultats des explications générées par LIME, SHAP et GradCam pour les classes "chat" et "chien" des modèles ViT, ResNet50, VGG16 et InceptionV3 sur l'image figure 4. Considérant ici pour LIME que les pixels ayant impacté négativement la prédiction faite par le modèle comme impactant positivement la classe non prédite.

3.2.1 Résultats LIME

L'analyse des explications fournies par LIME permet de mieux comprendre les différences entre les différents CNN et le ViT. En appliquant la méthode de segmentation Quickshift avec LIME sur l'image de chat présentée en figure 4, on observe des similitudes entre VGG16 et ResNet50. Sur l'ensemble des images testées, ces algorithmes sont expliqués par des masques identiques présentant les pixels impactant positivement la prédiction. Les masques présentés en figure 5.c et 5.e permettent de constater que les deux modèles se concentrent sur les mêmes zones de l'image pour prédire la classe correcte.

Les pixels impactant négativement les prédictions sont quant à eux différents entre ces deux modèles comme nous pouvons l'observer sur les masques 5.d et 5.f : les parties de l'image mises en évidence par les masques sont non seulement différentes mais tendent à signifier que l'arrière-plan est important dans ces deux cas de figure. En revanche, l'analyse des erreurs de prédiction du modèle InceptionV3 qui est moins performant, permet de constater (figure 5.h) que celui-ci se base souvent sur des parties du fond de l'image pour prendre sa décision. L'analyse des pixels ayant impacté négativement cette mauvaise prédiction, représentée figure 5.g, montre que l'algorithme InceptionV3 n'a pas pris en considération pour sa classification "chat" des parties essentielles du chat telles que le visage et le corps du chat (zone soulignée importante pour d'autres algorithmes, mais également qu'il a pris en compte des sections environnantes (l'arrière-plan à gauche et derrière le chat).

Lors de la comparaison entre ViT et l'ensemble des CNN, une différence significative émerge (figure 5.a et figure 5.b). Contrairement aux CNN qui analysent les pixels de manière unitaire, le ViT traite des parties d'images bien distinctes.

	Similarité de Jaccard J
ViT et ResNet50	≈ 0.577
ViT et VGG16	≈ 0.577
ResNet50 et VGG16	1.0
ViT, ResNet50 et VGG16	≈ 0.577

TABLE 2 – Similarité de Jaccard entre les masques des figures 5.a, figure 5.c et figure 5.e : LIME appliqué aux 3 algorithmes ResNet50, VGG16, ViT.

Pour étudier les similitudes entre les explications LIME de ViT, ResNet50 et VGG16, nous avons calculé le coefficient de similarité de Jaccard J sur les matrices représentant les masques associés aux bonnes prédictions de ces algorithmes. Les résultats, Table 2, présentent les valeurs obtenues pour différentes intersections de masque. L'analyse de ces valeurs a permis de confirmer que les masques générés par LIME pour Resnet50 et VGG16 étaient les mêmes (similarité de $J = 1$), et étaient eux-mêmes relativement similaires à celui généré pour ViT avec un coefficient à $J \approx 0.577$. L'intersection de l'ensemble de ces 3 masques est représentée visuellement dans la figure 6 et son ana-

lyse a mis en évidence que ces modèles utilisent une même partie de l'image de chat pour leur classification, à savoir, l'oreille gauche et un bout du museau. À l'inverse, aucun élément commun n'a été identifié dans ce qui impacte négativement la détection de ces 3 modèles avec un coefficient nul entre chacun des masques.

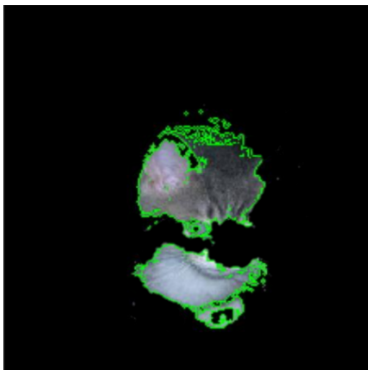


FIGURE 6 – Intersection des masques des figures 5.a, figure 5.c et figure 5.e : LIME appliqué aux 3 algorithmes ResNet50, VGG16, ViT

3.2.2 Résultats SHAP

SHAP a donné des résultats différents pour le ResNet50 notamment, comme le montre la figure 5.i et 5.j.
Des éléments similaires à ceux mis en avant par LIME ont été identifiés pour détecter la classe "chat", tels que les pattes et le visage, mais selon SHAP, l'oreille gauche impacte négativement la classification de l'image, et non positivement. De plus, SHAP met en évidence que ResNet50 prend aussi en compte une partie d'un objet (une conserve) présent dans l'image, alors que LIME non.
Cette explication se recoupe avec celle générée pour VGG16, à la différence que le masque généré par SHAP semble se focaliser plus sur le visage du chat et n'est que très peu impacté négativement par d'autres éléments de l'image.
Pour le modèle InceptionV3 qui s'est trompé dans la classification, les résultats de SHAP en figure 5.m et 5.n diffèrent de ceux de LIME en figure 5.g et 5.h : les caractéristiques déterminantes sont très parsemées et selon une grosse partie de l'image (en haut à droite de la tête du chat) a amené à classifier l'image comme "chien. Aucune explication du modèle ViT avec SHAP n'a pu être générée, car il nécessitait un trop grand nombre de caractéristiques à traiter pour être supporté par la RAM que nous avions à disposition.

3.2.3 Résultats GradCam

Les explications résultantes de GradCam sur le ResNet50 nuancent les résultats SHAP. Les figures 5.p et 5.q, présentant ces explications, montrent que la conserve aurait un impact négatif sur la prédiction car elle est de couleur claire sur la figure 5.p (qui met en avant les zones de chaleur pour la classe chat) et est mise en exergue avec des couleurs chaudes sur la figure 5.q (représentant les zones de chaleur pour la classe chien). Une zone orange sur le nez et les yeux du chat et jaune pour le contour du visage, constitue cependant un point commun avec les autres méthodes d'explicabilité.
La figure 5.s présentant le résultat de GradCam sur VGG16 montre une explication très similaire au ResNet50, ce qui rejoint les explications données par LIME et présentées dans la figure 5.e. La mauvaise prédiction donnée par InceptionV3 continue à se confirmer avec la carte de chaleur générée par GradCam. Une zone de chaleur qui pour la classe "chat" (figure 5.u) se répand sur la droite de l'image sans cibler d'objet et qui pour la classe "chien" (figure 5.v) prend en considération une grande partie de l'image. Soulignons également que dans les deux cas la zone de chaleur se répand sur toute l'image avec très peu de couleurs froides.
De son côté, GradCam appliqué au modèle ViT (figure 5.o) a mis en avant une zone bien plus importante de l'image, prenant en compte le chat dans son ensemble et de manière assez précise, ainsi que quelques éléments du fond de l'image. L'explication ici peut être interprétée comme étant plus juste car plus "complète" que celles des autres algorithmes d'XAI.

3.3 Discussion

3.3.1 Résumé des résultats

Nous avons mené une étude comparative des performances et des explications locales générées à partir de 4 algorithmes de classification d'images : ResNet50, VGG16, InceptionV3 et ViT. Pour notre étude, nous avons mesuré le temps nécessaire aux algorithmes pour le fine-tuning, le temps d'inférence et *accuracy*. Nous avons ainsi mis en évidence que parmi ces 4 algorithmes, le ViT était le plus performant, bien que le plus long à fine-tuner, et qu'il existait des différences notables en termes de performance entre les 3 architectures CNN différentes. InceptionV3 s'est avéré être, dans le cadre de notre étude, le moins performant. Nous avons ensuite implémenté 3 algorithmes d'explicabilité locale générant des explications visuelles, LIME, SHAP et GradCam, sur chacun des algorithmes de classification d'images dans le cas des classes "chat" et "chiens". Nous avons ainsi pu étudier les explications visuelles lorsque la classification est correcte pour ViT, ResNet50, et VGG16, et lorsque la classification est erronée pour InceptionV3. Pour chaque explication, nous avons analysé les superpixels contribuant positivement et négativement à la classification émise.

3.3.2 Discussion des explications par modèle

Plus en détails, concernant ViT, l'analyse des explications générées par LIME laisse supposer que ViT prend plus de temps pour s'ajuster aux données et pour effectuer une inférence car il traite les images par blocs plutôt que par pixels individuels, ce qui nécessiterait plus de calculs pour apprendre les relations spatiales entre les différentes parties de l'image. Cette approche semble permettre au modèle d'atteindre un seuil de précision important tout en lui permettant de prendre en compte l'intégralité de l'objet d'intérêt (ici le chat) pour la classification, comme le montrent les résultats avec la méthode GradCam (figure 5.o).

Les modèles ResNet50 et VGG16 présentent beaucoup plus de points communs. Ils semblent se concentrer sur des pixels similaires de l'image pour prédire la classe correcte, ce qui peut expliquer pourquoi, malgré leurs architectures différentes, ils ont des performances largement comparables. L'analyse de l'explication de la classification erronée d'InceptionV3, montre que l'algorithme ne semble pas avoir réussi à distinguer les formes présentes sur l'image et à se concentrer sur une partie en particulier. De plus, les zones mises en évidence par les 3 algorithmes d'XAI ne semblent pas cohérentes : les zones mises en lumière sont différentes d'un algorithme d'XAI à un autre. Ces résultats peuvent être mis en lien avec *accuracy* du modèle de 0,56 bien inférieure aux autres modèles, ce qui suggère que le modèle n'a pas réussi à converger en 10 epochs. Nous pensons donc que des simulations supplémentaires seraient nécessaires dans le cas de InceptionV3 pour explorer le lien entre l'évolution de *accuracy* du modèle et la précision des explications générées. Nous pourrions alors comparer nos résultats avec les explications d'une mauvaise classification sur un algorithme de classification d'image avec une forte *accuracy*. Un tel questionnement permettrait de mieux explorer la question de la représentation latente artificielle des algorithmes.

3.3.3 Discussion des méthodes d'explicabilité

Les différentes méthodes d'explicabilité que nous avons éprouvées lors de cette étude présentent à la fois des similarités et des différences. L'importance accordée au visage du chat est notablement présente dans l'ensemble des méthodes ayant bien classé l'image de chat. Mais des différences sont aussi ressorties, comme par exemple pour ResNet50 où SHAP considère un objet (conserve) comme important à la prédiction, là où GradCam ne le fait pas.

Ces différences peuvent s'expliquer par les approches utilisées par ces méthodes pour déterminer l'importance des caractéristiques. Chaque méthode se concentre sur différents aspects des modèles. SHAP prend en compte les interactions entre les pixels, tandis que GradCam se concentre sur les gradients des sorties du modèle par rapport aux entrées. LIME et SHAP sont des méthodes d'approximation locale : elles expliquent les prédictions du modèle pour des instances individuelles, mais ont plus de mal à capturer le comportement global du modèle. Alors que GradCam utilise les gradients de la sortie du modèle par rapport à ses entrées, ce qui peut fournir une vue plus globale de l'importance des caractéristiques.

Les résultats des méthodes d'XAI utilisées dans cette étude montrent que même si les performances des modèles de classification d'images peuvent être similaires, les caractéristiques sur lesquelles ils basent leur classification (autrement dit leur façons de classifier des images) peuvent être différentes. C'est pourquoi, le choix du modèle de classification, en plus de tenir compte de ces performances, doit prendre en compte les explications qu'on peut lui fournir. Nos résultats soulignent ainsi l'importance de considérer plusieurs méthodes d'explicabilité pour comprendre le comportement des modèles boîtes noires et d'éprouver leur robustesse.

Soulignons que nous avons relancé les simulations de LIME plusieurs fois et force est de constater que l'explicabilité locale fournie, peut varier et cela sans avoir modifié les paramètres de l'algorithme d'XAI ou la donnée d'entrée. Cela soulève la question de la robustesse des explications et de son impact sur la confiance des utilisateurs dans les explications générées. Nous souhaitons préciser ici que la robustesse des explications correspond à leur résilience face à des perturbations de l'algorithme d'XAI. C'est une notion distincte de la stabilité de l'algorithme d'XAI, qui correspond à la capacité du modèle à absorber des perturbations en dessous d'un seuil critique tout en maintenant un comportement stable et cohérent. Ces deux notions sont essentielles et complémentaires, et contribuent à la fiabilité d'un algorithme d'XAI [2, 27]). Des simulations supplémentaires seraient intéressantes à mener afin d'explorer plus en détails cette question de lien entre robustesse des explications et de stabilité des algorithmes d'XAI.

Enfin, nous pensons que l'alliance de méthodes d'ablation - dédiées à l'explication du comportement de modèles IA [28] - appliquées aux algorithmes de computer vision, et d'algorithmes d'explicabilité locale serait intéressant à explorer afin d'identifier les composants des modèles jouant un rôle dans la fiabilité des explications.

4 Conclusion

Notre étude est une introduction à la question de la confiance dans les algorithmes d'IA explicable. À travers ces travaux, nous avons questionné les limites des algorithmes d'explicabilité locale appliqués à la classification d'image. Nous avons montré que l'explicabilité peut être un outil pour questionner la représentation artificielle d'un algorithme et son comportement pour une classification lorsqu'elle est correcte ou non. Cependant, pour un même algorithme de classification d'images, il est important de multiplier les outils d'explicabilité afin de vérifier la fiabilité des explications et questionner les informations extraites.

Dans nos travaux futurs, nous souhaiterions explorer plus en détail les forces et faiblesses des algorithmes d'explicabilité en élargissant : (i) notre panel d'algorithmes de classification d'images, (ii) notre étude à un domaine d'application plus critique tel que le diagnostic médical et (iii) les dataset utilisés lors pour le fine-tuning pour en évaluer l'impact sur la représentation latente des algorithmes et l'impact sur les explications générées pour corroborer - ou non - les écarts de performance observés pour nos algorithmes. Par ailleurs, nous souhaiterions également explorer la question du choix de l'algorithme d'explicabilité locale selon la criticité du métier et des données notamment via des questionnaires d'évaluations des explications soumis auprès de différents métiers et niveaux d'expertise. Cela pourrait aider à la conception d'une cartographie des méthodologies d'explicabilité locale visuelles selon le besoin métier et permettre également l'élaboration de nouveaux algorithmes d'XAI qui tireraient profit des forces des différentes méthodologies existantes. Plus globalement, cela contribuera à la

question de la confiance dans les explications et par extension, à celle de l'appropriation de ces outils par la société civile.

Références

[1] IA GEHN. Les lignes directrices en matière d'éthique pour une ia digne de confiance, 2019.

[2] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad, Jose M Alonso-Moral, Roberto Confalonieri, Riccardo Guidotti, Javier Del Ser, Natalia Díaz-Rodríguez, and Francisco Herrera. Explainable artificial intelligence (xai) : What we know and what is left to attain trustworthy artificial intelligence. *Information fusion*, 99 :101805, 2023.

[3] Daniel S. Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Commun. ACM*, 62(6) :70–79, 2019.

[4] David Gunning. Darpa's explainable artificial intelligence (XAI) program. In *IUI*. ACM, 2019.

[5] Alejandro Barredo Arrieta, Natalia Diaz-Rodriguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58 :82–115, 2020.

[6] Vinay Chamola, Vikas Hassija, A. Razia Sulthana, Debshishu Ghosh, Divyansh Dhingra, and Biplab Sikdar. A review of trustworthy and explainable artificial intelligence (XAI). *IEEE Access*, 11 :78994–79015, 2023.

[7] Hans de Bruijn, Martijn Warnier, and Marijn Janssen. The perils and pitfalls of explainable AI : strategies for explaining algorithmic decision-making. *Gov. Inf. Q.*, 39(2) :101666, 2022.

[8] Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artif. Intell.*, 267 :1–38, 2019.

[9] Christian Meske and Enrico Bunde. Transparency and trust in human-ai-interaction : The role of model-agnostic explanations in computer vision-based decision support. In Helmut Degen and Lauren Reinerman-Jones, editors, *Artificial Intelligence in HCI - First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19-24, 2020, Proceedings*, volume 12217 of *Lecture Notes in Computer Science*, pages 54–69. Springer, 2020.

[10] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5) :93 :1–93 :42, 2019.

[11] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul. Asirra : a CAPTCHA that exploits interest-aligned manual image categorization. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 366–374. ACM, 2007.

[12] Jacquemont Mikaël. Cats and dogs sample, August 2021.

[13] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville : Deep learning - the MIT press, 2016, 800 pp, ISBN : 0262035618. *Genet. Program. Evolvable Mach.*, 19(1-2) :305–307, 2018.

[14] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks : an overview and application in radiology. *Insights into imaging*, 9 :611–629, 2018.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words : Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016.

[19] Aravinda S Rao, Tuan Nguyen, Marimuthu Palaniswami, and Tuan Ngo. Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. *Structural Health Monitoring*, 20(4) :2124–2142, 2021.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30 : Annual*

Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008, 2017.

[21] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you ?" : Explaining the predictions of any classifier. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016.

[22] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[23] Tobias Huber, Benedikt Limmer, and Elisabeth André. Benchmarking perturbation-based saliency maps for explaining atari agents. *Frontiers Artif. Intell.*, 5, 2022.

[24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam : Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.*, 128(2) :336–359, 2020.

[25] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 782–791. Computer Vision Foundation / IEEE, 2021.

[26] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.

[27] Ann-Kathrin Dombrowski, Christopher J. Anders, Klaus-Robert Müller, and Pan Kessel. Towards robust explanations for deep neural networks. *Pattern Recognit.*, 121 :108194, 2022.

[28] Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, and Tobias Meisen. Ablation studies in artificial neural networks. *CoRR*, abs/1901.08644, 2019.

Session 5 : Formalisation et systèmes à base de connaissances 1

Controllable Text Generation to Fight Disinformation

U. Oliveri[1,2], A. Dey[3], G. Gadek[2], D. Lolive[1], B. Costé [3], B. Grilheres[2], A. Delhay-Lorrain[1]

[1] Université de Rennes, CNRS, IRISA UMR6074
[2] Airbus Defence & Space, Elancourt
[3] Airbus Cybersecurity, Rennes

May 24, 2024

Abstract

During the 21st century, the global development of online platforms have led to their use in Strategic Influence Operations.
These operations cause deaths and put democratic processes at risk. To counter it, governmental and non-governmental entities emerged with analysts specialized in detecting and characterizing foreign disinformation campaigns. These analysts need constant training to be able to adopt new methodologies against the highly-evolving threats landscape. Trainings take place in simulated environments reproducing the Internet, providing faithful training environment that we call infospheres. This paper is about an ongoing PhD which aims at automating content generation to populate these infospheres. We highlight key challenges in this domain which are controllable text generation, fictive writings, preventing hallucinations and evaluation protocols to assess the used methodologies. We propose an idea based on a three levels modeling which structures knowledge and social dynamics associated with a training scenario. This modeling is used in a prompting methodology to automate content production with Large Language Models. Finally, we propose directions to continue this work, including social graph modeling to mimic how information propagates as well as new evaluation protocols.

1 Introduction

Online disinformation campaigns associated with Strategic Influence Operations [60] are a 21st century plague affecting democratic processes [45] and causing sanitary protocols failures [58]. Several initiatives are developed to fight it but it remains an open challenge [1]. Among these initiatives, governmental and non-governmental entities train specialized analysts. These analysts are in charge of detecting, characterizing and reporting on large scale disinformation campaigns, augmenting our knowledge on them and our capabilities to counter them. These analysts train in isolated environments which contain realistic emulations of the Internet that we call infospheres. These infospheres contain faithful reproductions of social media networks, press sites, blogs and other platforms present in the Internet.

To emulate the large scale of information flowing on the web, infospheres should be populated with large amounts of diverse contents (e.g., social media posts, press articles, micro-bloggings, etc.).

These infospheres are used here to align with legislations and ethical guidelines, removing the risk of creating potential harmful content in a public manner. These contents can be but are not limited to disinformationnal contents, political contents, religious points of views which are used to recreate what really happens within social medias. Legislations around this usage include training Defence Classification which make the generated content based on it unusable in public settings as well as legislations around content generation, for example GDPR [1].

In this ongoing PhD, we focus on generating textual contents for these infospheres and we highlight key challenges extracted from the literature to generate automatically these contents. Challenges include controllable text generation, fictive writings, preventing hallucinations and evaluation protocols.

A pre-requisite of the content generation for training purposes is to have it adhere to a training scenario provided by the system operator, which represents input constraints. The scenario is used to guide the generation, providing key points of interests for the training. The scenario contains realistic but fictive topics, events or information constituting the narrative arc of the generation. The fictive nature of these elements is necessary to reduce analysts biases due to past trainings, stereotypes towards populations as well as to comply with legislation. The sheer amount and diversity of contents to produce make it intractable for a human to manually write these contents for each diverse training scenario.

Furthermore, we shall create legitimate contents as well as the ones associated with reproduced disinformation campaign, calling attention to the need for global modeling used to generate both types with the same architecture.

In this paper, we will begin by reviewing the literature on the definitions of disinformation and the different attempts to mitigate it. With these related works, we find that pedagogy is one of the best method to reduce disinformation impact in the global population.

[1] https://gdpr-info.eu/

We apply this findings to our problem of training operational analysts leading us to stress out the need for a system similar to inoculation [54] to enhance operational analysts capabilities to detect, characterize and report disinformation campaigns. In this purpose, we study the works in Controllable Text Generation to provide contents adhering to a scenario, providing realism in the training. Finally, in order to assess the quality of the generated content, we study the evaluation methods in Controllable Text Generation and highlight that the current methods are not sufficient to solve this evaluation problematic.
Secondly, we will discuss about a first lead to solve the generation problem. We propose to upscale Information Definitions in order to apply it to Information Campaigns within social media platforms. This idea leads us to propose a three-stages modeling defining three distinct granularity levels. This modeling allow us to input macro-level input constraints and deriving the thinner constraints automatically, automating most of the work.

2 Related Work

2.1 Information & Disinformation

In the context of reproducing the exchanges within an infosphere, we can define information within online platforms as a communication process between a sender and a receiver of a message [42, 67]. Considering this definition, each individual is associated with contexts, or personas in relation to our work, with the sender having an associated intent [42]. The online platforms within the infosphere echoes to Shannon Information Theory, referred as the communication channel [59].
Truth, important in the context of disinformation, is often related to Information. We separate it from the definition, as an Information can be False or True, similar to the Ecological Approach of Information [46].
Having defined Information, we can discuss about what are the differences between Information and Disinformation.
Defining disinformation is subject to debates both in the scientific world and in the public area [25]. Wardle et al [68] has proposed a framework which has enlightened the diverse main terms [1] around what is called Information Disorder. These three terms are:

- Disinformation: Fake information spread with intent to harm.
- Misinformation: Fake information spread without intent to harm
- Malinformation: True information spread with intent to harm.

