



**HAL**  
open science

# Vers l'adaptation de comportements par le biais de l'évolution culturelle pour des essaims de robots autonomes

Erwan Plantec, Lionel Aquilanti, Rafaele Belorgey

► **To cite this version:**

Erwan Plantec, Lionel Aquilanti, Rafaele Belorgey. Vers l'adaptation de comportements par le biais de l'évolution culturelle pour des essaims de robots autonomes. [Travaux universitaires] LORIA, UMR 7503, Université de Lorraine, CNRS, Vandoeuvre-lès-Nancy. 2021. hal-03280188v2

**HAL Id: hal-03280188**

**<https://hal.inria.fr/hal-03280188v2>**

Submitted on 7 Jul 2021

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

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

# Vers l'adaptation de comportements par le biais de l'évolution culturelle pour des essaims de robots autonomes

PROJET TUTORÉ

Master 1 Sciences Cognitives

PRÉSENTÉ PAR

Rafaelle Belorgey ◊ Lionel Aquilanti ◊ Erwan Plantec

ENCADRÉ PAR

Christine Bourjot & Laurent Ciarletta

ORGANISME D'ACCUEIL

Équipe SIMBIOT - LORIA



Institut des Sciences du Digital, Management & Cognition

Université de Lorraine

Année universitaire : 2020 - 2021

# Table des matières

Introduction générale	1
<b>1. L'évolution culturelle comme source d'inspiration</b>	<b>2</b>
1.1. Introduction aux concepts-clés pour une approche inspirée par l'évolution culturelle .	2
1.1.1. Généralités sur l'évolution culturelle . . . . .	2
1.1.2. Principes des Language Games . . . . .	3
1.1.3. Intérêt d'une approche inspirée de l'évolution culturelle . . . . .	3
<b>2. Vers l'adaptation de comportements par le biais de l'évolution culturelle pour des essais de robots autonomes</b>	<b>5</b>
2.1. Conceptualisation de l'approche : cadre formel et modèle . . . . .	5
2.1.1. Un cadre formel pour une approche inspirée par l'évolution culturelle . . . . .	6
2.1.2. Introduction de biais dans les Language Games : vers une évolution adaptative	8
2.1.2.1. Description du Naming Game (NG) . . . . .	8
2.1.2.2. Introduction d'un biais de succès dans le NG . . . . .	9
2.1.2.3. Introduction d'un biais de nouveauté dans le NG . . . . .	9
2.1.2.4. Introduction de mutations dans le NG . . . . .	10
2.1.3. Analyse des propriétés du modèle . . . . .	10
2.1.3.1. Analyse du caractère adaptatif de l'évolution des comportements dans le cas d'un environnement stationnaire et non-stationnaire . . .	11
2.1.3.2. Analyse de la stabilité de la performance face à l'introduction d'aléa dans l'environnement . . . . .	12
2.1.3.3. Analyse de la stabilité de la performance face à la variation de la taille de la population . . . . .	13
2.1.3.4. Analyse du caractère continu de l'évolution des agents selon l'injection de mutations . . . . .	13
2.1.3.5. Synthèse de l'analyse des propriétés du modèle . . . . .	13
2.2. Vers une application concrète du modèle : recherche de cibles . . . . .	14
2.2.1. Description de la tâche . . . . .	14
2.2.2. Méthode expérimentale . . . . .	16
2.2.3. Présentation des résultats . . . . .	16
2.2.4. Interprétation des résultats . . . . .	19
<b>Conclusion</b>	<b>20</b>
<b>Bibliographie</b>	<b>22</b>
<b>Annexes</b>	<b>23</b>
<b>A. Exploration des possibilités offertes par le Category Game et tentatives de réponses à ses limites</b>	<b>i</b>
<b>B. Exploration de la simulation réseau pour l'étude de la fiabilité des communications intrinsèques au modèle</b>	<b>vi</b>

# Table des tableaux et figures

2.1.	Proposition d'une boucle de feedback positif dans la métaphore de l'évolution culturelle. . . . .	6
2.2.	Schématisation de l'architecture interne de l'agent dans cette approche ; l'agent agissant en boucle fermée dans l'environnement et interagissant avec le reste du groupe d'agents. . . . .	8
2.3.	Évolution du vocabulaire de 400 agents placés sur une grille. Chaque cellule est un agent et sa couleur représente la fitness associé au mot présent dans son vocabulaire. $t$ est le nombre de pas de temps. . . . .	10
2.4.	Évolution de la fitness moyenne de la population ( $N = 400$ ) en fonction de la fréquence de biais de succès (avec $f_{novelty} = 0$ ). . . . .	11
2.5.	Capacité de réadaptation face à une fonction de fitness non-stationnaire en fonction de la fréquence de biais de succès et de nouveauté. . . . .	11
2.6.	Évolution de la diversité du vocabulaire (des paramètres) en fonction de la fréquence de biais de nouveauté. . . . .	12
2.7.	Évolution de la fitness moyenne en fonction de la taille de la population et de l'écart type ( $\sigma$ ) de la fonction de fitness $f$ avec $f(x) \sim \mathcal{N}(x, \sigma)$ (le code couleur des courbes représente la valeur de $\sigma$ ). (a) $N = 400$ . (b) $N = 100$ . (c) $N = 25$ . . . . .	12
2.8.	Évolution de la fitness moyenne en fonction du nombre de parties par agent pour différentes tailles de population. . . . .	13
2.9.	Évolution de la fitness moyenne pour différentes fréquences de mutation. . . . .	14
2.10.	Illustration du positionnement des capteurs de l'agent. . . . .	15
2.11.	Tableau résumant les paramètres utilisés lors des expérimentations. . . . .	16
2.12.	Fenêtre de simulation NetLogo. . . . .	17
2.13.	Comparaison du nombre total de cibles trouvées sans et avec évolution culturelle (CE). $pvalue = 1.98e - 11$ (test t). Avec $n_{targets} = 20$ et $n_{dangers} = 0$ . . . . .	17
2.14.	Évolution du paramètre d'attraction au signal émis par les cibles ( $m_{target}$ ). Trajectoire moyenne sur 25 simulations. (CI = Confidence Interval.) . . . . .	17
2.15.	Évolution de l'attraction moyenne au signal émis par les cibles pour différentes tailles de population. . . . .	18
2.16.	Évolution du paramètre d'attraction au signal émis par les cibles lorsque la fonction de récompense est inversée à $t = 400$ . . . . .	18
2.17.	Comparaison du nombre total de cibles trouvées sans et avec évolution culturelle (CE). $pvalue = 1.92e - 05$ (test t). Avec $n_{targets} = 20$ et $n_{dangers} = 10$ . . . . .	19
2.18.	Évolution des paramètres d'attraction moyens pour chacun des signaux. . . . .	19
A.1.	Exemple d'une SOM entraînée sur présentation de $n = 10000$ stimuli suivant une loi normale centrée réduite en dimension 2 où chaque point représente un neurone et la position de ce neurone son prototype (le stimuli pour lequel il répond le plus fortement). . . . .	ii
A.2.	Comparaison des performances moyennes sur 10 essais entre les différentes stratégies sur $T = 15000$ itérations. Notons que durant ces 15000 itérations, <i>O2O</i> déclenche en moyenne 3 fois moins de parties. . . . .	iv
A.3.	Comparaison des performances en fonction du nombre de parties jouées. . . . .	v

# Introduction générale

Ce projet tutoré est le premier du nom, porté par l'équipe SIMBIOT du LORIA, et a pour objectif de trouver et d'explorer une approche s'inscrivant dans le domaine des systèmes multi-agents qui n'ait pas déjà été exploitée par l'équipe. En dehors de chercher des pistes en gardant dans l'idée de devoir travailler avec un groupe d'agents (et dans le futur de drones plus particulièrement), il n'y a pas plus de limites imposées quant à ce que peut être cette approche explorée.

Ce présent rapport est tout d'abord l'introduction d'une approche originale et particulièrement prometteuse inspirée par l'évolution culturelle. Cette approche a été imaginée dans l'optique d'obtenir des groupes d'agents capables d'adapter leur comportement à leur environnement, amenant la possibilité d'améliorer l'autonomie d'un tel groupe. De plus, ce processus d'adaptation et d'apprentissage reposant sur les dynamiques d'opinions au sein du groupe, il n'implique pas des agents ayant des capacités d'apprentissage et de raisonnement complexes, propriété primordiale dans le cadre de la robotique en essaim.

Pour cette approche est défini un cadre formel fondé sur les processus décisionnels de Markov partiellement observables (dits POMDP d'après la terminologie anglaise *Partially Observable Markov Decision Process*). C'est dans ce cadre qu'est proposé un modèle permettant de doter un groupe d'agents de la capacité à adapter leurs comportements vis-à-vis d'une tâche prescrite en autonomie, ce qui était l'objectif que nous avons défini en conclusion de notre rapport bibliographique. Ce modèle repose sur l'introduction de biais dans les jeux de langage, approche inspirée par les travaux menés dans le champ de l'anthropologie et de la psychologie évolutionnaire [5]. Ce modèle est d'abord analysé dans un cadre minimal puis éprouvé dans le cadre d'une tâche de recherche de cibles développée sur la plateforme NetLogo. Les résultats montrent que ce modèle permet à un groupe d'agents d'apprendre des corrélations simples entre des stimuli et des récompenses et ainsi améliorer ses performances relativement à la tâche. Les propriétés recherchées de ce modèle sont :

- l'évolution adaptative des comportements dans le cas d'un environnement stationnaire et non-stationnaire, ou la capacité du groupe d'agents à apprendre quels sont les bons comportements à avoir mais aussi ré-apprendre de nouveaux comportements dans le cas où l'environnement en vient à changer ;
- la robustesse face à l'aléa dans l'environnement, ou la capacité du système à ne pas être perturbé par l'aléa qui est toujours présent dans le monde réel, c'est un aspect important à étudier dans l'optique d'une application avec des agents physiques ;
- la robustesse face à la variation de la taille de la population, ou la capacité du système à conserver ses qualités pour différentes tailles de population ;
- l'évolution continue des agents, ou la capacité de la population à continuellement progresser, à trouver de meilleures solutions.

La première partie du rapport consiste en une présentation du sujet sous forme bibliographique, présentant le socle d'étude sur lequel repose la suite du travail présenté. En seconde partie est introduite une formalisation à partir de laquelle est développé un modèle en tant qu'instanciation et dont les propriétés sont par la suite analysées. L'instanciation de ce modèle dans un but de recherche de cibles vient confirmer l'applicabilité de celui-ci. Enfin, nous présentons en conclusion les perspectives de développements envisageables à partir de ce modèle selon différents prismes.

# 1. L'évolution culturelle comme source d'inspiration

Dans l'optique future de permettre à un groupe de drones de s'adapter à un environnement qui leur est inconnu de manière collaborative, nous avons employé une approche inspirée de l'évolution culturelle utilisant les *Language Games* pour leur permettre de communiquer et d'apprendre les uns des autres en s'interrogeant sur les comportements qu'ils adoptent selon ce qu'ils perçoivent de l'environnement.

## 1.1. Introduction aux concepts-clés pour une approche inspirée par l'évolution culturelle

L'évolution culturelle et les Language Games sont les deux socles théoriques importants à la construction de notre approche et nécessitent une explication de leurs principes fondamentaux pour le bien de la compréhension de ce qui suit.

### 1.1.1. Généralités sur l'évolution culturelle

L'évolution culturelle revêt dans nos sociétés un rôle adaptatif primordial. En effet, ce partage de connaissances permet à une population de s'adapter relativement rapidement à un environnement qui peut être changeant. De plus, ce mécanisme permet d'éviter en partie le coût de l'apprentissage individuel. Par exemple, lorsqu'une population doit choisir de la nourriture, il peut être coûteux que chaque individu doive apprendre quelles denrées sont comestibles et lesquelles ne le sont pas par essai-erreurs, alors que l'évolution culturelle permet à la majorité de la population de se baser sur les expériences d'autres individus.

Nous nous basons ici sur l'hypothèse des répliqueurs (ou des mêmes) qui veut que les traits culturels (ce qui compose une culture) soient des entités qui se transmettent entre individus par le biais de l'apprentissage social (l'imitation par exemple). Nous parlerons donc ici de culture comme l'ensemble des traits qui se transmettent entre individus par apprentissage social.

Cette idée de transmission de traits n'est pas sans rappeler l'évolution biologique. Premièrement, les traits culturels sous l'hypothèse des répliqueurs, tout comme les gènes, sont des entités capables de se transmettre d'individu en individu. La différence étant qu'ici, les traits culturels peuvent être transmis horizontalement (entre individus d'une même génération) alors que la transmission génétique est uniquement verticale. Deuxièmement, à l'instar des gènes qui subissent des erreurs de réplication (des mutations), les traits culturels sont également soumis à des variations aléatoires. Par exemple, si l'on regarde quelqu'un faire un gâteau pour le reproduire, il est possible que l'on n'emploie pas exactement la même technique que le modèle parce que l'on a mal regardé.

Afin que cette évolution revête un rôle adaptatif, il est nécessaire que la nature (le contenu) d'un trait culturel puisse impacter la propension de ce dernier à être transmis, tout comme un gène peut conférer un avantage reproducteur et ainsi être davantage transmis dans la génération suivante. Nous appellerons cet avantage la fitness du trait culturel. Ainsi, tout comme un « bon » gène se transmet davantage s'il confère un avantage reproducteur à son porteur, un bon trait culturel se transmet davantage s'il permet que l'individu porteur de ce trait soit imité plus fréquemment. Il en ressort alors une question : par quels mécanismes un individu va-t-il être plus limité qu'un autre en fonction de ses traits culturels et de leurs fitness respectives ?

