Travaux effectués

Laurent Feuilloley

Mes travaux de recherche se placent dans les domaines de l'algorithmique et des mathématiques discrètes. Dans ces domaines, j'ai travaillé sur des thématiques variées : calcul distribué, calcul en ligne, théorie des graphes, et dans une moindre mesure, algorithmes d'approximation, algorithmique du texte et géométrie algorithmique. Comme on le verra dans ce rapport, deux objets apparaissent de manière transversale : la *notion de graphe*, qui peut être un sujet d'étude, un modèle de calcul, ou un outil d'analyse, et la *notion de localité*, qui se traduit en algorithmique par des modèles où l'on a une connaissance partielle de l'entrée, et en théorie des graphes par l'étude de structures locales.

Mon expertise porte en premier lieu sur la *certification locale*, une notion venant de la tolérance aux fautes en calcul distribué. La première section de ce document résume mes travaux sur ce sujet. Le reste de mes recherches est ensuite présenté par thématique : calcul distribué, algorithmique en ligne et théorie des graphes.

Certains articles récents font l'objet d'un focus particulier et reflètent la diversité de mes thématiques de recherche ; ce sont les trois articles joints à ce document. D'autres travaux sont esquissés ou mentionnés rapidement. Enfin, dans un but de concision, plusieurs articles, plus isolés thématiquement ou plus anciens, ne sont pas mentionnés.

1 Certification locale

1.1 Introduction

La certification locale est un sujet de calcul distribué sur réseau. Dans ce domaine, un réseau est modélisé par un graphe, les machines étant représentées par les sommets et les canaux de communication par les arêtes. L'objectif général du domaine est de minimiser la communication pour résoudre une tache donnée [Pel00].

En algorithmique, la certification consiste à calculer, en plus d'une solution à un problème donné, un certificat qui permet de vérifier la solution plus rapidement [MMNS11]. En algorithmique distribuée, la certification est un sujet essentiel, pour deux raisons. D'abord, les systèmes distribués sont plus sujets à des erreurs, ce qui implique qu'il est nécessaire de pouvoir vérifier l'état du système. Ensuite, dans ce contexte, la vérification de la solution est souvent aussi coûteuse que sa construction, ce qui amène à chercher des moyens d'accélérer la vérification. Pour être concret, je vais développer un exemple de certification locale.

Exemple : certification d'un arbre couvrant. Une primitive essentielle dans les réseaux est la construction d'un arbre couvrant, qui sert par exemple à transmettre efficacement un message à tout le réseau (« broadcast »). Un arbre couvrant est conservé en mémoire de façon distribuée, sous la forme d'un pointeur vers le parent pour chaque nœud. En cas de faute, un pointeur peut être corrompu et un cycle peut apparaître. On aimerait pouvoir détecter une telle situation critique, en investissant un minimum de ressources. Malheureusement, détecter un cycle peut demander une communication à l'échelle du réseau entier (si le cycle est très long). En utilisant la certification locale, on peut faire en sorte que chaque sommet ne communique qu'avec ses voisins. L'idée est que l'algorithme qui construit l'arbre doit garder en mémoire un certificat (en plus de la solution) : la distance à la racine [AKY90]. Si chaque sommet vérifie la cohérence des distances et des pointeurs avec ses voisins, alors on a les deux propriétés clés suivantes :

- S'il n'y a pas de cycle, et que les distances ont été bien reportées, alors il n'y a pas d'incohérence détectée.
- S'il y a un cycle, quelles que soient les distances écrites (même si elles sont fournies par un adversaire), il y aura toujours un sommet pour détecter une incohérence, et ce sommet pourra lancer le calcul d'une nouvelle solution.

Ici, chacun des n sommets stocke un certificat de $O(\log n)$ bits. Cette taille est optimale, dans le sens où en utilisant $o(\log n)$ on ne peut pas assurer les deux propriétés clés [KKP10][1].

Contexte historique. La notion de certification locale été introduite sous le nom de *proof-labeling scheme* en 2005 [KKP10][2], dans le but d'étudier en profondeur les certificats, qui avait été étudiés jusque-là implicitement dans le contexte plus général des algorithmes dits autostabilisants (nous reviendrons plus tard sur cette notion). Depuis la certification locale est apparue naturellement dans d'autres contextes.

