Projet de recherche

La notion d'incertitude pour les systèmes aléatoires

Nathanaël Fijalkow

Organisation du document. Ce document est divisé en quatre parties. La première partie est une introduction. Les deuxième et troisième parties développent les deux lignes directrices du projet : la problématique de la robustesse et celle de la simulation en ligne. Enfin, la quatrième partie aborde la réalisation de ce programme de recherche : intégration dans les laboratoires et équipes, ainsi que collaborations nationales et internationales.

Introduction

Importance des systèmes aléatoires

L'aléa intervient en informatique dans deux contextes.

Le premier est lors de la construction d'algorithmes, où l'aléa est utilisé pour des raisons d'efficacité ; en effet, dans de nombreux cas les algorithmes randomisés sont plus efficaces que leurs homologues déterministes. Les algorithmes randomisés sont aujourd'hui très répandus en pratique et constituent un sujet de recherche actif [MR95].

Le second contexte est l'étude de systèmes dont l'évolution dépend de facteurs extérieurs ou incontrôlables, qui sont modélisés mathématiquement par un aléa. Ce cadre dépasse naturellement l'informatique fondamentale, et les systèmes aléatoires sont étudiés dans de nombreux domaines, que ce soit en linguistique, en biologie ou en économie.

Ces deux cadres témoignent de l'importance des systèmes aléatoires et de la nécessité de les modéliser, puis de développer des outils mathématiques et algorithmiques pour les analyser. L'algorithmique des systèmes aléatoires en informatique fondamentale a deux objectifs principaux :
- *Vérification* : étant donnés un modèle et une propriété, déterminer si le modèle satisfait la propriété.
- *Synthèse* : étant donnée une spécification, construire un modèle satisfaisant cette spécification.

Ces deux problèmes, décrits ici de manière générique, sont motivés par les méthodes formelles. L'objectif est de raisonner sur les modèles mathématiques, en décrivant leurs spécifications à l'aide de formalismes logiques. Cette approche, développée originellement par Clarke, Emerson et Sifakis, et récompensée d'un prix Turing en 2007, est aujourd'hui l'objet de nombreuses applications industrielles. L'utilisation de méthodes formelles pour l'étude des systèmes aléatoires n'est pas confinée à l'informatique fondamentale, elle est également au cœur de domaines connexes, dont la robotique et la théorie du contrôle.

Modélisation

Je décris plusieurs aspects apparaissant dans la modélisation de systèmes aléatoires.

Le premier aspect est *la dynamique d'évolution du système*, séparant systèmes continus et systèmes à temps discret : dans le premier cas, on observe l'évolution à tout moment, par exemple pour les systèmes temps réel, et dans le second, seulement à intervalles de temps fixés.

Le deuxième aspect est *le nombre d'agents* : plusieurs entités peuvent influencer l'évolution du système. Dans le cadre le plus simple, elle est entièrement aléatoire, il s'agit de chaînes de Markov. Selon les scénarios envisagés, on peut considérer un ou plusieurs contrôleurs, ainsi qu'un ou plusieurs agents extérieurs ; les modèles que l'on obtient sont appelés des jeux.

Le troisième aspect est *la notion d'observation* : chaque agent peut observer tout ou partie de l'évolution du système. Ainsi, s'il n'y a qu'un contrôleur, cette distinction implique deux modèles : d'un côté les processus de décision markoviens (Markov Decision Processes, MDP), où le contrôleur observe toute l'évolution du système, et de l'autre les automates probabilistes, où le contrôleur n'observe rien. Ces deux modèles sont unifiés par les processus de décisions markoviens à observation partielle (Partially Observable Markov Decision Processes, POMDP).

Les distinctions ci-dessus décrivent une très grande variété de modèles et de travaux. Je m'intéresse en particulier aux modèles *à observation partielle*.

De fait, les méthodes formelles pour les systèmes aléatoires *à observation parfaite* forment aujourd'hui un ensemble de techniques mûr et cohérent. De nombreux problèmes sur les chaînes de Markov [AAGT15] et les processus de décisions markoviens ont été analysés et compris, permettant leur insémination dans différents domaines. Par exemple, une grande partie des algorithmes construits ont été implantés dans l'outil PRISM [KNP07].
La situation est différente pour les systèmes aléatoires *à observation partielle*, par exemple pour les processus de décisions markoviens à observation partielle (POMDP). Même dans le cas particulier des automates probabilistes, les méthodes connues ne s'appliquent pas [GO10]. La notion d'observation partielle apparaît naturellement dans de nombreux domaines tels que l'apprentissage, l'intelligence artificielle et la planification de mouvements en robotique.

