Calcul et connaissance : l'information comme ressource algorithmique

Laurent Feuilloley

1 Cadre du projet

1.1 Introduction

L'algorithmique est essentielle en informatique, car elle apporte des garanties sur les calculs, autant en termes de correction que de complexité de calcul. Malheureusement, le cadre standard de l'algorithmique, « entrée – calcul – sortie », est trop restrictif pour capturer la plupart des contextes de calcul d'aujourd'hui. L'une des raisons de ce conflit entre la théorie et la pratique, est le fait que le cadre standard ne rend pas bien compte de l'information qui est vraiment accessible à l'algorithme. Parfois, on a moins d'information que dans ce modèle, et parfois plus d'information. Je vais commencer par développer rapidement ces deux points.

Information incomplète De nombreux modèles algorithmiques (calcul distribué local, calcul en ligne, mais aussi streaming, test de propriété, calcul massivement parallèle, etc.) sont des modèles dit *à information incomplète* : il y est nécessaire de prendre des décisions, alors même que l'entrée n'est connue que partiellement. Ces modèles sont justifiés par les limites du cadre algorithmique classique quand il s'agit de modéliser des systèmes distribués dans l'espace (les réseaux) ou dans le temps (les systèmes de prise de décision en temps réel). Dans ces systèmes, on a en effet une multiplicité de calculs coexistants, travaillant sur des morceaux de l'entrée et produisant un faisceau de sorties.

Information supplémentaire Une autre direction de recherche très active actuellement consiste à incorporer de l'*information supplémentaire* au calcul. Ici, c'est le fait que le cadre classique soit « isolé de l'extérieur » qui nécessite d'être dépassé, car un calcul existe quasiment toujours dans un contexte. Par exemple, pour un problème fixé, si l'on a une base de données d'instances déjà résolues, on peut souvent utiliser cette expérience pour accélérer le traitement d'une nouvelle instance. Notons que l'information supplémentaire dont on dispose est forcément partielle ou peu fiable, sinon il n'y aurait pas de difficulté algorithmique. De nombreux modèles ont été proposés selon les cas : on parle alors de conseils (pour les informations fiables mais partielles), de certificats (pour les informations qui accompagnent une solution et qu'il faut vérifier), de prédictions (pour les informations qui peuvent être fiables ou non, et peuvent être utilisées pour accélérer le calcul), etc.

Synthèse Ces deux aspects, information incomplète et information supplémentaire, semblent à première vue indépendants, voire opposés. Je propose au contraire de les rassembler dans une perspective générale, en considérant l'information comme une ressource à part entière de l'algorithme. La pertinence de ce point de vue est illustrée par [CCF+21], où nous avons étudié un modèle d'algorithmique en ligne avec échantillons, dans lequel le futur n'est pas connu exactement (information incomplète), mais une certaine connaissance est néanmoins accessible par l'échantillon, qui représente l'expérience acquise (information supplémentaire). Dans ce contexte, j'ai adapté les techniques du calcul distribué au calcul en ligne, ce qui montre que prendre un point de vue général sur l'information, au-delà des modèles spécifiques, est pertinent et prometteur d'un point de vue technique.

1.2 Questions et méthodologie

Si l'on pense à l'information utilisée par un algorithme, on peut distinguer deux postures. D'abord une posture passive, où l'information est donnée, et où il faut l'utiliser au mieux. Dans ce contexte, il y a deux questions naturelles : À quelle information a-t-on accès ? Et avec quelle efficacité peut-on résoudre le problème, étant donné cette information ? En prenant une posture active, on se demandera plutôt : Quelle serait l'information dont on aurait besoin pour atteindre une certaine performance ? Est-il facile d'obtenir cette information ?

Ma méthodologie pour répondre à ces questions est d'une part, pour chaque modèle, d'étudier l'information de manière quantitative et qualitative, et d'autre part de comparer et d'évaluer les modèles.

Approche quantitative En considérant comme deux ressources l'information et la puissance de calcul, l'objectif est d'établir des compromis entre la quantité d'information utilisée et la performance de l'algorithme étant donné cette information. Mon bagage technique apportera des outils adaptables à de nombreux contextes. Je pense en particulier au raisonnement par indistingabilité, allié à l'étude combinatoire du graphe des conflits associé, une approche générique et indépendante du modèle précis étudié [1]. À plus long terme, je compte me former en théorie de l'information, dans le but de capturer quantitativement la notion d'information utile, et d'étudier les intuitions que pourrait apporter la théorie des jeux, en voyant le système comme un jeu entre l'algorithme et un adversaire fournissant l'information de manière adaptative.