Quand j'ai commencé ma thèse en 2015, deux aspects de la certification locale avaient été étudiés. Pendant la période 2005-2011, une première vague d'articles a établi des bornes sur les tailles de certificats pour des problèmes spécifiques (*e.g.* arbre couvrant [KKP10], arbre couvrant de poids minimum [KK07], isomorphisme [GS16]). Ensuite, en 2010-2015, une approche plus structurelle a été développée, en se basant sur l'analogie entre la certification locale et la classe de complexité NP : dans les deux cas, il s'agit de dire que l'on peut vérifier efficacement une solution à l'aide d'un certificat (voir par exemple [FKP13]).

Dans les sous-sections qui suivent, je vais décrire mes travaux dans le domaine de la certification locale, et plus généralement de la vérification locale. La dernière sous-section porte sur mes surveys, exposés invités et rapports sur ce sujet.

1.2 Théorie de la complexité

Pendant mon stage de master et la première année de ma thèse, j'ai travaillé sur l'analogie avec la théorie de la complexité.

En collaboration avec Pierre Fraigniaud, je me suis intéressé à la généralisation d'un résultat fondamental de dérandomisation en calcul distribué : si l'on considère des problèmes dont les solutions peuvent être vérifiées en temps constant déterministe, alors la construction randomisée n'est pas plus puissante que la construction déterministe [NS95]. Nous avons étendu ce théorème aux problèmes dont les solutions peuvent être vérifiées en temps constant probabiliste (qui est un ensemble de problèmes strictement plus général) [FF21].

Ensuite, avec Juho Hirvonen, doctorant de l'équipe finlandaise où j'avais travaillé en 2015, nous avons défini et étudié un analogue distribué de la hiérarchie polynomiale [FFH21a]. Notre hiérarchie a amené d'autres auteurs à développer des analogues des classes Arthur-Merlin, ou PCP [KOS18, NPY20]).

---

[1] Je cite de préférences les versions journal des articles, ce qui peut modifier légèrement la chronologie. Par exemple, [KKP10] correspond à un article de PODC 2005.

[2] J'ai introduit le terme de « certification locale » dans ma thèse, pour englober les différentes variantes du concept.

1.3 Approfondissement de la notion

Dans un deuxième temps, j'ai étudié en profondeur la notion de certification locale. Un enjeu était d'explorer les liens entre différentes notions de localité. Inspiré par le test de propriété[3], j'ai proposé la notion de *sensibilité aux erreurs* : une nouvelle notion de certification où le nombre de nœuds qui rejettent doit être linéaire en le nombre de modifications à apporter à la solution pour la corriger. Cette sensibilité permet de détecter plus rapidement les configurations vraiment problématiques, et donc de lancer plus vite une procédure de secours. Dans ce travail avec Pierre Fraigniaud [FF17], j'ai notamment prouvé que pour l'arbre couvrant de poids minimum, on pouvait imposer cette propriété sans avoir à augmenter la taille des certificats. La preuve est technique et novatrice, car la certification utilise une hiérarchie d'arbres couvrants, qui rend complexe le suivi des erreurs détectées et des corrections possibles.

Un autre travail est inspiré de [OPR17] qui avait étudié les *compromis temps-espace en certification*. Par exemple, en autorisant les sommets à communiquer à distance t dans le réseau, on peut certifier un arbre couvrant avec $O(\log n/t)$ bits par sommet. Avec Juho Hirvonen, Pierre Fraigniaud, Ami Paz et Mor Perry, nous avons généralisé ces résultats à plus de propriétés, avec de meilleurs compromis, et en établissant des bornes inférieures [FFH+21b]. En particulier, j'ai montré que sur les cycles, les arbres et les grilles, on pouvait utiliser une méthode générique pour toutes les propriétés : donner des certificats qui ne concernent qu'une portion $1/t$ des nœuds et simuler les autres certificats au moment de la vérification. Cette technique a inspiré une généralisation dans un article récent [FOS21].