Le développement de méthodes formelles pour analyser les systèmes aléatoires à observation partielle est un enjeu important, qui est en plein essor depuis une dizaine d'années.

La notion d'incertitude

Mon projet de recherche suit deux lignes directrices liées à la notion d'*incertitude* :
- la *robustesse*, dont l'objectif est de capturer l'écart entre un système et son modèle, donc le degré d'incertitude que génère la modélisation,
- la *simulation en ligne*, dont l'objectif est de construire des algorithmes en ligne capables de reproduire le comportement d'un système à partir de son modèle. Ici l'incertitude est dans la façon dont l'algorithme reçoit son entrée, de manière séquentielle.

Vers la science des données

Je rejoins à partir de janvier 2017 l'Institut Alan Turing, dont l'objectif est de développer les différents aspects des sciences des données (Data Science), et de rassembler autour de ces problématiques des approches différentes, issues de l'informatique fondamentale, des statistiques, de la physique, et des sciences sociales.

La notion de *modèle* est centrale pour les sciences de données, mais elle a une interprétation différente de celle que l'on utilise en informatique fondamentale, et en particulier dans ce projet de recherche. Dans le cadre que j'étudie, le modèle est une abstraction mathématique représentant un système, par exemple un modèle physique, un ascenseur pour être concret. En science des données, un modèle est une abstraction mathématique représentant un ensemble de données. L'objectif de construire un modèle peut être d'analyser l'ensemble de données que l'on possède, mais également de faire des prédictions sur des données similaires.

Malgré cette différence de points de vue, de nombreuses problématiques se rejoignent. En particulier, celle de l'incertitude : dans les deux cas, la question de l'adéquation entre le modèle et ce qu'il représente est cruciale. Cette convergence de problématiques devrait permettre des échanges fructueux entre ces deux disciplines, et sera pour moi l'opportunité d'apprendre de nouvelles techniques et approches.
Plusieurs axes de recherche dans ce projet sont inspirées par des idées issues des sciences des données, et en particulier de l'apprentissage (Machine Learning), qui sont le fruit d'une collaboration commencée pendant mon post-doctorat à l'Université d'Oxford avec Borja de Balle, qui est chercheur en science des données à l'Université de Lancaster.

Robustesse

La notion de robustesse est au centre de la modélisation des systèmes aléatoires : elle quantifie l'écart entre système et modèle, en d'autres termes l'incertitude induite par la modélisation. Estimer la robustesse d'un modèle est une problématique importante dans de nombreuses applications, par exemple en chimie, en physique et en ingénierie.

J'étudie des modèles mathématiques de systèmes aléatoires. Pour plusieurs raisons, le comportement du système peut s'écarter de la loi de probabilité décrite par le modèle : soit le système en question est infiniment plus compliqué, soit il est sujet à des perturbations extérieures imprévisibles, soit encore il n'est pas connu de manière très précise. Un modèle pertinent se doit donc d'être robuste, c'est-à-dire décrire tous les comportements du système, y compris ses possibles variations.

Illustrons la notion de robustesse dans le cadre des algorithmes randomisés. On souhaite simuler le lancer d'une pièce équilibrée (ayant la même probabilité de tomber sur pile et sur face) à l'aide d'une pièce non équilibrée, disons qu'elle tombe sur pile avec probabilité x et face $1 - x$. Considérons l'algorithme proposé par von Neumann : on lance la pièce deux fois, puis

— si les deux lancers donnent le même résultat, on recommence,

— si les deux lancers donnent un résultat différent, retourner le résultat du premier lancer.

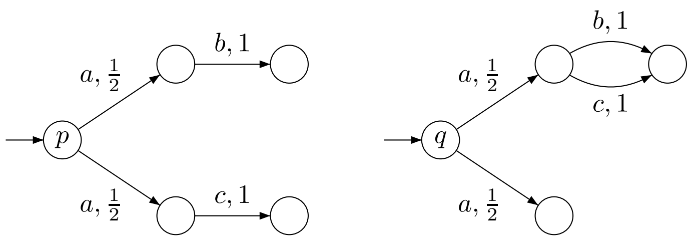
Il est facile de montrer que cet algorithme retourne pile et face avec la même probabilité. Mieux, cet algorithme est robuste, au sens suivant : même si x est modifié, l'algorithme demeure correct.