Dans *The Evolution of Cultural Evolution* [5], les auteurs proposent que l'évolution biologique ait amené l'apparition de certains biais de jugement favorisant l'évolution culturelle. Il est important de noter que les dynamiques de l'évolution culturelle sont grandement impactées par les biais de par la nature psychologique de l'environnement dans lequel les mêmes circulent. Ces biais de jugements peuvent être classés en deux classes distinctes, les biais basés sur le contenu et les biais basés sur le contexte. La première se réfère à la propension d'un individu à adopter un nouveau trait en fonction de son contenu, indépendamment de l'individu qui le porte. Concernant la deuxième, cette adoption est fonction du contexte social dans lequel l'individu évolue. Nous pouvons séparer cette dernière en deux sous-catégories, les biais basés sur le modèle (celui qu'on imite), et les biais basés sur la fréquence (le nombre de fois où le trait culturel a été observé). Dans la première, nous pouvons retrouver le biais dit de succès, biais qui augmente la propension d'un individu à imiter un autre individu qui est plus performant —dans un contexte donné— que lui. Ou encore le biais de prestige, qui se réfère au rang social du modèle. Concernant les biais basés sur la fréquence, nous pouvons trouver le biais de conformisme qui comme son nom l'indique va avantager les traits qui sont le plus répandus, et son contraire, le biais de nouveauté, qui avantage les traits nouveaux.

### 1.1.2. Principes des Language Games

Pouvoir discriminer un environnement complexe est primordial et les Language Games sont une approche utilisée ici en ce sens.

Historiquement, Ludwig Wittgenstein introduit les language games (jeux de langage) dans l'idée de proposer une autre vision de ce qu'est la signification d'un mot, alors que ses contemporains philosophes du langage considéraient communément qu'un mot réfère à un objet réel, ce qui impliquait que le mot n'était qu'une dénomination de l'objet réel [13]. Wittgenstein conçoit la signification du mot comme étant l'usage du mot dans un contexte, et non comme l'objet réel. C'est ici qu'interviennent les language games : un language game est ce contexte d'usage, qu'il définit comme étant constitué à la fois du langage et des actions avec lesquelles se lie le langage. Les language games sont regroupés sous cette dénomination sans pour autant partager un ensemble défini de caractéristiques, ils ont au contraire vocation à montrer l'hétérogénéité des usages du langage (et montrant par cette hétérogénéité que considérer la signification d'un mot comme étant directement l'objet du monde réel est un raccourci fallacieux).

À l'aide des réflexions de Wittgenstein —qui étaient purement philosophiques—, Luc Steels (linguiste et informaticien de formation) a introduit les Language Games (distingués des language games de Wittgenstein par des majuscules) dès 1995 : il est alors question du *Naming Game* (bien que celui-ci ne soit pas encore nommé ainsi dans son article *A Self-Organizing Spatial Vocabulary* qui est le premier de Steels s'appuyant sur les language games de Wittgenstein) [12]. Son objectif dans cet article était de démontrer qu'il était possible qu'un ensemble d'agents fasse émerger du vocabulaire afin que ceux-ci puissent s'identifier entre eux par un nom et une description de leur position spatiale : l'emphase était alors sur les dynamiques de négociation et l'émergence d'un consensus. Steels part du postulat de Wittgenstein disant que le comportement linguistique peut être imaginé comme une suite de language games ; il voit une partie comme une communication entre au moins un *speaker* (orateur) et un *hearer* (auditeur) qui partagent une perception commune de l'environnement où le speaker va désigner un objet de l'environnement, et le hearer cherche à identifier celui-ci (on retrouve le principe des language games où un mot prend sens selon son usage dans ce contexte, sachant que le contexte chez Wittgenstein et le contexte chez Steels sont différents : le premier l'interprète comme la conversation et le deuxième comme l'environnement).

### 1.1.3. Intérêt d'une approche inspirée de l'évolution culturelle

L'évolution culturelle semble être une source d'inspiration prometteuse dans l'optique du développement d'essaims de robots aptes à s'adapter à leur environnement. L'évolution culturelle présente des propriétés correspondant à nos besoins, principalement une adaptation rapide. L'évolution culturelle tire son efficacité du fait qu'une population est un réservoir de comportements. Ledit réservoir

se trouvant en constante évolution de par le fait que les comportements le composant se répandent plus ou moins en son sein selon leur caractère adaptatif dans un environnement donné et à un moment donné. Par ailleurs, du fait du couplage entre apprentissage individuel et apprentissage social, l'évolution culturelle est par essence Lamarckienne, en d'autres termes, le vécu d'un agent impacte ce qu'il transmet. Une évolution de ce type est par nature plus rapide que dans le cadre Darwinien.

Concernant les Language Games, accessibles facilement à l'entendement humain puisque se fondant sur nos interactions langagières, ceux-ci sont aussi algorithmiquement très simples. De plus, ils ont déjà démontré leur capacité à développer des vocabulaires communs à des groupes d'agents, ainsi que les possibilités étendues qu'ils offrent, nous permettant d'envisager de les modeler à notre convenance.



## 2. Vers l’adaptation de comportements par le biais de l’évolution culturelle pour des essaims de robots autonomes

Notre nouvelle approche inspirée de l’évolution culturelle prend comme point de départ la hiérarchisation des niveaux d’application des Language Games effectuée par Cambier *et al.* (2020) [4]. C’est le troisième niveau en particulier qui a permis de définir une boucle de rétroaction servant de fondation théorique au cadre formel s’inscrivant dans ladite approche. Le cadre en question s’inspire pour son aspect pratique du formalisme des processus décisionnels de Markov partiellement observables (POMDP). Le modèle utilisant le cadre formel qui vient d’être défini emploie quant à lui un Language Game particulier, le Naming Game, et va introduire des biais dans ce Language Game afin de modifier les dynamiques de langage dans la population et d’en faire émerger certaines propriétés qui seront décrites par la suite. Les propriétés définies dans l’introduction générale sont alors étudiées afin de déterminer si le modèle est suffisamment performant. Enfin il est aussi mis à l’épreuve par son instanciation dans une tâche de recherche de cibles.

### 2.1. Conceptualisation de l’approche : cadre formel et modèle

Dans le cadre de la robotique en essaim, les Language Games peuvent être utilisés selon 3 niveaux distincts [4], le troisième en particulier étant celui sur lequel se fonde notre cadre formel, niveau intéressant dans sa définition d’une approche dans laquelle le langage d’un agent influe sur ses comportements. Les trois niveaux sont les suivants :

1. Le niveau basique ou simple dans lequel le langage émerge/évolue de manière indépendante au comportement des agents.
2. Le niveau intermédiaire, où le comportement des agents influe sur le développement du vocabulaire/langage. Par exemple, l’étude de Miletitch *et al.* (2019) [8] où la création de nouvelles unités linguistiques dépend de la découverte de celles-ci et donc du comportement des agents. Dans cet article, il y a différents types de nourriture, à chaque fois qu’un agent en découvre une nouvelle il crée une nouvelle unité linguistique qui va être débattue au travers des jeux de langage afin de trouver un consensus sur le mot la décrivant.
3. Le troisième niveau, où le comportement influe sur le développement linguistique qui lui-même influe sur le comportement. Cambier *et al.* (2018) [3] ont étudié cette hypothèse et ont proposé un tel type de boucle. Dans leur papier les agents utilisent le *Minimal Naming Game* pour se mettre d’accord sur un seul et unique mot lors d’une tâche d’agrégation. Leur modèle de déplacement est défini par un paramètre que nous noterons  $p$ . L’astuce est qu’ici les mots générés définissent/encodent ce paramètre  $p$ . C’est pourquoi les agents qui sont le plus à même de s’agrèger avec un grand nombre d’individus (qui ont donc un paramètre  $p$  plus optimal) sont également plus aptes à transmettre leur vocabulaire au travers des jeux de langage (leur vocabulaire définissant ce paramètre  $p$ ). Ainsi on crée une boucle dans laquelle les individus qui ont un vocabulaire définissant le paramètre  $p$  le plus optimal sont le plus à même de transmettre ce dit paramètre. On a donc une boucle de feedback positif.

C’est cette boucle en particulier qui forme le fondement conceptuel de notre approche (cf. figure 2.1).

L’idée proposée par Cambier *et al.* (2018) [3] est très intéressante, mais n’est appliquée que sur la recherche d’un unique mot/paramètre. De plus, le choix de la tâche est particulièrement arrangeant

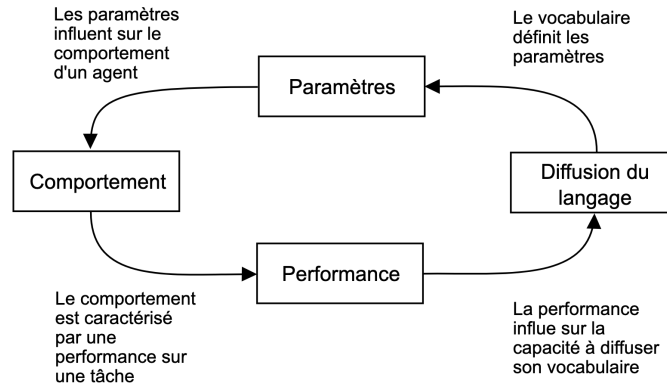


FIGURE 2.1. – Proposition d’une boucle de feedback positif dans la métaphore de l’évolution culturelle.

puisqu’elle ferme d’elle-même la boucle de feedback positif, une bonne performance entraînant directement une capacité accrue à communiquer (de par le fait qu’un agent performant est un agent qui est proche d’autres agents et qui par conséquent peut communiquer plus aisément).

Concernant le premier point, une approche où des mots définissent des stimuli, des concepts différents serait particulièrement intéressante dans l’optique de développer des comportements qui seraient dépendants de l’environnement dans lequel évoluent les agents. Deuxièmement, une problématique pour généraliser ce type d’approche est de trouver un mécanisme suffisamment général pour permettre de fermer cette boucle de rétroaction même lorsqu’une bonne performance à la tâche prescrite n’implique pas d’elle-même une meilleure capacité à transmettre son vocabulaire.

### 2.1.1. Un cadre formel pour une approche inspirée par l’évolution culturelle

Le travail présenté ci-après propose la formalisation du cas d’une population d’agents évoluant dans un environnement et apprenant à agir dans celui-ci via l’évolution culturelle, relativement à une tâche prescrite munie d’un critère de performance. Ce formalisme permet de définir un cadre à une approche particulièrement nouvelle et inspirée de l’évolution culturelle, cadre utile dans l’optique de servir d’appui à de futurs développements. De plus, en se basant sur le formalisme des POMDP, qui est notamment très répandu dans le domaine de l’apprentissage par renforcement, ce cadre pourrait alors permettre de faciliter la jonction entre différents champs de recherche et ainsi proposer un terrain fertile pour de nouvelles méthodes inspirées de ces derniers.

Dans ce cadre, les agents peuvent percevoir différents stimuli classés en concepts (e.g. des couleurs) présents dans leur environnement et possèdent un vocabulaire, soumis aux interactions avec les autres agents, décrivant ces premiers. Ces concepts peuvent être fixés a priori ou non. Les agents agissent en boucle fermée dans l’environnement et décident leurs actions selon un modèle comportemental (de décision) paramétrique.

Chaque mot du vocabulaire a une signification comportementale pour les agents, en d’autres termes, un mot définit une manière d’agir dans le monde (les paramètres du modèle de décision). Ainsi, lorsqu’un agent perçoit un certain concept de stimulus, le vocabulaire qu’il associe aux différents stimuli présents va conditionner sa manière d’agir.

Le choix d’un modèle de décision paramétrique vient de la visée première de ce type d’approche qui est de proposer une méthode permettant à un groupe d’agents de s’adapter post-déploiement. Il aurait été possible de définir le vocabulaire comme encodant directement les actions prises mais cela aurait impliqué des comportements totalement insatisfaisants dans les premières phases.

Dans ce cadre, les agents sont confrontés à un problème muni d’un critère de performance relatif à ce dernier. Ce problème peut être formalisé de manière générale par un POMDP qui est la donnée de  $\langle S, A, P, r, \Omega, O \rangle$  avec :

- $S$ , l’ensemble des états du monde ;

- $\Omega$ , l'ensemble des observations. Ce sont toutes les perceptions que peut avoir un agent. Une observation peut être composée de plusieurs concepts différents ;
- $A$ , l'ensemble des actions que peut réaliser un agent ;
- $O : S \rightarrow \Omega$ , la fonction d'observation qui à tout état du monde définit ce que l'agent perçoit ;
- $P : S \times A \times S \rightarrow [0, 1]$ , la fonction de transition entre états (les lois du monde) qui définit la probabilité de transition d'un état du monde à un autre après qu'un agent ait effectué une action ;
- $r : S \times A \times S \rightarrow \mathbb{R}$ , la fonction de récompense qui définit la récompense que reçoit un agent lorsqu'il est dans un certain état, réalise une action et se trouve dans un nouvel état donné.

Afin de formaliser les interactions entre agents, nous ajoutons ici 2 types :

- $M$ , un ensemble de messages définis dans un alphabet donné ;
- $RM$ , un ensemble de réponses définies dans un alphabet.

Dans le cadre défini par les POMDP, un agent appartenant à un groupe, dans cette approche, est la donnée de :

- $L$ , un vocabulaire ;
- $C$ , un ensemble de concepts différents (aussi nommés catégories perceptives) ;
- $\Theta$ , l'espace des paramètres comportementaux ;
- $c : \Omega \rightarrow E$ ,  $E \subseteq C$ , une fonction de classification qui à toute observation associe les différentes catégories/concepts de stimuli présentes. Une scène (une observation) peut être composée de plusieurs catégories/concepts de stimuli différents. Dans un cadre général, nous pourrions également créer des catégories/concepts sur n'importe quel état d'information et pas seulement l'observation courante (sur l'historique de l'agent par exemple) ;
- $I : C \rightarrow K$ ,  $K \subseteq L$ , une fonction qui à toute catégorie perceptive associe un mot ou plusieurs mots ;
- $D : P(L) \rightarrow L$ , où  $P(L)$  est l'ensemble des sous-parties de  $L$  (i.e. l'ensemble puissance de  $L$ ). Cette fonction permet de décider quel mot utiliser parmi l'ensemble des mots qui peuvent être associés à un concept (i.e. les mots dans  $I(c)$ ) ;
- $\phi : L \rightarrow \Theta$ , une fonction d'interprétation qui à tout mot associe des paramètres comportementaux.