Au-delà de l'optimisation de la taille des certificats, ces résultats de compromis mettent en lumière une certaine redondance dans les certificats. Dans la lignée de cette observation, j'ai étudié avec Juho Hirvonen, une nouvelle notion de *certification globale*, avec un unique certificat, accessible à tous les nœuds [FH18]. L'un des résultats saillants de ce travail est une nouvelle technique de bornes inférieures, qui fonctionne à la fois pour les certifications locales et globales.

1.4 Localité des classes de graphes

Dans cette sous-section, je présente mes travaux sur la *certification de propriétés du réseau lui-même*, et non d'une solution calculée. Ce sujet est important pour deux raisons. D'abord, de nombreux algorithmes distribués ont été conçus pour des réseaux ayant des propriétés spécifiques[4], et l'on voudrait vérifier efficacement que le réseau à la bonne forme avant de lancer un tel algorithme. Ensuite, la taille des certificats peut être vue comme une mesure de la localité de la propriété certifiée (de petits certificats impliquant que la propriété n'est pas loin d'être vérifiable localement), et c'est donc un concept intéressant pour des propriétés de graphes.

Focus sur « Compact distributed certification of planar graphs »
Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Algorithmica, 83(7):2215–2244, 2021.
doi: 10.1007/s00453-021-00823-w. See also: conference version at PODC 2020.

[3] Le test de propriété est un modèle de calcul dans lequel, pour une propriété donnée, on fait des requêtes d'adjacence sur un graphe, et l'on doit décider si le graphe satisfait la propriété, ou s'il faudrait le modifier en profondeur pour qu'il satisfasse la propriété [Gol10].

[4] Par exemple, j'ai compilé une bibliographie annotée des algorithmes d'approximation distribués sur les graphes planaires, d'arboricité bornée, d'expansion bornée, etc. [Feu20b].

En 2020, en parallèle de mon travail avec José Correa, j'ai travaillé avec plusieurs collègues sur la *certification de la planarité du réseau*. Un article récent avait proposé une certification compacte des graphes planaires, mais dans le cadre plus large d'une hiérarchie Arthur-Merlin. Le but était de déterminer si l'on pouvait obtenir le même résultat dans le cadre plus restreint et réaliste de la certification locale. Une approche naturelle pour ce problème est de certifier les faces d'un plongement planaire, mais cela ne semblait pas fonctionner. Alors, inspiré par mes travaux avec Michel Habib (détaillés plus loin), j'ai proposé de s'intéresser d'abord aux graphes planaires extérieurs, qui ont une caractérisation reposant sur un ordre spécifique des sommets. Après avoir établi cette certification, nous avons pu généraliser aux planaires, en certifiant une décomposition des graphes planaires en planaires extérieurs [FFM$^+$+21]. Cette certification utilise $O(\log n)$ bits par sommet, ce qui est remarquable : c'est la taille nécessaire pour certifier un arbre. Cette taille est optimale, ce qui était attendu, mais a quand même nécessité que je réutilise la méthode de borne inférieure développée dans [FH18].

Nous avons ensuite généralisé le résultat aux graphes de genre borné, ce qui a demandé de transférer les outils de découpages de surfaces, depuis la géométrie algorithmique (centralisé) vers le calcul distribué [FFM$^+$20].

Développements. À ce stade une question importante se dessine : existe-t-il une certification compacte de toutes les classes de graphes closes par mineurs[5] ? Ici, « compacte » signifie « avec des certificats de taille $O(\log n)$ ». À l'heure actuelle, les seules propriétés que l'on sait prouver comme étant difficiles à certifier sont des propriétés non héréditaires, ce qui laisse espérer une réponse positive, mais cette question reste très largement ouverte.

Avec Théo Pierron et Nicolas Bousquet, spécialistes de théorie des graphes, nous avons posé un premier jalon, en montrant que la conjecture était vérifiée pour tous les mineurs en dessous d'une certaine taille [BFP21a]. Pour ce travail, l'enjeu essentiel était de certifier des décompositions basées sur la connectivité, intéressantes en elle-même.

Poser la question des mineurs consiste à essayer d'avoir des résultats portant sur une famille de propriétés, plutôt que sur une propriété spécifique. Dans cette direction, nous avons voulu obtenir un métathéorème (comme le théorème de Courcelle en calcul centralisé), de la forme : « sur les graphes de la forme X, toutes les propriétés du type Y peuvent être certifiées efficacement ». Précisément, nous avons établi que dans les graphes de treedepth bornée, on pouvait certifier toutes les formules MSO, avec des certificats de taille $O(\log n)$ [BFP21b]. C'est le premier résultat de ce type dans ce domaine.