These definitions all refer to the intent, part of our information definition, as well as the truth value of the information.
Information Disorder spreading has been accelerated by social media networks, profiting of recommendations systems pushing forward high-click contents [3], creating echo chambers [62] diminishing impacts of debates on the web.
Information Disorder has been weaponized by States to disturb foreign adversaries. These attacks also known as Strategic Influence Operations are defined as *efforts by individuals and groups, including state and non-state actors, to manipulate public opinion and change how people perceive events in the world by intentionally altering the information environment* [60]. Studies about these operations highlight the targeting of foreign democratic processes [60] or geopolitical interests [41]. Several terms exist alluding to it, including Disinformation campaigns or Influence Campaigns.
Information Disorder in its entirety has proven being a massive danger for society, its diffusion speed and sheer amount resulting in multiple deaths by spreading erroneous facts during sanitary crises [58], widening the gaps within populations and undermining election processes [45].
Hence, it is necessary to find ways to counter it and reduce its impact.

2.2 Fighting Disinformation

As disinformation explodes, several developments to try to counter it have emerged. Four categories appear [55]: Algorithmic, Corrective, Legislative and Psychological solutions.
Among them, we regroup algorithmic and corrective solutions as technical solutions as we believe that emerging new technologies will make both categories converge.
A wide range of technical solutions is available. Within this field, lots of research is done on detecting disinformationnal content [72]. Among this research, we can find uses of neural networks used to detect disinformation [77] or disinformational networks inside platforms [66]. Algorithm tweaking by social media network providers, whose goal is to disincentive disinformation and reduce their exposure [55] via their recommendation systems, is also commonly done. Natural language processing algorithms can also be used to counter disinformation content through automatic content fact-checking within platforms [23]. Manual fact checking is also widely developed [20] but is limited by the work it requires to verify disinformational contents and by the limits of its audience and effects [5]. Both types of fact-checking suffer from a phenomenon known as continued influence effect [14], which states that fact-checking on already disinformation-exposed people results in little to no effect [57]. Finally, inspired from the military world, frameworks characterizing Strategic Influence Operations allow the classification of attacks and storage of the various modus operandi in structured knowledge databases. These works are particularly useful to enable quick reactions against a threat.
Among these frameworks, DISARM [63], derived from MITRE ATTACK Matrix [61], allows the classification of Tactics, Techniques and Procedures from disinformation campaigns. ABCDE [49] describes the Actors, Behaviors, Contents, Degree and Effects, which are the "who", "what"

and "how" produced by a disinformation campaign [4]. Applied to particular threat actors, the 4Ds [47] describe the Russian operations against foreign States. However, it is mostly designed against Information Manipulation by the Russian State, which is limited. Similarly, BEND [6] only focuses on mass cognitive exploitation techniques. These frameworks now make it possible to structure Strategic Influence Operations and understand their mechanisms, allowing us to fight against them.

Legislative Solutions are legislations or regulations created by actor of powers, head of states or global social media providers. The European Digital Service Act [16] makes Internet Service Providers, Cloud Providers and Social Media networks responsible for the content hosted on their services, especially in disinformation matters with its Code of Practice against Disinformation [2]. However, laws against disinformation needs to state what is false and what is true which can result in abuses by favoring the "truth" emitted by the executive power [19], referencing censorship.

Terms of Service by Social Media Network legitimize the suppression of accounts in case of disinformational activities after detecting them with technical solutions [3], referring to the precedent description.

One essential part to fight disinformation relies on training the population to detect and develop critical thinking in order to diminish the negative impacts [9].

Pedagogy solutions include all the techniques used to increase one's resiliency against disinformation and inability to detect it. Some initiatives aim at integrating curriculum lessons for young people on how to fight disinformation [15] with limitations including gaps in technology literacy between teachers and students or lacks of training for the teachers on the disinformation thematic.

Among Pedagogy techniques, we can find inoculation [55, 54] which is the closest to our work. The main idea from inoculation is to expose the reader to disinformation in a controlled environment to enhance their capability to detect it and their resiliency against it. However, we diverge from the Bad News Game [54] by introducing fictive writings and massive amounts of diverse data to our training.

Progresses in disinformation fighting have been made, but are not sufficient. To accomplish our objectives, new technologies such as controlled text generation with Large Language Models (LLMs) are essential.

LLMs will help us obtain a huge amount of diverse contents - both in topics addressed and text diversity -, as well as adhering to pre-defined training scenario, increasing realism.

However, producing automated content with an Artificial Intelligence can exacerbate biases towards populations or nations [24]. Hence, it is necessary to contain usage with both controlled environments (i.e infospheres) and through fictive writings, which is required not to marginalise populations, reducing at the same time the intrinsic biases of users toward past experiences.

[2] https://digital-strategy.ec.europa.eu/en/policies/code-practice-disinformation

[3] https://support.google.com/youtube/answer/10834785?

2.3 Controllable Text Generation

Controllable Text Generation is a subfield from Natural Language Generation. It aims at guiding the generation to solve constraints on the output. It usually falls into two categories.

The first category, called soft constraints, mostly addresses emotions, topics or style or detoxification [10, 56, 37]. It is done by adding control codes [30], external classifiers [70, 35, 10, 38] or smaller language models used as guides [33, 56, 37]. These techniques, while having proven results, suffer from decreasing generation quality when multiplying the constraints [73]. Furthermore, when adding external models or classifiers to re-rank model's predictions, it multiplies the calls to external models. In fact, for each token, a subset of the model's predicted probability density is passed to external models [70], resulting in slower systems [73].

In the second category, hard constraints are used to enforce structural constraints in a generation, for example by enforcing the presence of explicit knowledge in the model using knowledge graphs [39, 53], requiring the presence of particular keywords [28], particular structures [29] or length control using text-adapted diffusion models [35].

Recently, advances in Large Language Models have been tremendous. Previous works like LLama [65] or InstructGPT [48] have demonstrated the ability of models to follow prompted instructions, resulting in new State of The Art (SOTA) results in hard, soft or combined constraints [75]. However, it suffers from the same difficulties where the generation quality decreases when multiplying the number of constraints.

An important factor in controllable text generation applied to training is fictive writings. It is used to diminish analysts biases towards populations or past experience. Furthermore, it also exposes analysts to realistic but new scenarios.

Fictive writings, which are used here to generate texts about realistic but fake events or topics are necessary to reduce biases including past experiences impacts. It is required for the analyst to focus on the proper methodology to detect disinformation and not relying only on past operations. The challenge here is to use realistic fictive writings without making explicit references to the world knowledge induced by the model's training. It is then different from generating fictive stories [69, 71] or paraphrase generation [76]. It can be seen as paraphrasing realistic events, which causes fictive discussions within the infosphere. To the best of our knowledge, no work has been done on this particular topic. To automate this part, it is necessary to tackle the hallucination problem, to remove the possibility that real arguments can harm training participants while reducing experience biases impact.

We refer to hallucinations as texts that are nonsensical or unfaithful to the provided source of content [27]. From the perspective of training, hallucinations can be quite a problem in stating erroneous facts in a non-controlled manner, as well as making explicit links with

real informations coming from model pre-training dataset, cutting short the analyst's immersion. The latter is particularly a problem because doing non-requested links with the real worlds makes the use of fictive writings obsolete. To link it with the literature definitions, the "reality" of the operational training is what is contained in the scenario. Refering to external knowledges from the real world that opposes the scenario can thus be qualified as hallucinations.

Several works have been trying to address this problem, including using external knowledge bases to have the model adhering to a ground truth [43] or relying on prompting methodologies to have the model verify its own generation [12]. Other methods include building faithful datasets [18] or usage of Reinforcement Learning with Human Feedback (RLHF) [48] to align the model generation with human behavior, less prone to hallucinations. However, RLHF needs training of a Reward Model which is cost-intensive, both in human time and hardware, although research is being done to diminish the costs [21, 51]. To this day, hallucination mitigation remains an open scientific problem.

Finally, to assert our generation results, it is necessary to have an objective function to evaluate.

2.4 Evaluation

In Text Generation, several key points need to be addressed, including the intrinsic quality of the text, fluency, grammaticality and diversity [73]. Assessing how much the constraints are respected have been proven difficult in Controllable Text Generation [17]. Two main branches exist in this domain which are human and automatic evaluations.

Human evaluation are generally used with Likert or RankMe scales [73]. It is considered the best way to evaluate generation quality but fails in evaluating diversity [22] while being time-consuming and expensive.

Due to these constraints, automatic metrics have been developed. In text generation, a lot of them are based on n-grams overlap like BLEU [50] or ROUGE [36]. However, these metrics fail to evaluate paraphrase, which is particularly present in Controllable Text Generation [74]. Quality can also be calculated using external models as "oracles" to judge if the generation is qualitative.

Perplexity [26] is used with causal models to assess the uncertainty of the generation, hence how "surprised" the model is when presented to new tokens. Nevertheless, perplexity does have its own flaws and is not representative enough of quality [64].

Recently, methods using SOTA Large Language Models as evaluators [8, 7] start to emerge, demonstrating real capabilities in evaluating quality, paving the way for further research.

Usually, to evaluate constraints respect, external classifiers are adopted and are in charge of scoring the generation [70, 10, 35]. Language Models embeddings [2] can also be used. For instance, BertScore [74], Bertr and Yisi [44] compute a distance between the embeddings of the generation and the reference to assess if they are in the same semantic field thus respecting the constraints.

Within our research, generating high diversity contents is needed. To evaluate it, two main metrics exist in the literature: Distinct-n [34] for evaluating the diversity within a generation and SELF-BLEU [78] which assesses the n-gram overlap between all generations pair to pair, evaluating diversity within a generated dataset.

Instruction-Tuned models, which are SOTA models at the moment, have their constraints specified as textual instructions. To have diversity within our generations, we need to provide them with with instruction scenarios, bringing variance.

To increase realism, we want that each generation innovates from the scenario, while respecting the constraints. This idea is analogous to evaluating serendipity [32, 31] which in text generation is done only in story generation [71] as we know.

3 Methodology

3.1 Reasoning

To populate a realistic infosphere which can be used to train analysts, it is necessary to have both normal content and realistic disinformation campaigns. Moreover, to increase realism, it is necessary to have quantities of data similar to what a social media user can be confronted to in real life. Hence, huge amounts are necessary. Unfortunately, it is not possible to manually design all topics and events addressed in a realistic manner. It is necessary to reflect upon this necessity to craft every content with little to no manual work from external system user.

Using previous definitions, we can start designing a modeling based on Information Definition which we want to upscale for mass scale communications, necessary for our work. We focus in this work on structured information campaigns.

We propose a training-free and model-agnostic Prompting Methodology based on an extension of our Information Definition, bringing us characteristics from disinformation frameworks and normal communication processes. Inspired from the three levels decomposition from DISARM [63], namely Tactics, Techniques and Procedures, we develop a three stages modeling: Global Actors, Local Actors and Individuals. Each stage is characterized by different granularity of objectives. Global Actors define directions of an information scenario and global objectives. Local Actors describe mid-term objectives and the content to push. Finally, Individuals are characterized by short-sighted objectives, which are described as intent. Having identified three groups, we need to complete their description to have a more concrete view of what happens in reality, allowing us to craft all contents with minimal user work.

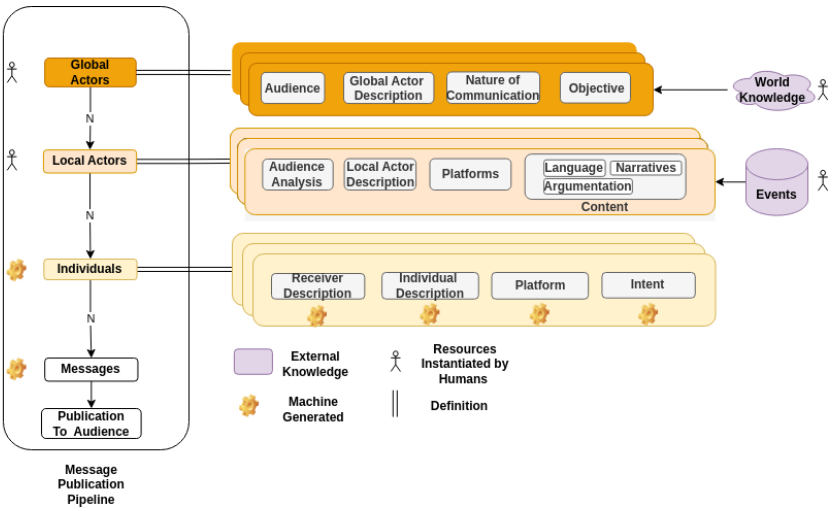


Figure 1: Modeling overview used to semi-automate the population of an infosphere

3.2 Modeling

3.2.1 Global Actors

The goal of having a Global Actor description Level is to describe from a Macro-Level perspective who are the main Actors of the campaign. As seen in Figure 1, we also model what are their long-term objectives, what is the population targeted by their communication campaign and what is the nature of the communications wanted by the Actors, be it micro-blogging, video platforms, blogs, etc. For example, an instantiate of a scenario may begin with the description of a fictive country called Truantia, which has an objective to establish economic relations with the neighboring countries by crafting an information campaign on microblogging platforms to spread the benefits of linking with Truantia.

From this level, we can derive Local Actors, which are subsets of the Main Actor.

Encompassing all Actors, is passed externally a short description of what is currently happening in the World, focusing on the global thematic, referred in Figure 1 as world knowledge. With our example, we can imagine an history of Truantia past economic ties and description of its neighbors. This description is limited to the geopolitical landscape the scenario focuses on and can be compiled with external resources for example with press sites.

3.2.2 Local Actors

Local Actors are groups deriving from a Global Actor, whose job is to accomplish tactical operations to accomplish the final objective. We detail here the audience, including analysis of the audience or micro targeting, the Local Actor description, the platforms used in this group and what is called here the Content this group pushes. The Content, inspired by ABCDE framework [49] illustrates what is the language used, the narrative to push, including an arguments justifying the stance on the narrative.

An event Database is Plugged in this description, based on [13] event description. This Database will need manual work by humans to instantiate fictive events. Creating this database is non-trivial and necessitates a thorough human analysis work of creating event with interest for the scenario, to create realistic exchanges. In fact, these events are the main subjects of discussion in the infosphere, where each individual talks upon it according to his belongings.

For example, a company from Truantia called Compania leverages its influence and fame to push on micro-blogging social networks an english narrative that having partnership with Compania can help Truantia's neighbors companies to flourish due to its past impacts. Individuals in this group will have an intent derived from this narrative and will talk on the specified event.

3.2.3 Individuals

Based on our initial definition of Information, an Individual is composed by a persona (sender description), a receiver description, a platform and its description as well as an intent which is a short-term objective. We base our definition of persona on [42] which includes:

- Geographical Context: what is the nation, culture, language, physical community of the individual.
- Social Context: what are the individual interests, including political interests in our context.
- Educational Context: what is the individual level of study.
- Professional Context: what is the individual past and current occupations.

Role-playing enhances generation quality and diversity [40] by deriving messages from unique personas. It also mimics what happens in real life with real users behind accounts, interacting in respect of their beliefs.

3.3 Direction

3.3.1 Automating content generation

The goal is to automate the content production of an infosphere given a description from Global Actors and Local Actors, which are given by our input training scenario. Starting from this human input, production of individuals is automated according to the given definition granting a large diversity and enhancing quality thanks to role-play [40]. After this first step, each individual description generation is an input for the generation of messages, resulting in diverse views of the same event from the Events Database.

In the first step, a Large Language Model will be prompted with Global Actors and Local Actors, and asked to generate a Persona, an Intent and a Platform to generate a message on.

In the second step, a LLM is prompted with Global Actors, Local Actors and the new information generated at Step 1) and is asked to generate a message according to its instructions.

Considering the created generation methodology, a perspective of improvement bears on improving the realism by mimicking real user behaviours and how information propagates. Hence, it is necessary to do further research on social graph modeling [52] to increase realism and thus immersion for analysts.

3.3.2 Evaluation

This whole process constitutes a large amount of constraints. As discussed before, the evaluation of controllable text generation is quite difficult and remains an open challenge.

In our case, assessing the respect of the constraints with SOTA methods is not possible due to the large number of constraints. As reviewed, several methods use external classifiers trained to assess presence of certain constraints. However, given the goal to address a wide range of constraints, it is not affordable to re-train classifiers for each constraint and for each scenario. In the same vein, BertScore methods use pre-trained Bert [11] Models. The objective of generating fictive content would require to re-train Bert Models for each scenario, without having previously assessed the quality of the generation. This is intractable.

For the moment, we consider to assess the respect of the constraints via Human Evaluation, which is considered the best evaluator [22].

For straight quality measurements, several methods are usable. Here, our goal is to assess that the prompting methodology, which stacks constraints, does not degrade the model capacity to generate good quality text. This quality measurement is attainable via Perplexity calculation [26]. By assessing it with a model prompted with our methodology and the same model with a simple straight instruction - asking to write on the same platform -, it is possible to measure this degradation. Finally, due to time constraints, the fictive Writings evaluation and Hallucination mitigation evaluation have not been explored yet.

4 Conclusion

In this work, we have presented an ongoing PhD which aims at automating content production to populate an infosphere. The purpose of this infosphere is to be used by entities specialized in fighting Strategic Influence Operations or disinformation campaigns. These entities train operational analysts to detect, characterize and report these operations.

To populate this infosphere, we highlight the need to use Large Language Models to facilitate the creation of massive amounts of content as well as enhancing its diversity. Furthermore, generations need to adhere to constraints from scenarios, which requires the use of Controllable Text Generation techniques. Finally, fictive writings need to be used to comply with legislation, reduce biases from past trainings and stereotypes towards populations from the operational analyst.

From the literature, we have extracted key unsolved challenges. These challenges are expressed by the ability to control text generation, fictive writings generation, preventing hallucinations and finally defining evaluation protocols to assess the generation adherence to constraints.

To automate both the generation of normal contents and reproduced disinformation campaigns, we have proposed a first lead to upscale Information definitions and adapt it to modelize Information Campaigns, first step to produce mass scale content. This idea leads us to a three-stage modeling used to reproduce information campaigns in order to do massive scale generation. This modeling is used with a prompting methodology, using SOTA Large Language Models to constraint generation and produce diverse, qualitative contents.

However, currently, this work is limited to structured campaigns and can not reproduce non-structured casual exchanges that are usually found in a social network (e.g day to day discussions about life, sports, etc.). Addressing non-structured casual exchanges add layers of complexities regarding topics discussed within the infosphere as well as the likelihood of individuals talking about said topics. Extensions are planned during this PhD to include these types of exchanges, enabling the production of contents unrelated to the scenario.

We highlight the need for further research in this direction, including but not limited to social graph modeling allowing imitations of typical social network behaviours, hallucination mitigation and evaluation protocols.

5 Ethic Statement

The stakes are high on the topic of text generation, with numerous potential misuses. To mitigate possible negative impacts of our work, we do not plan to release it in an uncontrolled fashion.

Measures will be taken to reduce the risks. All the work will be hosted within an air-gap environment to mitigate content leaking danger. Use of fictive writings will reduce

the risks of defamation, hate or harassment. To follow current regulations, all participants will be aware that the content is generated by Artificial Intelligence and that the purpose of this exercise is to fight disinformation.
Unintended risks are harder to measure and detect, but we believe that studying disinformational content and its impact in order to train operationals are among the best manners to fight it.
Last but not least, we plan to follow ethics recommendations in the domain as well as upcomings regulations to update our work to comply with in effect guidelines.

Notes

Dear Reviewers, we wanted to thank you for the thorough reviews done and the insightful commentaries. The main takeaways as we understood is that the paper is too light on explaining context. The main problem needs to be better defined and explained as well as the key points around it as pedagogy, legislations and globally who is the main target of the training. Thus, we provided extensive context on the points highlited by the reviewers remarks, trying to make it clearer. Please find below the logs of the changes done according to your remarks. Thanks again.

Changes

Reviewer	Section	Commentaries	Change
3	Abstract	l'abstract seul permet difficilement de comprendre le contexte de formation d'acteurs à la détection de campagne de désinformation.	Agree, adding content to Abstract
1	Introduction	La notion d'Infosphere et la modélisation proposées sont un peu floues au début du papier	Agree, adding content to Introduction
3	Introduction	These infospheres are used here to align with legislations, removing the risk of creating potential harmful content » n'est pas très évident pour un non-expert du domaine et mériterait d'être plus détaillée ou enrichie avec des exemples	Agree, Adding examples of what are the possible harmful contents generated
3	Introduction	Quelles sont les législations qui impactent la génération d'infosphères fictives ? Quels sont les enjeux et les obligations ?	Agree, adding examples of involved legislations
3	Introduction	le besoin de modélisation globale n'est pas très évident pour un non-expert du domaine. Si j'ai bien compris, il s'agit ici de modéliser des campagnes de désinformation et leurs acteurs mais cela pourrait être précisé dès l'introduction.	Agree, adding precision. The goal is to produce reproduced disinformation campaigns within a platform populated with "normal" content, hence generating both types of content.
3	Introduction	L'introduction mentionne plusieurs fois « this problem » sans que celui-ci soit clairement explicité. S'agit-il de la modélisation globale ? De la génération de texte ? Des deux aspects combinés ?	Agree, the first one is evaluation problematic, the second one is generation problematic, changes done in Introduction.
3	Introduction	Le lien entre ces deux phrases ne me paraît pas évident car la première me semble être liée à une population globale qui doit être éduquée pour réduire l'impact de la désinformation tandis que la deuxième s'intéresse à la formation d'analystes à détecter / combattre (?) la désinformation.	Agree, adding better transition

Reviewer	Section	Commentaries	Change
3	Related Works	La nécessité du multi-linguisme n'a pas été introduite précédemment et je me demande donc son besoin et son impact (ce qui est peut-être lié au manque d'une définition précise du problème et des acteurs visés)	Removal of this part
3	Related Works	La notion de barrière de l'âge pourrait également être davantage discutée car elle apparaît seulement ici sans être discutée	Removal of the part due to the focus on operational analysts
3	Related Works	Des hallucinations ne devraient donc pas faire des liens avec des informations réelles ? Peut-être que le rationnel derrière cette phrase mériterait d'être détaillés	Agree, adding context to explain the hallucination problem in 2.3
3	Related Works	Je ne suis pas sûr de comprendre le rationnel derrière cette phrase notamment l'utilisation de « our capabilities » dans un contexte de revue d'état de l'art.	Removal of the part
3	Methodology	Je pense qu'il manque une discussion sur la justification d'utiliser des LLM uniquement pour générer des individus et des messages. On pourrait envisager les utiliser pour générer artificiellement des acteurs locaux ou même globaux ?	Global And Local are input from the scenario, hence the goal is not to be generated. Adding further context in Methodology
3	Methodology	On pourrait également discuter de la difficulté pour des humains de modéliser le « world knowledge » ou la base « d'events ». Est-ce facile ? Peut-on réutiliser des ressources existantes ?	Agree, adding short explaination in Methodology
3	Conclusion	[...] Je me demande comment simuler des acteurs non-structurés et sans intention précise pour compléter l'infosphère et si les auteurs pourraient en dire quelques mots.	Agree, adding explaination in Conclusion

Table 1: Adressed Changes for the final revision of the paper

References

[1] Esma Aïmeur, Sabrine Amri, and Gilles Brassard. Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining*, 13(1):30, February 2023. Number: 1.