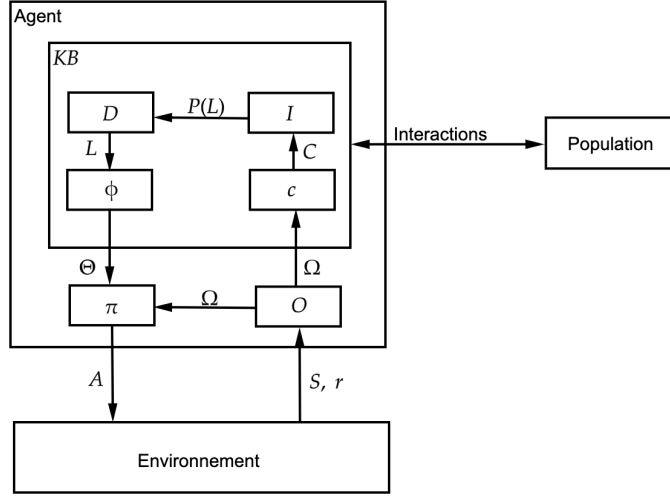
Le comportement des agents se décline sur deux niveaux distincts. Le niveau actionnel, qui définit les actions qu'ils effectuent sur l'environnement, et le niveau interactionnel, qui définit comment ils interagissent avec les autres agents et comment ils mettent à jour leurs connaissances du fait de ces interactions. Nous définissons ici une interaction comme un élément de  $M \times RM$ . Pour des raisons de simplicité, nous noterons ici la base de connaissance  $KB$  d'un agent comme étant la donnée de  $\langle L, C, c, I, D, \phi \rangle$ .

Définition du niveau interactionnel :

- $Send : KB \rightarrow M$ , la fonction qui définit ce qu'envoie un agent pour initier une interaction.
- $Ans : M \times KB \rightarrow RM$ , une fonction qui définit la réponse d'un agent à un message reçu relativement à ses connaissances ;
- $U : M \times RM \times KB \rightarrow KB$ , la fonction de mise à jour qui définit comment un agent modifie ses connaissances à la suite d'une interaction  $(M, RM)$  ;

Définition du niveau actionnel :

- $\pi : \Theta \times \Omega \rightarrow A$ , la fonction de décision qui, selon les paramètres comportementaux, associe à toute observation une action.



**FIGURE 2.2.** – Schématisation de l’architecture interne de l’agent dans cette approche ; l’agent agissant en boucle fermée dans l’environnement et interagissant avec le reste du groupe d’agents.

Ce cadre formel est décrit dans la figure 2.2. Le niveau interactionnel n’y est pas spécifié par souci de garder la figure lisible.

Un critère de performance  $\rho_\tau : \mathbb{R}^\tau \rightarrow \mathbb{R}$  défini sur une durée donnée  $\tau$ , donc prenant en compte les  $\tau$  derniers pas de temps, doit permettre de hiérarchiser les agents. Si  $\tau = \infty$ , la performance d’un agent est fonction de l’ensemble des récompenses reçues depuis l’initialisation.

Soient  $r_i^t$  la récompense reçue par l’agent  $i$  au temps  $t$  et  $p_i^t$ , sa performance définie selon  $\rho_\tau$  et soient  $i$  et  $j$  deux agents, alors un critère de performance  $\rho_\tau$  doit vérifier la propriété suivante :

$$\sum_{\Delta t=0}^{\tau} r_i^{T-\Delta t} > \sum_{\Delta t=0}^{\tau} r_j^{T-\Delta t} \Rightarrow p_i^T \geq p_j^T \quad (2.1)$$

La performance d’un agent peut simplement être définie comme la somme des récompenses qu’il a reçues au cours du temps.

### 2.1.2. Introduction de biais dans les Language Games : vers une évolution adaptative

Notre modèle, dans le cadre présenté précédemment, utilise les jeux de langage comme modèle d’interaction entre agents. Plus précisément, il repose sur l’introduction de biais dans ces derniers, particulièrement le biais de succès ainsi que le biais de nouveauté, biais qui vont modifier les dynamiques de langage dans la population. L’hypothèse est que le biais de succès permet de diriger les dynamiques de vocabulaires de manière à augmenter la fréquence des mots amenant de meilleures performances et que le biais de nouveauté permettrait de sauvegarder de la variabilité afin de tendre vers une évolution qui soit continue (qui ne s’arrête pas sauf en cas d’optimum global) et qui soit en mesure de s’adapter à des changements.

Nous présentons le Naming Game, choisi pour sa simplicité, auquel sont introduits des biais.

#### 2.1.2.1. Description du Naming Game (NG)

Le NG (*Naming Game*, jeu de dénomination) est organisé en trois phases dans l’algorithme 1. D’abord, le speaker choisit une catégorie (le topic) de stimulus et un mot associé à cette catégorie (en général le dernier enregistré), il envoie le topic ainsi que le mot choisi au hearer. Si le hearer possède déjà ce mot associé à cette catégorie alors la partie est un succès et les deux parties ne gardent que le mot impliqué comme associé à ce topic. Sinon, le jeu est un échec et le hearer ajoute le mot à l’ensemble des mots associés à cette catégorie.

---

**Algorithme 1** : Algorithme du Naming Game.

---

**Données** : speaker, hearer

- 1  $c \leftarrow U(C)$  avec  $U$  la loi uniforme
  - 2  $w \leftarrow D_{\text{speaker}}(I_{\text{speaker}}(c))$
  - 3 **si**  $w \in I_{\text{hearer}}(c)$  **alors**
  - 4    $I_{\text{speaker}}(c) \leftarrow \{w\}$
  - 5    $I_{\text{hearer}}(c) \leftarrow \{w\}$
  - 6 **sinon**
  - 7    $I_{\text{hearer}}(c) \leftarrow I_{\text{hearer}}(c) \cup \{w\}$
- 

### 2.1.2.2. Introduction d'un biais de succès dans le NG

Soit un critère de performance  $\rho_\tau$  donné relativement à une tâche et soit  $p_i$  la performance de l'agent  $i$ . Un agent  $i$  avec un biais de succès, lorsqu'il est hearer, n'accepte de jouer que si le speaker  $j$ , a une performance plus élevée que la sienne ( $p_j > p_i$ ) (cf. algorithme 2).

---

**Algorithme 2** : Algorithme du Naming Game avec biais de succès chez le hearer.

---

**Données** : speaker, hearer

- 1  $c \leftarrow U(C)$
  - 2  $w \leftarrow D_{\text{speaker}}(I_{\text{speaker}}(c))$
  - 3 **si**  $p_{\text{speaker}} \leq p_{\text{hearer}}$  **alors**
  - 4   fin
  - 5 **sinon si**  $w \in I_{\text{hearer}}(c)$  **alors**
  - 6    $I_{\text{speaker}}(c) \leftarrow \{w\}$
  - 7    $I_{\text{hearer}}(c) \leftarrow \{w\}$
  - 8 **sinon**
  - 9    $I_{\text{hearer}}(c) \leftarrow I_{\text{hearer}}(c) \cup \{w\}$
- 

### 2.1.2.3. Introduction d'un biais de nouveauté dans le NG

Un agent avec un biais de nouveauté garde le mot envoyé par le speaker uniquement s'il n'a jamais rencontré ce mot au cours des  $n_{\text{memory}}$  dernières parties impliquant le topic choisi. Pour ce faire, cet agent enregistre en mémoire les  $n_{\text{memory}}$  derniers mots reçus pour chaque catégorie de stimulus. Lorsqu'un speaker lui envoie un mot et une catégorie, il regarde s'il a déjà reçu ce mot durant les  $n_{\text{memory}}$  dernières parties pour lesquelles le sujet était la catégorie reçue, si oui, alors il n'enregistre pas le mot s'il ne l'a pas déjà associé à la catégorie actuelle (cf. algorithme 3).

---

**Algorithme 3** : Algorithme du Naming Game avec biais de nouveauté chez le hearer.

---

**Données** : speaker, hearer

- 1  $c \leftarrow U(C)$
  - 2  $w \leftarrow D_{\text{speaker}}(I_{\text{speaker}}(c))$
  - 3 **si**  $w \in I_{\text{hearer}}(c)$  **alors**
  - 4    $I_{\text{speaker}}(c) \leftarrow \{w\}$
  - 5    $I_{\text{hearer}}(c) \leftarrow \{w\}$
  - 6 **sinon si**  $w \notin \text{memory}(c)$  **alors**
  - 7    $I_{\text{hearer}}(c) \leftarrow I_{\text{hearer}}(c) \cup \{w\}$
  - 8 **sinon**
  - 9   fin
-

#### 2.1.2.4. Introduction de mutations dans le NG

A l’instar de l’évolution génétique, mais comme expliqué auparavant pas non plus étranger à l’évolution culturelle, nous avons introduit dans notre modèle de la variation par le biais de mutations. Lors de chaque partie de NG jouée, il advient une erreur de transmission avec probabilité  $p_{mut}$  qui change aléatoirement l’une des lettres du mot envoyé par le speaker. Ce type de variation permet d’éviter des convergences trop rapides dans des optimums locaux et plus généralement dans des sous-optimums. De plus, conserver un certain niveau de variabilité permet également de s’adapter plus aisément lorsque les conditions environnementales sont amenées à changer.

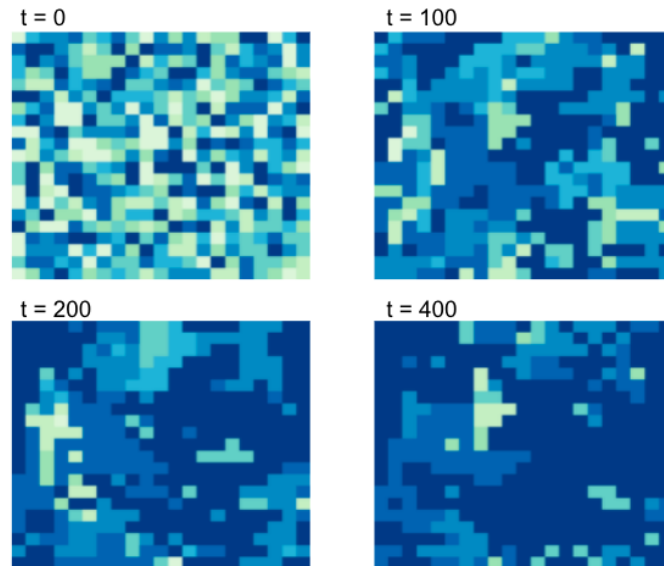
#### 2.1.3. Analyse des propriétés du modèle

Afin d’étudier plus en profondeur les propriétés du modèle proposé, nous avons confronté notre modèle à des problèmes d’optimisation afin d’en analyser les caractéristiques dans un environnement qui soit le plus contrôlé possible. Nous nous intéressons particulièrement ici à 4 propriétés primordiales dans le cas de systèmes multi-agents autonomes :

- l’adaptation des comportements dans le cas d’un environnement stationnaire et non-stationnaire ;
- la robustesse face aux fluctuations aléatoires dans l’environnement ;
- la robustesse face à la variation de la taille de la population ;
- l’évolution continue des agents.

Pour ce faire nous avons disposé des agents sur une grille de dimension  $n \times n$ . Les agents ne peuvent faire une partie de NG qu’avec l’un de leurs 4 voisins directs. A chaque itération,  $n_{games}$  parties sont déclenchées.

Parmi les  $N = n^2$  agents,  $N_{success} = N \times f_{success}$  ont un biais de succès, et,  $N_{novelty} = N \times f_{novelty}$  ont un biais de nouveauté, le reste des agents n’ont pas de biais. Une simulation a été réalisée en NetLogo et observable en figure 2.3.

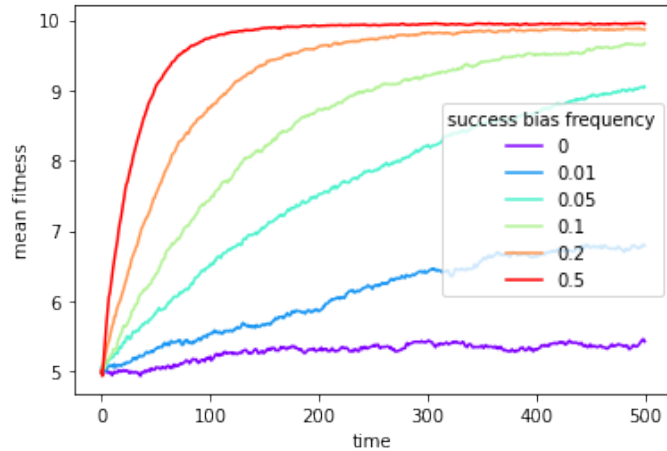


**FIGURE 2.3.** – Évolution du vocabulaire de 400 agents placés sur une grille. Chaque cellule est un agent et sa couleur représente la fitness associé au mot présent dans son vocabulaire.  $t$  est le nombre de pas de temps.

Les mots échangés via le NG sont ici des nombres binaires de longueur 10 pouvant encoder 1024 valeurs possibles qui sont mises à l’échelle dans l’intervalle  $[0 ; 10]$ . La fonction d’évaluation (ou de fitness) est ici simplement la fonction identité, le paramètre optimal étant donc 10 et le mot associé “1111111111”. Dans ce cadre simplifié, nous passons directement à la notion de fitness sans passer par une fonction de récompense. Ici, la fitness d’un individu peut être interprétée comme son espérance de performance sachant son vocabulaire.