1.5 Diffusion de la notion de certification locale

Je m'emploie à populariser et vulgariser la notion de certification locale. J'ai écrit un survey sur l'approche « théorie de la complexité » (avec Pierre Fraigniaud) [FF16], et une introduction au domaine, d'un point de vue historique et technique [Feu21]. J'ai aussi été invité à donner un « gem talk » sur la notion à PODC 2021, et un exposé sur la certification de classes de graphes au workshop ADGA à DISC 2021. Enfin, je publie régulièrement sur mon blog et sur arxiv des notes et explications sur des sujets spécifiques du domaine[6].

[5] La notion de mineurs est centrale en théorie des graphes, et les familles closes par mineurs sont la généralisation naturelle des arbres, et des graphes planaires et de genre borné.

[6] Voir par exemple [Feu19] et *Diameter lower bound in local certification* sur Discrete Notes.

2 Autres travaux en calcul distribué

Ma spécialité en calcul distribué est la certification locale, mais je participe plus généralement à la recherche en calcul distribué sur réseaux, présente dans les conférences PODC/DISC. Je vais évoquer deux thématiques sur lesquelles j'ai travaillé.

2.1 Autostabilisation

Un algorithme distribué est *autostabilisant* s'il peut converger vers une solution correcte en partant d'une configuration arbitraire, ce qui le rend tolérant aux fautes. On peut montrer que si l'on veut que l'algorithme converge vers une solution, puis conserve cette solution, alors l'algorithme doit utiliser, au moins implicitement, une certification locale [BFP14].

Quand j'ai abordé ce domaine en 2018, une question était ouverte depuis plusieurs années : est-il possible d'utiliser moins d'espace mémoire en autostabilisation si l'on s'autorise à donner une réponse non pas exacte, mais approchée ? En particulier, est-ce le cas pour le problème classique de l'arbre couvrant de poids minimum ? Avec Lélia Blin et Swan Dubois, nous avons montré que c'était le cas : nous avons établi un compromis entre l'espace mémoire et l'approximation pour ce problème [BDF20], en se basant sur une connaissance fine de la certification de [KK07], que j'avais détaillé dans [Feu19].

Un algorithme autostabilisant peut utiliser moins de mémoire que la certification optimale, si l'on s'autorise à être en permanence en train de recalculer la solution. Par exemple, le problème classique de l'élection d'un leader peut-être résolu avec $O(\log\log n)$ bits de mémoire, au lieu de $O(\log n)$ pour la certification. Avec Lélia Blin et Gabriel Le Bouder, nous avons montré que $O(\log\log n)$ bits était optimal en poussant les techniques de borne inférieure de [BGJ99], contredisant le folklore qui conjecturait une borne supérieure plus basse. Cet article [BFB21] a reçu le prix du meilleur article étudiant à OPODIS 2021.

2.2 Localité en moyenne

Pendant mon stage de master en 2014, en parallèle de mon travail avec Pierre Fraigniaud, j'ai introduit et étudié de manière autonome une notion de complexité en moyenne pour le calcul local. J'ai montré que les bornes inférieures classiques en $\Omega(\log^* n)$ [Lin92] étaient encore valides en moyenne pour le calcul déterministe, mais que si l'on utilisait en plus la randomisation, alors la complexité devenait constante [Feu20b]. Ce travail a trouvé deux échos dans la communauté. D'abord, plusieurs articles ont développé un modèle où les machines peuvent se mettre à l'arrêt temporairement, et dans ce contexte, la complexité en moyenne était pertinente et a été utilisée (*e.g.* [BT19, CGP20][7]). Ensuite, des connexions entre le calcul local et la combinatoire descriptive sont en train d'être construites, notamment pour les graphes infinis, où l'on doit utiliser la complexité moyenne et non le pire cas [GR21].