[2] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural Probabilistic Language Models. In Dawn E. Holmes and Lakhmi C. Jain, editors, *Innovations in Machine Learning: Theory and Applications*, Studies in Fuzziness and Soft Computing, pages 137–186. Springer, Berlin, Heidelberg, 2003.

[3] Paul Bernal. Fakebook: why Facebook makes the fake news problem inevitable. *Northern Ireland Legal Quarterly*, 69(4):513–530, December 2018. Number: 4.

[4] Sam Blazek. SCOTCH: a framework for rapidly assessing influence operations, 2021.

[5] Mato Brautovic and Romana John. Limitations of Fact-checking on Debunking Covid-19 Misinformation on Facebook: Case of Faktograf.hr. *Central European Journal of Communication*, 16:40–58, October 2023.

[6] Kathleen M. Carley. Social cybersecurity: an emerging science. *Computational and Mathematical Organization Theory*, 26(4):365–381, December 2020. Number: 4.

[7] Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. INSTRUCTEVAL: Towards Holistic Evaluation of Instruction-Tuned Large Language Models, June 2023. arXiv:2306.04757 [cs].

[8] Cheng-Han Chiang and Hung-yi Lee. Can Large Language Models Be an Alternative to Human Evaluation? 2023.

[9] Theodora Dame Adjin-Tettey. Combating fake news, disinformation, and misinformation: Experimental evidence for media literacy education. *Cogent Arts & Humanities*, 9(1):2037229, December 2022. Number: 1 Publisher: Cogent OA _eprint: https://doi.org/10.1080/23311983.2022.2037229.

[10] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. December 2019. arXiv: 1912.02164 Publisher: arXiv.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[12] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-Verification Reduces Hallucination in Large Language Models, September 2023. arXiv:2309.11495 [cs].

[13] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva, editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[14] Ullrich K. H. Ecker, Stephan Lewandowsky, John Cook, Philipp Schmid, Lisa K. Fazio, Nadia Brashier, Panayiota Kendeou, Emily K. Vraga, and Michelle A. Amazeen. The psychological drivers of misinformation belief and its resistance to correction. *Nature Reviews Psychology*, 1(1):13–29, January 2022. Number: 1.

[15] European Council. Faire face à la propagande, à la désinformation et aux fausses nouvelles - Des écoles démocratiques pour tous - www.coe.int, July 2023.

[16] Parliament European Parliament. Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and amending Directive 2000/31/EC (Digital Services Act) (Text with EEA relevance), October 2022. Legislative Body: EP, CONSIL.

[17] Cristina Garbacea and Qiaozhu Mei. Why is constrained neural language generation particularly challenging?, June 2022. Issue: arXiv:2206.05395 arXiv:2206.05395 [cs].

[18] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating Training Corpora for NLG Micro-Planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada, 2017. Association for Computational Linguistics.

[19] Leslie Gielow Jacobs. Freedom of Speech and Regulation of Fake News. *The American Journal of Comparative Law*, 70(Supplement_1):i278–i311, October 2022.

[20] Lucas Graves and Federica Cherubini. The Rise of Fact-Checking Sites in Europe. 2016.

[21] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling, August 2023. Issue: arXiv:2308.08998 arXiv:2308.08998 [cs].

[22] Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying Human and Statistical Evaluation for Natural Language Generation, April 2019. Issue: arXiv:1904.02792 arXiv:1904.02792 [cs, stat].

[23] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812, Halifax NS Canada, August 2017. ACM.

[24] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing Sentiment Bias in Language Models via Counterfactual Evaluation, October 2020. Issue: arXiv:1911.03064 arXiv:1911.03064 [cs].

[25] Jean-Baptiste Jeangène Vilmer. La lutte contre la désinformation russe : contrer la propagande sans faire de contre-propagande ? *Revue Défense Nationale*, 801(6):93–105, 2017. Number: 6 Publisher: Comité d'études de Défense Nationale.

[26] Frederick Jelinek, Robert L. Mercer, Lalit R. Bahl, and Janet M. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62, 1977.

[27] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, December 2023. Number: 12 arXiv:2202.03629 [cs].

[28] Sagar Joshi, Sumanth Balaji, Aparna Garimella, and Vasudeva Varma. Graph-based Keyword Planning for Legal Clause Generation from Topics. January 2023. arXiv: 2301.06901.

[29] Mihir Kale and Abhinav Rastogi. Text-to-Text Pre-Training for Data-to-Text Tasks, July 2021. Issue: arXiv:2005.10433 arXiv:2005.10433 [cs].

[30] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. CTRL: A Conditional Transformer Language Model for Controllable Generation. September 2019. arXiv: 1909.05858 Publisher: arXiv.

[31] Yuri Kim, Bin Han, Jihyun Kim, Jisoo Song, Seoyeon Kang, and Seongbin Park. A Quantitative Model to Evaluate Serendipity in Hypertext. *Electronics*, 10(14):1678, July 2021.

[32] Denis Kotkov, Alan Medlar, and Dorota Glowacka. Rethinking Serendipity in Recommender Systems. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pages 383–387, Austin TX USA, March 2023. ACM.

[33] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative Discriminator Guided Sequence Generation. September 2020. arXiv: 2009.06367 Publisher: arXiv.

[34] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A Diversity-Promoting Objective Function for Neural Conversation Models, June 2016. arXiv:1510.03055 [cs].

[35] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-LM Improves Controllable Text Generation. May 2022. arXiv: 2205.14217 Publisher: arXiv.

[36] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[37] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online, 2021. Association for Computational Linguistics.

[38] Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. Composable Text Controls in Latent Space with ODEs. August 2022. arXiv: 2208.00638.

[39] Jin Liu, Chongfeng Fan, Fengyu Zhou, and Huijuan Xu. Syntax Controlled Knowledge Graph-to-Text Generation with Order and Semantic Consistency. July 2022. arXiv: 2207.00719.

[40] Hongyuan Lu, Wai Lam, Hong Cheng, and Helen Meng. Partner Personas Generation for Dialogue Response Generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5200–5212, Seattle,

United States, 2022. Association for Computational Linguistics.

[41] Rida Lyammouri and Youssef Eddazi. Russian Interference in Africa: Disinformation and Mercenaries. June 2020.

[42] A.D. Madden. A definition of information. *Aslib Proceedings*, 52(9):343–349, November 2000. Number: 9.

[43] Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. Fine-Grained Analysis of Propaganda in News Article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019.

[44] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Putting Evaluation in Context: Contextual Embeddings Improve Machine Translation Evaluation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy, July 2019. Association for Computational Linguistics.

[45] Sadiq Muhammed T and Saji K. Mathew. The disaster of misinformation: a review of research in social media. *International Journal of Data Science and Analytics*, 13(4):271–285, May 2022. Number: 4.

[46] Julian Newman. Some Observations on the Semantics of "Information". *Information Systems Frontiers*, 3(2):155–167, June 2001. Number: 2.

[47] Ben Nimmo. Anatomy of an Info-War: How Russia's Propaganda Machine Works, and How to Counter It, May 2015. Section: Context.

[48] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, March 2022. Issue: arXiv:2203.02155 arXiv:2203.02155 [cs].

[49] James Pamment. The EU's Role in Fighting Disinformation: Crafting A Disinformation Framework. 2020.

[50] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Philadelphia, Pennsylvania, 2001. Association for Computational Linguistics.

[51] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, May 2023. Issue: arXiv:2305.18290 arXiv:2305.18290 [cs].

[52] Alireza Rezvanian and Mohammad Reza Meybodi. Stochastic graph as a model for social networks. *Computers in Human Behavior*, 64:621–640, November 2016.

[53] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating Pretrained Language Models for Graph-to-Text Generation, September 2021. Issue: arXiv:2007.08426 arXiv:2007.08426 [cs].

[54] Jon Roozenbeek and Sander van der Linden. Fake news game confers psychological resistance against online misinformation. *Palgrave Communications*, 5(1):1–10, June 2019. Number: 1 Publisher: Palgrave.

[55] Sander van der Linden Roozenbeek, Jon. Psychological Inoculation Against Fake News. In *The Psychology of Fake News*. Routledge, 2020. Num Pages: 23.

[56] Punyajoy Saha, Kanishk Singh, Adarsh Kumar, Binny Mathew, and Animesh Mukherjee. CounterGeDi: A controllable approach to generate polite, detoxified and emotional counterspeech, May 2022. Issue: arXiv:2205.04304 arXiv:2205.04304 [cs].

[57] Sebastian Schuetz, Tracy Sykes, and Viswanath Venkatesh. Combating COVID-19Fake News on Social Media through Fact Checking: Antecedents and Consequences, November 2021.

[58] Shadi Shahsavari, Pavan Holur, Tianyi Wang, Timothy R. Tangherlini, and Vwani Roychowdhury. Conspiracy in the time of corona: automatic detection of emerging COVID-19 conspiracy theories in social media and the news. *Journal of Computational Social Science*, 3(2):279–317, November 2020. Number: 2.

[59] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948. Conference Name: The Bell System Technical Journal.

[60] Kate Starbird, Ahmer Arif, and Tom Wilson. Disinformation as Collaborative Work: Surfacing the Participatory Nature of Strategic Information Operations. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–26, November 2019.

[61] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.

[62] Cass R. Sunstein. *Echo Chambers: Bush V. Gore, Impeachment, and Beyond*. Princeton University Press, 2001. Google-Books-ID: sEgHAAAACAAJ.

[63] SJ Terp and Pablo Breuer. DISARM: a Framework for Analysis of Disinformation Campaigns. In *2022 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 1–8, June 2022. ISSN: 2379-1675.

[64] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models, April 2016. arXiv:1511.01844 [cs, stat].

[65] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. Issue: arXiv:2302.13971 arXiv:2302.13971 [cs].

[66] Luis Vargas, Patrick Emami, and Patrick Traynor. On the Detection of Disinformation Campaign Activity with Network Analysis. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 133–146, Virtual Event USA, November 2020. ACM.

[67] Arjan Vreeken. Notions of Information: A Review of Literature. 2002.

[68] Claire Wardle. INFORMATION DISORDER : Toward an interdisciplinary framework for research and policy making, August 2023.

[69] Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. MEGATRON-CNTRL: Controllable Story Generation with External Knowledge Using Large-Scale Language Models, October 2020. arXiv:2010.00840 [cs].

[70] Kevin Yang and Dan Klein. FUDGE: Controlled Text Generation With Future Discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021.

[71] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: Story Writing With Large Language Models. In *27th International Conference on Intelligent User Interfaces*, pages 841–852, Helsinki Finland, March 2022. ACM.

[72] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending Against Neural Fake News, December 2020. Issue: arXiv:1905.12616 arXiv:1905.12616 [cs].

[73] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models. January 2022. arXiv: 2201.05337 Publisher: arXiv.

[74] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT, February 2020. Issue: arXiv:1904.09675 arXiv:1904.09675 [cs].

[75] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, November 2023. arXiv:2311.07911 [cs].

[76] Jianing Zhou and Suma Bhat. Paraphrase Generation: A Survey of the State of the Art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[77] Xinyi Zhou and Reza Zafarani. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys*, 53(5):1–40, September 2021. Number: 5 arXiv:1812.00315 [cs].

[78] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A Benchmarking Platform for Text Generation Models, February 2018. arXiv:1802.01886 [cs].

Détection de Communautés Floues et Chevauchantes via l'Analyse Formelle de Concepts

Martin WAFFO KEMGNE[1], Christophe DEMKO[1], Karell BERTET[1], Jean-Loup GUILLAUME[1]

[1] L3i, La Rochelle Université, La Rochelle, France

Résumé

Cet article introduit une nouvelle méthode pour identifier des communautés floues et chevauchantes dans les graphes via l'AFC (Analyse Formelle des Concepts). Alors que de nombreuses méthodes de détection de communautés existent, elles se concentrent généralement sur des structures non-chevauchantes, supposant qu'un nœud appartient à une seule communauté. Cette hypothèse est limitée face à la complexité des réseaux réels où des nœuds peuvent être partagés entre plusieurs communautés, créant des structures chevauchantes ou floues. Nous examinons d'abord les approches existantes et leurs limites. En réponse, nous adaptons l'AFC, traditionnellement utilisée pour des communautés non-chevauchantes, pour détecter des communautés floues et chevauchantes. L'efficacité de notre proposition est validée par des simulations sur des réseaux synthétiques et réels.

Mots-clés

Communautés chevauchantes, Communautés floues, AFC, graphes

Abstract

This paper introduces an innovative method for identifying fuzzy and overlapping communities in graphs through FCA (Formal Concept Analysis). While numerous community detection methods exist, they generally focus on non-overlapping structures, assuming a node belongs to only one community. This assumption is limited in the face of real network complexities where nodes can be shared among multiple communities, creating overlapping or fuzzy structures. We first examine existing approaches and their limitations. In response, we adapt FCA, traditionally used for non-overlapping communities, to detect fuzzy and overlapping communities. The effectiveness of our proposal is validated through simulations on synthetic and real networks.

Keywords

Overlapping communities, Fuzzy communities, FCA, graphs

1 Introduction

Dans une ère marquée par une prolifération exponentielle de données, la modélisation de systèmes complexes à travers les réseaux ou graphes s'est imposée comme un pilier fondamental dans la compréhension des structures sous-jacentes qui régissent les dynamiques sociales, biologiques, et technologiques. Ces réseaux, qui modélisent des connexions entre éléments ou acteurs, sont le reflet de la complexité intrinsèque des systèmes qu'ils représentent, allant des interactions entre protéines aux réseaux sociaux et aux infrastructures technologiques. Leur étude, ancrée dans les principes de la science des réseaux, dévoile non seulement la morphologie mais aussi les fonctionnalités émergentes de ces systèmes, offrant ainsi une vision plus intégrée et holistique de leur organisation [3].

Au cœur de cette exploration se trouve la notion de structure communautaire, une caractéristique omniprésente dans les réseaux qui manifeste la tendance des nœuds à s'agglomérer en groupes ou communautés, où les liens internes sont nombreux tandis que ceux intercommunautaires se font plus rares. L'étude des structures communautaires au sein des réseaux complexes a suscité une attention significative dans le domaine de la science des réseaux, offrant des aperçus profonds sur l'organisation, la dynamique et la fonction de divers systèmes réels [19]. Il est regrettable de reconnaître que la définition de ce qu'est une communauté reste peu précise et sujette à interprétation [8], chaque algorithme construisant ses fondements sur des idées intuitives variées.

Traditionnellement, les algorithmes de détection de communautés ont cherché à partitionner les réseaux en groupes distincts ou communautés, avec l'hypothèse sous-jacente que chaque nœud est membre d'une seule communauté [8]. Cette approche, bien qu'utile, échoue souvent à capturer les relations nuancées inhérentes à de nombreux réseaux complexes où les nœuds peuvent appartenir à plusieurs communautés, reflétant un scénario plus réaliste de chevauchement ou d'appartenances communautaires floues [33].

L'AFC [10] a émergé comme un outil puissant pour l'analyse de données et la représentation de connaissances, particulièrement dans le contexte de la détection de communautés non chevauchantes [2, 7, 9, 13, 15, 17, 31]. Cependant, l'application de l'AFC à la détection de communautés floues et chevauchantes reste largement inexplorée. Cette lacune dans la littérature présente une opportunité d'étendre l'utilisation de l'AFC au-delà de ses limites traditionnelles. En incorporant le concept d'ensembles flous, où l'appartenance communautaire de chaque nœud est caractérisée par un degré d'appartenance plutôt que par une classification binaire, nous pouvons mieux modéliser la complexité

et l'ambiguïté présentes dans les réseaux réels [34].
Dans cet article, nous proposons une adaptation de l'AFC pour la détection de communautés floues et chevauchantes. Notre approche s'appuie sur les travaux existants en AFC pour la détection de communautés non-chevauchantes [2, 7, 9]. Nous démontrons l'efficacité de notre méthode à travers des simulations étendues sur des réseaux artificiels d'une part et réels d'autre part, mettant en évidence sa capacité à révéler les structures communautaires chevauchantes que les méthodes traditionnelles pourraient négliger. De plus, les retombées de nos recherches touchent également le domaine de l'intelligence artificielle (IA), en particulier pour le *clustering*. Étant donné que nous étendons certaines approches basées sur l'AFC qui travaillaient sur des graphes non orientés et non pondérés, notre approche continue dans la même perspective en se limitant aux mêmes types de graphes. Cependant, celle-ci peut être facilement étendue aux cas de graphes orientés et pondérés.
La structure de cet article est organisée comme suit. La section 2 offre un panorama des travaux antérieurs concernant la détection de communautés non-chevauchantes et chevauchantes ; puis une présentation détaillée des approches existantes basées sur l'AFC et leurs limites est présentée en section 3. La section 4 détaille notre approche, introduisant la méthodologie et les innovations que nous proposons, ainsi que des simulations menées sur des réseaux artificiels et un *benchmark* reconnu. Enfin, nous concluons dans la section 5, en récapitulant les contributions principales de notre travail, ses implications et les perspectives futures de recherche.

2 Travaux antérieurs

La compréhension et l'analyse des structures communautaires dans les réseaux ont été des sujets d'intérêt croissant dans le domaine de la science des réseaux. Les travaux antérieurs ont exploré diverses méthodes pour détecter des communautés en distinguant entre les communautés chevauchantes et non chevauchantes.

2.1 Définitions préliminaires

Graphe : Un graphe $G = (V, E)$ est représenté par deux ensembles : V, qui correspond aux nœuds, et $E \subseteq V \times V$, qui correspond aux arêtes. Le nombre de nœuds dans le graphe est donné par $n = |V|$ et le nombre d'arêtes est donné par $m = |E|$.
Clique : Dans un graphe, une clique est identifiée comme un sous-ensemble de nœuds $M \subseteq V$ où chaque paire de nœuds est connectée par une arête. Une clique est dite maximale (CM) si elle ne peut être agrandie par l'ajout d'autres nœuds tout en restant une clique, ce qui implique qu'elle ne se situe pas intégralement à l'intérieur d'une clique de plus grande taille.
Partition : Une partition d'un graphe $G = (V, E)$ est une division de l'ensemble des nœuds V en plusieurs sous-ensembles non vides, appelés communautés, de telle sorte que chaque nœud appartient à exactement une communauté. Formellement, une partition est une collection $V_1, V_2, \ldots, V_k$ de sous-ensembles de V telle que $\bigcup_{i=1}^{k} V_i = V$ et $V_i \cap V_j = \emptyset$ pour tout $i \neq j$. Cette propriété assure qu'aucun nœud n'est laissé sans affectation à une communauté, et qu'il n'existe pas de chevauchement entre les communautés.
Partition floue : À l'opposé d'une partition classique, une partition floue permet à un nœud d'appartenir à plusieurs communautés simultanément, avec des degrés d'appartenance variables. Dans ce cadre, le degré d'appartenance α_{ic} d'un nœud i à une communauté c est un nombre réel compris entre 0 et 1, où 0 signifie qu'il n'appartient pas à la communauté et 1 signifie qu'il en est un membre à part entière. Une partition floue est alors définie par une matrice d'appartenance $A = [\alpha_{ic}]$ pour tous les nœuds $i \in V$ et toutes les communautés $c \in C$. Contrairement à la partition classique, où chaque nœud est strictement affecté à une seule communauté, la partition floue reflète la nature souvent nuancée des affiliations dans les réseaux sociaux, biologiques et autres systèmes complexes, permettant une représentation plus fidèle de la structure communautaire.
Dans la suite, nous noterons par C l'ensemble des communautés. Dans le cas flou particulièrement, pour une communauté $c \in C$ et $i \in V$, α_{ic} désigne le degré d'appartenance de i à la communauté c et $c(i)$ signifie que la communauté c contient le sommet i.

2.2 Détection de communautés non-chevauchantes

La détection de communautés non-chevauchantes est l'un des premiers problèmes abordés dans l'analyse des réseaux. Plusieurs algorithmes ont été proposés dans cette direction. L'un des plus anciens et historiques est l'algorithme de Newman et Girvan [21], et l'un des plus connus à nos jours est l'algorithme de Louvain développé par Blondel et al. [4]. De plus, un état de l'art a été proposé par Santo Fortunato [8] dans lequel il présente plus d'une cinquantaine de méthodes. Certaines se basent sur la maximisation d'une métrique spécifique, comme la modularité de Newman et Girvan [11], et d'autres méthodes se basent sur d'autres principes intuitifs.

2.2.1 Méthodes basées sur l'optimisation de la modularité

La modularité est une métrique généralement utilisée pour évaluer la qualité d'une partition des sommets en communautés. Elle a initialement été introduite par Newman et Girvan [11]. Elle compare le nombre de liens au sein des communautés au nombre attendu de ces liens dans un graphe aléatoire possédant le même nombre de nœuds, le même nombre de liens et la même distribution de degrés que le graphe initial. Elle se définit, pour un graphe $G = (V, E)$ et une partition (un ensemble de communautés) C de V, par :

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c(i), c(j)) \\ &= \frac{1}{2m} \sum_{c \in C} \sum_{i,j \in V_c} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \end{aligned} \quad (1)$$

Ici, A est la matrice d'adjacence, k_i est le degré du sommet i, m est le nombre d'arêtes dans le réseau, V_c est l'ensemble des sommets dans la communauté c et δ est le delta de Kronecker, égal à 1 si les nœuds i et j appartiennent à la même communauté, et 0 dans le cas contraire.