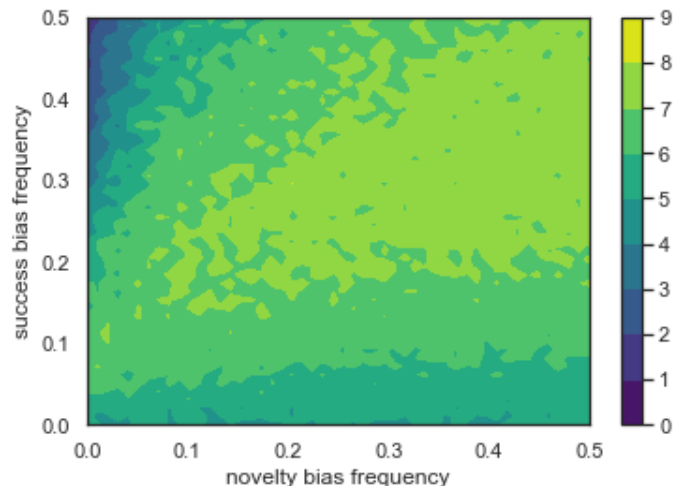
### 2.1.3.1. Analyse du caractère adaptatif de l'évolution des comportements dans le cas d'un environnement stationnaire et non-stationnaire

La présence de biais de succès amène la population à converger vers de bons paramètres même avec de faibles fréquences. La figure 2.4 permet de constater qu'il y a apprentissage dès 5% d'individus avec un biais de succès dans un environnement stationnaire (qui ne change pas).



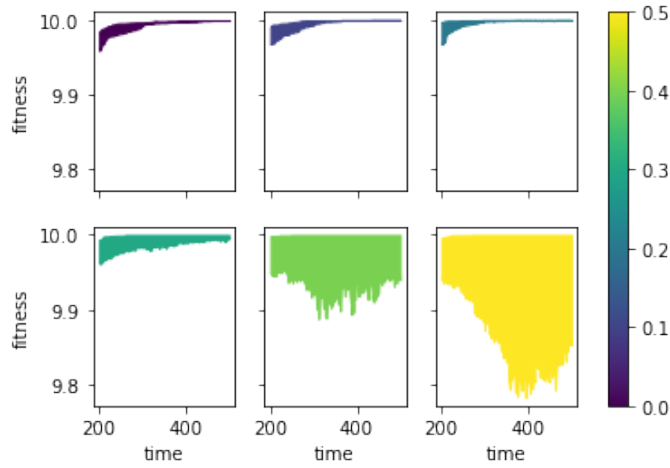
**FIGURE 2.4.** — Évolution de la fitness moyenne de la population ( $N = 400$ ) en fonction de la fréquence de biais de succès (avec  $f_{novelty} = 0$ ).

La capacité d'un groupe à s'adapter à des contraintes environnementales pouvant être changeantes est une propriété très intéressante dans l'optique d'agents plus autonomes. Nous avons rendu la fonction de fitness de l'environnement non-stationnaire —c'est-à-dire changeante dans le temps—, afin d'évaluer la capacité de ré-adaptation de ce modèle. Nous avons pour cela laissé la population évoluer pendant 200 pas de temps (ce qui permet d'atteindre une bonne convergence), puis la fonction de fitness a été inversée, les moins bons paramètres devenant les meilleurs et inversement. Nous avons évalué la capacité d'adaptation comme la performance atteinte 500 pas de temps après l'inversion. Les résultats montrent que l'interaction entre les biais de succès et de nouveauté favorise l'adaptativité de la population (cf. figure 2.5). Cela est dû à la capacité des biais de nouveauté de conserver de la variabilité dans les idées au sein de la population, et des biais de succès à exploiter ces variations lorsqu'elles sont bénéfiques.



**FIGURE 2.5.** — Capacité de réadaptation face à une fonction de fitness non-stationnaire en fonction de la fréquence de biais de succès et de nouveauté.

Les biais de nouveauté montrent une propriété intéressante qui est de générer de la variabilité principalement après convergence du vocabulaire (cf. figure 2.6). Cet effet est intéressant dans le sens où les individus porteurs de ce biais ne freinent pas la convergence initiale.



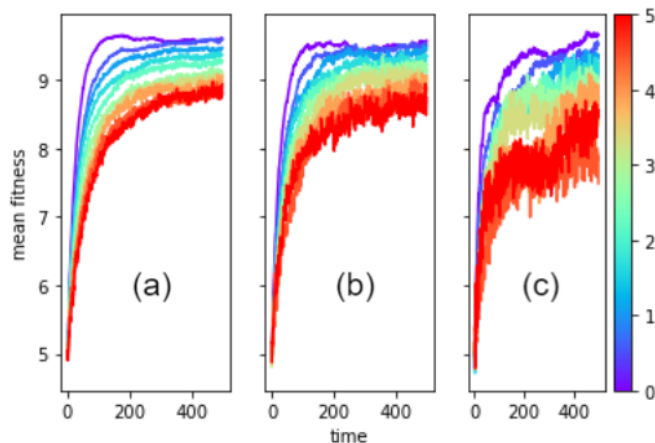
**FIGURE 2.6.** – Évolution de la diversité du vocabulaire (des paramètres) en fonction de la fréquence de biais de nouveauté.

### 2.1.3.2. Analyse de la stabilité de la performance face à l'introduction d'aléa dans l'environnement

La robustesse d'un système, autrement dit la capacité de celui-ci à conserver ses propriétés en présence de fluctuations aléatoires dans l'environnement, est une propriété fondamentale lorsque l'objectif du système est de tendre à être fonctionnel dans un environnement plus proche du réel, dans lequel les erreurs et l'aléatoire sont toujours présents.

Afin d'évaluer la robustesse du système, de l'erreur est introduite dans la fonction de fitness. Ainsi, la fitness d'un paramètre à un moment donné suit une loi normale dont l'espérance est la fitness réelle et d'écart type  $\sigma$  ( $f(x) \sim \mathcal{N}(x, \sigma)$ ). Le comportement du modèle est ici évalué pour différentes tailles de population et différents  $\sigma$ .

Les résultats en figure 2.7 montrent que plus la population est petite, plus elle est sensible à l'aléa. Ce phénomène s'explique par le fait que dans une population de petite taille, il est possible qu'un individu avec de mauvais paramètres ait une meilleure performance que les autres agents du fait de l'aléa et ainsi transmette efficacement son vocabulaire. Dans une population de grande taille, au contraire, il est très peu probable qu'un individu avec de mauvais paramètres performe mieux que tous les individus avec de meilleurs paramètres, ceux-ci transmettront par conséquent plus efficacement leur vocabulaire que ce premier.



**FIGURE 2.7.** – Évolution de la fitness moyenne en fonction de la taille de la population et de l'écart type ( $\sigma$ ) de la fonction de fitness  $f$  avec  $f(x) \sim \mathcal{N}(x, \sigma)$  (le code couleur des courbes représente la valeur de  $\sigma$ ). (a)  $N = 400$ . (b)  $N = 100$ . (c)  $N = 25$ .

Ce type de phénomène s'explique sous le prisme de la loi des grands nombres. Soit  $m$  et  $k$  deux traits



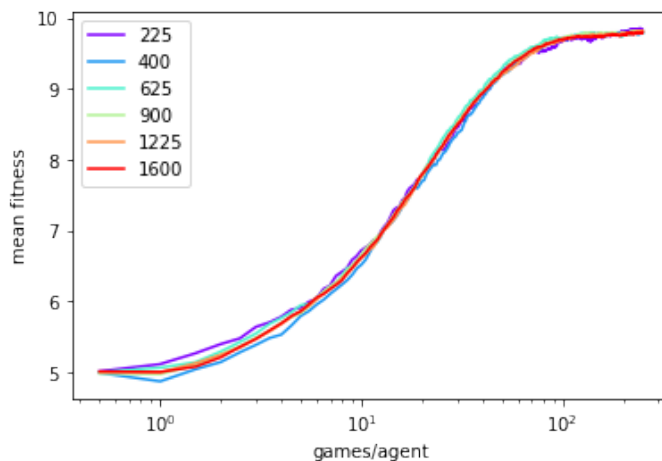
culturels distincts et  $f_m$  et  $f_k$  leurs fitness réelles respectives. Notons  $\overline{P}_i = \frac{1}{n_i} \sum p_x$  la performance moyenne des individus porteurs du trait  $i$  ( $n_i$  étant le nombre d'individus portant le trait  $i$ ) avec  $p_x \sim \mathcal{N}(f_i, \sigma)$  la performance de l'agent  $x$  porteur du trait  $i$ . Alors, selon la loi des grands nombres, nous avons :

$$\lim_{N \rightarrow \infty} [\mathbb{P}(\overline{P}_m > \overline{P}_k \mid f_m < f_k)] = 0 \quad (2.2)$$

Avec  $N$  la taille de la population, les populations porteuses de chaque trait étant supposées proportionnelles à la taille totale. En partant de l'hypothèse qu'un trait se transmet mieux qu'un autre s'il octroie une meilleure performance à ses porteurs en moyenne, alors, cette équation nous dit que plus la population est grande, plus la probabilité qu'un trait  $m$  ayant une moins bonne fitness qu'un trait  $k$  se transmette mieux que ce dernier est faible.

### 2.1.3.3. Analyse de la stabilité de la performance face à la variation de la taille de la population

Le passage à l'échelle du modèle, soit sa capacité à conserver ses qualités pour des tailles de population croissantes, est une propriété importante et recherchée pour l'intelligence en essaim. Le modèle ne montre aucune détérioration lorsque la taille augmente. En effet le temps de convergence reste constant sur toute les tailles de population testées lorsque le nombre de parties jouées à chaque itération est proportionnel à la taille de la population (cf. figure 2.8).



**FIGURE 2.8.** – Évolution de la fitness moyenne en fonction du nombre de parties par agent pour différentes tailles de population.

### 2.1.3.4. Analyse du caractère continu de l'évolution des agents selon l'injection de mutations

Bien que tout système adaptatif tende à améliorer sa performance au cours du temps, il persiste le risque de converger vers des solutions sous-optimales. Ainsi la capacité du système à générer continuellement de la nouveauté et donc potentiellement des améliorations est très importante.

À l'instar des algorithmes génétiques, la génération de variabilité via les mutations (voir section 2.1.2.4, page 10) permet à la population d'agents de ne pas rester bloquée au niveau du meilleur individu initial, mais de trouver de meilleures solutions, d'innover en quelque sorte. Les résultats visualisables en figure 2.9 montrent effectivement que sans ou avec trop peu de variation, la fitness moyenne atteint un plateau alors qu'avec suffisamment elle tend vers la fitness optimale.

### 2.1.3.5. Synthèse de l'analyse des propriétés du modèle

Le modèle montre en effet une capacité à s'adapter et se ré-adapter à un environnement (stationnaire ou non), que pour des tailles de population suffisamment grandes il est robuste face à l'aléa, que

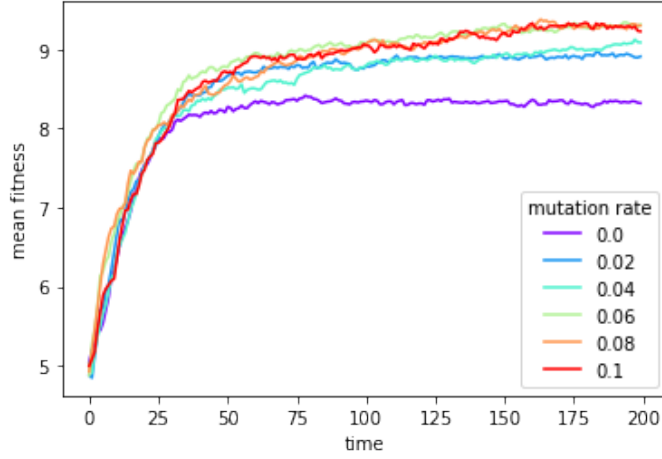


FIGURE 2.9. – Évolution de la fitness moyenne pour différentes fréquences de mutation.

les temps de convergence restent constants pour toute taille de la population ainsi qu’une capacité à s’adapter de manière continue grâce à l’injection de variation par le biais de mutations.

Les propriétés analysées sont donc avérées selon les résultats obtenus, cependant cela reste des premiers résultats et le modèle mériterait d’être exploré plus en profondeur dans le futur.

## 2.2. Vers une application concrète du modèle : recherche de cibles

Afin d’éprouver ce modèle, nous avons développé sous NetLogo une tâche classiquement étudiée dans le champ de la robotique en essai : la recherche de cibles. Cette tâche fut créée en s’inspirant d’un cas d’usage où les cibles recherchées sont des victimes dans un environnement post-tremblement de terre où le terrain est inconnu et risqué. Néanmoins bien d’autres cas d’usage ne menant pas à cette tâche en particulier auraient pu servir de source d’inspiration étant donné que le modèle en lui-même est tout à fait général, qu’il s’agisse fondamentalement d’une tâche de recherche de cibles est fortuit. Est décrit dans un premier temps cette tâche selon le formalisme précédemment introduit. Ensuite, nous présenterons les résultats obtenus, que nous discuterons dans un dernier temps.