3 Algorithmique en ligne

Lors de mon deuxième postdoc, à l'Universidad de Chile, j'ai rejoint une équipe de recherche opérationnelle, spécialisée en théorie algorithmique des jeux. Dans ce contexte, j'ai travaillé sur un problème de décision en ligne, un modèle assez loin de travaux précédent, pour lequel mes outils de calcul distribué se sont révélés très utiles.

[7] Tzalik Maimon effectue actuellement une thèse avec Leonid Barenboim sur ce sujet.

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

Le *problème des secrétaires* est un problème classique de l'algorithmique en ligne [Fre83]. Étant donné une suite de nombres sélectionnés par un adversaire, présentés un par un dans un ordre aléatoire, il faut s'arrêter sur le maximum. On peut montrer qu'un algorithme, d'une famille appelée « algorithmes à seuil », résout le problème avec la probabilité optimale.

Inspiré par un article récent [KNR20], nous avons étudié une variante où *un échantillon des nombres est connu à l'avance*. Cet échantillon modélise l'information que l'on a pu accumuler dans le passé, et qui guide les choix présents. Pour cette variante, on peut considérer que l'ordre de présentation est aléatoire ou bien choisi par l'adversaire. Il est vite apparu que les techniques classiques permettaient de bien comprendre l'ordre aléatoire, mais pas l'ordre adversaire. Dans ce dernier cas, un algorithme à seuil semblait bien fonctionner, mais les méthodes classiques de borne inférieure ne s'adaptaient pas au nouveau contexte.

Grâce à une nouvelle technique, utilisant des intuitions venues du calcul distribué, j'ai pu établir les bornes inférieures que nous attendions. L'idée principale est de raisonner sur l'indistingabilité : montrer qu'un algorithme plus efficace qu'un certain seuil devrait nécessairement faire des choix différents dans des contextes où sa vue du système est identique, ce qui est impossible. Pour simplifier l'approche, j'ai d'abord supposé que l'algorithme ne connaissait pas la taille de la suite de nombres à l'avance (hypothèse classique en calcul distribué, en ce qui concerne la taille du système, mais peu commune en calcul en ligne)[8]. La preuve consiste ensuite à définir un graphe, appelé « graphe des conflits » en calcul distribué, tel que l'on puisse mesurer la performance optimale d'un algorithme comme la taille d'une structure dans le graphe (ici une sorte d'ensemble indépendant). Une étude combinatoire de ce graphe permet de montrer une borne sur la performance optimale. Cette borne est égale à la performance de l'algorithme à seuil de départ, ce qui clôt la question.

4 Théorie des graphes

Je vais maintenant décrire certains de mes travaux concernent la théorie structurelle des graphes. Comme décrit précédemment la connaissance des graphes est l'un de mes outils pour obtenir des résultats algorithmiques.

4.1 Classes et sous-graphes ordonnés interdits

Je vais parler ici de caractérisations par motifs interdits, et pour cela je vais commencer par un exemple. Les graphes d'intervalles forment une classe de graphes bien étudiée (voir *e.g.* [Gol85]), définie de manière géométrique. Ils peuvent être caractérisés par l'existence d'un ordre sur les sommets, tel qu'il n'existe pas trois sommets $a < b < c$, tel que (a, c) est une arête, mais pas (a, b). J'appelle ce genre de résultat une *caractérisation par motif interdit*. Une telle caractérisation est apparue naturellement lors de mon stage de 2013, en optimisation combinatoire et recherche opérationnelle, et une fois arrivé en thèse, j'ai proposé à Michel Habib de travailler sur ce sujet, ce qui a abouti à l'article suivant.

[8] Dans un deuxième temps, on peut renforcer la preuve pour se passer de cette hypothèse.

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

La littérature ne contient que quelques publications relativement méconnues sur les motifs interdits [Dam90, HMR14]. Avec Michel Habib nous avons fixé un objectif ambitieux : classifier toutes les classes définies par des motifs à trois sommets. C'est ce que nous avons fait dans [FH21a], avec l'aide d'un petit code pour réduire la liste des configurations à étudier. De cette classification ressortent deux messages importants. Premièrement, les classes ainsi décrites sont quasiment toutes des classes connues (graphes cordaux, d'intervalles, split, bipartis, etc.) Ce type de caractérisation est donc pertinent, et de plus il d'avoir des caractérisations compactes et des preuves d'inclusions standardisées (et non ad hoc comme dans la littérature). Deuxièmement, ces classes peuvent presque toutes être reconnues en temps linéaire, ce point de vue est donc aussi pertinent d'un point de vue algorithmique.