L'objectif des algorithmes utilisant cette métrique est de trouver la ou les partitions qui maximisent la modularité. Partant de ce principe, le problème de détection des communautés peut être vu comme un problème d'optimisation et il a été prouvé que la maximisation de la modularité est un problème NP-difficile [5]. Plusieurs algorithmes heuristiques pour approcher ce maximum ont été proposés [20] ; Parmi les algorithmes les plus utilisés, figure l'algorithme de Louvain particulièrement rapide [4].

Cependant, les méthodes basées sur la modularité souffrent d'un grand problème de résolution limite [8] ; en effet, elles ne permettent pas de déterminer les communautés d'une certaine taille particulière en fonction du graphe. De plus, d'autres problèmes inhérents aux méthodes basées sur la modularité, ont été discutés dans la littérature, comme le souligne Peixoto [24]. Néanmoins, il existe d'autres méthodes qui ne se basent pas sur l'optimisation de la modularité.

2.2.2 Autres Méthodes

Plusieurs autres algorithmes n'utilisant pas la modularité existent. Par exemple, l'algorithme *Infomap* [26] de Rosvall et Bergstrom, considéré comme l'un des algorithmes les plus efficaces et utilisés, ne se base pas sur l'optimisation de la modularité, mais plutôt sur le *map equation* et sur le concept de marche aléatoire dans un graphe. Un algorithme également largement connu est le *Label Propagation* [25], qui est une méthode simple et efficace pour détecter les communautés dans les réseaux. Il repose sur l'idée que les nœuds d'un réseau tendent à adopter les étiquettes (ou labels) qui sont majoritairement présentes dans leur voisinage immédiat.

D'autres méthodes, telles que les Modèles des Blocs Stochastiques (SBM), remplacent les méthodes basées sur la modularité pour l'analyse des réseaux, grâce à leur capacité à modéliser efficacement les structures communautaires complexes [14]. De même, d'autres méthodes basées sur le deep learning pour la détection des communautés sont discutées dans une revue exhaustive par Su et al. [30].

Pour en savoir plus sur d'autres méthodes, il faut consulter la revue de Santo Fortunato [8], qui présente une étude très détaillée d'un grand nombre d'algorithmes de détection des communautés. En dépit de toutes les méthodes citées précédemment, il en existe une dont nous n'avons pas parlé, qui est le cœur de cet article. Ces méthodes se basent sur la notion de cliques et d'AFC pour détecter les communautés [2, 7, 9, 15, 16, 17].

Ces méthodes, bien qu'efficaces pour identifier des structures communautaires, ignorent la possibilité qu'un nœud puisse appartenir à plusieurs communautés simultanément. A titre illustratif, en appliquant l'algorithme de Louvain au graphe de la figure 1, on obtient comme meilleure partition $\mathcal{P} = \{\{1, 2, 3, 4\}, \{a, b, c, d\}\}$, qui a une modularité de 0.41, ce qui est intuitif pour un bon algorithme de détection de communautés. Cependant, si on considère la figure 2, on constate qu'en appliquant la méthode de Louvain, la meilleure partition est soit $\mathcal{P}_1 = \{\{1, 2, 3, x\}, \{a, b, c\}\}$ soit $\mathcal{P}_2 = \{\{1, 2, 3\}, \{a, b, c, x\}\}$, avec la même modularité de 0.21. La méthode de Louvain étant non-déterministe, on obtiendra l'une ou l'autre de ces partitions dans lesquelles le nœud x peut se trouver dans différentes communautés. Ceci est intuitif et on peut observer que le nœud x ne possède pas de position très claire, dans le sens où il pourrait former une communauté en s'associant avec les nœuds de $C_1 = \{1, 2, 3\}$ ou de $C_2 = \{a, b, c\}$. On pourrait présager que le nœud x appartient en même temps à C_1 et C_2. Cependant, les méthodes non-chevauchantes comme Louvain sont limitées pour détecter ces types de communautés. Dans la section suivante, il sera question de l'exploration approfondie des méthodes et des techniques dédiées à la détection de communautés chevauchantes.

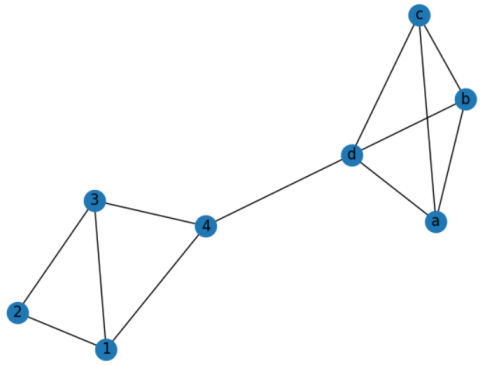


FIGURE 1 – Graphe avec deux communautés clairement identifiables

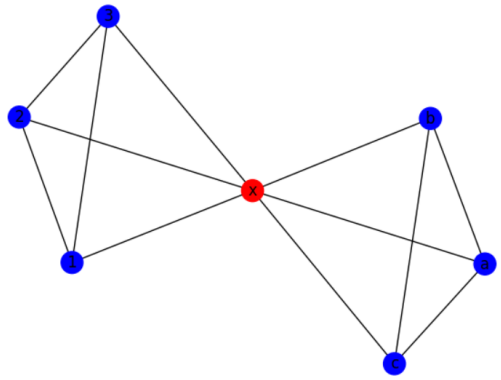


FIGURE 2 – Graphe avec deux communautés floues et un sommet ayant le même degré d'appartenance à deux communautés

2.3 Détection de communautés chevauchantes

Malgré l'efficacité prouvée des méthodes citées précédemment pour identifier des communautés non chevauchantes,

elles semblent insuffisantes pour saisir la complexité des interactions dans de nombreux domaines. Prenons les réseaux sociaux, où une personne peut tisser des liens avec divers groupes, ou la recherche interdisciplinaire, qui voit les chercheurs naviguer à travers plusieurs champs d'études. En écologie, les espèces interagissent avec de multiples écosystèmes, tandis que dans le domaine médical, les symptômes des patients peuvent recouvrir plusieurs pathologies. Ces scénarios illustrent des environnements où les entités ne se cantonnent pas à des catégories uniques et bien délimitées.

Reconnaissant ces limitations, des recherches plus récentes se sont concentrées sur la détection de communautés chevauchantes. Les premières approches sont celles basées sur le *line graph* [1, 6]. Il existe aussi des méthodes qui se fondent sur l'extension ou la contraction de graines [32].

Une autre méthode, parmi les plus connues ne passant pas par le *line graph*, est l'algorithme CPM (*Clique Percolation Method*) [23]. Plusieurs autres méthodes de détection de communautés chevauchantes existent, particulièrement les communautés floues [12] où, en plus d'être chevauchantes, un degré d'appartenance à une communauté est associé à chaque nœud. Plus précisément, le degré d'appartenance d'un sommet v à une communauté c est généralement exprimé par un coefficient d'appartenance α_{vc}, un nombre réel compris entre 0 et 1, de sorte que la partition C soit floue et tel que, pour chaque v,

$$\forall v \in V,\ \sum_{c \in C} \alpha_{vc} = 1.$$

À cet effet, plusieurs versions modifiées de la modularité ont été proposées par différents auteurs [18, 22, 28, 29] pour étendre la modularité de Newman et Girvan [11], mais celle que nous utiliserons dans cet article est celle de Shen et al. [28], ayant pour expression :

$$Q = \frac{1}{2m} \sum_{c \in C} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \alpha_{ic} \alpha_{jc}.$$

Cette expression est une extension de l'équation 1, où $\delta(c(i), c(j))$ est remplacé par $\alpha_{ic}\alpha_{jc}$. Pour voir à quel point cette extension est pertinente, considérons encore notre figure 2, on peut remarquer que si nous considérons la partition floue $\mathcal{P} = \{\{(1;1),(2;1),(3;1),(x;0.5)\},\{(a;1),(b;1),(c;1),(x;0.5)\}\}$, qui signifie intuitivement que le nœud x appartient simultanément aux communautés $\tilde{C_1} = \{1,2,3,x\}$ et $\tilde{C_2} = \{a,b,c,x\}$, nous obtiendrons une modularité de 0.25 qui est supérieure à 0.21, qui est la valeur obtenue en appliquant la modularité floue aux partitions $\mathcal{P}_1 = \{\{(1;1),(2;1),(3;1),(x;1)\},\{(a;1),(b;1),(c;1)\}\}$ ou $\mathcal{P}_2 = \{\{(1;1),(2;1),(3;1)\},\{(a;1),(b;1),(c;1),(x;1)\}\}$ (qui représentent en réalité les partitions $\mathcal{P}_1 = \{\{1,2,3,x\},\{a,b,c\}\}$ ou $\mathcal{P}_2 = \{\{1,2,3\},\{a,b,c,x\}\}$ respectivement) de la figure 2 obtenues par l'algorithme de Louvain, d'où la pertinence de ces communautés floues. Ainsi, de nombreuses méthodes de détection de communautés, qu'elles soient recouvrantes ou non, existent dans la littérature. Dans la section suivante, nous présenterons en détail celles se basant sur l'AFC.

3 Détection de communautés à base de l'AFC

L'AFC [10] est une méthode d'analyse de données et de représentation des connaissances qui traite une information sous forme de hiérarchies de concepts construites à partir de relations binaires. Elle utilise deux formalismes importants : la notion de contexte formel et de concept formel. L'AFC a été utilisée dans plusieurs études pour la détection de communautés non-chevauchantes, offrant un cadre rigoureux basé sur la théorie des treillis pour identifier des groupes d'éléments partageant des attributs communs [2, 7, 9]. Dans l'AFC, la détection des communautés peut se résumer à la recherche des attributs cachés partagés par certains objets. Il est question dans cette section de présenter les approches utilisées dans [2, 7, 9]. De manière historique, Freeman a été le pionnier en 1996 dans l'utilisation de l'AFC pour analyser les réseaux sociaux et identifier les communautés en se concentrant sur les cliques maximales chevauchantes [9]. Néanmoins, cette approche écartait les acteurs des cliques intermédiaires. Falzon, en 2000, a proposé une amélioration en incluant ces acteurs dans l'analyse sans les exclure [7]. Ali en 2014 a poussé plus loin cette idée en intégrant tous les acteurs du réseau, même ceux n'appartenant pas aux cliques maximales [2]. Une extension de la même approche se trouve pour les graphes orientés dans [31]. Toutefois, son application à la détection de communautés chevauchantes et floues demeure peu explorée. En effet, comme on peut le voir pour le graphe 2, il y a intuitivement au moins deux communautés. Notre objectif est de développer une méthode basée sur l'AFC qui surmonte ces limitations en révélant les communautés floues chevauchantes cachées.

3.1 Définitions préliminaires

En entrée, l'AFC utilise un contexte formel pour représenter l'ensemble des objets selon les attributs qu'ils possèdent. Ces attributs sont binaires, indiquant pour un attribut donné si un objet le possède ou non.

Contexte Formel : Un contexte formel est un triplet $\mathbb{K} = (G, M, I)$ où :

- G est un ensemble fini d'objets,
- M est un ensemble fini d'attributs,
- $I \subseteq G \times M$ est une relation d'incidence qui spécifie quels objets possèdent quels attributs.

Opérateurs de Dérivation et Connexion de Galois

L'AFC définit deux opérateurs de dérivation, $\alpha : 2^G \to 2^M$ et $\beta : 2^M \to 2^G$, formant une connexion de Galois entre les ensembles 2^G et 2^M. Ces opérateurs sont définis comme suit :

- $\alpha(A) = \{a \in M | \forall o \in A, (o,a) \in I\}$ pour tout $A \subseteq G$,
- $\beta(B) = \{o \in G | \forall a \in B, (o,a) \in I\}$ pour tout $B \subseteq M$.

L'opérateur α associe à un ensemble d'objets l'ensemble de

tous les attributs partagés par ces objets, tandis que β associe à un ensemble d'attributs l'ensemble de tous les objets qui les partagent.
Concept Formel : Un concept formel est une paire (A, B), avec $A \subseteq G$ et $B \subseteq M$, qui satisfait $\alpha(A) = B$ et $\beta(B) = A$. A est appelé l'extension du concept et B son intension.
Les concepts formels, organisés selon la relation $(A_1, B_1) \geq (A_2, B_2) \Leftrightarrow A_1 \supseteq A_2 \Leftrightarrow B_2 \supseteq B_1$, forment un treillis complet muni de cette relation d'ordre.

3.2 Les méthodes de Freeman et Falzon pour la détection de communautés

Comme nous l'avons mentionné plus haut, la méthode de Falzon [7] est une amélioration de la méthode de Freeman [9], qui prend en compte les acteurs des cliques intermédiaires dans la phase de détection.

3.2.1 Définition de la notion de chevauchement

Nous généralisons la définition des ensembles de nœuds qui se chevauchent de Freeman pour les ensembles de nœuds c-chevauchants. Il définit un ensemble de nœuds comme étant l'ensemble des acteurs représentés par un nœud particulier du treillis de Galois. Il rappelle que, au niveau 1 du treillis, les ensembles de nœuds correspondent aux ensembles de cliques maximales (CM). En substituant "ensemble de nœuds" à "clique", Freeman donne la définition suivante.
Soit $L_k = \{n_1, n_2, \ldots, n_m\}$ l'ensemble des ensembles de nœuds dans le niveau k du treillis, qui comprend m nœuds. Falzon définit une relation binaire *chevauchement*, $o \subseteq L_k \times L_k$ telle que :

(i) $(n_i, n_i) \in o$ — chaque ensemble de nœuds se chevauche lui-même, donc o est réflexive.

(ii) $n_i \cap n_j \neq \emptyset \Rightarrow (n_i, n_j) \in o$ — deux ensembles de nœuds ayant un élément commun se chevauchent, donc o est symétrique.

(iii) $(n_i, n_j) \in o$ et $(n_j, n_k) \in o \Rightarrow (n_i, n_k) \in o$ — la relation de chevauchement est transitive.

En suivant la même idée d'ensembles de nœuds qui se chevauchent, nous la modifions légèrement pour obtenir une notion plus généralisée que nous appelons ensembles de nœuds *c-chevauchants*.
La relation binaire *c-chevauchement*, $o_c \subseteq L_k \times L_k$, est définie comme suit :

(i) $(n_i, n_i) \in o_c$ — chaque ensemble de nœuds se chevauche lui-même, donc o_c est réflexive.

(ii) $|n_i \cap n_j| > c \Rightarrow (n_i, n_j) \in o_c$ — deux ensembles de nœuds ayant plus de c éléments en commun se chevauchent, donc o_c est symétrique.

(iii) $(n_i, n_j) \in o_c$ et $(n_j, n_k) \in o_c \Rightarrow (n_i, n_k) \in o_c$ — la relation de c-chevauchement est transitive.

Notons que pour $c = 0$, le *0-chevauchement* coïncide avec le *chevauchement*. Il est clair que o_c est une relation d'équivalence et partitionne L_k en sous-ensembles (disjoints pour $c = 0$) qui forment la base des groupes de niveau k, qui sont les communautés.

3.2.2 Principe de la méthode de Freeman

Soit $G = (V, E)$ un graphe et $C = \{C_1, C_2, ..., C_n\}$ l'ensemble des cliques maximales de taille au moins 3 du graphe G. La première étape consiste à construire le contexte $\mathbb{K} := (V, C, I)$ où I est la relation binaire définie par : pour tout $x \in V$ et $M \in C$, $I(x, C_i) = 1$ si le nœud x appartient à la clique maximale C_i et $I(x, C_i) = 0$ sinon, $\forall i \in \{1, 2, ..., n\}$.
La deuxième étape consiste à construire le treillis de concept associé au contexte précédent. Ensuite, pour détecter des communautés, Freeman [9] se base sur la notion de chevauchement de CM dans le treillis de Galois. Il détermine, à partir du treillis de Galois du contexte précédemment construit, un ensemble de CM où au moins deux chemins allant de leur position courante dans le treillis vers l'infimum ne sont pas de la même longueur, qu'il appelle cliques intermédiaires. C'est le cas, par exemple, avec la figure 5 où la clique $C5$ en rouge est considérée comme intermédiaire car elle possède des chemins vers l'infimum (traits rouges dans la figure 5) qui ne sont pas de même longueur. Ensuite, il procède à l'élimination des arêtes partant de ces nœuds pour obtenir des groupes disjoints (en vert sur la figure 5). Ces groupes représentent les communautés. Pour les obtenir, il suffit d'appliquer la relation de chevauchement sur les CM restantes après l'élimination des CM intermédiaires ; ces CM sont au premier niveau du treillis. Les classes d'équivalence pour cette relation représentent les communautés (en faisant bien sûr l'union des éléments ou CM de la même classe d'équivalence).
Pour le graphe de la figure 3, par exemple, le contexte est donné par la figure 4 et le treillis par la figure 5. Après l'élimination de $C5$, les communautés obtenues sont $C1 \cup C2 \cup C3 \cup C4 = \{a, b, c, d, e\}$, $C6 \cup C7 = \{1, 2, 3, 4\}$, et $C8 = \{A, B, C\}$. Cependant, dans de grands réseaux, il peut arriver qu'il y ait plusieurs CM intermédiaires et, souvent, en les éliminant, certains acteurs du réseau sont également éliminés. C'est la raison pour laquelle Falzon [7] a proposé un autre formalisme pour pallier cette limite et prendre en compte les acteurs des CM intermédiaires.

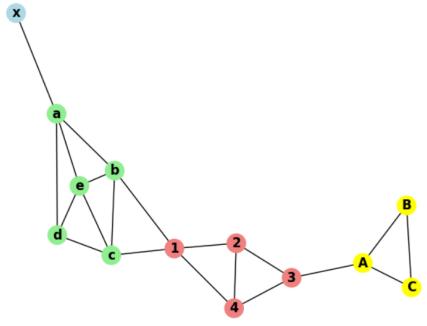


FIGURE 3 – Graphe ayant une clique intermédiaire

3.2.3 Principe de la méthode de Falzon

L'étape de construction du contexte est similaire à celle décrite dans la sous-section 3.2.2.

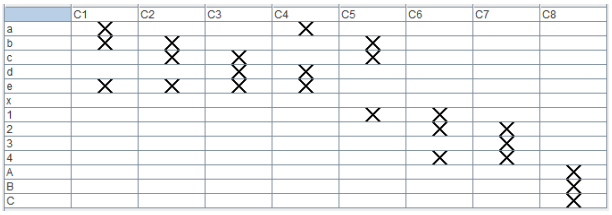


FIGURE 4 – Contexte du Graphe ayant une clique intermédiaire

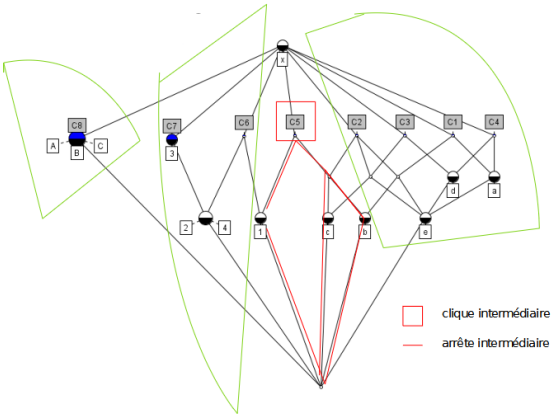


FIGURE 5 – Treillis du Graphe ayant une clique intermédiaire

La deuxième étape consiste à construire le treillis de concept associé au contexte précédent. La construction du treillis de concepts, basée sur l'algorithme 1 de Falzon [7], débute avec l'ensemble des CM noté L_1. Pour chaque niveau k, un nouvel ensemble L_k est formé par les intersections des paires de nœuds du niveau précédent $k-1$. Parallèlement, pour chaque niveau k, une liste spécifique $LS[k]$ est constituée, regroupant les nœuds absents des niveaux supérieurs à k. Enfin, une évaluation comparative entre les niveaux adjacents du treillis permet d'identifier et d'éliminer les nœuds communs au niveau supérieur, affinant ainsi la structure du treillis.

Dans la phase finale, Falzon [7] propose une méthode pour détecter les communautés en exploitant les propriétés des cliques maximales au sein du treillis de concepts. L'algorithme 2 génère, pour chaque niveau du treillis, des groupes G_k correspondant à des classes d'équivalence basées sur la relation de chevauchement entre les éléments d'un même niveau particulier du treillis, qui sont interprétés comme des communautés. Cette approche permet une détection efficace et intuitive des structures communautaires dans le réseau tout en prenant en compte les nœuds qui appartiennent aux cliques intermédiaires, ce qui n'était pas le cas avec la méthode de Freeman [9] qui éliminait les nœuds des cliques intermédiaires.

Algorithm 1 Construction d'un treillis de Galois à partir des cliques du réseau

Formez L_1 à partir des cliques du réseau, c'est-à-dire, chaque ensemble de nœuds représente une clique.
Initialisez $k := 1$.
repeat
  $k := k + 1$.
  $L_k := \{\}$.
  Effectuez l'intersection par paires sur chaque paire d'ensembles de nœuds dans L_{k-1}.
  Ajoutez tous les ensembles de nœuds non vides à L_k.
  for all ensembles de nœuds n_i dans L_k **do**
    if n_i est un sous-ensemble de tout autre ensemble de nœuds n_j dans L_k **then**
      Retirez n_i de L_k.
    end if
  end for
  $U := \bigcup_{n_j \in L_{k-1}} n_j$.
  L'ensemble d'étiquettes de nœuds devient $n_i - U$.
  Construisez une liste d'ensembles des nœuds de L_{k-1}, qui ont des ensembles d'étiquettes de nœuds non vides, nommez cet ensemble $LS[k-1]$.
until L_k est vide (nœud inférieur du treillis de Galois).
$Maxniveau := k$.