### 2.2.1. Description de la tâche

L’environnement est un tore de taille  $l \times l$  dans lequel sont disposées  $n_{targets}$  cibles aléatoirement. Des zones de danger sont également placées aléatoirement au nombre de  $n_{dangers}$ . Ces zones, si un agent passe trop près, ont pour effet de ralentir ce dernier fortement. Les concepts sont ici définis a priori comme étant les dangers et les cibles ( $c = \{danger, cible\}$ ). Les dangers ainsi que les cibles émettent un signal (que l’on peut par exemple interpréter par la chaleur émise par un incendie, et les cris poussés par une victime), respectivement  $s_{danger}$  et  $s_{target}$ , qui se diffusent à une vitesse  $diff$  et s’évaporent à une vitesse  $evap$ . Les  $n_{agents}$  agents sont également disposés à des positions aléatoires dans l’environnement. Un état du monde appartenant à  $S$  est donc la donnée de :

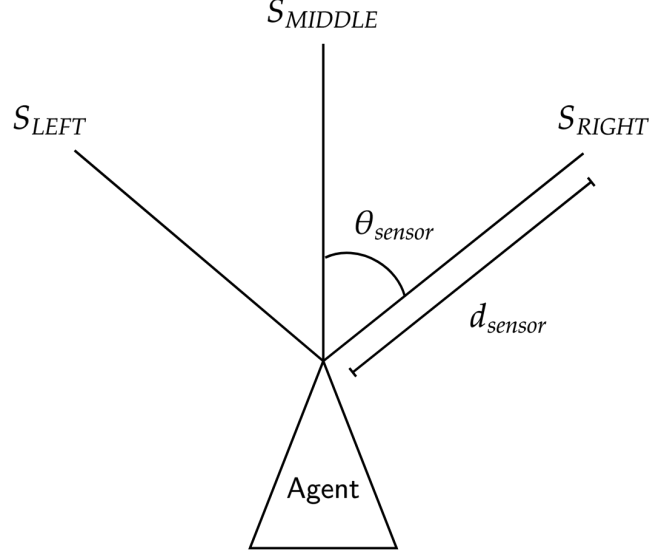
- la position de tous les agents ;
- la position de chaque cible ;
- la position de chaque danger ;
- l’intensité  $s_{target}$  et  $s_{danger}$  sur chaque cellule (un patch dans NetLogo).

En s’inspirant du modèle d’agent de Jones (2010) [6], nous avons défini la perception d’un agent comme étant composée de l’intensité de chaque signal mesurée par le biais de 3 capteurs (*LEFT*, *MIDDLE*, *RIGHT*) placés devant eux ( $\Omega$ ) (cf. figure 2.10). Afin de déterminer la direction à suivre les agents comparent les valeurs  $S_k$ ,  $k \in \{LEFT, MIDDLE, RIGHT\}$  de chaque capteur. Chacune de ces valeurs est définie comme la somme pondérée de chaque signal. Les poids étant les

valeurs encodées par les mots décrivant ces stimuli. Ainsi, soit  $s_x^k$  l'intensité du signal  $x$  au niveau du capteur  $k$ , avec  $k \in \{LEFT, MIDDLE, RIGHT\}$  et  $x \in \{target, danger\}$  :

$$S_k = \sum_{x \in C} m_x \times s_x^k, \forall k \in \{LEFT, MIDDLE, RIGHT\} \quad (2.3)$$

$$m_x = \phi(D(I(x))) \quad (2.4)$$



**FIGURE 2.10.** – Illustration du positionnement des capteurs de l'agent.

Les mots du vocabulaire sont ici des nombres binaires de longueur 10 encodant une valeur (un paramètre) dans l'intervalle  $[-1; 1]$ . Un agent utilise toujours le dernier mot qu'il a associé à un concept.

Ainsi, une fois la valeur de chaque capteur calculée, l'agent agit suivant l'algorithme 4.

---

**Algorithme 4 :** Algorithme de décision.

---

**Données :**  $S_{LEFT}$ ,  $S_{MIDDLE}$ ,  $S_{RIGHT}$

- 1 **si**  $S_{MIDDLE} > S_{LEFT}$  **et**  $S_{MIDDLE} > S_{RIGHT}$  **alors**
  - 2   └ Continuer dans la même direction
  - 3 **sinon si**  $S_{MIDDLE} < S_{LEFT}$  **et**  $S_{MIDDLE} < S_{RIGHT}$  **alors**
  - 4   └ Tourner de  $\theta_{turn}$  degrés aléatoirement vers la droite ou la gauche
  - 5 **sinon si**  $S_{LEFT} < S_{RIGHT}$  **alors**
  - 6   └ Tourner vers la droite de  $\theta_{turn}$  degrés
  - 7 **sinon**
  - 8   └ Tourner vers la gauche de  $\theta_{turn}$  degrés
- 

Les agents déclenchent des parties de NG en tant que speaker à une fréquence  $f_{games}$ . Un speaker ne peut jouer qu'avec un individu situé dans un rayon de communication  $r_{comm}$ . Parmi les agents,  $N_{success} = N \times f_{success}$  ont un biais de succès, et,  $N_{novelty} = N \times f_{novelty}$  ont un biais de nouveauté, le reste des agents n'ont pas de biais.

À chaque pas de temps, si un agent se trouve à une distance inférieure à  $d_{percept}$  d'une cible, il reçoit une récompense de 1 et la cible disparaît, une autre cible apparaissant aléatoirement dans l'environnement. La performance  $p_i^t$  d'un agent  $i$  au temps  $t$  est la somme des récompenses reçues depuis le départ.

### 2.2.2. Méthode expérimentale

Chaque ensemble de paramètres évalué est simulé 25 fois sur 1000 itérations. Est analysée ensuite l'orbite moyenne du système sur ces 25 simulations. Nous avons compté entre 4 et 6 heures pour réaliser chaque expérience (composées de 25 simulations). Les paramètres utilisés pour mener les expériences qui suivent sont résumés dans le tableau 2.11. La capture d'écran en figure 2.12 présente l'interface de notre modèle NetLogo.

Symbole	Valeurs testées	Signification
$n_{agents}$	[20, 100]	Nombre d'agents
$n_{targets}$	20	Nombre de cibles
$n_{dangers}$	{0, 10}	Nombre de dangers
$r_{comm}$	25	Rayon de communication
$f_{games}$	0.4	Fréquence de déclenchement de partie
$evap$	0.06	Vitesse d'évaporation
$diff$	0.98	Taux de diffusion
$f_{success}$	0.5	Fréquence du biais de succès
$f_{novelty}$	0.4	Fréquence du biais de nouveauté
$d_{percept}$	5	Distance d'un capteur par rapport à un agent
$d_{sensor}$	1	Distance à laquelle un agent détecte une cible
$\theta_{sensor}$	45	Angle entre les capteurs latéraux et l'orientation de l'agent
$\theta_{turn}$	10	Angle de rotation de l'agent
$l$	200	Taille de l'environnement
$\tau$	$\infty$	Échelle temporelle sur laquelle est définie la performance
$P_{mut}$	0.1	Probabilité de mutation

TABLE 2.11. – Tableau résumant les paramètres utilisés lors des expérimentations.

### 2.2.3. Présentation des résultats

Deux principales expériences ont été menées : avec introduction ou non de dangers au sein de l'environnement, d'une part sans dangers afin de montrer la validité du modèle dans un cas simple où les seuls éléments émettant un signal sont les cibles, et d'autre part avec dangers (émettant aussi un signal) dans le but de constater la capacité du modèle à apprendre une corrélation entre une récompense et un comportement lorsque celle-ci est indirecte —en effet, les dangers dans le cadre présenté n'affectent pas directement la performance par une récompense négative mais en ralentissant les agents.

La première expérience ne fait donc pas apparaître de dangers dans l'environnement, seulement des cibles à atteindre. La figure 2.13 montre une différence significative dans le nombre de cibles trouvées par les agents lorsque leur comportement s'adapte grâce à l'évolution culturelle.

La capacité des agents à pouvoir être davantage attirés par les cibles grâce à l'évolution culturelle semble se confirmer par l'analyse de la trajectoire du paramètre d'attraction moyen. En effet, il est possible d'observer une médiane se stabilisant plus ou moins à partir de 200 itérations dans des

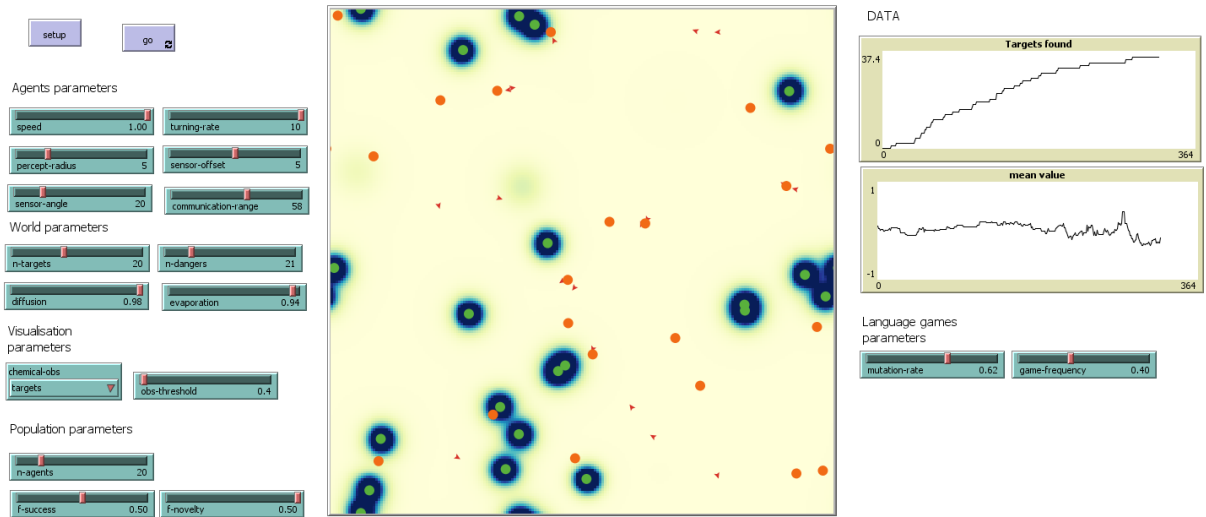


FIGURE 2.12. – Fenêtre de simulation NetLogo.

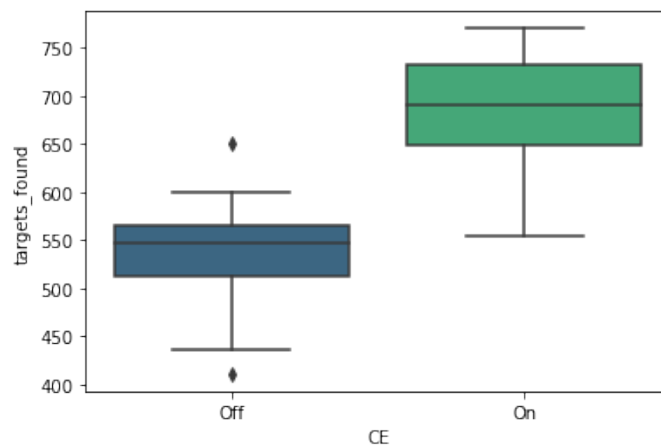


FIGURE 2.13. – Comparaison du nombre total de cibles trouvées sans et avec évolution culturelle (CE).  
 $pvalue = 1.98e - 11$  (test t). Avec  $n_{targets} = 20$  et  $n_{dangers} = 0$ .

valeurs positives, exhibant une certaine capacité d'apprentissage des agents à être attirés par les cibles, comme montré par la figure 2.14.

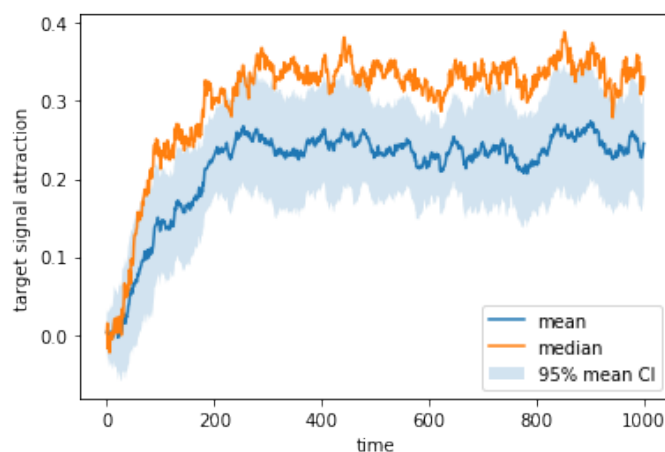
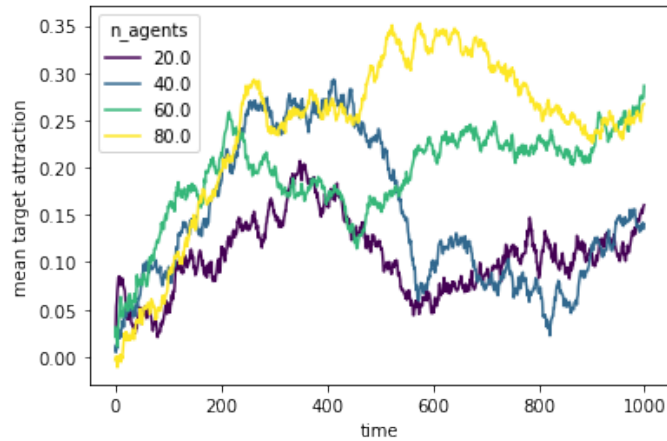


FIGURE 2.14. – Évolution du paramètre d'attraction au signal émis par les cibles ( $m_{target}$ ). Trajectoire moyenne sur 25 simulations. (CI = Confidence Interval.)