Axe 1 : algorithmes pour évaluer la distance entre modèles aléatoires

Étudier la robustesse d'un modèle peut être vu de manière plus générale ; il s'agit de définir une notion de distance entre modèles. Définir la proximité entre modèles est un enjeu important ayant de nombreuses applications. Par exemple, ceci ouvre la voie vers la minimisation de modèles : étant donné un modèle, peut-on construire un modèle proche, plus petit, et donc dont l'analyse est facilitée ?

Afin de parler de distance, il faut commencer par décrire les modèles équivalents, c'est-à-dire à distance zéro. C'est l'objectif de la notion de bisimilarité : deux systèmes sont bisimilaires s'ils ont exactement les mêmes comportements. Il existe plusieurs notions de bisimilarité entre systèmes aléatoires, dont la plus connue et étudiée est la bisimulation probabiliste de Larsen et Skou [LS89], dont j'ai étudié les propriétés logiques en collaboration avec Bartek Klin et Prakash Panangaden [FKP16].

Récemment, de nouvelles notions ont emergé, dont par exemple la notion de bisimilarité sur les distributions [HKK14], permettant une analyse plus fine que les approches précédentes.



La figure ci-dessus montre deux systèmes aléatoires bisimilaires pour les distributions mais pas au sens de Larsen et Skou. En effet, depuis p comme depuis q la probabilité de générer a est 1, et celle de ab ainsi que de ac sont $\frac{1}{2}$.

Dans le cadre de mon post-doctorat à Oxford, j'ai étudié avec Stefan Kiefer et Mahsa Shirmohammadi les propriétés algorithmiques de la notion de bisimilarité sur les distributions [FKS16], puis lors de mon séjour à Berkeley ses propriétés logiques, en collaboration avec Bartek Klin [FK16].

De nombreuses questions restent ouvertes, et en particulier celles de nature algorithmiques. Les algorithmes connus sont peu efficaces (exponentiels) et seulement pour des classes de systèmes restreintes. Des progrès récents démontrent des connections avec des techniques algébriques, ouvrant des perspectives algorithmiques intéressantes.

Objectif : construire un algorithme efficace (en temps polynomial) pour décider la relation de bisimilarité sur les distributions.

En s'appuyant sur de tels algorithmes, nous pouvons envisager de définir une notion de robustesse pour laquelle on peut vérifier si un modèle est robuste, et entreprendre le développement de méthodes formelles pour les modèles aléatoires robustes.

Axe 2 : introduire la notion de confiance

Ainsi que l'ont observé Ouaknine et Worrell [AAOW15], l'étude des chaînes de Markov est reliée au problème de Skolem. Ce problème étant ouvert depuis 80 ans, et l'objet de nombreuses conjectures en théorie des nombres algébriques, il n'est pas raisonnable de chercher à le résoudre de front. De manière similaire, de nombreux problèmes naturels pour les automates probabilistes sont indécidables.

Dans les deux cas, la faute est à la modélisation, qui génère des modèles irréalistes car trop précis. De fait, ces modèles ne prennent pas en compte les possibles déviations du système original. Le deuxième axe de cette ligne de recherche est l'introduction de modèles intégrant la notion de confiance, qui est un paramètre décrivant l'écart entre un système et son modèle.

Objectif : définir une notion de modèles aléatoires avec intervalle de confiance, et construire des algorithmes pour les analyser.

La notion de confiance est couramment définie et étudiée dans le domaine des statistiques, et dans de nombreux modèles il existe des méthodes efficaces pour la calculer ou l'approximer. Je souhaite m'inspirer de ces notions, en particulier de celles issues de la théorie de l'apprentissage (Machine Learning) pour introduire la notion de confiance dans les automates probabilistes. Concrètement, le concept d'algorithmes Probably Approximately Correct (PAC) introduit par Valiant [Val84] est le point de départ de cette investigation.

J'ai commencé à aborder ces questions avec Borja de Balle, en s'appuyant sur des algorithmes connus, dont par exemple ERM (Empirical Risk Minimisation).

Objectif à long terme et intégration

Développer une théorie mathématique des systèmes aléatoires robustes, étendant les modèles et les résultats des systèmes aléatoires en y intégrant la notion de confiance. Cette théorie va permettre de prendre en compte naturellement les erreurs de précision, d'assurer des propriétés de robustesse, et de résoudre des problèmes que l'on ne sait pas résoudre dans les modèles classiques.