Algorithm 2 Algorithme de la structure de groupe selon la méthode de Falzon

$LS[0] \leftarrow \emptyset$
for $i \leftarrow 0$ to $Maxniveau - 1$ **do**
  $L \leftarrow L_{i+1} \cup LS[1] \cup LS[2] \cup \ldots \cup LS[i]$;
  $k \leftarrow 1$;
  while L non vide **do**
    Soit n le premier ensemble nœud de L;
    $GS \leftarrow n$;
    Déterminez tous les ensembles nœud n_j, tels que $(n, n_j) \in \theta$ (fonction de chevauchement) ;
    $G_k \leftarrow \bigcup n_j$; $GS \leftarrow \{n_j\}$;
    $L \leftarrow L - GS$;
    Ajoutez G_k à la liste de groupes pour le niveau i;
    $k \leftarrow k + 1$;
  end while
  $i \leftarrow i + 1$;
end for

La méthode a été appliquée sur plusieurs graphes et a donné de bons résultats ; voir [7] pour plus d'informations. Malgré les différentes améliorations de cette méthode, son utilisation pour les communautés floues reste inexplorée. C'est l'objet de la section suivante.

4 Approche proposée

Si les méthodes traditionnelles basées sur l'AFC sont efficaces pour identifier des communautés non chevauchantes, elles semblent insuffisantes pour détecter les communautés chevauchantes. En particulier, la méthode de Falzon [7], bien que révolutionnaire à son époque, rencontre des limites face à de nombreuses configurations. Considérons le graphe 6 ; il est intuitif de postuler l'existence d'au moins deux communautés.

Or, la méthode de Falzon, appliquée à ce même graphe, ne parvient pas à déceler ces communautés. Face à ce constat, notre objectif est de proposer une nouvelle méthode qui pallie ces insuffisances et qui soit donc capable de détecter les communautés floues chevauchantes, offrant ainsi une lecture plus fidèle de la structure complexe des réseaux concernés ; une étude comparative est donnée dans le tableau 1.

4.1 Principe de la méthode

Le principe de la méthode se rapproche de celui de Falzon [7].

Soit $G = (V, E)$ un graphe et $C = \{C_1, C_2, ..., C_n\}$ l'ensemble des cliques maximales du graphe. La première étape consiste à construire le contexte $\mathbb{K} := (V, C, I)$ en suivant le même principe qu'à la sous-section 3.2.3. Donc l'étape de construction du treillis est identique à celle de [7]. La deuxième étape est celle de génération des communautés.

1^{er} **cas :** Si pour tout $C_i, C_j \in C$ tels que $i \neq j$, $C_i \cap C_j = \emptyset$, alors l'ensemble des communautés est $\pi = C = \{C_1, C_2, ..., C_n\}$.

2^{eme} **cas :** Dans le cas contraire, il existe $C_i, C_j \in C$ tel que $i \neq j$ et $C_i \cap C_j \neq \emptyset$. Déterminer dans un premier temps le cardinal de chaque élément non vide du treillis. Soit c le plus petit cardinal ; $c \geq 1$, par définition. Soit $\mathcal{B}$ le treillis de concept précédemment construit.

Ensuite, appliquer l'algorithme de Falzon à notre treillis $\mathcal{B}$, où la fonction de chevauchement a été modifiée par le *c-chevauchement*.

Ainsi, les communautés qu'on doit obtenir sont chevauchantes. Dans la phase finale, il est question de donner le degré d'appartenance d'un élément à une communauté. Pour cela, supposons qu'à un niveau particulier de $\tilde{L}$, nous obtenons une famille de communautés $\pi = \{K_1, K_2, ..., K_m\}$. Alors pour un $K_i \in \pi$ et $x \in K_i$, le degré d'appartenance de x à K_i est donné par $\alpha_{xK_i} = \frac{card(\{y \in K_i \mid (x,y) \in E\})}{deg_{\tilde{G}}(x)}$, où $deg_{\tilde{G}}(x)$ désigne le degré de x dans le sous-graphe $\tilde{G}$, où $\tilde{G}$ désigne la restriction du graphe G aux nœuds appartenant à au moins une clique maximale.

4.2 Exemple d'illustration

Dans tous les exemples qui suivent nous nous sommes limités à extraire uniquement les groupes du premier niveau. Pour des graphes plus grands, il pourrait cependant être nécessaire d'explorer les niveaux supérieurs pour avoir des communautés plus fines.

Exemple 1 : Graphe artificiel

Dans cet exemple, nous appliquons notre méthode au graphe 6. Comme nous l'avons souligné précédemment, la méthode de Falzon ne trouve aucune communauté. Pourtant, intuitivement, elles existent. La première étape consiste à générer le treillis de concept $\mathcal{B}$ associé à ce graphe, que nous avons représenté à la figure 7. En appliquant l'algorithme 2 au treillis 7, nous obtenons comme communautés la famille $\pi_3 = \{\{1, 2, 3, 4, x\}, \{a, b, c, x\}\}$ dont nous pouvons associer la version floue $\tilde{\pi_3} = \{\{(1;1), (2;1), (3;1), (4;1), (x;4/7)\}, \{(a;1), (b;1), (c;1), (x;3/7)\}\}$, ce qui est plus intuitif. Pour se convaincre de la pertinence de notre approche, les partitions trouvées par la méthode de Louvain [4] ont une modularité floue de 0.250. Pourtant, en appliquant la modularité floue [28] à la famille trouvée avec notre approche, on obtient $0.254 > 0.250$. D'où une éventuelle supériorité de notre approche à détecter des attributs cachés.

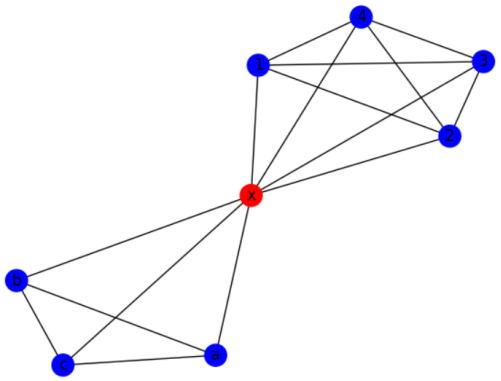


FIGURE 6 – Graphe avec communautés floues et différents degrés d'appartenance

Exemple 2 : Graphe artificiel

Considérons le graphe 8. La méthode de Falzon [7] et de Louvain [4] détectent 4 communautés ; $\{1, 2, 3, 4\}$, $\{I, II, III, IV\}$, $\{a, b, c, d\}$, et $\{A, B, C, D\}$. Cependant, notre approche, en plus de détecter ces 4 communautés, détecte une $5^{\text{ème}}$ communauté cachée $\{1, a, I, A\}$, qui n'était pas détectable par les approches précédentes.

Exemple 3 : Graphe artificiel

En faisant de même avec le graphe 9. La méthode de Falzon [7] détecte 5 communautés : $\{1, 2, 3, 4\}$, $\{I, II, III, IV\}$, $\{a, b, c, d\}$, $\{A, B, C, D\}$ et $\{x\}$, avec une communauté constituée d'un seul élément. La méthode de Louvain [4], quant à elle, détecte 4 communautés : $\{1, 2, 3, 4\}$, $\{I, II, III, IV\}$, $\{a, b, c, d\}$, et $\{A, B, C, D\}$, en plaçant x dans une des quatre communautés de manière

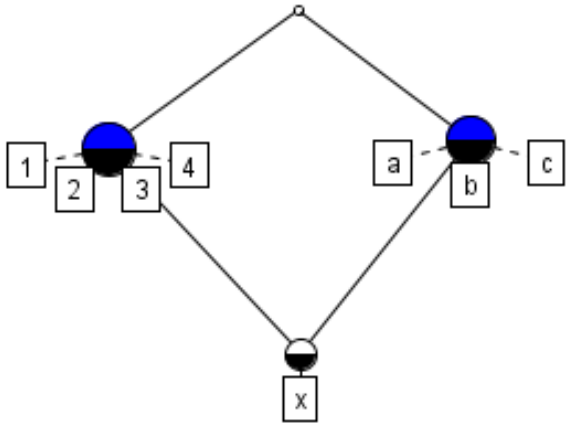


FIGURE 7 – Treillis de concept de L

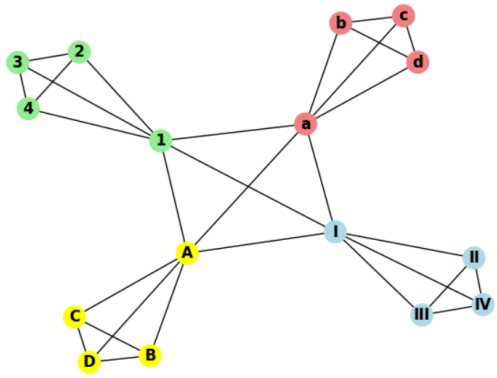


FIGURE 8 – Graphe avec une communauté cachée

non-déterministe. Mais le seul élément qui peut changer de communauté est x, et le nombre de communautés est fixe et égal à 4 à chaque fois. Cependant, notre approche, en plus de détecter les 4 communautés détectées par la méthode de Falzon, détecte en plus une $5^{\text{ème}}$ communauté cachée $\{1, a, I, A, x\}$, ce qui n'était pas détectable par les approches précédentes. De plus, ces communautés sont très intuitives. Pour ce cas particulier, le fait que la méthode de Falzon [7] détecte un singleton ($\{x\}$) comme communauté est un très grand défaut, car une communauté est intuitivement supposée posséder au moins deux éléments. D'autre part, la méthode de Louvain ne parvient pas à trouver cette communauté cachée, ce qui cause l'instabilité ou la fluctuation de x en fonction de l'ordre d'exécution.

Exemple 4 : Graphe réel

Afin d'illustrer notre approche sur un exemple réel, considérons les données du monastère de Sampson telles que décrites par Freeman [9]. Dans son étude de ces données, Sampson [27] identifie trois groupes qu'il nomme *Loyal Opposition (LO)*, *Young Turks (YT)* et *Outcasts (O)*, définis respectivement par : $LO = \{4, 5, 6, 8, 9, 10, 11\}$, $YT = \{1, 2, 7, 12, 14, 15, 16\}$ et $O = \{3, 17, 18\}$. De plus, Sampson décrit des liens entre ces groupes. Un novice, le numéro 13, n'était clairement assignable à aucun groupe ; il était considéré comme un "Hésitant" naviguant entre **LO** et **O**.

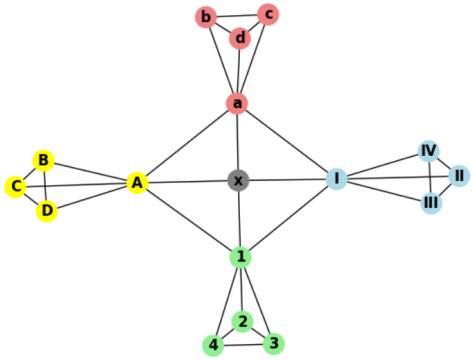


FIGURE 9 – Graphe avec une communauté cachée non-singleton

Nous nous appuyons sur l'étude réalisée par Freeman [9] sur ces données. Le processus d'extraction du graphe, ainsi que le treillis 10 généré avec lequel nous travaillons, se trouvent dans [9].

Ensuite, Falzon [7] a également appliqué sa méthode et a obtenu des communautés presque similaires à celles de Freeman.

En effet, en raison du principe utilisé, Freeman détecte les communautés **LO**, **YT** et **O** et exclut le nœud 13 lors de la détection, de sorte que 13 n'appartient à aucune communauté. Tandis que la méthode de Falzon [7] détecte toutes les communautés **LO**, **YT** et **O** identifiées par Freeman, mais ajoute une communauté supplémentaire, $\{13\}$, ce qui n'est pas très intéressant car cela n'apporte pas beaucoup d'informations sur le nœud 13. Ainsi, les communautés détectées par les approches de Freeman et Falzon sont cohérentes avec le rapport de Sampson [27], mais ne parviennent pas à assigner le nœud 13 à l'une des communautés. Cependant, notre approche détecte les communautés $LO = \{4, 5, 6, 8, 9, 10, 11\}$, $YT \cup \{13\} = \{1, 2, 7, 12, 13, 14, 15, 16\}$ et $O \cup \{13\} = \{3, 13, 17, 18\}$. Comme nous pouvons le voir, le fait que notre méthode soit capable de détecter des communautés chevauchantes est très cohérent avec le rapport de Sampson et se distingue en outre par le fait qu'elle détecte le nœud 13 comme un élément flou appartenant simultanément à **YT** et **O**, ce qui est très conforme à l'affirmation de Sampson selon laquelle 13 était un hésitant naviguant entre **LO** et **O**. Ainsi, notre approche est capable de révéler des informations ou attributs cachés que les approches précédentes ne pouvaient pas dévoiler, car aucune des méthodes de Falzon ou de Freeman n'était capable de révéler une position cachée du nœud 13 de la manière dont notre approche l'a fait.

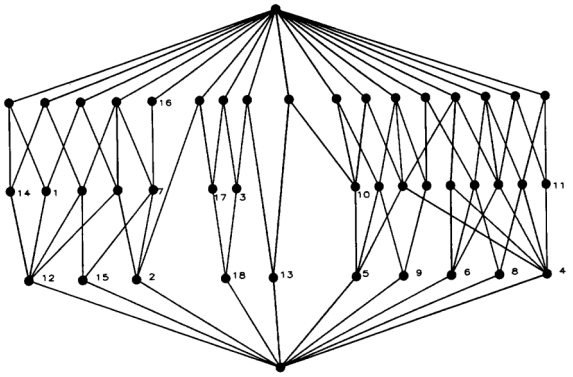


FIGURE 10 – Treillis des données de Sampson généré par Freeman [9]

4.3 Garanties offertes par notre approche

Notre approche, comparée à plusieurs autres approches, offre plusieurs garanties comme on peut le voir dans le tableau 1.

Chevauchante : Le fait que notre approche détecte des communautés chevauchantes lui permet de trouver certaines communautés cachées, comme on l'a vu dans les exemples plus haut, ce qui n'était pas le cas avec les approches existantes comme celle de Louvain et celles se basant sur l'AFC, comme l'approche de Freeman, Falzon et Ali. De plus, elle doit hériter de la plupart des avantages des communautés chevauchantes en général.

Déterminisme : L'aspect déterministe de notre méthode lui confère un caractère canonique, ce qui est particulièrement bénéfique en science. En effet, la possibilité de garantir l'unicité des solutions permet une caractérisation plus précise. Ce n'est pas le cas avec la méthode de Louvain, qui bien que rapide, n'est pas déterministe.

Non-paramétrique : Notre méthode étant non-paramétrique, assure qu'aucun type de communauté ne sera privilégié en fonction d'un paramètre, alors que CPM, par exemple, prend également un paramètre k en entrée et les types de communautés retournés vont dépendre du k, ce qui n'est pas toujours avantageux, car cela implique que pour certaines configurations complexes du graphe, il faut exécuter plusieurs fois en changeant les valeurs de k afin de voir les différentes communautés possibles, puis décider ce que l'on veut faire des différents résultats. Cela peut être très coûteux ou pas du tout très intéressant.

Clique :

L'utilisation des cliques, au lieu de la modularité, garantit que notre méthode ne souffre pas de certains problèmes intrinsèques à la modularité comme les problèmes de résolution limites énoncés plus haut. Les méthodes basées sur la modularité, telles que la méthode de Louvain standard, ne peuvent pas détecter des communautés d'une certaine taille particulière en fonction de la configuration du graphe à cause de ce problème de résolution limite.

TABLE 1 – Aperçu des garanties fournies par notre méthode en comparaison avec certaines approches existantes.

Critères	Ch	De	To	Cl	Mo	N-Pa
Freeman		✓		✓		✓
Falzon		✓		✓		✓
Ali		✓	✓	✓	✓	✓
Louvain			✓		✓	✓
Clique percolation	✓		✓		✓	
Approche proposée	✓	✓		✓		✓

Légende : Ch : Chevauchante, To : Total, Cl : Clique, Mo : Modularité, De : Déterministe, N-Pa : Non-paramétrique

5 Conclusion

Dans cet article, nous avons présenté une méthode nouvelle pour la détection de communautés floues et chevauchantes au sein de réseaux complexes grâce à l'AFC. Cette approche permet d'améliorer les méthodes traditionnelles en tenant compte de la nature multi-facettes des réseaux sociaux et d'autres systèmes complexes, où les entités peuvent simultanément appartenir à plusieurs groupes. Les tests effectués sur des réseaux à la fois synthétiques et réels attestent de l'efficacité de notre méthode. Cette recherche pave la voie à une compréhension approfondie des structures communautaires dans une variété de domaines, y compris dans le secteur de l'intelligence artificielle, où elle trouve des applications potentielles telles que le clustering. De plus, notre étude propose plusieurs axes de recherche futurs. Notamment, nous avons identifié qu'un graphe peut présenter des nœuds n'appartenant à aucune clique maximale. C'est le cas du nœud x dans le graphe de la figure 3. Ces derniers ne sont pas pris en compte dans notre modèle actuel. Une amélioration future consisterait donc à intégrer tous les nœuds du graphe en développant une méthode pour assigner ceux n'appartenant à aucune clique maximale à des communautés cohérentes, c'est la raison pour laquelle nous avons spécifié que notre méthode n'est pas totale dans le tableau 1. En outre, une analyse approfondie de la complexité algorithmique de notre méthode s'avère cruciale. De même, quelques extensions de notre méthode pourraient s'appliquer aux cas des graphes orientés, pondérés et dynamiques. Cela permettrait de généraliser notre approche et de l'adapter à une plus grande variété de réseaux complexes, ouvrant ainsi la voie à de nouvelles applications et améliorations.

Remerciements

Ce travail de recherche est supporté par l'Agence Nationale de la Recherche à travers le projet "SmartFCA - ANR-21-CE23-0023 : Analyse Formelle de Concepts : un outil intelligent pour l'analyse de données complexes"

Références

[1] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in

networks. *nature*, 466(7307) :761–764, 2010.

[2] Selmane Sid Ali, Fadila Bentayeb, Rokia Missaoui, and Omar Boussaid. An efficient method for community detection based on formal concept analysis. In *Foundations of Intelligent Systems : 21st International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, 2014. Proceedings 21*, pages 61–72. Springer, 2014.

[3] Albert-László Barabási. Linked : The new science of networks, 2003.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, 2008(10) :P10008, 2008.

[5] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On finding graph clusterings with maximum modularity. In *Graph-Theoretic Concepts in Computer Science : 33rd International Workshop, WG 2007, Dornburg, Germany, June 21-23, 2007. Revised Papers 33*, pages 121–132. Springer, 2007.

[6] Tim S Evans and Renaud Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical review E*, 80(1) :016105, 2009.

[7] Lucia Falzon. Determining groups from the clique structure in large social networks. *Social networks*, 22(2) :159–172, 2000.

[8] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5) :75–174, 2010.

[9] Linton C Freeman. Cliques, galois lattices, and the structure of human social groups. *Social networks*, 18(3) :173–187, 1996.

[10] Bernhard Ganter and Rudolf Wille. *Formal concept analysis : mathematical foundations*. Springer Science & Business Media, 2012.

[11] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12) :7821–7826, 2002.

[12] Steve Gregory. Fuzzy overlapping communities in networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2011(02) :P02017, 2011.

[13] Soumaya Guesmi, Chiraz Trabelsi, and Chiraz Latiri. Fca for common interest communities discovering. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, pages 449–455. IEEE, 2014.

[14] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social networks*, 5(2) :109–137, 1983.

[15] Mohamed-Hamza Ibrahim, Rokia Missaoui, and Abir Messaoudi. Detecting communities in social networks using concept interestingness. In *Proceedings of the 28th annual international conference on computer science and software engineering*, pages 81–90, 2018.

[16] Rokia Missaoui, Abir Messaoudi, Mohamed Hamza Ibrahim, and Talel Abdessalem. Detecting communities in complex networks using formal concept analysis. In *Advances in Knowledge Discovery and Management : Volume 9*, pages 77–105. Springer, 2022.

[17] Rokia Missaoui and Idrissa Sarr. *Social network analysis-Community detection and evolution*. Springer, 2015.

[18] Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1) :016107, 2008.

[19] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2) :167–256, 2003.

[20] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6) :066133, 2004.

[21] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2) :026113, 2004.

[22] Vincenzo Nicosia, Giuseppe Mangioni, Vincenza Carchiolo, and Michele Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics : Theory and Experiment*, 2009(03) :P03024, 2009.

[23] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043) :814–818, 2005.

[24] Tiago P Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4(1) :011047, 2014.

[25] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3) :036106, 2007.

[26] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4) :1118–1123, 2008.

[27] Samuel Franklin Sampson. *A novitiate in a period of change : An experimental and case study of social relationships*. Cornell University, 1968.

[28] Hua-Wei Shen, Xue-Qi Cheng, and Jia-Feng Guo. Quantifying and identifying the overlapping community structure in networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2009(07) :P07042, 2009.

[29] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A : Statistical Mechanics and its Applications*, 388(8) :1706–1712, 2009.

[30] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, et al. A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[31] Norbert Tsopze and Gamgne Domgue Félicité. Détection des communautés dans les réseaux orientés à l'aide des concepts formels. 2016.

[32] Xiaohua Wang, Licheng Jiao, and Jianshe Wu. Adjusting from disjoint to overlapping community detection of complex networks. *Physica A : Statistical Mechanics and its Applications*, 388(24) :5045–5056, 2009.

[33] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks : The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4) :1–35, 2013.

[34] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.

Session 6 : Formalisation et systèmes à base de connaissances 2

Mise en place d'une chaîne de traitement pour l'analyse explicable de séries temporelles via l'Analyse Formelle de Concept par compression en chronogrammes

G. Savarit[1], C. Demko[1], K. Bertet[1]

[1] La Rochelle Université, L3i

24 mai 2024

Résumé

Dans cet article, nous présentons une chaîne de traitement générique pour l'analyse interactive de séries temporelles. En définissant des propriétés temporelles*, nous mettons en place une forme de représentation basée sur la logique temporelle appelée* chronogramme*, pouvant être traduit en séquence d'intervalle temporelle pour une analyse explicable et hétérogène par l'Analyse Formelle des Concepts en utilisant l'outil GALACTIC.*

Mots-clés

Séries temporelles, Logique temporelle, Analyse Formelle des Concepts

Abstract

In this paper, we create a processing chain for interactive time series analysis. Using temporal properties*, we create a temporal logical representation called a* chronogram*, which can be interpreted as interval-based sequence for explicable and heterogeneous analysis with Formal Concept Analysis by using the framework GALACTIC.*

Keywords

Time series, Temporal logic, Formal Concept Analysis

1 Introduction

Les séries temporelles sont la forme de données temporelles la plus commune. La captation d'un signal au cours du temps créant des séries temporelles, celles-ci sont devenues archétypales dans la représentation de l'information au cours du temps dans de nombreux domaines, tels que l'économie, la médecine ou le fonctionnement opérationnel de l'industrie par le biais des tableaux de bord. Une série temporelle est la représentation discrète d'un signal réel continu évoluant au cours du temps [14], et est constituée d'une série $\langle t_k, x_k \rangle$ de paire temps/valeur.