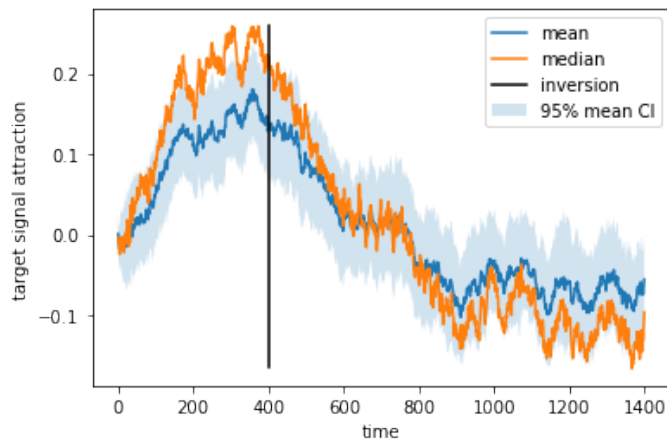
Nous avons ensuite analysé le comportement du système pour différentes tailles de population. Nous ne pouvons pas ici nous fier au nombre de cibles trouvées du fait que l'environnement n'a pas été agrandi proportionnellement au nombre d'agents amenant ainsi des effets de bord rendant

ces données peu exploitables. Cependant, nous pouvons tout de même regarder l'évolution des paramètres. Bien qu'il ne soit pas possible de l'affirmer de manière certaine, les trajectoires produites semblent indiquer que sous une certaine taille de population le comportement du système devient très instable (cf. figure 2.15).



**FIGURE 2.15.** — Évolution de l'attraction moyenne au signal émis par les cibles pour différentes tailles de population.

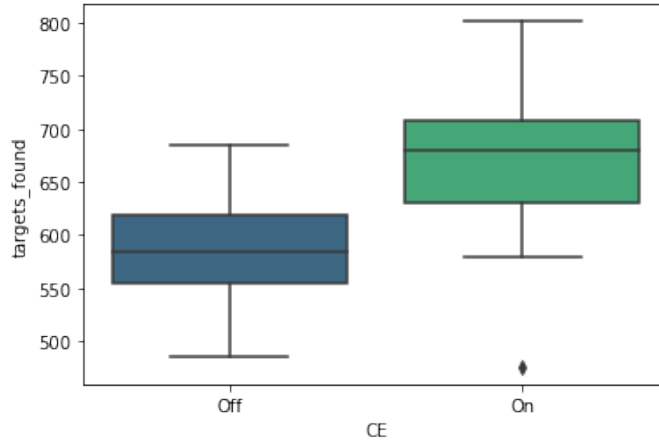
Enfin, nous avons analysé le comportement du système face à une fonction de récompense non stationnaire (i.e. qui peut changer), ceci dans le but d'évaluer la capacité du groupe à se réadapter. Pour ce faire, nous avons laissé le système évoluer pendant 400 pas de temps, puis inversé la fonction de récompense : trouver une cible rapportant dorénavant une récompense de -1 au lieu de 1. Comme indiqué en figure 2.16, le groupe se réadapte rapidement : le paramètre d'attraction aux cibles passant d'une valeur moyenne positive à une valeur négative en 400 pas de temps environ.



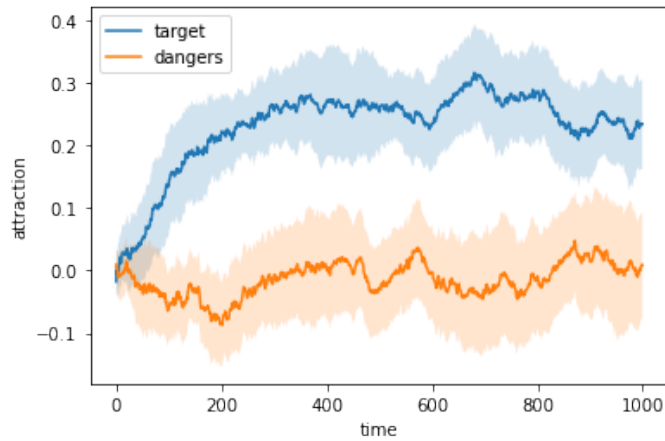
**FIGURE 2.16.** — Évolution du paramètre d'attraction au signal émis par les cibles lorsque la fonction de récompense est inversée à  $t = 400$ .

Dans un second temps, nous avons introduit dans l'environnement des dangers —au nombre de 10—, dangers qui comme expliqué auparavant ralentissent les agents qui entrent dans son rayon d'action. Tout comme dans la première expérience, la performance du groupe est significativement meilleure lorsque l'évolution culturelle est activée (cf. figure 2.17).

Lorsque que nous nous intéressons aux paramètres d'attraction (cf. figure 2.18), nous pouvons observer que comme dans la première expérience les agents deviennent attirés par le signal cible. En revanche, ils ne sont pas particulièrement repoussés par le signal danger, la valeur moyenne oscillant aux alentours de 0 suggérant que les agents ignorent ce signal dans le meilleur des cas.



**FIGURE 2.17.** – Comparaison du nombre total de cibles trouvées sans et avec évolution culturelle (CE).  $pvalue = 1.92e - 05$  (test t). Avec  $n_{targets} = 20$  et  $n_{dangers} = 10$ .



**FIGURE 2.18.** – Évolution des paramètres d'attraction moyens pour chacun des signaux.

#### 2.2.4. Interprétation des résultats

Ces expériences viennent confirmer la capacité du modèle à exhiber de l'adaptation. En effet, les agents apprennent assez rapidement (i.e. environ 200 pas de temps) à être attirés par le signal émis par les cibles. En revanche, il ne semble pas réellement amener les agents à être repoussés par le signal émis par les dangers. La corrélation entre dangers et baisse de performance étant indirecte, il aurait été intéressant d'analyser l'évolution des paramètres sur des temps plus longs.

Bien que l'effet de la taille de la population ne soit pas formellement identifiable dans ces expériences, les résultats suggèrent que des tailles de population trop faibles peinent à amener des résultats stables. Cette observation que nous avons déjà suggérée lors de l'analyse du modèle rejoint les résultats obtenus par Cambier *et al.* [3], qui n'ont pas observé de réel gain de performance pour des populations inférieures à 100 individus.

# Conclusion et ouvertures

À travers le travail présenté précédemment, nous avons pu montrer qu’une approche culturelle basée sur l’échange d’un vocabulaire en rapport aux stimuli perçus permettait de doter un groupe d’agents de la capacité d’adapter leurs comportements en vue de maximiser leur performance dans une tâche de recherche de cibles au sein d’un environnement inconnu et en toute autonomie. Pour cela, nous avons défini un cadre formel dans lequel nous avons proposé un modèle reposant sur l’introduction de biais dans les Language Games. Dans l’évaluation du modèle nous avons montré que ce dernier était robuste face à l’aléa, capable de passer à l’échelle, de s’adapter et de se réadapter et également d’évoluer continuellement sous certaines conditions.

Nous avons montré que cette approche inspirée par l’évolution culturelle est viable et prometteuse. Le cadre formel dans lequel cette approche s’inscrit a par ailleurs été pensé de manière à être peu restrictif. Le modèle qui en est issu est solide et exploitable de par ses propriétés fondamentales pour la robotique en essaim (montrées en section 2). De plus, cette approche et par extension notre modèle ne requièrent aucune faculté de raisonnement et d’apprentissage complexes de la part des agents, ce qui est aussi un pilier de la robotique en essaim.

De nombreux éléments peuvent encore être étudiés dans ce cadre. Parmi ceux-ci, le critère de performance, défini dans notre travail comme étant la somme des récompenses reçues par les agents irrespectivement du temps qui passe et donc des changements qui peuvent survenir. Par conséquent, étudier différentes définitions de critères de performance ou bien modifier l’échelle temporelle sur laquelle ce critère est défini ( $\tau$ ) pourrait permettre l’obtention d’une évolution qui se veut plus flexible. Cette flexibilité viendrait d’une part d’un critère de performance se définissant sur une fenêtre de temps plus courte, permettant d’éviter de tenir compte de ce qui n’est plus exactement valable dans l’état du monde actuel et, dans un même temps, d’un critère de performance se définissant sur une fenêtre de temps suffisamment ample pour éviter les fluctuations statistiques lorsque la récompense suit un processus aléatoire ; il s’agirait donc de trouver un compromis entre ces deux aspects.

Une autre problématique apparaissant dans le modèle s’intitule le problème de l’assignement de crédit (*credit assignment problem*). Cette problématique concerne la difficulté d’un agent à identifier quel comportement fait qu’un autre agent est performant et donc dans quels aspects doit-il l’imiter. Dans notre travail les agents se contentent d’imiter globalement les comportements des agents performants, mais il serait intéressant de rechercher une méthode qui leur permettent d’isoler approximativement ce qui fait qu’un agent est performant à un moment donné.

L’approche proposée ici amène de nombreuses perspectives d’évolution dont nous présentons quelques exemples. Une évolution particulièrement importante serait l’ajout d’une forme d’apprentissage individuel, permettant d’exploiter davantage le potentiel de l’évolution culturelle. Cet ajout rendrait à l’évolution culturelle sa nature Lamarckienne (i.e. les expériences d’un agent modifient ce qu’il transmet) qui est théoriquement plus rapide qu’une évolution reposant exclusivement sur la sélection. Dans cette optique, l’utilisation des POMDP dans notre cadre formel permettrait de faciliter la liaison avec des méthodes issues de l’apprentissage par renforcement. Potentiellement, les POMDP pourraient aussi permettre de faire le lien avec d’autres champs de recherche et donner naissance à de nouvelles méthodes.

Un autre axe de travail porterait sur l’amélioration de l’autonomie des agents qui, actuellement, est limitée par la définition a priori de concepts inhérents à l’environnement, en l’occurrence dans la tâche mise en œuvre les concepts de “signal-cible” et de “signal-danger”. Dans l’optique de groupes autonomes, l’objectif serait de leur donner le moins de connaissances a priori. Une méthode telle que le Category Game [11] leur permettrait de développer leurs propres concepts (dites catégories),



cependant cette méthode présente actuellement de nombreuses limites que nous avons tenté de lever dans un travail préliminaire, consultable en annexe A. Le Category Game tel qu'abordé dans cette annexe reste conceptuellement particulièrement approprié dans ce but.

Une autre piste, qui a été explorée en travail préliminaire et disponible en annexe B, porterait sur la simulation des communications réseaux au sein d'un groupe de drones, amenant à envisager une approche orientée co-simulation en vue de permettre la simulation parallèle et décentralisée d'un ensemble de modèles qui constituent l'étude d'un seul et unique système selon différentes perspectives.

À un niveau plus conceptuel, les Language Games sont notre porte d'entrée à l'émergence d'une langue dans un groupe d'agents ; nous nous sommes attachés à les faire discriminer des concepts de l'environnement de même nature, mais il serait possible de décrire des concepts d'autres natures : par exemple —sans être exhaustif— des relations entre des objets, des comportements, ou encore des relations spatiales. Avec ces concepts de natures différentes, il deviendrait possible de laisser les agents composer des énoncés transmettant des informations de plus en plus complexes.

Le présent travail est un premier pas dans la direction du développement de systèmes d'essaims de robots plus autonomes de par l'adaptation de leurs comportements et ce grâce à des mécanismes inspirés de l'évolution culturelle. Les pistes suggérées précédemment dévoilent des chemins passionnants pouvant mener à cet idéal.

# Bibliographie

- [1] B. AL NAHAS, S. DUQUENNOY et O. LANDSIEDEL, « Network-Wide Consensus Utilizing the Capture Effect in Low-Power Wireless Networks, » in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, sér. SenSys '17, Delft, Netherlands : Association for Computing Machinery, 2017, ISBN : 9781450354592. DOI : 10.1145/3131672.3131685.
- [2] E. BONABEAU, M. DORIGO et G. THERAULAZ, « Swarm Intelligence : From Natural to Artificial Systems / E. Bonabeau, M. Dorigo, G. Theraulaz., » jan. 2001.
- [3] N. CAMBIER, V. FRÉMONT, V. TRIANNI et E. FERRANTE, « Embodied Evolution of Self-Organised Aggregation by Cultural Propagation, » in *11th International Conference on Swarm Intelligence (ANTS 2018)*, Rome, Italy, oct. 2018, p. 351-359.
- [4] N. CAMBIER, R. MILETITCH, V. FRÉMONT, M. DORIGO, E. FERRANTE et V. TRIANNI, « Language Evolution in Swarm Robotics : A Perspective, » *Frontiers in Robotics and AI*, t. 7, p. 12, 2020, Publisher : Frontiers Media S.A. DOI : 10.3389/frobt.2020.00012.
- [5] J. HENRICH et R. MCELREATH, « The evolution of cultural evolution, » *Evolutionary Anthropology : Issues, News, and Reviews*, t. 12, n° 3, p. 123-135, 2003. DOI : <https://doi.org/10.1002/evan.10110>.
- [6] J. JONES, « Characteristics of pattern formation and evolution in approximations of physarum transport networks, » en, *Artificial Life*, t. 16, n° 2, mar. 2010, Publisher : Massachusetts Institute of Technology Press (MIT Press), ISSN : 1064-5462, 1530-9185. DOI : 10.1162/artl.2010.16.2.16202.
- [7] O. LANDSIEDEL, F. FERRARI et M. ZIMMERLING, « Chaos : Versatile and Efficient All-to-All Data Sharing and in-Network Processing at Scale, » in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, sér. SenSys '13, Roma, Italy : Association for Computing Machinery, 2013, ISBN : 9781450320276. DOI : 10.1145/2517351.2517358.
- [8] R. MILETITCH, A. REINA, M. DORIGO et V. TRIANNI, « Emergent naming of resources in a foraging robot swarm, » *arXiv :1910.02274 [cs]*, oct. 2019, arXiv : 1910.02274.
- [9] V. POIROT, B. A. NAHAS et O. LANDSIEDEL, « Paxos made wireless : consensus in the air, » p. 13,
- [10] Y. PRUDENT et A. ENNAJI, « An incremental growing neural gas learns topologies, » jan. 2005, 1211-1216 vol. 2, ISBN : 978-0-7803-9048-5. DOI : 10.1109/IJCNN.2005.1556026.
- [11] A. PUGLISI, A. BARONCHELLI et V. LORETO, « Cultural route to the emergence of linguistic categories, » *Proceedings of the National Academy of Sciences of the United States of America*, t. 105, p. 7936-40, 10 juin 2008. DOI : 10.1073/pnas.0802485105.
- [12] L. STEELS, « A Self-Organizing Spatial Vocabulary, » *Artificial Life*, t. 2, n° 3, p. 319-332, 1<sup>er</sup> avr. 1995, ISSN : 1064-5462. DOI : 10.1162/artl.1995.2.3.319.
- [13] L. WITTGENSTEIN, *Philosophical investigations*. Oxford : Basil Blackwell, 1953, 250 p., ISBN : 978-0-631-11900-5.