Cet axe de recherche est aligné avec les thématiques de l'ANR STOCH-MC (dont je fais partie), qui rassemble en particulier Nathalie Bertrand et Blaise Genest à l'IRISA, et Hugo Gimbert au LaBRI.

L'équipe SuMo à l'IRISA étudie les systèmes aléatoires, en rassemblant plusieurs points de vue : le point de vue de l'informatique fondamentale, mais également du traitement du signal et de l'ingénieurie en électronique. Cette combinaison de compétences permet d'élargir les perspectives, et intégrer cette équipe serait pour moi une excellente opportunité de développer cet axe de recherche. En particulier, Ocan Sankur a étudié en profondeur la notion de robustesse pour les systèmes temporisés [San13].

L'équipe Méthodes Formelles au LaBRI étudie l'application des méthodes formelles dans les systèmes complexes, et en particulier les systèmes aléatoires. De par la diversité des modèles étudiés, mêlant concurrence, aspects temporisés et aléatoires, c'est un terreau idéal pour mener l'étude des systèmes aléatoires robustes.

Simulation en ligne

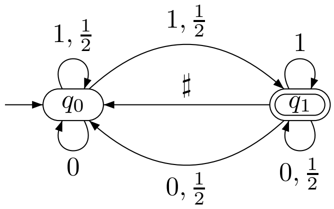
Les algorithmes en ligne prennent connaissance avec leur entrée de manière séquentielle. Cette forme incertitude apparaît dans de nombreux scénarios ; la notion d'algorithme en ligne a été identifiée comme un objet d'étude fondamental depuis les années 80.

La question de Rabin

Rabin a introduit les automates probabilistes en 1963 [Rab63], c'est une généralisation naturelle des automates où chaque transition est déterminée par une distribution aléatoire. Ainsi, un automate probabiliste associe à chaque mot une valeur entre 0 et 1, qui est la probabilité qu'une exécution soit acceptante. Dans son article fondateur, Rabin pose de nombreuses questions, qui ont eu d'importants échos et ont permis le développement de ce modèle de calcul.

La dernière partie de cet article concerne la complexité spatiale en ligne. La problématique est la suivante : considérons un automate probabiliste, on souhaite construire un algorithme en ligne simulant cet automate. Concrètement, l'algorithme reçoit des lettres l'une après l'autre, et doit à chaque instant être capable de déterminer la probabilité d'acceptation du mot lu. Le résultat principal de Rabin dans cette partie de l'article est d'exhiber une sous-classe d'automates probabilistes ayant une faible complexité spatiale en ligne. En d'autres termes, pour chaque automate dans cette sous-classe on peut construire un algorithme en ligne pour le simuler utilisant une quantité d'espace raisonnable. Dans la conclusion de cette partie, Rabin annonce que cette propriété n'est pas vraie en général, sans donner plus de détails, laissant cette question ouverte.

J'ai montré dans l'article [Fij16a] que l'intuition de Rabin est exacte : il existe un automate probabiliste dont la complexité spatiale en ligne est maximale.



Illustrons sur un exemple la notion de complexité spatiale en ligne. Considérons des mots de la forme $u\sharp v$, où u et v sont deux mots sur l'alphabet $\{0, 1\}$ et $\sharp$ est un séparateur. On voit u et v comme la représentation binaire de deux nombres décimaux, notons $\text{bin}(u)$ la valeur de la représentation u. L'automate probabiliste ci-dessus accepte le mot $u\sharp v$ si $\text{bin}(u) \cdot \text{bin}(v) > \frac{1}{2}$, c'est-à-dire si le produit des valeurs est supérieur à $\frac{1}{2}$.
On souhaite simuler cet automate probabiliste par un algorithme en ligne. Le mot $u\sharp v$ est donné lettre à lettre, considérons la situation après avoir lu u. La question est : quelle quantité d'information est nécessaire sur u pour déterminer pour tout mot v si $u\sharp v$ doit être accepté ? Intuitivement, il est nécessaire de connaître avec précision la valeur de $\text{bin}(u)$, ce qui implique d'utiliser un état différent pour chaque mot u. En d'autres termes, la complexité spatiale en ligne de ce langage est maximale.

La question de Rabin constitue un programme de recherche que l'on peut formuler ainsi : étant donné un automate probabiliste, quelle quantité d'espace est nécessaire pour le simuler au moyen d'un algorithme en ligne ?