De par sa représentation dans des domaines fortement mathématiques, la série temporelle a été premièrement analysée comme une fonction au cours du temps [7]. Les analyses des séries temporelles ont ainsi très rapidement porté sur une problématique de prédiction de celles-ci. En effet, s'il est possible de définir une fonction portant la série, alors il est possible de l'étendre temporellement celle-ci. Des méthodes d'agrégation générique telles qu'*ARIMA* [13] ont ainsi été mises en place dans les années 70 pour modéliser les séries temporelles. Ce qui va amener à des découvertes sur les natures mêmes des séries, dont les traitements diffèrent selon leur caractère univarié (ne représentant qu'une variable), ou multivarié [6] [16]. Plus récemment, la modélisation a été effectuée en utilisant les travaux sur les réseaux de neurones profonds [12].

De la même manière, la recherche de corrélations entre séries temporelles s'est rapidement développée dès les années 20 dans les travaux de Persons [14], le temps pouvant servir de base commune pour la recherche de correspondances, en faisant néanmoins attention à ce que la structure même soit adaptée à la corrélation [15].

Les séquences temporelles ont été utilisées pour modéliser des événements successifs en utilisant des fenêtres glissantes d'intervalles de temps. L'analyse de séquences, la fouille de séquences, a débuté en utilisant l'ordre, mais sans utiliser la composante temporelle [1], via des algorithmes comme *GSP (Generalized Sequential Pattern)* , avant de s'intéresser à la temporalité par le biais des séquences d'intervalles temporels [17]. L'analyse de ces dernières pouvant se faire en utilisant les équations d'Allen [2].

Ces analyses de corrélations et de prédiction ont des limites. L'analyse de séries ou de séquences temporelles est rarement hétérogène, chaque type de traitement étant conçu pour un type de données spécifique. La prédiction possède par ailleurs des aspects dits "boîte noire" qui complexifie la possibilité d'extraire du sens des séries et des séquences. Il est difficle de faire de la prédiction et d'extraire du sens en même temps.

Pour palier à ces problématique, des approches de type *XAI* (*eXplainaible AI*) ont émergés pour obtenir du sens dans ces différentes méthodes. Nous souhaitons nous placer dans ces approches en utilisons des méthodes hiérarchiques directement explicable par nature.

L'analyse formelle de concept (*Formal Concept Analysis* ou *FCA*) a été conceptualisée en 1982 [18], et repose sur la théorie des ensembles. À partir d'un contexte composé d'objet et d'attributs binaires [11], ainsi que des relations d'incidence entre les deux, il est possible de générer un

treillis des concepts. Ce treillis, hiérarchique, va représenter les différents sous-groupes d'objets/attributs. Ces sous-groupes sont explicables, résolvant les soucis "boîte noire" et permettent d'extraire des corrélations.
L'extension à des attributs non binaires [10] va permettre d'ouvrir le développement de plateformes d'analyse pour des données complexes. L'une de ces plateformes est GALACTIC, une implémentation de l'algorithme NEXTPRIORIRYCONCEPT [9] qui s'inspire de travaux sur les *pattern structure* pour permettre une construction générique des treillis, pilotée par l'utilisateur. Les attributs binaires de la *FCA* sont obtenus par le biais de descriptions sur des prédicats, ce qui permet une utilisation pour des données hétérogènes. Des stratégies pilotant la génération des treillis permettent de réduire le déluge de *pattern*. Via la plateforme GALACTIC, il a été ainsi possible d'analyser des séquences d'intervalle temporels [5]. Les données sont décrites sous la forme de *pattern*, qui par leur récurrence dans les objets permettent de définir des sous-ensembles communs.
Notre approche a pour objectif de permettre génériquement d'extraire de séries temporelles des séquences temporelles d'intervalles, pour pouvoir ensuite analyser celles-ci dans le cadre de la FCA en utilisant GALACTIC. Pour cela nous avons mis en place une première chaîne de traitement décrite dans la section 2 permettant d'extraire de toute série temporelle des *chronogrammes*, qui sont des représentations de logique temporelle binaire de propriétés sous la forme de séquences temporelles d'intervalles. Nous y définissons des propriétés présentes dans les séries temporelles. Nous allons ensuite présenter dans la section 3 une seconde chaîne de traitement permettant l'utilisation de ces *chronogrammes* pour l'analyse via GALACTIC. Enfin dans la section 4 nous nous intéresserons à un jeu de données réels, celui des capteurs de niveaux, éoliens, et de l'analyse des surcotes detectées par le Port Atlantique de La Rochelle.

2 Compression par *chronogrammes*

Avant de présenter la chaîne de traitement, nous allons définir des notations qui seront utilisées par la suite.

Notations. Nous allons utiliser les notations d'ensembles suivantes :

- $\mathbb{R}$: Ensemble des nombres réels
- $\mathbb{T}$: Ensemble de réels représentant le temps sous forme de *timestamps*.
- $\mathbb{B} = \{\top, \bot\} = \{0, 1\}$: Ensemble des booléens.
- $\mathbb{I}_n = [\![0, n[\![$: Entiers naturels inférieurs à n, utilisé pour représenter des espaces d'indices

2.1 Du signal aux séries temporelles

Une **série temporelle** peut s'interpréter comme la représentation discrète d'un **signal** réel continu évoluant au cours du temps.

Définition 1 *Un **signal** est une fonction continue $\dot{s}$ qui au temps associe une valeur x.*

$$\begin{aligned} \dot{s} : \mathbb{T} &\to \mathbb{R} \\ t &\mapsto x = \dot{s}(t) \end{aligned} \tag{1}$$

Nous utilisons la notation $\dot{s}$ pour marquer l'aspect continu. De manière graphique, il s'agit d'une courbe continue. La figure 1 en est un exemple pour des valeurs $\dot{s}(t)$ entre 0 et 10 unités de temps.
La captation du signal consiste à récupérer les données à des intervalles définis, réguliers ou non, sur une durée finie. Ce processus entraîne la discrétisation du signal. Le signal devient alors une série temporelle.

Par construction, une série temporelle est une suite strictement croissante sur le temps. Cette suite est définie sur un ensemble $\mathbb{I}_n$ d'indices de taille n, qui correspond au nombre de captations.

Définition 2 *Une **série temporelle** s est une famille ordonnée de temps discrets t_k et de valeurs x_k.*

$$\begin{aligned} s : \mathbb{I}_n &\to \mathbb{T} \times \mathbb{R} \\ k &\mapsto (t_k, x_k) \text{ avec } t_k < t_{k+1} \end{aligned} \tag{2}$$

Toute série s peut être représentée sous forme de suite. Les éléments $s(k)$ sont notés s_k. Nous allons définir deux formes pour les suites. La forme complète composée de la paire (t_k, x_k), et par abus de langage une forme réduite composé uniquement des valeurs x_k :

$$\textbf{Forme complète : } s = \langle (t_k, x_k)_{k \in \mathbb{I}_n} \rangle \in \mathbb{S}_n$$
$$\textbf{Forme réduite : } s = \langle x_k \rangle_{k \in \mathbb{I}_n} \in \mathbb{S}_n$$

Cette série représente plus ou moins fidèlement le signal. La fidélité est impactée par la fréquence d'échantillonnage et l'écart entre les captations. Plus la fréquence est importante et l'écart faible, plus la série est fidèle, mais aussi plus volumineuse.
Cela tend vers une série temporelle parfaite, possédant une fréquence d'échantillonnage sans écart, de tous les instants, pour un ensemble de captation infini $\mathbb{I}_\infty$ qui serait identique au signal réel.

$$\lim_{n \to \infty} s = \dot{s}$$

De manière graphique, nous obtenons des points discrets. La figure 2 représente ainsi 10 relevés du signal de la figure 1.
La série résultante est la suivante :

$$\alpha \in \mathbb{S}_{10} = \left\langle \begin{matrix} (0.7 & 0.7), & (1.2 & 1), \\ (2.5 & 0.6), & (4 & -0.6), \\ (4.6 & -1), & (5 & -1), \\ (6.5 & 0.3), & (7.2 & 0.6), \\ (8 & 0.9), & (9.6 & -0.2) \end{matrix} \right\rangle$$

On considère souvent la fréquence d'échantillonnage comme fixe, ce qui permet de représenter la série sans utiliser la composante de temps [15] [6] [13], ou par un temps décrit via l'écart $\delta > 0$ fixe entre deux captations, et à partir d'une première captation t_0.

$$\begin{aligned} s : \mathbb{I}_n &\to \mathbb{T} \times \mathbb{R} \\ k &\mapsto (t_0 + k\delta, x_k) \end{aligned} \tag{3}$$

Par construction, $t_k = t_0 + k\delta < t_0 + (k+1)\delta = t_{k+1}$. La forme réduite est ainsi plus adaptée pour un échantillonnage régulier.

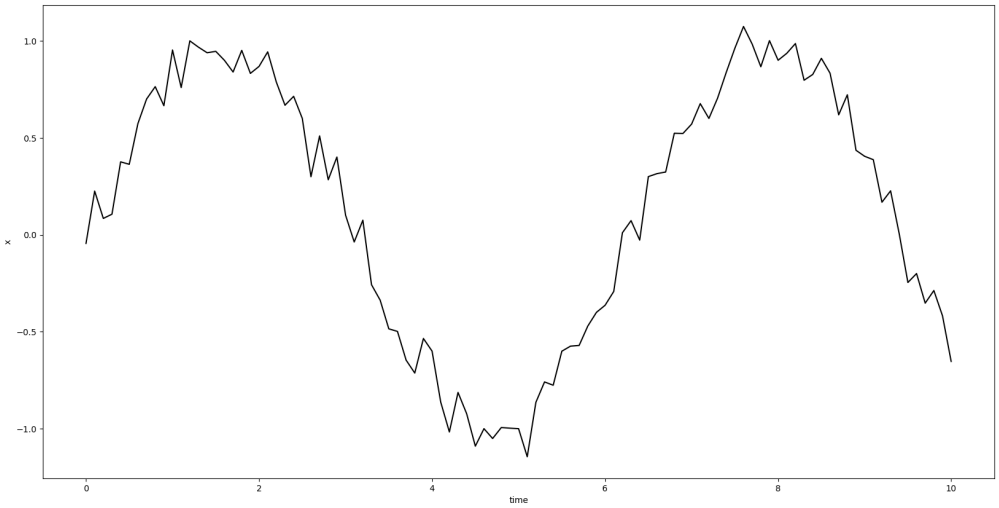


FIGURE 1 – Exemple de signal réel, $\dot{s} : [0, 10] \to \mathbb{R}$

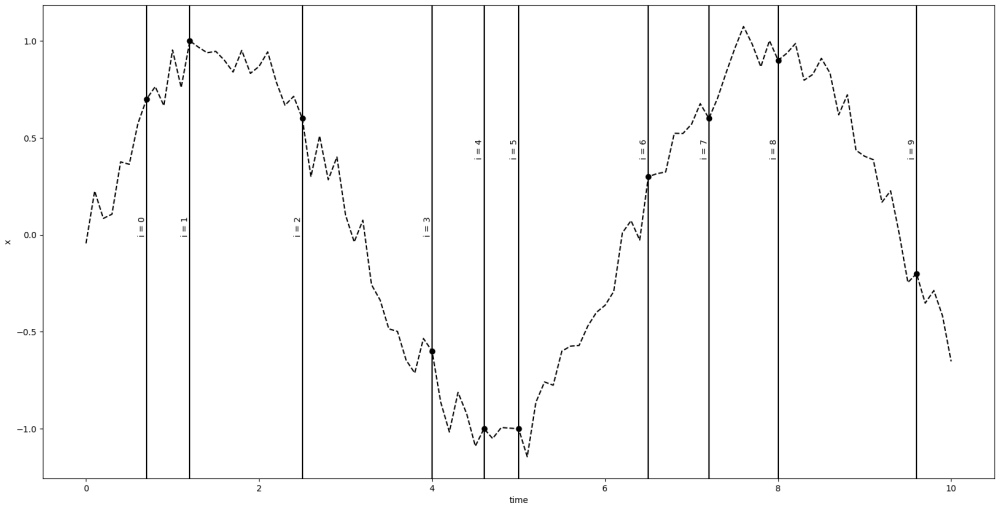


FIGURE 2 – Exemple de série temporelle $\alpha : \mathbb{I}_{10} \to [0, 10] \times \mathbb{R}$

Projections. Il est possible de projeter une série temporelle sur un intervalle temporel $T = [\underline{t}, \overline{t}[$ pour obtenir une sous-suite de la série :

$$\textbf{Forme complète :}\ s||T = \langle (t_k, x_k) \mid \underline{t} \leq t_k < \overline{t} \rangle$$
$$\textbf{Forme réduite :}\ s||T = \langle x_k \mid \underline{t} \leq t_k < \overline{t} \rangle$$

Cette projection peut se faire de deux manière, *absolue* en utilisant les temps réels, présenté au dessus, et *relative* en utilisant des temps relatifs à l'intervalle de projection :

$$\textbf{Forme complète :}\ s||T = \langle (t_k - \underline{t}, x_k) \mid 0 \leq t_k < \overline{t} - \underline{t} \rangle$$
$$\textbf{Forme réduite :}\ s||T = \langle x_k \mid 0 \leq t_k < \overline{t} - \underline{t} \rangle$$

2.2 De la série temporelle à la série temporelle booléenne

Nous allons extraire, à partir des séries temporelles, des propriétés temporelles et définir des fonctions de logique temporelles permettant leur représentation sous la forme d'une série temporelle booléenne.

Propriétés temporelles. Une *propriété temporelle* représente une vérité calculée en chaque point d'une série temporelle.

Définition 3 *Une* **propriété temporelle** *est, pour une série temporelle $s = \langle (t_k, x_k) \rangle \in \mathbb{S}_n$, une propriété booléenne vraie ou fausse pour chaque instant $k \in \mathbb{I}_n$, pouvant se définir par une fonction ou du pseudo code selon des paramètres θ telle que :*

$$\text{PROPERTY}_{s=\langle (t_k, x_k) \rangle}[\theta](k) := \%\textit{Définition} \qquad (4)$$

Par exemple, VALUESUP, VALUEINF et VALUEEGAL sont des propriétés qui représente respectivement les valeurs au-dessus, en-dessous et égales à un seuil, et sont décrites

comme :

$$\begin{aligned}\text{VALUESUP}_s[\text{seuil}](k) &:= x_k > \text{seuil}\\ \text{VALUEINF}_s[\text{seuil}](k) &:= x_k < \text{seuil}\\ \text{VALUEEGAL}_s[\text{seuil}](k) &:= x_k = \text{seuil}\end{aligned}$$

Une *propriété temporelle* permettant de représenter des intervalles répétitifs FREQUENCYTIMES est décrite par :

$$\text{FREQUENCYTIMES}_s\begin{bmatrix} t_{begin}\\ t_{end}\end{bmatrix}(k) := t_{begin} \geq\ t_k > t_{end}$$

Enfin une *propriété temporelle* qui représente la variation positive de la dérivée en un point DERIVATIVEPOSITIVE est décrite par :

$$\text{DERIVATIVEPOSITIVE}_s[\,](k) := \quad \frac{x_{k+1}-x_{k-1}}{t_{k+1}-t_{k-1}} > 0$$

Les séries temporelles sont composé d'une paire d'élements temporels et numériques. On peut donc définir trois sortes de *propriétés temporelles* :

— Les ***propriétés temporelles*** **quantitatives** dépendent uniquement des valeurs x.
Ex : VALUESUP, VALUEINF, VALUEEGAL

— Les ***propriétés temporelles*** **chroniques** dépendent uniquement des temps t.
Ex : FREQUENCYTIMES...

— Les ***propriétés temporelles*** **mixte** dépendent des deux.
Ex : DERIVATIVEPOSITIVE...

Il est possible à l'aide d'opérations logiques $\wedge, \vee, \neg$ de combiner les *propriétés temporelles* pour en construire des nouvelles. Ainsi la propriétés VALUEINTERVAL qui représente les valeurs dans un intervalle peut se définir comme :

$$\text{VALUEINTERVAL}_s\begin{bmatrix}\min\\ \max\end{bmatrix}(k) := \begin{matrix}\text{VALUESUP}_s[\max](k)\\ \wedge\\ \text{VALUEINF}_s[\min](k)\end{matrix}$$

***Chronogrammes*.** Chaque instant d'une série temporelle peut valider ou non une *propriété temporelle*. Nous introduisons ainsi la notion de *chronogramme* d'une *propriété*, défini comme une discrétisation de fonction de logique temporelle.

Définition 4 *Une **fonction de logique temporelle** $\dot{\chi}$ associe un booléen au temps pour les valeurs d'un signal $\dot{s}$.*

$$\begin{aligned}\dot{\chi}_{\dot{s}} : \mathbb{T} \to &\quad \mathbb{R} &&\to \mathbb{B}\\ t \mapsto &\quad \dot{s}(t) &&\mapsto \dot{\chi}_{\dot{s}}(t) = \dot{\chi}_{\dot{s}}(\dot{s}(t))\end{aligned} \tag{5}$$

Définition 5 *Une **série temporelle booléenne** χ est une fonction qui associe un booléen a chaque élément d'une série temporelle.*

$$\begin{aligned}\chi_s : \mathbb{I}_n \to &\quad \mathbb{T}\times\mathbb{R} &&\to \mathbb{T}\times\mathbb{B}\\ k \mapsto &\quad (t_k, x_k) &&\mapsto (t_k, \chi_s(k)) = \chi_s(s(k))\end{aligned} \tag{6}$$

Il est aussi possible de représenter cela sous la forme d'une suite de la même manière que les séries temporelles. Par simplification la forme réduite est celle permettant la définition des propriétés, mais la forme complète est retrouvable directement.

Forme complète : $\chi_s = \langle (t_k, \chi_s(k))_{k\in\mathbb{I}_n}\rangle = \langle (t_k, b_k)_{k\in\mathbb{I}_n}\rangle$
Forme réduite : $\chi_s = \langle \chi_s(k)_{k\in\mathbb{I}_n}\rangle = \langle (b_k)_{k\in\mathbb{I}_n}\rangle$

En utilisant les notations précédentes, il est ainsi possible de définir χ_1 qui représente la propriété d'un seuil supérieur à 0, VALUESUP, sur une série s par :

$$(\chi_1)_s = \text{VALUESUP}_s[\,\text{seuil} \leftarrow 0\,]$$

χ_2 représente les saisons d'hiver boréal, du 7 novembre au 4 février. Nous utilisons la norme ISO-8601 pour écrire le temps dans cette exemple :

$$(\chi_2)_s = \text{FREQUENCYTIMES}_s\begin{bmatrix} t_{begin} \leftarrow \text{11-07},\\ t_{end} \leftarrow \text{02-04}\end{bmatrix}$$

Enfin χ_3 représente la positivité du signe de la dérivé :

$$(\chi_3)_s = \text{DERIVATIVEPOSITIVE}_s[\,]$$

Considérons toujours la série α représentée sur la figure 2. La série temporelle booléenne des valeurs strictement positive de la série temporelle α est ainsi :

$$\text{VALUESUP}_\alpha[\text{seuil} \leftarrow 0] = \langle 1,1,1,0,0,0,1,1,1,0\rangle \tag{7}$$

Cette représentation peut être utilisée pour extraire des séquences temporelles d'intervalles.

Projections. Comme pour les séries temporelle, il est possible de projeter un *chronogramme* sur un intervalle temporel $T = [\underline{t}, \bar{t}[$, de telle sorte que :

Forme complète : $\chi_s||T = \langle (t_k, b_k) \mid \underline{t} \leq t_k < \bar{t}\rangle = \chi_{s||T}$
Forme réduite : $\chi_s||T = \langle b_k \mid \underline{t} \leq t_k < \bar{t}\rangle = \chi_{s||T}$

2.3 De la série temporelle booléenne au *chronogramme*

De la série booléenne sous sa forme complète, il est possible de construire une séquence d'intervalles temporelles K_χ correspondant aux valeurs "vraies" et "fausses" que nous appellerons par abus de langage un *chronogramme*.

Chronogramme . Un *chronogramme* est une répresentation simplifié des séries temporelles booléenne équivalente à des séquences d'intervalles temporels définies par des *propriétés temporelles*.

Définition 6 *Une **séquence d'intervalles temporels** K est une série d'événements qui associe un intervalle non nul $T = (\underline{t}, \bar{t})$, de telle sorte que $\underline{t} < \bar{t}$.*

Nous allons définir des séquences correspondant à la *propriété temporelle*. Les intervalles seront construit en utilisant les points de bascules entre les valeurs $\top$ et $\bot$ du *chronogramme*.

$$K_\chi : \mathbb{I}_n \to (\mathbb{T} \times \mathbb{T})$$
$$s_\chi \mapsto \langle (\underline{t_k}, \overline{t_k}) \mid \chi_s(t_i) = \top \text{ pour } \underline{t_k} \leq t_i < \overline{t_k}, \quad (8)$$
$$\chi_s(\underline{t_{k-1}}) = \chi_s(\overline{t_k}) = \bot \rangle$$

Comme nous avons fait pour les séries temporelles booléennes, nous allons pouvoir distinguer deux formes pour représenter K_χ. Nous ajustons la forme réduite de telle sorte que seul les éléments "vrais" soit representé par les intervalles temporelles. Les éléments "faux" sont par construction aussi présent dans les intervalles intercalaires.

Forme complète : $K_\chi = \langle b_k, (\underline{t_k}, \overline{t_k}) \rangle = \langle \chi_s, (\underline{t_k}, \overline{t_k}) \rangle$

Les formes réduites seront double. Une forme *temporelle* qui est résumé uniquement par les temps de début des intervalles "vrais" et des intervalles "faux", et une forme *indicielle* qui correspond aux indices des temps de la forme temporelle.