# Annexes

# A. Exploration des possibilités offertes par le Category Game et tentatives de réponses à ses limites

Le *Category Game* (jeu des catégories) [11] a pour objectif de permettre à des agents de catégoriser leur environnement ; contrairement au Naming Game de Steels (le Language Game original) [12], les agents n'ont pas d'office des concepts de fournis, et c'est à eux de déterminer ce qui différencie des parties de leur environnement les unes des autres. Le Category Game prend ses racines en 2008 mais n'a pas été particulièrement exploré depuis, et c'est l'objectif de cette section de déterminer comment il est possible de pousser son principe de manière à être utile à une application avec des drones comme agents, et si finalement ce Language Game correspond à nos besoins.

## Description du Category Game originel

Le Category Game est développé par Puglisi *et al.* en 2008 [11]. Son objectif est de permettre aux agents de développer un vocabulaire commun permettant de catégoriser un ensemble de stimuli en un ensemble de termes. Pour ce faire chaque agent va diviser son espace perceptif (espace des stimuli) en catégories perceptives auxquelles sera associé un vocabulaire. Le vocabulaire de chaque agent va évoluer dynamiquement par itération du Category Game.

Les travaux de Puglisi utilisent un paramètre *JND* (*Just Noticeable Difference*, soit en français un seuil de discrimination) qui est la distance en dessous de laquelle deux stimuli ne peuvent être différenciés (les stimuli  $s$  d'une scène  $S$  sont tous à une distance supérieure au *JND* les uns des autres). Cette valeur permet de borner le nombre de catégories perceptives nécessaires pour discriminer tous les stimuli, par exemple pour des stimuli dans l'intervalle  $[0; 1]$ ,  $1/JND$  catégories perceptives sont nécessaires.

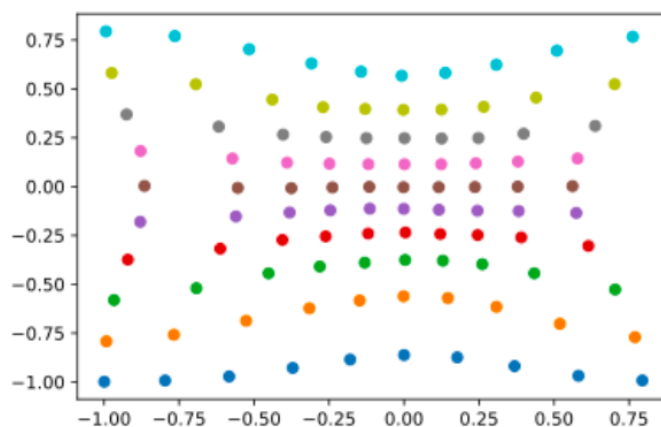
L'implémentation de cette approche ne s'est pas avérée aussi efficace que l'implémentation d'origine bien que tous les résultats soient qualitativement similaires à ceux obtenus par Puglisi *et al.* : d'une part, la convergence est extrêmement lente, il faut environ  $10^6$  partie par agent pour avoir des résultats convenables (ce qui sur la machine qui exécutait les essais représentait 3 heures de simulation) avec  $JND = 0.1$ , d'autre part l'implémentation présente une faiblesse au niveau de l'adaptabilité puisque les catégories perceptuelles sont totalement dépendantes des premiers stimuli utilisés et ne peuvent changer (ce qui est problématique si un changement important altère totalement l'environnement). Par ailleurs, il est notable de relever que dans cette approche les agents ont toute la même perception de l'environnement et il est aussi considéré que les agents impliqués dans une partie de Category Game savent exactement quels sont les stimuli empruntés dans la scène (sans possibilité de se tromper).

Andrea Puglisi fut contacté afin d'en savoir plus sur l'algorithme voire l'implémentation d'origine, dans l'idée de pouvoir obtenir des détails qui n'étaient pas disponibles dans l'article de 2008 et qui permettraient d'améliorer cette nouvelle implémentation. Cependant celui-ci n'avait plus à disposition le code originel (ou même les documents de l'époque), et la recherche a dû s'arrêter ici concernant la ré-implémentation la plus fidèle possible du Category Game de Puglisi *et al.* (2008).

## Introduction des cartes de Kohonen pour discrétiser des espaces perceptifs de dimension quelconque

Nous avons choisi d'utiliser les cartes de Kohonen, ou cartes auto-organisées (dites SOM de l'anglais *Self-Organized Map*) afin de représenter les catégories perceptives d'un agent dans le cadre du Category Game. Nous avons choisi cette méthode car elle permet de discrétiser un espace d'entrée de dimension quelconque en unités discrètes appelées prototypes ou neurones.

Une SOM, après présentations successives de vecteurs de l'espace d'entrée suivant une distribution donnée, s'auto-organise afin que les neurones de la carte représentent le plus fidèlement possible la distribution d'entrée. Par exemple, nous pouvons voir en figure A.1 les neurones d'une SOM entraînée avec des vecteurs en dimension 2 suivant une loi normale centrée réduite. Cet exemple permet de voir que la densité des neurones correspond relativement bien à une loi normale centrée réduite en dimension 2.



**FIGURE A.1.** – Exemple d'une SOM entraînée sur présentation de  $n = 10000$  stimuli suivant une loi normale centrée réduite en dimension 2 où chaque point représente un neurone et la position de ce neurone son prototype (le stimuli pour lequel il répond le plus fortement).

Nous avons donc expérimenté cette approche pour le Category Game, chaque neurone de la carte se voit attribuer un vocabulaire que les parties de Category Game vont faire évoluer. Nous avons essayé l'approche dans un premier temps avec des stimuli en dimension 1, puis en dimension 2.

Les premiers résultats montrent l'émergence de catégories linguistiques satisfaisantes. Cependant, cette approche présente toujours le problème de la contamination, à savoir, le décalage inter-individuel des catégories perceptives entraîne la diffusion des catégories linguistiques de proche en proche jusqu'à ce qu'un unique mot décrive tous les stimuli possibles.

Afin d'améliorer la stabilité et la vitesse de convergence, nous avons développé une méthode permettant de renforcer les vocabulaires déjà développés. Pour ce faire, le speaker d'une partie, au lieu de définir une scène aléatoirement, définit la scène sur la base de ses propres prototypes auxquels un mot est attribué. Cette méthode permet des convergences de vocabulaire bien plus franches mais locales, cela est dû au fait que les agents ne traitent par conséquent pas tous les stimuli possibles. Nous avons donc introduit un paramètre  $\epsilon$  qui définit la probabilité que le speaker choisisse une scène aléatoire plutôt qu'exploite ses connaissances. Cette méthode, qui trouve un compromis entre exploration et exploitation, a résulté en des améliorations notables dans la convergence du vocabulaire.

Enfin nous avons utilisé des SOM incrémentales, et plus précisément l'algorithme *Incremental Growing of Neural Gas* (IGNG) [10], l'avantage de telles méthodes est que les agents gagnent en adaptabilité puisque moins de paramètres doivent être fixés par l'homme et qu'elles nécessitent moins de données d'entraînement. Cette approche a donné des résultats aussi bons voir meilleurs qu'avec des SOM classiques.

## Spécialisation des agents linguistiques pour accélérer et stabiliser la convergence du vocabulaire

Dans un second temps, nous avons étudié la possibilité que le vocabulaire (les différents mots) puisse s'auto-organiser de lui-même au sein des cartes auto-organisées en créant des liens avec les neurones de la carte.

Pour ce faire, nous nous sommes inspirés de la spécialisation chez les insectes sociaux. Ce phénomène de spécialisation repose sur un mécanisme très simple : plus un individu effectue une tâche (avec succès) plus il sera enclin à refaire cette même tâche.

Bonabeau *et al.* [2] ont proposé un modèle pour ce phénomène qui repose sur les réponses seuillées. Disons que chaque tâche est liée à un stimulus particulier (par exemple faire le ménage quand c'est en désordre), alors la probabilité qu'un individu s'engage dans la tâche suit une fonction sigmoïde centrée sur un seuil. Alors, si un individu réalise sa tâche avec succès, il va baisser ce seuil, ainsi, il s'engagera dans cette même tâche pour une intensité de stimulus associé plus faible.

Plus formellement ce modèle peut être décrit comme suit.

Soit  $\theta(i, j)$  le seuil de réponse d'un individu  $i$  à un stimulus associé à la tâche  $j$ . Alors, à chaque fois que  $i$  effectue la tâche  $j$  :

$$\begin{cases} \theta(i, j) \leftarrow \theta(i, j) - \Delta_0 \\ \theta(i, k) \leftarrow \theta(i, k) + \Delta_1 \forall k \notin \{j\} \cup \{n(j)\} \end{cases} \quad (\text{A.1})$$

Après un certain temps, chaque individu de la population se spécialise pour certaines tâches.

Notre idée ici est que les individus soient les mots du langage, les tâches soient de se déclencher lorsque un prototype de la SOM est activé, et le stimulus associé à ces tâches est l'activation du neurone de la SOM. Afin que cette organisation soit alignée avec le reste de la population d'agents, nous avons intégré un signal de renforcement qui est lié au résultat des parties de Language Games.

$$\begin{cases} \theta(i, j) \leftarrow \theta(i, j) - \Delta_0 * R \\ \theta(i, j') \leftarrow \theta(i, j') - \Delta_1 * R \forall j' \in n(j) \text{ le voisinage de } j \\ \theta(i, k) \leftarrow \theta(i, k) + \Delta_2 \forall k \notin \{j\} \cup \{n(j)\} \\ R < 0 \text{ si la partie est un échec, } R > 0 \text{ sinon} \end{cases} \quad (\text{A.2})$$

Ce modèle montre une stabilité accrue sans doute due à la compétition entre les unités linguistiques. Cependant une forte synonymie apparaît, qui en soit n'affecte pas la capacité des agents à se comprendre.

Nous avons ensuite proposé une méthode pour moduler la probabilité qu'un agent déclenche une partie en tant que speaker. L'idée ici est que nous voulons favoriser la transmission des connaissances en créant des vagues d'activité qui pourraient propager le vocabulaire, vagues d'activité que l'on peut observer dans de nombreux systèmes auto-organisés comme chez les fourmis ou les populations neuronales.

La méthode est assez simple, soit  $s$  le speaker,  $h$  le hearer et  $P_i$  la probabilité de déclenchement de l'individu  $i$  alors :

$$\begin{cases} P_h = P_{max} \text{ et } P_s = P_s + \alpha(P_0 - P_s) & \text{si la partie est un succès} \\ P_h = P_h + \alpha(P_0 - P_h) \text{ et } P_s = P_s + \alpha(P_0 - P_s) & \text{sinon} \end{cases} \quad (\text{A.3})$$

Avec  $P_0$  la probabilité dite de repos.

Cette méthode a montré une amélioration significative des performances du Category Game.

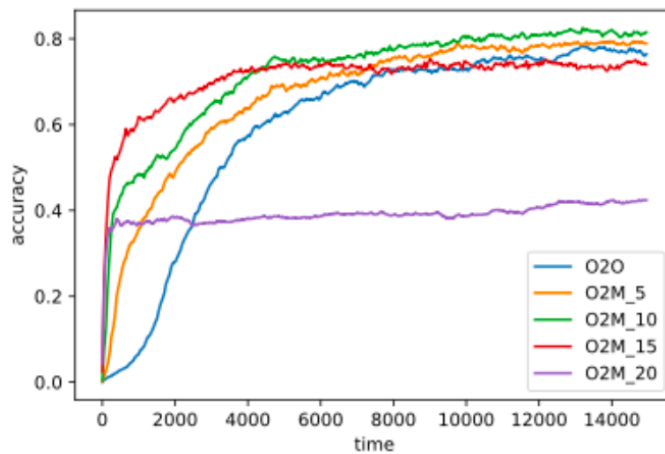
## Parties de Category Game d’un-à-un à un-à-plusieurs

Nous nous intéressons ici à l’étude d’une variation du jeu de langage. Le principe est ici de proposer des jeux qui ne sont non plus en dialogue (un hearer et un speaker), mais en groupe (un speaker mais plusieurs hearers), et ce afin de rapprocher le Language Game de l’approche réseau où la communication entre plus de deux agents est plus pertinente que de un à un.

Nous émettons l’hypothèse qu’un tel système pourrait montrer une vitesse de convergence du vocabulaire plus élevé. Mais pourrait également être sujet à des convergences locales trop rapides au sein de la population ce qui pourrait mener à des problèmes.

Nous proposons ici de comparer le mode de communication en dialogue classique que nous noterons  $O2O$  (*One to One*), à un processus de jeu en groupe que nous noterons  $O2M_k$  (*One to Many*), avec  $k$  un entier strictement positif définissant le nombre maximal de “hearers” auxquels s’adresse un speaker.

Le mécanisme de déclenchement de partie reste similaire (probabilité de déclencher une partie adaptative). Lorsque un agent décide de déclencher une partie, il propose à ses  $k$  voisins les plus proches qui acceptent s’ils ne sont pas déjà en train de jouer une partie. Ensuite les  $k$  parties se jouent en parallèle de manière classique.



**FIGURE A.2.** – Comparaison des performances moyennes sur 10 essais entre les différentes stratégies sur  $T = 15000$  itérations. Notons que durant ces 15000 itérations,  $O2O$  déclenche en moyenne 3 fois moins de parties.