Importance des algorithmes en ligne

Considérons deux scénarios dans lesquels la notion d'algorithme en ligne intervient.

Le premier exemple est celui des ensembles de données à très grande échelle, par exemple en bases de données, ou dans de très larges réseaux. Le volume de ces données est si important qu'il n'est pas possible de les stocker afin de les analyser à loisir. Il y a essentiellement deux solutions à ce problème : ne considérer qu'une partie des données, c'est le point de vue du test de propriété (Property Testing), ou traiter l'information en ligne, au fur et à mesure de son arrivée, c'est le point de vue du calcul en ligne.
Une réalisation concrète de cette problématique apparaît dans le traitement de fichiers XML, qui est le langage de balise utilisé pour l'échange de fichiers entre systèmes d'informations hétérogènes, et sur Internet en particulier. Plusieurs approches (API) permettent de lire et traiter un document XML ; les méthodes DOM et SAX sont deux

interfaces de programmation instanciant les deux approches les plus courantes. La méthode DOM charge l'intégralité d'un document XML dans une structure de données arborescente, ce qui peut s'avérer impossible lorsque le document est trop volumineux. À l'inverse, la méthode SAX apporte une alternative en parcourant le document XML élément par élément, c'est-à-dire au moyen d'un algorithme en ligne.

Le deuxième exemple est celui des systèmes temps réel, c'est-à-dire évoluant au cours du temps. Ceci inclut une grande variété de systèmes, qui apparaissent en robotique, dans les réseaux de télécommunications ou en développement matériel. Lorsque l'on souhaite vérifier ou contrôler l'évolution d'un tel système, les décisions doivent être prises immédiatement, on doit donc construire des algorithmes en ligne. Pour cette raison, les algorithmes en ligne sont étudiés en théorie du contrôle.

Modélisation

Aujourd'hui, une problématique majeure pour l'étude des algorithmes en ligne est celle de l'analyse de compétitivité. La question, proposée par Karp [Kar92], est la suivante : considérons un problème d'optimisation, où à chaque solution est associée sa qualité, qui est une valeur numérique que l'on cherche à maximiser. Le ratio de compétitivité de ce problème est l'écart entre la solution optimale est celle obtenue par un algorithme en ligne. Les techniques pour évaluer ce ratio forment un domaine de recherche actif [FW98; BE98], ayant de nombreuses applications.

La complexité spatiale en ligne, que je me propose d'étudier dans cette ligne du projet de recherche, s'appuie sur le même modèle d'algorithmes en ligne que l'analyse de compétitivité, mais l'attention est portée sur la quantité d'*espace* utilisée. La notion d'espace pour un algorithme couvre de nombreux aspects pertinents pour son analyse ; par exemple, la précision des valeurs numériques manipulées par l'algorithme est bornée par l'espace qu'il utilise.
La définition de la complexité spatiale en ligne est également due à Karp [Kar67] : la complexité d'un algorithme en ligne est la fonction $\mathbb{N} \to \mathbb{N}$ associant à un entier n le nombre d'états accessibles depuis l'état initial en lisant un mot de longueur au plus n.

La complexité spatiale en ligne a connu un développement rapide dans les années 60, couronné par exemple par le résultat de Hartmanis et Shank [HS69] donnant des bornes précises sur la complexité de déterminer la primalité d'un nombre entier. Les techniques et problématiques qu'elle soulève sont réapparues avec l'avènement des algorithmes de streaming, introduits par une série de papiers (Munro et Paterson [MP80], puis Flajolet et Martin [FM85], suivi du papier fondateur de Alon, Matias et Szegedy [AMS96]). Un algorithme de streaming est limité à la fois en espace et en temps de calcul, et tâche de calculer des statistiques sur l'entrée lue, par exemple sur les fréquences de distributions de lettres et de motifs. Le modèle utilisé pour les algorithmes de streaming est celui des machines de Turing, pour lesquelles il est très difficile d'établir des bornes inférieures. De fait, la plupart des bornes inférieures connues pour les algorithmes de streaming s'appliquent plus généralement à la complexité spatiale en ligne. C'est la raison pour laquelle cette ligne de mon projet de recherche s'appuie sur cette dernière, qui offre une formalisation mathématique plus facile à appréhender.