Forme réduite temporelle : $K_\chi = \langle (t_i)_{i \in \mathbb{I}_{m \leq n}} \rangle$
Forme réduite indicielle : $K_\chi = \langle (i)_{i \in \mathbb{I}_{m \leq n}} \rangle$

Ces formes devront respecter des propriétés de telle sorte que :

$$\chi_s(t_0) = \top,$$
$$\chi_s(t_{2i}) \neq \chi_s(t_{2i+1}),$$

Nous appellerons ces formes des *chronogrammes réduits*. Le *chronogramme réduit* de notre série α est ainsi :

Forme temporelle : $K_{ValueSup_\alpha} = \langle 0.7, 2.5, 6.5, 8 \rangle$
Forme indicielle : $K_{ValueSup_\alpha} = \langle 0, 2, 6, 8 \rangle$

Toutes ses formes représentent des séquences d'intervalles temporels, et sont équivalentes. Il est possible de retrouver chaques formes via une autre. Ainsi nous conserverons surtout les formes réduites temporelle et indicielle, car celle-ci ont la particularité d'être plus **compressées** que les formes complète ou de séquence d'intervalles temporels.

Il est aussi possible de représenter graphiquement cela sous la forme de ce que nous appellerons une *chronographie*.

Définition 7 *Une* **chronographie** *est la représentation graphique d'un* chronogramme*. Comme pour les représentations graphiques de série temporelle, elle est considérée comme continue entre les points.*

La figure 3 montre notre série exemple et la *chronographie* représentant la propriété $x \geq 0$.

Le *chronogramme réduit* ainsi construit permet de retrouver les informations issues de la propriété choisie en réduisant le volume de donnée nécessaire pour les représenter. Il permet aussi une représentation graphique simplifiée sous forme de *chronographie* pour faciliter l'analyse.

3 Recherche de corrélation avec GALACTIC

La recherche de corrélation entre séries temporelles peut se faire de plusieurs manières. Néanmoins, ces recherches se focalisent sur la détection de structure au sein des séries, et les approches statistiques employées reposent uniquement sur les séries temporelles. Pour pouvoir dans le futur nous concentrer sur les questions de prédiction et d'analyse hétérogène des données, nous allons utiliser pour nos recherches de corrélation les travaux récents de Boukhetta [4] sur les séquences d'intervalles temporels via l'Analyse Formelle des Concepts en utilisant l'outil GALACTIC.

GALACTIC. L'algorithme NEXTPRIORITYCONCEPT [8] permet une approche centrée sur l'utilisateur de la fouille de *pattern*, applicable sur des données hétérogènes G. Il s'agit d'une approche générique décrivant les objets par des prédicats. Ces prédicats sont construit par des *descriptions*. L'utilisateur peut ensuite piloter la fouille en utilisant des *stratégies*.

Une *description* δ est une application générant les prédicats décrivant un ensemble d'objets $A \subseteq G$. Chaque prédicat dépend des *caractéristiques* des objets manipulés. La description finale δ est l'union des prédicats pour chaque *caractéristique*.

Chaque concept $(A, \delta(A))$ est composé d'un sous-ensemble d'objet A et d'un ensemble de prédicat $\delta(A)$, permettant de traiter des données hétérogènes.

Inspiré par Bordat [3], un treillis de concept est généré niveau par niveau en utilisant des *stratégies* σ pour générer et sélectionner les prédécesseurs, en enrichissant la *description* $\delta(A)$ pour un ensemble d'objet $A' \subseteq A$. Les stratégies pouvant réduire le nombre de prédécesseur, il est possible d'obtenir des treillis plus petit que lors d'utilisation des stratégies naïves de l'Analyse Formelle des Concepts, et donc résoudre le problème du déluge de *pattern*.

GALACTIC intègre les *caractéristiques*, *descriptions* et *stratégies* pour l'analyse des séquences temporelle d'intervalle.

Données. L'ensemble initial des données que nous considérons pour l'analyse est un ensemble de *chronogrammes* Σ issus d'une ou plusieurs séries-temporelles : $\Sigma = \{\chi_s = \langle (t_k, b_k) \rangle\}$
Nous souhaitons rechercher des corrélations temporelles. Pour cela, nous considérons un ensemble d'intervalles focus $F = \{T = (\underline{t}, \overline{t})\}$ correspondant à nos objets. Cela impose une temporalité sur nos objets, qui doivent être définis sur des intervalles temporelles.

Nous pouvons aussi construire ces *focus* par le choix d'un *chronogramme focus* χ_f en utilisant les intervalles temporelles validant la propriété, de telle sorte qu'en utilisant la

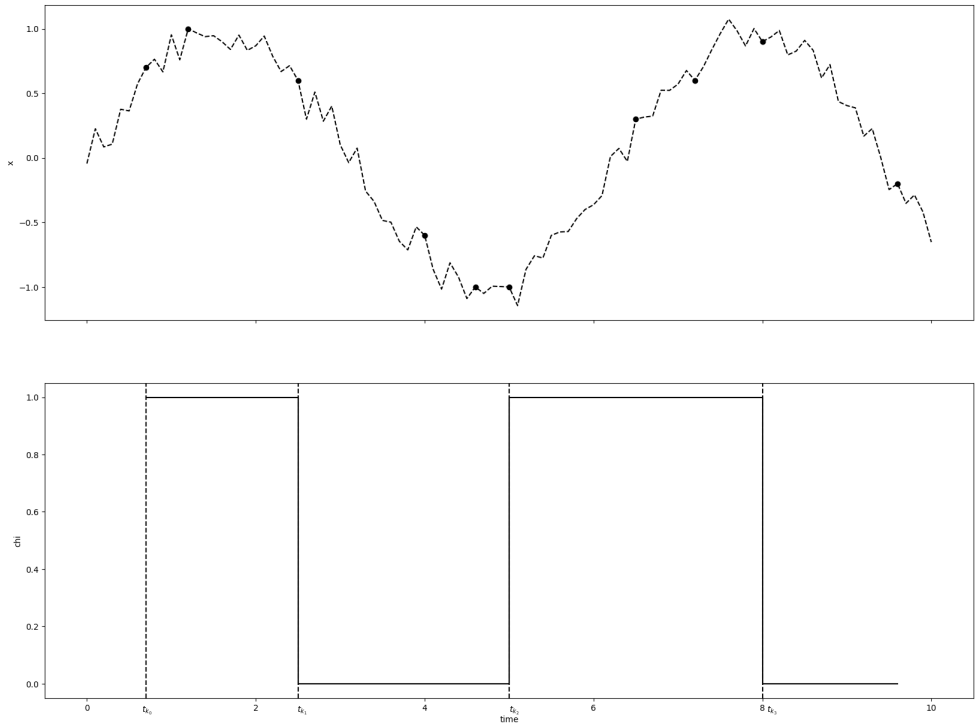


FIGURE 3 – Exemple de représentation par *chronographie* en bas

forme réduite sur les valeurs "vraies" de K_{χ_f}, $(k_{\chi_f}) = (\underline{t_k}, \overline{t_k}) \in F$.

Contexte. Les *caractéristiques* pour chaque intervalle $T \in F$ sont les projections des *chronogrammes* de Σ sur T. Ceux-ci peuvent être relatives ou absolues.

Les séquences temporelles ainsi formé sont *prédicats* des objets :

$$K_\Sigma = \{K_{\chi_s} || T = \langle (\underline{t_k}, \overline{t_k}) \rangle \mid K_{\chi_s} \neq \emptyset, \chi_s \in \Sigma, T \in F\} \tag{9}$$

Les séquences peuvent être de deux types, absolues ou relatives, et vont impacter l'analyse.

- Les séquences **absolues** se concentrent sur un espace des temps général. Il s'agit de vérifier les corrélations entre temps précis. Dans le langage courant, il s'agit de séquences d'intervalle temporel qui auront eu lieu *en même temps*. Par exemple, *les deux individus ont, le 12 octobre 2018 à 16h14, acheté une voiture*.
- Les séquences **relatives** se concentrent sur un espace des temps propres aux objets. Il s'agit de vérifier les corrélations entre temps relatifs. Par exemple, *les deux individus ont, à 20 ans, acheté une voiture.*

4 Expérimentation

Nous allons utiliser les données du Port Atlantique La Rochelle. Il s'agit de données de type série temporelle, composé des sorties de capteur installé près des portes d'écluses menant au bassin à flot. C'est une zone critique du port impliquant une surveillance accrue, et de nombreux capteurs pour vérifier en temps réel les conditions.

Notre jeu de donnée est composé de trois séries temporelles :

- $vent$ qui représente la captation des directions des vents entre mars 2021 et mars 2022.
- $maree$ qui représente la captation des niveaux d'eau de l'océan auprès du port entre mars 2021 et mars 2022.
- $shom$ qui représente la prédiction des niveaux d'eau de l'océan au port calculées par le Service Hydrographique et Océanographique de la Marine (*SHOM*).

Une quatrième série temporelle est construite :

- $surcote$ qui représente la différence entre la prédiction $shom$ et la série $maree$.

À partir de la série $vent$, nous définissons les 4 *chronogrammes* suivant :

$$WEST_{vent} = \textsc{ValueInterval}_{vent} \begin{bmatrix} \text{min} \leftarrow 45, \\ \text{max} \leftarrow 135 \end{bmatrix}$$

$$SOUTH_{vent} = \textsc{ValueInterval}_{vent} \begin{bmatrix} \text{min} \leftarrow 135, \\ \text{max} \leftarrow 225 \end{bmatrix}$$

$$EAST_{vent} = \textsc{ValueInterval}_{vent} \begin{bmatrix} \text{min} \leftarrow 225, \\ \text{max} \leftarrow 315 \end{bmatrix}$$

$$NORTH_{vent} = \textsc{ValueInf}_{vent} \left[\text{seuil} \leftarrow 45\right] \vee \textsc{ValueSup}_{vent} \left[\text{seuil} \leftarrow 315\right]$$

Nous construisons un *chronogramme* représentant les anomalies de surcote positives supérieures à 40 cm :

$$ANOMALY_{surcote} = \text{VALUESUP}_{surcote}\left[\text{seuil} \leftarrow 40\right]$$

Sur l'année de mars 2021 à mars 2022, nous obtenons 10 anomalies detectées. Une surcote à 40 cm est une surcote importante, ce faible nombre d'anomalie est donc attendu. Nous obtenons donc 10 anomalies aux temporalités variables de 10 minutes à 3 heures, avec une moyenne de 69 minutes et une médianne à 50 minutes.

Un résumé des données est fourni table 1.

Compression par chronogramme. Nous allons utiliser les chronogrammes issues de $vent$.
Pour comparer les tailles des différentes données, nous allons utiliser une représentation simplifiée de celles-ci. Ainsi, les données d'orientation du vent sont représentées par des entiers int64 sur un octet, associés aux timestamps int64 correspondants, et les données des *chronogrammes* seront elles-aussi représentées par des entiers int64.

Nous comparons ainsi la taille de la série à la taille de quatre *chronogrammes* associés, en distinguant sur 1 an, 6 mois, 3 mois et 1 mois.
Les résultats sont présentés dans le tableau 2.
Nous pouvons constater que les taux de compression avoisine les 85% dans toutes les configurations. Le volume de données en entrée variant en nombre de jours ne modifie pas significativement le taux de compression.
Pour compenser la perte des informations numériques, nous avons dû construire plusieurs *chronogrammes*, quatre ici, ce qui implique des taux de compression plus faible que pour un seul *chronogramme*. Néanmoins cela n'a pas significativement impacté la compression.

Nous présentons aussi les résultats en comparant uniquement avec la taille des données numériques x des séries temporelles. Cette expérimentation permet de vérifier la compression des données dans les cas où l'information temporelle est induite ou négligée, par exemple lors de l'analyse de multiples séries temporelles définies sur les mêmes temporalités.
Il est d'usage dans les logiciels de traitement pour ces cas de ne représenter qu'une fois les temps pour toutes les séries. t_k est commun à toutes les séries. Nous voulons montrer que même dans ces cas d'analyse multiple, les *chronogrammes* réduisent le volume des données.
Nous voyons que nous avoisinons les 70% de compression.

Sur la compression des données nous pouvons donc conclure que cette démarche permet même dans des cas où de multiples *chronogrammes* doivent être utilisés pour représenter les *propriétés* des séries temporelles de réduire le volume de données significativement.

Analyse des vents. Nous allons ensuite analyser la série $surcote$ dans le contexte des $vents$.

Le *chronogramme* $ANOMALY_{surcote}$ décrit plus haut sera notre *focus*.

Nous ajoutons ensuite le contexte en utilisant une combinaisons des *chronogrammes* de $vent$ décrit plus tôt. Nous allons analyser chaque anomalie de manière *relative*.

Nous pouvons ensuite utiliser GALACTIC sur les données ainsi formées. Nous utiliserons les descriptions MAXIMALCOMMONINTERVALDESCRIPTION car nous voulons comparer les intervalles communs. Nous utiliserons la statégies AUGMENTEDMINIMUMCARDINALITYSTRATEGY pour sélectionner les prédécesseurs nous permettant d'augmenter la taille des sous-groupe formés.

Les résultats sont présenté dans l'annexe 4. Les objets, les anomalies, d'un concept se retrouvent en parcourant le treillis vers le bas. Les prédicats se retrouvent en parcourant vers le haut.
Le treillis ainsi formé est composé de 19 concepts. Dans le treillis les concepts sont définis par des identifiants préfixés d'un $, le nombre d'objet contenus dans chaque concept est quant à lui préfixé d'un #. Les objets sont en gris foncés et les prédicats descriptifs en blanc.
3 sous-groupes ressortent de l'analyse. Un premier sous-groupe est lié au concept 16, et est composé de l'anomalie 0.

$16 : #1
vent match [[0;120] :[sud], [120;150] :[est], [150;180] :[sud], [180;210] :[est], [210;270] :[sud], [270;300] :[est], [300;330] :[sud], [230;510] :[est], [510;540] :[sud], [540;3000] :[est]]
[anomaly0]

Ce sous-groupe est caractérisé par une correlation avec des vents du sud, et une alternance entre est et sud. Ceci semble correspondre à des vents sud-est.

Le concept 17 correspond à un deuxième sous-groupe composé de l'anomalie 18. Il s'agit d'un concept composé de vent alternant entre est, nord, ouest et sud. Les intervalles de temps sont faibles pour les vents d'est et de sud, et semble montrer une forte composante de vent du nord-ouest. Une analyse plus poussée de ce sous-groupe semblerais intéressante pour comprendre cette différence.

Enfin un troisième sous-groupe plus important semble correler les vents d'est à la plupart des anomalies (8/10). Il s'agit du sous-groupe central dans le treillis, qui est composé des concepts 15 et de ces successeurs.

$15 : #4
vent match [[0;30] :[est], **[0;120] :[est]**, [120;150] :[est], [180;210] :[est], [270;300] :[est], **[330;510] :[est]**, [540;600] :[est], [1020;1050] :[est], [1110;1140] :[est], [1530;1560] :[est], [1530;1560] :[est], [1950;1980] :[est]]
[anomaly10, anomaly16, anomaly6, anomaly8]

Ce sous-groupe est composé uniquement de vents d'est, et semble montrer une corrélation entre les anomalies et cette direction.
Ces résultats étaient attendu au vu des données, et confirme ainsi la pertinence de l'analyse. Pour les experts du port, il s'agit d'un phénomène attendu : les vents venant d'est "poussent" les eaux vers les terres. En utilisant les chrono-

Descriptif	Fréquence	Nombre de captation $\mathbb{I}_n$	Taille en mémoire (Kio)
$vent$	30 secondes	1 052 310	16 442,34
$maree$	Entre 30 secondes et 1 minute	981 070	15 329,22
$shom$	10 minutes	52 616	2 886,16
$surcote$	10 minutes	52 616	2 886,16

TABLE 1 – Résumé des données

Taille relative des données en jours	a	b	c	Taux de compression (%) c/a	Taux de compression (%) c/b
366 jours	16 442,34	8 221,17	2 484,26	84,89	69,78
183 jours	8 235	4 117,5	1 228,80	85,08	70,16
91 jours	4 085	2 047,5	614,87	84,98	69,97
31 jours	1 395	697,5	221,74	84,10	68,21

TABLE 2 – Compression par *chronogramme* des données d'orientation du vent.
a : Taille des données (Kio) $vent = \langle (t_k, x_k) \rangle$,
b : Taille des données (Kio) $vent = \langle x_k \rangle$,
c : Taille des chronogrammes (Kio) $T_{WEST_{vent}}$+$T_{SOUTH_{vent}}$+ $T_{EAST_{vent}}$+$T_{NORTH_{vent}}$

grammes des vents dans toutes les directions, l'information que les vents d'est corrèlent est bien ressorti.

5 Conclusion

Par l'introduction de *chronogramme* et leur représentation graphique *chronographie*, nous avons mis en place une structure de données permettant la visualisation et la représentation pour un analyste de propriété issu de série temporelle. Nous avons présenté des algorithmes d'extraction de propriétés génériques sur les séries temporelles, et les propriétés associées.
De ces *chronogrammes*, il a été possible d'extraire directement des séquences temporelles d'intervalle symbolisées par les propriétés représentées. Nous avons montré l'apport de la compression sur les séries de la représentation par *chronogramme* des séquences, ainsi que la possibilité de les exploiter directement dans la FCA.
Notre chaîne de traitement ainsi formée permet d'utiliser des séries temporelles dans la plateforme GALACTIC.
L'utilisation de la plateforme GALACTIC et de la FCA nous permettent d'obtenir une explicabilité et une intéractivité directe.
Dans nos prochains travaux, nous nous intéresserons à ajouter de nouvelles propriétés génériques, spécialisées par les intéractions avec les experts métiers, ou calculées tel que les moyennes, médiannes ou des quantiles. Nous nous intéresserons ainsi aux résultats différents obtenues selon des propriétés choisies par l'utilisateur ou calculés sur les données. Nous continuerons l'analyse en profondeur des résultats obtenus via l'utilisation de séries temporelles dans GALACTIC.

Remerciements

Ce travail de recherche est supporté par l'Agence Nationale de la Recherche par "SmartFCA - ANR-21-CE23-0023 : Analyse Formelle de Concepts : un outil intelligent pour l'analyse de données complexes"

Références

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, March 1995.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. 26(11) :832–843, November 1983.

[3] J. P. Bordat. Calcul pratique du treillis de Galois d'une correspondance. *Mathématiques et sciences humaines*, 96 :31–47, 1986.

[4] S. E. Boukhetta, C. Demko, K. Bertet, J. Richard, and C. Cayèré. Temporal Sequence Mining Using FCA and GALACTIC. In *Graph-Based Representation and Reasoning*, Lecture Notes in Computer Science, pages 185–199, Cham, 2021.

[5] S. E. Boukhetta, J. Richard, and C. Demko. Interval-based sequence mining using FCA and the NextPriorityConcept algorithm. page 12, 2020.

[6] K. R. W. Brewer. Some consequences of temporal aggregation and systematic sampling for ARMA and ARMAX models. *Journal of Econometrics*, 1(2) :133–154, June 1973.

[7] H. T. Davis. *Analysis of economic time series. Principia Press, Bloomington, 1941.* 1941.

[8] C. Demko, K. Bertet, C. Faucher, J-F. Viaud, and S. Kuznetsov. Next Priority Concept : A new and generic algorithm computing concepts from complex and heterogeneous data. *Theoretical Computer Science*, 845 :1–20, December 2020. arXiv : 1912.11038.

[9] C. Demko, S. E. Boukhetta, J. Richard, G. Savarit, K. Bertet, C. Faucher, and D. Mondou. GALACTIC : towards a generic and scalable platform for complex and heterogeneous data using formal concept analysis. 2022.

[10] S. Ferré and O. Ridoux. A logical generalizationof formal concept analysis. In *Conceptual Structures : Logical, Linguistic, and Computational Issues*, Lecture Notes in Computer Science, pages 371–384. Springer, 2000.

[11] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, 1999.

[12] Hassan Ismail F., G. Forestier, J. Weber, L. Idoumghar, and P-A. Muller. Deep learning for time series classification : a review. *Data Mining and Knowledge Discovery*, 33(4) :917–963, July 2019.

[13] H. Lütkepohl. Forecasting Aggregated Vector ARMA Processes, 1987.

[14] W. M. Persons. Correlation of Time Series. *Journal of the American Statistical Association*, 18(142) :713–726, June 1923. Publisher : Taylor & Francis.

[15] M. H. Quenouille. The Comparison of Correlations in Time-Series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(1) :158–164, 1958.

[16] D. E. Rose. Forecasting aggregates of independent arima processes. 5(3) :323–345, May 1977.

[17] Po shan K. and Ada Wai-Chee F. Discovering temporal patterns for interval-based events. In *International Conference on Data Warehousing and Knowledge Discovery*, 2000.

[18] R. Wille. Restructuring lattice theory : an approach based on hierarchies of concepts. In *Formal Concept Analysis*, Lecture Notes in Computer Science, pages 314–339. Springer, 1982.