Approche qualitative L'approche quantitative doit se doubler d'une approche qualitative, dans le sens où l'on ne peut pas se contenter de mesurer la taille de l'information. D'abord, dans le but d'améliorer notre compréhension, et de faire des transferts de techniques, il est bon d'étudier la nature de l'information utile ou accessible. Par exemple, dans le cadre de [FFM+21], le fait de considérer les ordres sur les graphes comme une information utile (inspiré par [FH21a]) m'a permis de faire le premier pas vers une certification optimale de la planarité. Ensuite, pour des informations de même taille et de même utilité, il est essentiel de faire une différence entre celles qui sont faciles à obtenir et celles qui ne le sont pas.

Comparaison et évaluation Les modèles algorithmiques à information incomplète ou supplémentaire, ont jusqu'ici été étudiés indépendamment les uns des autres. Les étudier ensemble offre l'opportunité de transférer des techniques, mais permet aussi de faire des comparaisons et d'établir des hiérarchies, pour une meilleure compréhension globale. Dans cette direction, un enjeu sera aussi de mesurer la pertinence des modèles comme reflets de la pratique, et éventuellement de proposer des approches plus adaptées.

Dans les sections à venir, je vais me concentrer sur trois directions de recherche, pour illustrer de manière concrète l'application de ma méthodologie.

2 Direction 1 : Le pouvoir de la certification compacte

Comme décrit dans le rapport d'activité, ma spécialité est la certification locale en calcul distribué. La certification locale est un exemple marquant d'interaction entre information et calcul. Lors d'une première phase de calcul, une solution a été calculée, mais sa correction

1. Voir la discussion de [CCF+21] dans le rapport des travaux effectués.

n'est pas assurée à long terme, du fait des fautes du système. Dans la phase qui suit le calcul, on veut donc pouvoir vérifier que la solution est toujours correcte, mais les ressources sont beaucoup plus limitées. On demande alors à avoir, en plus de la solution, un certificat facilement vérifiable de sa validité.

Deux grandes questions sur la certification compacte En 2011, Göös et Suomela ont identifié le fait d'avoir des certificats de $O(\log n)$ bits par sommet, comme étant un standard de certification compacte. Depuis, la question de déterminer quelles propriétés ont une certification compacte a reçu une attention notable. Dans cette direction, deux questions plus spécifiques se distinguent. Dans les deux cas, il s'agit de dépasser l'approche classique qui consiste à n'étudier qu'une propriété donnée, pour étudier une large famille de propriétés. Nous avons obtenu des résultats partiels sur ces questions [BFP21a, BFP21b] (détaillés dans le rapport), mais elles restent encore très ouvertes.

Question : Existe-t-il une certification compacte pour toutes les classes de graphes closes par mineurs ?

Si la réponse à cette question est positive, alors on gagnera beaucoup en intuition : les propriétés de graphes n'ayant pas de certification compacte seraient pathologiques, dans le sens où elles ne seraient pas stables par des opérations naturelles dans des réseaux, comme le retrait de nœuds ou la déconnexion de liens.

Ensuite, les formules logiques dites MSO, permettent d'exprimer la plupart des propriétés intéressantes en théorie des graphes et en optimisation combinatoire. Elles jouent un rôle central en calcul centralisé, et en particulier en complexité paramétrée, car vérifier ces propriétés peut être fait en temps polynomial dans de nombreux cas.

Question : Quand peut-on certifier de manière compacte les formules MSO ?

Première approche : comprendre la treewidth Pour les deux questions, une étape importante est de comprendre le paramètre de graphe appelé treewidth [2]. Un article très récent [FMRT21] introduit une certification de la treewidth, mais celle-ci utilise $\Theta(\log^2 n)$ bits ; elle n'est donc pas compacte au sens strict. Il est cependant possible que $\Theta(\log^2 n)$ soit optimal pour cette propriété, et pour comprendre si c'est le cas je propose de revisiter la certification de l'arbre couvrant de poids minimum, qui est le seul cas connu pour lequel cette taille est optimale [KK07]. Cette preuve est complexe et utilise des outils ad hoc, il s'agira de déterminer si elle peut être généralisée.