Axe 1 : développer des techniques de bornes inférieures

Les algorithmes en ligne déterministes sont souvent trop faibles, c'est la raison pour laquelle l'étude des algorithmes en ligne s'appuie souvent sur des modèles plus expressifs, tels que les algorithmes randomisés ou les algorithmes alternants.
Il est facile de donner des bornes inférieures sur la complexité spatiale en ligne déterministe ; il s'agit essentiellement de compter les classes d'équivalence de la relation de Myhill-Nerode [Fij16a]. Dès que l'on considère des algorithmes randomisés ou alternants, très peu de techniques sont connues pour donner des bornes inférieures.

Récemment, j'ai développé une technique de bornes inférieures sur la complexité spatiale en ligne des machines alternantes [Fij16b]. Cette technique permet d'obtenir de nouvelles bornes inférieures, mais elle ne permet pas de traiter tous les exemples. Par exemple, le langage décrit dans l'exemple ci-dessus est étudié dans le livre fondateur de Salomaa et Sotoila [SS78] parce que reconnu par un automate probabiliste à deux états, mais on ne connaît pas de bornes inférieures pour sa complexité spatiale en ligne alternante.

Objectif : montrer que le langage décrivant la multiplication de deux nombres en binaire a une complexité spatiale en ligne alternante exponentielle.

Je prévois d'étudier les différentes approches introduites autour de la complexité de la communication. Dans ce domaine, de nombreuses bornes inférieures existent qui s'appuient sur des théories algébriques avancées, et ont été appliquées à des problèmes pour les algorithmes de streaming.

Axe 2 : classification des automates probabilistes

Les automates probabilistes habitent toutes les classes de complexité spatiale en ligne déterministe [Fij16a]. Ce résultat est une étape vers une question plus ambitieuse, qui classifie les automates probabilistes selon leur complexité spatiale en ligne. En d'autres termes, on souhaite décrire la classe des automates probabilistes pour lesquels il existe un algorithme en ligne efficace de simulation. Cette description peut être de différente nature, par exemple structurelle, algébrique, ou au moyen d'un formalisme logique.

Objectif : caractériser la classe des automates probabilistes ayant une complexité spatiale en ligne polynomiale.

Décrire précisément la classe ci-dessus peut s'avérer être une question très difficile. Dans un premier temps, je commencerai par sous approximer cette classe, c'est-à-dire par décrire des conditions suffisantes pour qu'un automate probabiliste ait une faible complexité spatiale en ligne. Par exemple, restreindre l'ambiguité d'un automate probabiliste implique des résultats positifs sur sa capacité à être simulé en ligne. J'ai obtenu en collaboration avec Cristian Riveros et James Worrell des résultats partiels dans cette direction [FRW16].

Classifier les automates probabilistes selon leur complexité spatiale en ligne signifie décrire de manière générique des algorithmes de simulation pour des classes d'automates probabilistes. Ces techniques sont fondamentales pour développer l'utilisation des automates probabilistes dans les domaines où intervient la notion d'algorithme en ligne.

Objectif à long terme et intégration

L'étude de la complexité spatiale en ligne apporte une compréhension théorique sur l'utilisation de l'espace dans les algorithmes en ligne. Elle exhibe des familiarités entre problèmes et montre des limites intrinsèques pour leur résolution. À terme, son développement permettra de décrire des algorithmes génériques, structurant notre compréhension des algorithmes en ligne.
Les automates probabilistes sont couramment utilisés pour l'étude des langues naturelles. Être capable de simuler en ligne des automates probabilistes permettrait donc d'effectuer du traitement de langues naturelles en temps réel, ce qui constitue un enjeu majeur des décennies à venir.

Cet axe de recherche est un projet personnel. Le modèle a été proposé et étudié auparavant, et les questions que je propose sont motivées par divers enjeux contemporains, en écho à des travaux majeurs réalisés dans des cadres connexes. Bien que les questions soient nouvelles, les outils mobilisés me sont familiers, et sont étudiés dans les équipes Méthodes Formelles du LaBRI et SuMo à l'IRISA.

Olivier Gauwin au LaBRI coordonne le projet ANR ExStream, dont l'objectif est de charactériser les tâches pouvant être effectuées en ligne, et de construire des algorithmes pour ces tâches. Un exemple concret de tâche est la validation de fichiers XML, qui est étudiée en particulier par Gabriele Puppis, Thomas Place et Anca Muscholl dans l'équipe Méthodes Formelles. Par ailleurs, la problématique des algorithmes en ligne apparaît en théorie du contrôle, étudié dans l'équipe SuMo. Je prévois d'étudier les liens entre la complexité spatiale en ligne et ces modèles et problématiques.