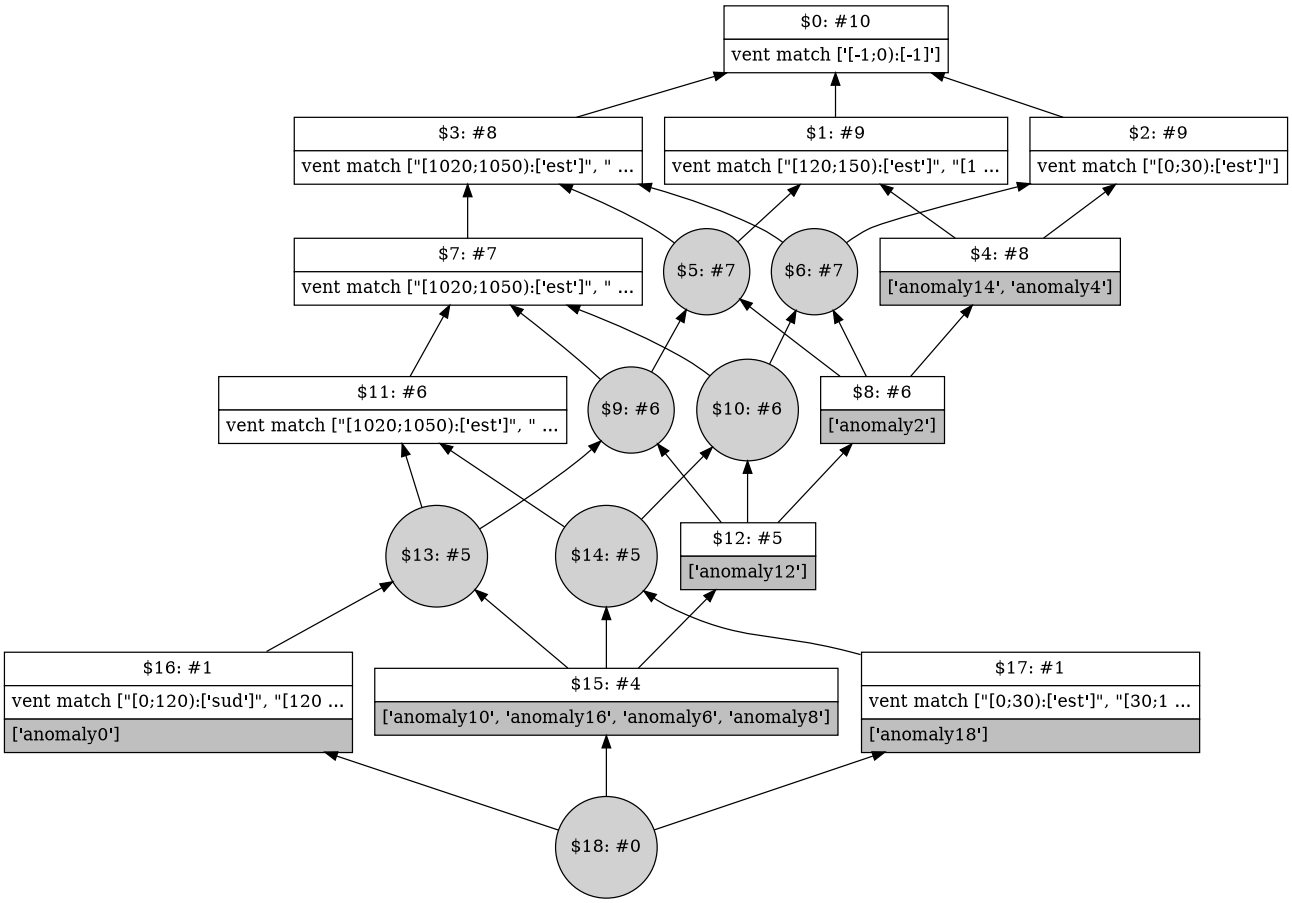


FIGURE 4 – Treillis des concepts

Une nouvelle logique de description NP-complet sous sémantique catégorielle

Ludovic Brieulle[1], Chan Le Duc[1]

[1] Université Sorbonne Paris Nord, LIMICS, U1142, F-93000, Bobigny, France

{ludovic.brieulle}, {chan.leduc}@univ-paris13.fr

Résumé

Nous introduisons dans cet article une nouvelle logique de description ainsi qu'une procédure de raisonnement pour celle-ci. La construction de cette nouvelle logique et de l'algorithme de raisonnement sont basés sur une réécriture de la sémantique ensembliste de la logique $\mathcal{ALC}$ avec des TBox générales, en utilisant la théorie des catégories ; où les concepts sont des objets et les subsomptions de concepts des flèches. Cette réécriture nous donne une description plus modulaire de la sémantique qui nous permet d'obtenir une nouvelle sous-logique NP*-complet de $\mathcal{ALC}$.*

Mots-clés

Logique de Description, Théorie des Catégories, Ontologies.

Abstract

We introduce in this paper a novel Description Logic and a reasoning procedure for it. The construction of this novel logic and the reasoning algorithm are both based on a rewriting of the usual set semantics of the DL $\mathcal{ALC}$ with general TBoxes in categorical language. In this approach, we use objects and arrows to represent DL concepts and subsumptions respectively. Providing a more modular description of the semantics, this approach allows us to define a sublogic of $\mathcal{ALC}$ that is NP*-complete.*

Keywords

Description Logics, Category Theory, Ontologies.

1 Introduction

Les Logiques de Description (DL) sont utilisées pour décrire des connaissances représentées dans les ontologies. Pour donner une sémantique à ces logiques, on utilise habituellement la théorie des ensembles. Cependant, les définitions fournies par cette *sémantique ensembliste* font que l'interaction entre certains constructeurs provoque une explosion en complexité pour le raisonnement. Par exemple, la logique $\mathcal{ALC}$ est EXPTIME-complet dans le cas d'une TBox générale [1], à cause, entre autres, de l'interaction entre les constructeurs $\exists$ et $\forall$ [1].

Dans le domaine médical, l'utilisation de la négation est importante ([3]) bien qu'elle ne soit pas intégrée aux logiques utilisées habituellement comme $\mathcal{EL}$ pour SNOMED [5]. Supposons qu'un patient X soit allergique à l'aspirine et qu'on lui en prescrive, il est préférable que le concept extrait du rapport de sortie d'hôpital : $\exists \mathsf{medicatedWith}.\mathsf{Aspirin} \sqcap \forall \mathsf{medicatedWith}.\neg \mathsf{Aspirin}(X)$ soit insatisfiable et remonte une erreur. Cependant, si la sémantique ensembliste de $\forall$ est utilisé sans changement, l'interaction avec le constructeur $\exists$ est conservée, ce qui entraine la perte de la tractabilité de la logique $\mathcal{EL}$.

Nous proposons alors d'utiliser la théorie des catégories pour obtenir une nouvelle logique qui nous permettra d'affaiblir une de ces interactions en ignorant une propriété spécifique d'un constructeur. Grâce à cette nouvelle *sémantique catégorielle*, nous définissons une nouvelle sous-logique de $\mathcal{ALC}$ NP-complet que nous appelons $\mathcal{ALC}$ *affaibli*, notée $\mathcal{ALC}_{\overline{\forall}}$. Dans cette nouvelle sous-logique, un raisonneur pour $\mathcal{ALC}_{\overline{\forall}}$ trouvera toujours que : $\exists \mathsf{medicatedWith}.\mathsf{Aspirin} \sqcap \overline{\forall} \mathsf{medicatedWith}.\neg \mathsf{Aspirin}(X)$ est insatisfiable mais plus rapidement qu'un raisonneur pour $\mathcal{ALC}$.

L'utilisation de la théorie des catégories dans le domaine des DL n'est pas récent, une de ses utilisations notable par [4] consiste à se servir de la théorie des catégories pour établir une relation entre l'expressivité et les structures des modèles d'une logique.

Cependant, notre approche diffère de celle-ci dans le sens où nous utilisons la théorie des catégories pour encoder la sémantique ensembliste usuelle et non pas pour décrire des modèles. Cela nous permet entre autres de décomposer la sémantique de chaque constructeur logique en plusieurs propriétés de flèches d'une catégorie particulière. Certaines de ces flèches entrainent les intéractions qui sont reponsables de l'intractabilité de la logique $\mathcal{ALC}$, on va alors les « ignorer » pour obtenir une nouvelle sous-logique de complexité inférieure à $\mathcal{ALC}$ pour la satisfiabilité, tout en conservant une partie de son expressivité.

Dans cet article, nous présenterons la DL $\mathcal{ALC}$ en sémantique ensembliste, puis nous introduirons notre nouvelle sémantique utilisant la théorie des catégories (Section 2). Pour finir, nous décrirons notre nouvelle sous-logique ainsi qu'un algorithme permettant de fournir un candidat pour répondre au problème de la satisfiabilité sous cette sous-logique (Section 3).

2 Sémantiques de $\mathcal{ALC}$

Dans cette section, nous décrivons dans les grandes lignes notre sémantique catégorielle de $\mathcal{ALC}$ équivalente à la sémantique ensembliste.

Logique de description $\mathcal{ALC}$. Soit $\mathbf{C}$ et $\mathbf{R}$ deux ensembles non vides de *concept atomique* et de *rôle* respectivement. Les constructeurs suivants sont utilisés pour la logique $\mathcal{ALC}$: $\sqcap, \sqcup, \neg, \exists, \forall, \bot, \top$. Alors un concept en $\mathcal{ALC}$ est défini récursivement de la façon suivante : pour tout $A \in \mathbf{C}$, A est un concept et si C, D sont des concepts en $\mathcal{ALC}$ alors $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists R.C$, $\forall R.C$ sont des concepts en $\mathcal{ALC}$ où $R \in \mathbf{R}$ – $\top$ et $\bot$ sont également des concepts en $\mathcal{ALC}$. Une *Inclusion Générale de Concept* (GCI en anglais) est une relation de la forme $C \sqsubseteq D$. Une *TBox* est un ensemble *fini* $\mathcal{O}$ de GCI.

Sémantique ensembliste. Pour définir la sémantique ensembliste, nous définissons une *interprétation* $\mathcal{I}$ qui consiste en un domaine d'interprétation $\Delta^{\mathcal{I}} \neq \varnothing$, et une fonction $\cdot^{\mathcal{I}}$ qui à chaque concept C associe un ensemble $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ et à chaque rôle R un ensemble $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. On dit que $\mathcal{I}$ est un *modèle* d'une TBox $\mathcal{O}$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ est vraie pour toutes GCI $C \sqsubseteq D$ de $\mathcal{O}$. Un concept C est dit *satisfiable* par rapport à une TBox $\mathcal{O}$ s'il existe un modèle $\mathcal{I}$ de $\mathcal{O}$ tel que $C^{\mathcal{I}} \neq \varnothing$. Pour une description plus détaillé de l'interprétation de chaque constructeur voir [1].

Sémantique catégorielle. Les ensembles $\mathbf{C}$ et $\mathbf{R}$ nous servent à définir ce que nous appelons des *catégories de syntaxe* : une *catégorie de concepts*, notée $\mathscr{C}_c$, dont tous les $C \in \mathbf{C}$ sont des objets de cette catégorie, et une *catégorie de rôles* $\mathscr{C}_r$, dont tous les $R \in \mathbf{R}$ sont des objets de celle-ci – dans les deux cas, les flèches représentent la subsomption (de concept ou de rôle). Nous avons également besoin de deux *foncteurs* : $\Pi_\ell, \Pi_r : \mathscr{C}_r \longrightarrow \mathscr{C}_c$ qui envoient des objets de $\mathscr{C}_r$ vers des objets de $\mathscr{C}_c$, en conservant les flèches de la première catégorie à la deuxième. Ces deux foncteurs serviront à donner la sémantique des constructeurs $\exists$ et $\forall$ – ce sont les équivalents des projections pour les relations binaires dans le contexte de la théorie des ensembles. Ces deux catégories sont également munies de deux objets spéciaux chacunes, qu'on appelle *objets initiaux*, $\bot$ pour les concepts et $R_\bot$ pour les rôles, et *objets terminaux*, $\top$ pour les concepts et $R_\top$ pour les rôles. Dans la suite, nous écrivons $\mathsf{Ob}(\mathscr{C})$ pour parler de l'ensemble des objets d'une catégorie $\mathscr{C}$ et $\mathsf{Hom}(\mathscr{C})$ pour l'ensemble de ses flèches.

Afin de représenter la sémantique des différents constructeurs logiques, il faut ajouter de la structure à ces catégories de syntaxe pour obtenir un nouveau type de catégorie appelée *catégorie d'ontologie*. Ces nouvelles catégories ont pour paramètres un concept de base en $\mathcal{ALC}$, C_0, et une TBox, $\mathcal{O}$ – elles sont écrites $\mathscr{C}_c\langle C_0, \mathcal{O}\rangle$ et $\mathscr{C}_r\langle C_0, \mathcal{O}\rangle$. Les objets de $\mathscr{C}_c\langle C_0, \mathcal{O}\rangle$ sont alors des concepts en $\mathcal{ALC}$ – pour $\mathscr{C}_r\langle C_0, \mathcal{O}\rangle$, les objets, en plus de ceux déjà mentionnés, seront des objets-rôles associés aux différentes restrictions existentielles et universelles présentes dans la TBox $\mathcal{O}$.

Les flèches de ces deux types catégories sont déterminées à partir de l'ontologie $\mathcal{O}$ et d'un système de propriétés qui définissent la sémantique des constructeurs $\sqcap, \sqcup, \neg, \exists, \forall, \bot$ et $\top$ sous la nouvelle sémantique catégorielle – de sorte que les subsomptions et équivalences de la sémantique ensembliste soient toujours respectées. Par exemple, pour tout axiome $E \sqsubseteq F \in \mathcal{O}$, la flèche $\top \longrightarrow \neg E \sqcup F$ est dans $\mathscr{C}_c\langle C_0, \mathcal{O}\rangle$. Ou encore, pour représenter le constructeur $\sqcap$ on utilise l'objet $C \sqcap D$ défini comme le produit (catégoriel) de C et D. Cela nous donne les flèches $C \sqcap D \longrightarrow C, D$ et pour tout objet X tel que $X \longrightarrow C, D$ alors on a la flèche $X \longrightarrow C \sqcap D$ – ces deux propriétés englobent la sémantique complète de la conjonction, plus de détails à ce sujet dans [2].

Grâce à ces différentes propriétés, nous pouvons, entre autres, déduire la GCI $\exists R.C \sqcap \forall R.D \sqsubseteq \exists R.(C \sqcap D)$ qui en semantique catégorielle s'écrit :

$$\exists R.C \sqcap \forall R.D \longrightarrow \exists R.(C \sqcap D) \tag{1}$$

La propriété (1) est une des responsables de l'intractablité de $\mathcal{ALC}$ [1], elle est conséquence d'une propriété dites *indépendante* de la définition du constructeur $\forall$ – indépendance caractérisée par le fait que la flèche représentant cette propriété n'est pas déduite des autres flèches de la catégorie. C'est cette propriété qu'on souhaite « ignorer » pour pouvoir définir notre nouvelle sous-logique, Section 3.

Une catégorie d'ontologie (de concepts) $\mathscr{C}_c\langle C_0, \mathcal{O}\rangle$ est *saturée* si elle vérifie toutes les propriétés pour définir les différents constructeurs de la logique $\mathcal{ALC}$. Elle est dite *minimale* si elle ne contient que des flèches (et donc des objets) issues de ces propriétés et non pas d'ajout arbitraire ou artificiel. Les définitions de la négation et des autres constructeurs nous permettent également d'obtenir les lois de De Morgan dans le contexte catégorielle. Ces lois permettent de traduire en un temps polynomial les concepts en *forme négative normale* (NNF en anglais), *i.e.* il n'y a de négation que devant des concepts atomiques – et donc de réduire le nombre de propriétés nécessaires pour caractériser la sémantique de $\mathcal{ALC}$ en terme de satisfiabilité.

Une catégorie de concept est NNF *saturée et minimale* si elle respecte ces nouvelles propriétés et ne contient que des flèches venant de l'application de celles-ci. *Un concept C_0 en $\mathcal{ALC}$ est* insatisfiable sous la sémantique catégorielle *par rapport à $\mathcal{O}$ s'il existe une catégorie* NNF *saturée minimale $\mathscr{C}_c\langle C_0, \mathcal{O}\rangle$ telle que $C_0 \longrightarrow \bot \in \mathsf{Hom}(\mathscr{C}_c\langle C_0, \mathcal{O}\rangle)$.*

Théorème 1 *Soit $\mathcal{O}$ une TBox en $\mathcal{ALC}$ et C_0 un concept en $\mathcal{ALC}$. Le concept C_0 est insatisfiable sous la sémantique catégorielle en rapport avec $\mathcal{O}$* ssi *il est insatisfiable sous la sémantique ensembliste en rapport avec $\mathcal{O}$.*

3 Satisfiabilité dans $\mathcal{ALC}_{\overline{\forall}}$

Lorsque nous parlons de la complexité d'une logique, nous parlons de la complexité partagée de ses problèmes de raisonnement les plus difficiles. C'est pour cela que nous disons de la logique $\mathcal{ALC}$ qu'elle est EXPTIME-complet

avec des TBox générale puisque tous les problèmes de raisonnement peuvent être réduits à un problème de satisfiabilité [1]. Pour diminuer cette complexité, nous avons décidé d'ignorer la propriété (1) qui provoque l'intéraction entre les retrictions exitentielles et universelles – cependant, de l'information pouvant venir de cette propriété pourrait être perdue. Supposons que les flèches $C \sqcap D \longrightarrow \bot$ et $C_0 \longrightarrow \exists R.C \sqcap \forall R.D$ soient dans une catégorie de concept NNF minimale saturée. Si le regroupement n'a plus lieu, il est possible d'imaginer ne plus pouvoir déduire que C_0 est insatisfiable dans cette logique. Nous avons trouvé que les différentes propriétés des constructeurs logiques en sémantique catégorielle nous permettent d'obtenir le résultat suivant :

$$C \sqcap D \longrightarrow \bot \Longrightarrow \exists R.C \sqcap \forall R.D \longrightarrow \bot \qquad (2)$$

sans faire intervenir la propriété (1). Nous définissons alors une nouvelle sous-logique de $\mathcal{ALC}$ en utilisant une version affaiblie de $\forall$, notée $\bar{\forall}$ – nouvelle sous-logique écrite $\mathcal{ALC}_{\bar{\forall}}$. Une catégorie de concept est alors $\mathcal{ALC}_{\bar{\forall}}$ saturée et minimale si elle respecte toutes les propriétés de cette nouvelle logique et que toutes ses flèches sont issues de ces propriétés uniquement.

Vérification de la satisfiabilité. Soit C_0 un concept $\mathcal{ALC}_{\bar{\forall}}$, et $\mathcal{O}$ une TBox en $\mathcal{ALC}_{\bar{\forall}}$. Pour raisonner dans cette nouvelle logique et obtenir la complexité NP pour la vérification de la satisfiabilité, il nous faut une structure intermédiaire qu'on appelle *graphe d'objets* G. Un graphe d'objets est composé d'un ensemble de nœuds $\mathbf{V}$, d'arêtes $\mathbf{E}$, d'une racine $v_0 \in \mathbf{V}$ et d'une fonction L qui a chaque $v \in \mathbf{V}$ associe un sous-ensemble $L(v)$ de $\mathsf{sub}\langle C_0, \mathcal{O}\rangle$ qui est l'ensemble des sous-concepts de $\mathcal{O}$ et C_0, appelé étiquette de v.

Algorithme 1

Entrée : C_0, un concept en $\mathcal{ALC}_{\bar{\forall}}$ et $\mathcal{O}$, une TBox en $\mathcal{ALC}_{\bar{\forall}}$.
Sortie : G, un graphe d'objets.
Étape 1. Initialiser le graphe : Créer un nœud v_0 de sorte que $C_0 \in L(v_0)$ et $\mathsf{NNF}(\neg F \sqcup F') \in L(v_0)$ pour tout axiome $F \sqsubseteq F'$ dans $\mathcal{O}$.
Étape 2. Pour chaque nœud $v \in \mathbf{V}$ et chaque concept dans $L(v)$:

- Pour chaque conjonction $C \sqcap D$ dans $L(v)$, ajouter C et D s'ils ne sont pas déjà dans l'étiquette $L(v)$; pour chaque disjonction $C \sqcup D$, *choisir* d'ajouter C ou D si aucun des deux n'est pas déjà dans $L(v)$.
- Si $\exists R.C$ et/ou $\forall R.C'$ est/sont dans $L(v)$ et que ni C ou $\{C, C'\}$ ne sont déjà dans un label de G, alors créer un nouveau nœud v_C ou $v_{C,C'}$ initialisé de la même façon qu'à l'**Étape 1** en ajoutant C ou C, C' à $L(v_C)$ ou $L(v_{C,C'})$ respectivement, et $\{v, v'\}$ à $\mathbf{E}$ où $v' \in \{v_C, v_{C,C'}\}$.

Étape 3. Dès qu'il n'y a plus de modification à faire sur G, retourner alors $G = \langle \mathbf{V}, \mathbf{E}, L, v_0\rangle$.

Le critère de satisfiabilité revient alors à parcourir tous les nœuds v de G à la recherche d'une étiquette $L(v)$ contenant à la fois A et $\neg A$ pour A un concept atomique en $\mathcal{ALC}_{\bar{\forall}}$ ou $\bot$. Si c'est le cas, G contient un *clash*. Si *tous* les graphes générés contiennent un clash, il en découle que C_0 est insatisfiable sous la logique $\mathcal{ALC}_{\bar{\forall}}$. Inversement, si *au moins un* graphe d'objets G généré ne contient pas de clash, alors C_0 est satisfiable.

Nous avons prouvé que pour un concept C_0 en $\mathcal{ALC}_{\bar{\forall}}$, et une TBox $\mathcal{O}$, notre algorithme retourne un graphe G sans clash *si et seulement si* C_0 est satisfiable par rapport à $\mathcal{O}$ sous la logique $\mathcal{ALC}_{\bar{\forall}}$. Notons que l'algorithme 1 est non-déterministe, mais retourne un graphe en un temps polynomial, ce qui nous donne la complexité NP. En réduisant le problème SAT à un problème de satisfiabilité en $\mathcal{ALC}_{\bar{\forall}}$, on montre que la satisfiabilité dans $\mathcal{ALC}_{\bar{\forall}}$ est NP-difficile et de là est déduit le théorème suivant :

Théorème 2 *$\mathcal{ALC}_{\bar{\forall}}$ est* NP*-complet.*

4 Conclusion et perspectives

Nous avons construit une nouvelle sous-logique NP-complet de $\mathcal{ALC}$ en effectuant une réécriture de la sémantique ensembliste utilisant la théorie des catégories. Par la suite, nous implémenterons l'algorithme pour l'évaluer sur des ontologies médicales volumineuses.

Il est également envisagé d'explorer une nouvelle sous-logique de $\mathcal{ALC}$ en enlevant la distributivité qui est également une propriété indépendante. Nous conjecturons que $\mathcal{ALC}_{\bar{\forall}}$ avec une version affaiblie de $\sqcup$ notée $\bar{\sqcup}$, *i.e.* sans distributivité, est tractable. On ajoutera alors les constructeurs affaiblis $\bar{\forall}$ et $\bar{\sqcup}$ à la logique $\mathcal{EL}$++ tout en conservant sa tractabilité.

Références

[1] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[2] Ludovic Brieulle, Chan Le Duc, and Pascal Vaillant. Reasoning in the description logic ALC under category semantics (extended abstract). In *Proceedings of the 35th International Workshop on Description Logics (DL 2022)*, volume 3263 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022.

[3] Werner Ceusters, Peter Elkin, and Barry Smith. Negative findings in electronic health records and biomedical ontologies : A realist approach. *International Journal of Medical Informatics*, 76 :S326–S333, 2007.

[4] Clemens Kupke and Dirk Pattinson. Coalgebraic semantics of modal logics : An overview. *Theoretical Computer Science*, 412(38) :5070–5094, 2011.

[5] K.A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption : Experience with snomed-rt. *J. of the American Medical Informatics Association.*, Fall Symposium Special Issue, 2000.