Deuxième approche : arbres couvrants et réduction Une deuxième approche est plus qualitative et provient du constat qu'une primitive omniprésente en certification compacte est la certification d'arbres couvrants. Je propose d'étudier une restriction du modèle, où la certification ne peut être qu'une conjonction d'arbres couvrants. Ceci est intéressant pour trois raisons. D'abord, restreindre la forme des certificats pourrait permettre d'obtenir des bornes inférieures plus facilement, pour répondre partiellement aux grandes questions ci-dessus. Ensuite, imposer une forme standard aux certificats permet d'assurer qu'ils seront faciles à calculer. Enfin, cela revient à introduire une notion de réduction entre problèmes (*e.g.* à quel point est-ce que le problème est plus dur que de certifier un arbre couvrant ?), notion

2. En particulier, en comprenant la treewidth, on comprend les familles excluant des mineurs planaires [RS86], et on se place dans le cadre exact du théorème de Courcelle en calcul centralisé [Cou97].

centrale en théorie de la complexité standard, mais absente de la théorie de la complexité de la vérification distribuée [FKP13].

3 Direction 2 : Algorithmique en ligne et expérience

Comme déjà évoqué, en calcul en ligne, l'information accessible à l'algorithme est un sujet essentiel : le futur n'est pas connu à proprement parler, mais il est loin d'être complètement inconnu, du fait de l'expérience accumulée.

Problématique Dans ce contexte, une question importante est : à quelle information peut-on avoir accès, et à quel prix ? En particulier, si l'on a une grande base de données d'où tirer de l'information, quelle est la complexité de l'accès à cette information ?

Optimisation de paramètres Si l'on veut s'interroger sur la complexité d'accès à une information, il est pertinent de spécifier la manière dont l'algorithme utilise l'information. Par exemple, pour le problème des secrétaires, étudié dans [CCF+21], l'algorithme a une forme très simple et prend un unique paramètre, que nous optimisons en fonction de l'échantillon. De façon générale, on peut fixer un algorithme ayant quelques paramètres et considérer ces paramètres comme l'information supplémentaire que l'on va calculer à partir de la base de données. Cette approche est aussi justifiée par la pratique, où l'on préférera souvent utiliser l'algorithme standard d'une bibliothèque dont on optimisera simplement les paramètres.

La question de l'optimisation des paramètres est un domaine de recherche très actif, en particulier dans l'optique de la complexité en nombre de requêtes/échantillons, par exemple dans le cadre de l'apprentissage PAC [3]. Je propose de prendre une approche un peu différente, où l'on veut optimiser le paramètre en faisant une passe sur la base de données en streaming. Par exemple, notre algorithme [CCF+21] ne demande que de conserver les k valeurs les plus élevées, ce qui peut se faire facilement en streaming. Une généralisation du problème des secrétaires est le problème des couplages, qui est fondamental dans le domaine et très utilisé dans la pratique, ce qui amène à la question suivante.

Question : Peut-on résoudre efficacement le problème du couplage en ayant un accès en streaming à un échantillon des arêtes ?

Comparaison entre les modèles Pour le calcul en ligne, une variété de modèles d'information supplémentaire a été introduite. Un objectif important est de comparer et de hiérarchiser ces modèles. Par exemple, la performance de notre algorithme (dans [CCF+21]) tend vers la performance des algorithmes ayant une connaissance exacte de la distribution sous-jacente, ce qui est naturel, mais nous n'avons pas encore d'outils pour montrer la convergence du premier modèle vers le deuxième.

Extension au-delà du calcul en ligne Enfin, les idées développées ici s'appliquent au-delà du calcul en ligne, en particulier aux modèles où l'on accède à l'entrée « par morceaux » mais pas forcément de manière séquentielle. Un exemple typique est la recherche dans un tableau trié avec une prédiction, pour laquelle un algorithme simple atteint une complexité $O(\log \eta)$ au lieu de $O(\log n)$, où η est la différence entre la position prédite et la position réelle [MV20].

3. Voir [Hau90] pour une référence sur l'apprentissage PAC, et [BDD+21] pour un article récent sur l'hyperparamétrage.

4 Direction 3 : Ordre de sommets dans les graphes

Pour finir ce projet, je vais renforcer le volet qualitatif, en me focalisant sur un type d'information dans les graphes : un ordre total des sommets. Cette information est à la fois générique et pratique d'un point de vue algorithmique : elle peut être efficacement stockée au niveau des sommets (dans une base de données ou dans un réseau), et peut souvent être générée au fur et à mesure, lors d'un parcours du graphe.

Calcul de l'ordre Une question naturelle est : quelle est la complexité du calcul d'un ordre satisfaisant certaines propriétés ? Dans le domaine des motifs interdits discuté dans le rapport des travaux effectués [FH21a, FH21b], Hell, Mohar et Rafiey ont proposé en 2014 une conjecture toujours ouverte [HMR14] :

Conjecture : Pour chaque motif, le problème du calcul de l'ordre associé est ou bien polynomial, ou bien NP-complet.