Réalisation du programme de recherche

Je souhaite développer mon projet de recherche dans l'un des deux laboratoires suivants :

1. Le Laboratoire Bordeaux de Recherche en Informatique (**LaBRI**, Bordeaux), dans l'équipe Méthodes Formelles, dirigée par Jérôme Leroux.

   Au sein de cette équipe, Hugo Gimbert se spécialise dans l'étude des systèmes aléatoires. J'ai commencé à collaborer avec lui avant même de commencer ma thèse, et la plupart de mes travaux sur les systèmes aléatoires pendant ma thèse sont issus de nos interactions ou de questions qu'il a posées. Je souhaite continuer cette collaboration, au sein de l'ANR STOCH-MC notamment, dont nous faisons tous les deux partie.

   Le projet ANR ExStream dirigé par Olivier Gauwin étudie les algorithmes de streaming, avec des problématiques très proche de celles développées dans la deuxième partie de mon projet. Le traitement de XML et les logiques sur les arbres, étudiés par Diego Figueira, Anca Muscholl et Gabriele Puppis, aborde également des questions liées. . Ces connections permettent d'élargir le spectre d'applications de la complexité spatiale en ligne, et motivent de nouvelles questions.

   Enfin, Jérôme Leroux, Gabriele Puppis et Marc Zeitoun étudient des systèmes à compteurs avec des problématiques proches de celles que j'ai étudiées pendant ma thèse, ce qui pourrait amener à de futures collaborations.

2. L'Institut de Recherche en Informatique et Systèmes Aléatoires (**IRISA**, Rennes), dans l'équipe SuMo, dirigée par Éric Fabre.

   La première ligne directrice de mon projet est directement reliée aux travaux des membres de l'équipe SuMo. Ocan Sankur a étudié en profondeur la notion de robustesse pour les automates temporisés. L'étude de distances entre systèmes aléatoires est influencée et motivée par les travaux récents d'Éric Fabre et Blaise Genest. C'est déjà l'objet d'une collaboration avec Nathalie Bertrand, Blaise Genest et S. Akshay, commencée l'année dernière grâce au soutien de l'ANR STOCH-MC.

   Enfin, la théorie du contrôle, qui est un des thèmes développés dans l'équipe SuMo, entre en résonnance avec la seconde ligne directrice de mon projet, permettant de créer des ponts entre ces deux domaines.

Collaborations nationales et internationales

Au niveau national, en dehors des deux laboratoires cités ci-dessus, j'entretiens des liens scientifiques avec l'équipe Vérification au LACL (Youssouf Oualhadj), l'équipe Automates de l'IRIF (Thomas Colcombet, Florian Horn, Mai Gehrke, Jean-Éric Pin et Olivier Serre), l'équipe PLUME de l'ENS-LIP (Denis Kuperberg, Matteo Mio et Damien Pous), l'équipe MoVe du LIF (Benjamin Monmège, Pierre-Alain Reynier et Jean-Marc Talbot) et le LSV (Laurent Doyen et Stefan Göller).

Au niveau international, je souhaite pérenniser mes liens avec l'équipe Automates de l'Université de Varsovie, dirigée par Damian Niwiński, le groupe de Jean-François Raskin de l'Université libre de Bruxelles, le groupe de Krishnendu Chatterjee de IST Austria, le groupe de Christof Löding de RWTH Aachen, l'équipe Vérification de l'Université d'Oxford et Borja de Balle de l'Université de Lancaster.

Candidature dans un laboratoire de l'INSMI, concours 06/04

Je candidate également sur le concours 06/04 pour une affectation dans un laboratoire relevant de l'INSMI. Je souhaite à ce titre développer mon programme de recherche au département Mathématiques et Interactions de Nice.

Le programme de recherche que je décris dans ce document appartient à l'informatique fondamentale, dans le sens où les modèles et les problématiques sont issues de l'informatique, et en particulier de l'algorithmique. Les réponses cependant sont souvent mathématiques, elles nécessitent de faire le lien entre notions informatiques et théories mathématiques avancées. Concrètement, j'ai étudié des notions issues de la topologie (complétions profinies, espaces polonais), de l'algébre (théorie des semigroupes) et de la théorie algébrique des nombres (approximations Diophantiennes). Ces domaines sont représentés à l'Université de Nice, et intégrer le département de mathématiques me permettraient d'étendre ces liens.