Les résultats en figure A.2 montrent une convergence plus rapide pour  $O2M$  que  $O2O$ , et plus rapide quand  $k$  est grand excepté pour  $k = 20$  qui montre un comportement de butée. Notre hypothèse est que  $k$  module le compromis vitesse-justesse, alors quand  $k$  est trop grand, les vocabulaires convergent localement trop rapidement ce qui mène à des conflits entre des sous-populations d’agents. Il est également important de noter que beaucoup plus de parties sont jouées en  $O2M$  à chaque pas de temps.

Nous pouvons voir en figure A.3 que pour un même nombre de parties jouées,  $O2O$  affiche de meilleures performances.

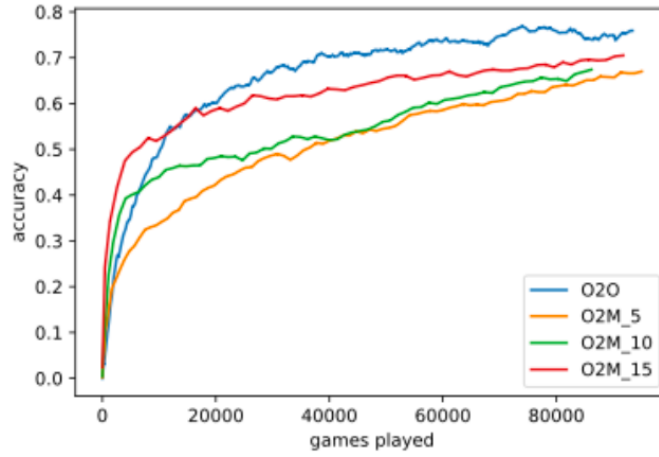


FIGURE A.3. – Comparaison des performances en fonction du nombre de parties jouées.

## Bilan de l’exploration du Category Game

Bien que le Category Game soit conceptuellement très intéressant dans l’optique de doter un essaim de drone d’une autonomie maximale, sa mise en place effective pose de nombreux problèmes techniques. Le principal problème se trouve dans l’alignement des catégories perceptives entre les agents qui, comme nous l’avons vu, amène le problème de la contamination. Les temps de convergence très longs représentent également un problème qui atteint actuellement l’intérêt pratique de cette approche. Nous avons ici essayé de proposer un ensemble de méthodes pour combler ces lacunes, méthodes qui ont montré des améliorations partielles. Cette approche prometteuse mériterait d’être plus approfondie.



## B. Exploration de la simulation réseau pour l'étude de la fiabilité des communications intrinsèques au modèle

L'exploration de la simulation réseau s'est révélée comme une piste intéressante à explorer dans le but d'étudier précisément l'évolution des communications entre drones dans le cas d'une instantiation physique de notre modèle. Le comportement évolutif du modèle développé ce semestre se fonde sur une interaction directe entre les agents, sous la forme d'un court dialogue centré autour de l'échange de mots. Les interactions sont donc nombreuses et un taux d'erreur important impliquerait un écart important entre le comportement théorique et le comportement observé du modèle. Cette problématique est inhérente à la frontière entre simulation et instantiation physique du fait de l'environnement complexe au sein duquel les drones peuvent être amenées à évoluer (i.e. obstacles), mais aussi par les communications elles-mêmes (e.g. collisions de paquets). C'est pourquoi la fiabilité des transmissions dans le but de garder une consistance dans l'échange des données a très vite retenu notre attention.

### L'imbrication de plusieurs mécanismes afin de fiabiliser les communications réseaux

Durant la partie bibliographique, une partie avait été dédiée à la présentation du protocole Wireless Paxos [9], qui se présente comme étant particulièrement adapté aux nombreuses problématiques soulevées précédemment dans le cas des réseaux sans fil, ici en l'occurrence un réseau constitué de drones aériens. L'avantage étant d'assurer le maintien des communications entre les drones d'un même essaim malgré les nombreuses perturbations physiques et matérielles. L'acquisition de données précises en rapport aux communications et dans divers scénarios aurait donc permis de franchir un pas supplémentaire vers le réalisme de notre modèle. Les propriétés de ce protocole sont :

- La garantie d'une prise de décision consensuelle entre tous les éléments du réseau, ici appelés nœuds (i.e. des drones dans le cas présent) ;
- L'obtention de faibles délais de transmission ;
- Une fiabilité accrue des communications malgré les diverses perturbations environnantes ;
- Le maintien d'une consistance des données contre toute défaillance ;
- Une meilleure gestion de la consommation énergétique des drones.

Pour ce faire, sont utilisés plusieurs mécanismes complémentaires les uns des autres, dont les principaux sont brièvement décrits au sein de cette section. Pour un détail plus précis des travaux sur lesquels reposent ses mécanismes, le lecteur peut se référer aux travaux de Landsiedel *et al.* (2013) [7], Al Nahas *et al.* (2017) [1], Poirot *et al.* (2019) [9]. L'usage de ces trois mécanismes offre la possibilité qu'un nœud puisse agir selon trois logiques :

- Proposer : initie une requête de vote au sein du groupe et devient leader jusqu'à acceptation ou refus de la requête engagée ;
- Acceptor : participe au vote ;
- Learner : apprend la valeur finale sans participer au consensus.

Ces rôles sont attribués automatiquement pour toute la durée d'un *round*. Un round représente le déroulement d'une opération, c'est-à-dire depuis son lancement jusqu'à son agrément final (acceptation ou refus). Une opération est décomposée en une quantité variable de *slots* qui correspondent aux intervalles fixes énoncés précédemment. Durant un slot, un noeud peut être dans un état parmi trois :

- Transmit : il transmet une information selon une politique définie (vote, dissémination agressive d'une décision finale, requête d'arrivée ou de départ au sein du groupe) ;
- Receive : il écoute les transmissions avoisinantes ;
- Sleep : sa radio est éteinte jusqu'à sa réactivation par une application, un service ou un autre protocole.

Lorsqu'un noeud doit transmettre une information, il le fait selon une politique inspirée des modèles épidémiques, c'est-à-dire uniquement si l'information qu'il reçoit de son voisinage est nouvelle ou plus ancienne, impliquant la notion d'état local afin de pouvoir comparer l'information locale avec celle nouvellement réceptionnée. Une chronologie de l'ensemble des opérations du protocole exécuté et basée sur un incrément de compteur permet de constituer une forme d'historique. Cet historique accorde alors aux nœuds la possibilité d'apprendre les décisions passées.

## Une base de code déjà existante mais inexploitable dans le cadre du projet

Bien que les auteurs de Wireless Paxos [9] proposent une base de code du protocole en question sous forme d'un dépôt GitHub, cette dernière se révèle inexploitable dans le cadre de notre projet. En effet, le code source présent - qui est majoritairement écrit en langage C - a été développé en vue d'être compilé pour Contiki<sup>1</sup>. Contiki est un système d'exploitation orienté pour les réseaux de capteurs sans fils, donc conçu pour les spécificités matérielles de ces capteurs, qui diffèrent de celles que l'on peut communément trouver dans un drone aérien, qu'il soit destiné au grand public ou à l'industrie. Cette compilation permettant alors l'obtention d'un binaire exécutable prêt à l'emploi. Par conséquent, cela soulève trois problématiques principales :

- La compilation nécessaire du code source vers un firmware (i.e. un microprogramme décrivant le fonctionnement interne du matériel en question) prêt à être chargé puis exécuté par le matériel cible ;
- L'impossibilité de ne pouvoir simuler le comportement réseau de ce firmware avec un simulateur autre que celui proposé par Contiki, en l'occurrence Cooja ;
- L'existence d'un nombre important d'instructions spécifique à Contiki au sein-même du code source, impliquant de grandes difficultés à pouvoir extraire correctement les instructions propres au protocole Wireless Paxos.

De plus, l'utilisation d'un paradigme procédural et destiné à de la programmation embarqué complexifie toute imbrication avec la simulation du modèle multi-agents que nous proposons. Le choix s'est alors porté vers une étude des différents travaux composant la solution Wireless Paxos en vue d'implémenter ses principaux mécanismes (i.e. ceux réellement nécessaires à l'étude des communications réseaux de notre modèle) directement au sein du simulateur réseau *Network Simulator 3* (NS, version 3). Ce simulateur possède l'avantage d'être proposé en tant que logiciel libre, permettant l'existence d'un support communautaire très large, mais aussi de permettre un développement avec le langage C++, proposant ainsi davantage de modularité du fait du paradigme objet.

---

1. <http://www.contiki-os.org/>

## Une démarche de génie logiciel nécessaire à l'implémentation effective du protocole Wireless Paxos

Du fait d'une trop grande complexité dans l'implémentation du protocole, la mise en œuvre de ce dernier n'a finalement pu aboutir avec succès. Il était en effet possible d'inférer et implémenter plusieurs mécanismes de la couche réseau, sans pour autant pouvoir réussir à implémenter l'ensemble de la logique. La majorité des cas d'applications qui existent sur les ressources communautaires réutilisent des modules existants, là où une implémentation des principaux mécanismes de Wireless Paxos nécessite la création d'un module à part entière et spécifique à ce protocole pour être fidèle à ce dernier, et donc hors de portée de nos compétences dans le contexte de notre projet. En effet, l'implémentation des modules les plus spécifiques au protocole nécessite une connaissance approfondie du simulateur en question en vue de commencer par les conceptualiser via une méthodologie de génie logiciel avant de pouvoir les implémenter. La conceptualisation efficace du protocole fait notamment référence à une conception pouvant servir d'architecture à une couche applicative extérieure au simulateur, ici représenté par l'ensemble du code constituant la simulation des Language Games et qui est écrit en Python. NS-3 permet de simuler ce type de couche applicative (couche la plus haute de la pile réseau), mais sans pour autant faciliter son imbrication au sein d'une architecture ne reposant pas sur du code déjà existant au sein des nombreux modules déjà proposés.

La complexité de cette réalisation étant fonction des nombreuses dépendances entre les éléments abstraits constituant le simulateur et un changement dans la logique de l'un de ces éléments influe sur l'ensemble de la chaîne reliant ces éléments. Toute compilation d'un binaire exécutable s'en retrouve impossible, augmentant la difficulté à pouvoir tester les ajouts au sein du code source. Il est à noter qu'un débogage basé sur l'exploration des pointeurs et des références a été longuement mené sans pour autant permettre d'avancer davantage dans la conceptualisation de l'architecture du protocole Wireless Paxos au sein du simulateur NS. Le choix a été l'arrêt de cette exploration pour ne se concentrer plus que sur le développement de l'approche culturelle. L'implémentation effective d'une telle simulation n'étant par conséquent non réalisable en si peu de temps et sans une grande étape d'étude méticuleuse du simulateur cible. Une démarche de développement piloté par les tests couramment appelé TDD (de l'anglais *Test Driven Development*) est une piste envisageable pour assurer le maintien de la compatibilité de toutes les fonctionnalités implémentées durant le processus de développement. Cette démarche consiste à utiliser des tests unitaires dès le début de ce développement, permettant de pouvoir tester des fonctionnalités sans pour autant nécessiter la compilation d'un binaire exécutable. Ces tests peuvent être ensuite intégrés au sein d'un pipeline d'intégration continue, permettant l'automatisation de la chaîne de développement afin d'assurer sa linéarité. Cependant, ces pistes nécessitent toujours une certaine connaissance du simulateur en question en vue d'être menées correctement.

## La simulation réseau en tant que perspective supplémentaire à l'étude et l'observation d'un système

Bien que la simulation réseau n'ait pu concrètement aboutir, d'autres outils de simulation ont été utilisés dans le cadre du projet. NetLogo, un simulateur multi-agents à pas de temps discret, a été utilisé pour représenter les modèles d'agents de perception / action des agents, tandis que Python a été utilisé pour représenter les modèles d'agents relatifs aux Language Games. L'avantage de cette combinaison étant de pouvoir profiter des avantages de NetLogo pour obtenir rapidement un comportement qualitatif, tout en exploitant la simplicité de Python pour implémenter les nombreuses interactions nécessaires à ces jeux de langages, sans pour autant devoir développer un environnement de simulation permettant l'observation de comportements qualitatifs. Par ailleurs, l'utilisation du simulateur de réseaux IP Network Simulator avait pour but d'étudier l'évolution de l'ensemble des communications inhérentes au modèle. En conséquence, l'utilisation de différents environnements de simulation en vue d'analyser voire d'observer les propriétés et comportements de notre modèle a

permis de se confronter à une problématique non soupçonnée. En effet, l'étude d'un même système selon différentes perspectives en simultanée permet de comprendre son évolution, tant au niveau spatial que temporel et à des niveaux de résolutions différents, du plus macroscopique au plus microscopique. Il en résulte une certaine hétérogénéité des différents simulateurs (i.e. outils) utilisés, chacun possédant son propre formalisme, ses avantages et ses inconvénients. L'étude d'un seul et même système peut donc se faire sur base de modèles continus ou discrets et selon des paradigmes différents, alors que ces outils ne sont initialement pas conçus pour être combinés ensemble. Bien qu'il fût possible d'exécuter du code Python depuis NetLogo, cela ne s'est pas fait sans devoir modifier une partie du code pour qu'il soit effectivement exécutable par NetLogo, mais aussi sans compter les nombreux effets de bords provoqués par ce pont entre les deux outils (i.e. Python et NetLogo). Par conséquent, il pourrait revêtir un certain intérêt à l'intégration d'une démarche de co-simulation en vue de pouvoir échanger correctement les données entre ces outils hétérogènes et ce, grâce à un formalisme suffisamment générique qui permettrait de les synchroniser sans avoir à nécessité de les modifier en profondeur. La plateforme de co-simulation MECSYCO<sup>2</sup> semble alors constituer un candidat intéressant à ce type d'approche, permettant la simulation parallèle et décentralisée d'un ensemble de modèles qui constituent l'étude d'un seul et unique système, en se basant notamment sur un formalisme de modélisation et simulation qui se veut particulièrement modulaire.

---

2. <http://www.mecsyco.com/fr/> .