Dans cette direction, mon projet est d'établir des critères sur les motifs (par exemple sur la connexité) qui permettent de classer le problème de reconnaissance dans P ou NP.

Ordre et optimisation combinatoire Pour certaines classes de graphes spécifiques, un ordre calculé en temps linéaire permet de résoudre une variété de problèmes d'optimisation combinatoire avec un algorithme glouton générique [4]. Dans [CFPS15], nous avons montré que pour les *grounded rectangle graphs* (une classe de graphes venue de la recherche opérationnelle), un ordre permet aussi de résoudre des problèmes d'optimisation, mais en passant cette fois par une programmation dynamique plus complexe. Cette deuxième approche est beaucoup moins bien comprise et serait a priori plus puissante, d'où la question suivante.

Question : Dans quel cas peut-on calculer un ordre qui permettent de résoudre de manière générique des problèmes d'optimisation par programmation dynamique ?

Vérification de l'ordre Étant donné un ordre, on veut aussi pouvoir vérifier efficacement (dans différents modèles) qu'une propriété donnée est satisfaite. Un exemple concret est de vérifier qu'il ne contient pas un certain motif, au sens de [FH21a]. C'est toujours possible en $O(n^k)$ où k est la taille du motif, mais le développement récent de la « complexité dans P » montre que minimiser l'exposant du polynôme est à la fois difficile et très intéressant [5]. Ici, des bornes inférieures pourraient être obtenues grâce aux techniques d'indistingabilité dans des modèles où la vue du graphe est partielle. Enfin, on peut faire une analogie prometteuse entre la recherche de structure dans un graphe ordonné, et celle de motif dans un texte, ce dernier modèle ayant une très riche littérature [6].

4. Voir par exemple le survey [Cor04]
5. Voir par exemple le court tutoriel [Bri19].
6. J'ai eu l'occasion en 2019 de travailler sur l'algorithmique du texte et d'y développer des bornes inférieures [CFS19].

5 Intégration

Je souhaite mettre en place ce projet de recherche dans l'un des trois laboratoires suivants.

Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)
Lyon, équipe GOAL, dirigée par Eric Duchêne.
Cette équipe est celle dans laquelle j'effectue actuellement un postdoc. J'y ai commencé une collaboration au long cours avec Théo Pierron et Nicolas Bousquet, sur la certification locale de graphes [BFP21a, BFP21b], et j'ai adopté leurs thématiques de reconfigurations [BFHR21, BFHR21]. Mes intérêts sont aussi très proches de ceux d'Aline Parreau et Eric Duchêne (théorie des jeux, optimisation combinatoire). Enfin, comme mon projet consiste en partie à mieux comprendre l'information utilisable en pratique, la collaboration avec les membres de l'équipe ayant une recherche plus appliquée, notamment Mohammed Haddad et Hamida Seba, sera naturelle.

Laboratoire de l'Informatique du Parallélisme (LIP)
Lyon, équipe MC2, dirigée par Michaël Rao.
Grâce aux « rencontres graphes à Lyon » que j'organise deux fois par mois, je connais bien les membres de cette équipe intéressés par les graphes (Édouard Bonnet, Christophe Crespelle, Stephan Thomassé, Nicolas Trotignon et Rémi Watrigant). Les classes définies par des structures interdites (motifs, sous-graphes, mineurs) font partie de nos intérêts en commun. La direction actuelle de ma recherche en certification, notamment l'étude de paramètres de graphes, comme la treedepth ou la treewidth, sont aussi des sujets très pertinents dans le cadre de cette équipe qui a développé une expertise sur ces objets. Enfin, Omar Fawzi serait un excellent point d'entrée pour la théorie de l'information utile à mon projet.

Laboratoire Sciences pour la conception, l'Optimisation et la Production (G-SCOP)
Grenoble, équipe Optimisation Combinatoire, dirigée par Louis Esperet.
Dans cette équipe, je suis en contact régulier avec Louis Esperet, spécialiste de théorie des graphes, qui a travaillé sur la certification locale et l'optimisation distribuée. Tous les autres membres de l'équipe sont aussi des collaborateurs pertinents : nous partageons le même intérêt pour les graphes, l'optimisation combinatoire et les structures discrètes en général. Par ailleurs, Karine Altisen, spécialiste d'auto-stabilisation, avec qui j'ai pu échanger dans le cadre de l'ANR ESTATE, est aussi présente à Grenoble (au laboratoire Vérimag).