Une autre connexion naturelle est avec les statistiques. L'axe de recherche de mon projet de recherche lié à la robustesse, et en particulier à la notion de confiance s'inspire d'idées issues des statistiques, que je pourrais développer à l'Université de Nice.

Références bibliographiques personnelles citées (liste partielle)

[Fij16a] Nathanaël Fijalkow. *Lower Bounds for Alternating Online Space Complexity*. Tech. rep. 2016.

[FK16] Nathanaël Fijalkow and Bartek Klin. *A Logical Viewpoint on Probabilistic Bisimulation over Distributions*. Tech. rep. 2016.

[FKP16] Nathanaël Fijalkow, Bartek Klin, and Prakash Panangaden. *Logical Characterisation of Probabilistic Simulation*. Tech. rep. 2016.

[FRW16] Nathanaël Fijalkow, Cristian Riveros, and James Worrell. *Probabilistic Automata of Bounded Ambiguity*. Tech. rep. 2016.

[Fij16b] Nathanaël Fijalkow. "Online Space Complexity of Probabilistic Automata". In: *LFCS*. 2016, pp. 106–116.

[FKS16] Nathanaël Fijalkow, Stefan Kiefer, and Mahsa Shirmohammadi. "Trace Refinement in Labelled Markov Decision Processes". In: *FoSSaCS*. 2016, pp. 303–318.

Références bibliographiques

[AAGT15] Manindra Agrawal, S. Akshay, Blaise Genest, and P. S. Thiagarajan. "Approximate Verification of the Symbolic Dynamics of Markov Chains". In: *J. ACM* 62.1 (2015).

[AAOW15] S. Akshay, Timos Antonopoulos, Joël Ouaknine, and James Worrell. "Reachability problems for Markov chains". In: *Inf. Process. Lett.* 115.2 (2015).

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. "The Space Complexity of Approximating the Frequency Moments". In: *STOC*. 1996.

[BE98] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[FW98] Amos Fiat and Gerhard J. Woeginger, eds. *Online Algorithms, The State of the Art*. Vol. 1442. Lecture Notes in Computer Science. Springer, 1998.

[FM85] Philippe Flajolet and G. Nigel Martin. "Probabilistic Counting Algorithms for Data Base Applications". In: *J. Comput. Syst. Sci.* 31.2 (1985).

[GO10] Hugo Gimbert and Youssouf Oualhadj. "Probabilistic Automata on Finite Words : Decidable and Undecidable Problems". In: *ICALP*. 2010, pp. 527–538.

[HS69] Juris Hartmanis and H. Shank. "Two Memory Bounds for the Recognition of Primes by Automata". In: *Mathematical Systems Theory* 3.2 (1969).

[HKK14] Holger Hermanns, Jan Krčál, and Jan Křetínský. "Probabilistic Bisimulation : Naturally on Distributions". In: *CONCUR*. Vol. 8704. Lecture Notes in Computer Science. Springer, 2014, pp. 249–265.

[Kar92] Richard M. Karp. "On-Line Algorithms Versus Off-Line Algorithms : How Much is it Worth to Know the Future ?" In: *IFIP*. 1992.

[Kar67] Richard M. Karp. "Some Bounds on the Storage Requirements of Sequential Machines and Turing Machines". In: *J. ACM* 14.3 (1967).

[KNP07] Marta Kwiatkowska, G. Norman, and D. Parker. "Stochastic Model Checking". In: *SFM*. Ed. by M. Bernardo and J. Hillston. Vol. 4486. LNCS (Tutorial Volume). Springer, 2007.

[LS89] Kim G. Larsen and Arne Skou. "Bisimulation Through Probabilistic Testing". In: *POPL*. 1989, pp. 344–352.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[MP80] J. Ian Munro and Mike Paterson. "Selection and Sorting with Limited Storage". In: *Theor. Comput. Sci.* 12 (1980).

[Rab63] Michael O. Rabin. "Probabilistic Automata". In: *Information and Control* 6.3 (1963), pp. 230–245.

[SS78] Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Texts and Monographs in Computer Science. Springer, 1978.

[San13] Ocan Sankur. "Robustness in Timed Automata : Analysis, Synthesis, Implementation". PhD thesis. Laboratoire Spécification et Vérification, ENS Cachan, France, 2013.

[Val84] Leslie G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pp. 1134–1142.