Note : Seuls les travaux mentionnés dans ce projet sont listés ci-dessous. Mes travaux sont séparés en publiés et non publiés, mais pas en journaux et conférences. Cette distinction est faite dans la liste des publications.

Références personnelles non publiées

BFP21b Nicolas Bousquet, Laurent Feuilloley, and Théo Pierron. Local certification of MSO properties for bounded treedepth graphs, 2021. arxiv: 2110.01936.

Références personnelles publiées

BFHR21 Nicolas Bousquet, Laurent Feuilloley, Marc Heinrich, and Mikaël Rabie. Distributed recoloring of interval and chordal graphs. In *25th International Conference on Principles of Distributed Systems, OPODIS 2021*, 2021. arxiv: 2109.06021.

BFP21a Nicolas Bousquet, Laurent Feuilloley, and Théo Pierron. Local certification of graph decompositions and applications to minor-free classes. In *25th International Conference on Principles of Distributed Systems, OPODIS 2021*, 2021. arxiv: 2108.00059.

CCF+21 José R. Correa, Andrés Cristi, Laurent Feuilloley, Tim Oosterwijk, and Alexandros Tsigonias-Dimitriadis. The secretary problem with independent sampling. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 2047–2058. SIAM, 2021. doi:10.1137/1.9781611976465.122.

CFPS15 José R. Correa, Laurent Feuilloley, Pablo Pérez-Lantero, and José A. Soto. Independent and hitting sets of rectangles intersecting a diagonal line : Algorithms and complexity. *Discret. Comput. Geom.*, 53(2) :344–365, 2015. doi:10.1007/s00454-014-9661-y. Voir aussi la version conférence à LATIN 2014.

CFS19 Vincent Cohen-Addad, Laurent Feuilloley, and Tatiana Starikovskaya. Lower bounds for text indexing with mismatches and differences. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 1146–1164, 2019. doi:10.1137/1.9781611975482.70.

FFM+21 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Compact distributed certification of planar graphs. *Algorithmica*, 83(7) :2215–2244, 2021. doi: 10.1007/s00453-021-00823-w. Voir aussi la version conférence à PODC 2020.

FH21a Laurent Feuilloley and Michel Habib. Graph classes and forbidden patterns on three vertices. *SIAM J. Discret. Math.*, 35(1) :55–90, 2021. doi: 10.1137/19M1280399.

FH21b Laurent Feuilloley and Michel Habib. Classifying grounded intersection graphs via ordered forbidden patterns, 2021. arxiv: 2112.00629.

Autres références

BDD+21 Maria-Florina Balcan, Dan F. DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms ? generalization guarantees for data-driven algorithm design. In *STOC '21 : 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 919–932. ACM, 2021. doi:10.1145/3406325.3451036.

Bri19 Karl Bringmann. Fine-grained complexity theory (tutorial). In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126, pages 4 :1–4 :7, 2019. doi: 10.4230/LIPIcs.STACS.2019.4.

Cor04 Derek G. Corneil. Lexicographic breadth first search - A survey. In *Graph-Theoretic Concepts in Computer Science, 30th International Workshop,WG 2004*, volume 3353, pages 1–19, 2004. doi:10.1007/978-3-540-30559-0_1.

Cou97 Bruno Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In Grzegorz Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1 : Foundations*, pages 313–400. World Scientific, 1997.

FKP13 Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *J. ACM*, 60(5) :35, 2013.

FMRT21 Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. A meta-theorem for distributed certification, 2021. arxiv: 2112.03195.

Hau90 David Haussler. Probably approximately correct learning. In Howard E. Shrobe, Thomas G. Dietterich, and William R. Swartout, editors, *Proceedings of the 8th National Conference on Artificial Intelligence. AAAI 90*, pages 1101–1108, 1990. url: AAAI90-163.

HMR14 Pavol Hell, Bojan Mohar, and Arash Rafiey. Ordering without forbidden patterns. In *Algorithms - ESA 2014 - 22th Annual European Symposium*, volume 8737, pages 554–565, 2014. doi:10.1007/978-3-662-44777-2_46.

KK07 Amos Korman and Shay Kutten. Distributed verification of minimum spanning trees. *Distributed Computing*, 20(4) :253–266, 2007.

MV20 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 646–662. Cambridge University Press, 2020. doi:10.1017/9781108637435.037.

RS86 Neil Robertson and Paul D. Seymour. Graph minors. v. excluding a planar graph. *J. Comb. Theory, Ser. B*, 41(1) :92–114, 1986. doi:10.1016/0095-8956(86)90030-4.