Développements Plus récemment, nous nous sommes tournés vers les motifs à quatre sommets. Ce sujet n'est pas un prolongement anecdotique du cas précédent : plusieurs classes importantes (par exemple, les graphes 3-colorables et planaires extérieurs) ont une caractérisation avec un motif à quatre sommets. Mais le paysage est beaucoup plus complexe : la reconnaissance de certaines classes est NP-complète et les classes sont beaucoup plus nombreuses. Nous avons montré une connexion forte entre motifs à quatre sommets et géométrie : les hiérarchies de classes définies par des motifs, et celle définies par des intersections géométriques (comme les graphes outerstring, très étudiés dans le domaine) sont très proches [FH21b]. Ce travail améliore notre connaissance des caractérisations par motif, mais apporte aussi une dimension combinatoire à des classes géométriques, parfois difficile à manipuler. Ainsi nous avons pu établir des séparations entre des classes en énumérant des ordres de sommets, ce qui est beaucoup plus facile que d'énumérer des géométries.

4.2 Reconfiguration

Mon postdoc actuel est financé par une ANR portant sur la reconfiguration, une thématique de théorie des graphes très liée à la localité. La reconfiguration est l'étude des modifications d'une structure combinatoire en une autre. Concrètement, une question classique est : peut-on passer d'une k-coloration d'un graphe à une autre, en modifiant les couleurs une à une, en conservant une coloration propre, et si oui, quel est le chemin le plus court ?

Sur ce sujet, j'ai participé à trois travaux, l'un publié et deux en rédaction. Avec Nicolas Bousquet, Marc Heinrich et Mikael Rabie [BFHR21], nous avons montré que l'on pouvait améliorer l'état de l'art en recoloration distribuée (où les sommets du graphe doivent calculer eux-mêmes la série de recolorations, en minimisant la communication) sur les réseaux bien structurés (graphes cordaux et d'intervalles). Avec Vladyslav Hlembotskyi et Konrad K. Dabrowski, nous avons montré une connexion inattendue entre la reconfiguration d'indépendants et la notion de primalité de graphe, qui est issue d'un sous-domaine très différent de la théorie des graphes [DFH21]. Enfin, avec Nicolas Bousquet, Valentin Gledel, Marc Heinrich et Mikaël Rabie, nous avons étudié une question fondamentale en recoloration : le nombre maximum de pas nécessaires entre deux $\Delta + 1$ colorations, dans les graphes de degré maximum Δ. Un résultat aujourd'hui classique du domaine montre que ce nombre de pas est au plus quadratique, et nous avons montré que l'on pouvait baisser cette borne à être linéaire, en utilisant des intuitions du calcul distribué [BFG$^+$22].

Note : Seuls les travaux mentionnés dans ce rapport 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

BFG$^+$22 Nicolas Bousquet, Laurent Feuilloley, Valentin Gledel, Marc Heinrich, and Mikaël Rabie. Short and local transformations between $(\Delta+1)$-colorings, 2022.

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

DFH21 Konrad K. Dabrowski, Laurent Feuilloley, and Vladyslav Hlembotskyi. 2-independent set reconfiguration and primality, 2021.

Feu19 Laurent Feuilloley. Note on distributed certification of minimum spanning trees, 2019. arxiv: 1909.07251.

Feu20b Laurent Feuilloley. Bibliography of distributed approximation beyond bounded degree, 2020. arxiv: 2001.08510.

FFM$^+$20 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Eric Rémila, and Ioan Todinca. Local certification of graphs with bounded genus, 2020. arxiv:2007.08084.

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

Références personnelles publiées

BDF20 Lélia Blin, Swan Dubois, and Laurent Feuilloley. Silent MST approximation for tiny memory. In *Stabilization, Safety, and Security of Distributed Systems - 22nd International Symposium, SSS 2020*. 2020. doi:10.1007/978-3-030-64348-5_10.

BFB21 Lélia Blin, Laurent Feuilloley, and Gabriel Le Bouder. Optimal space lower bound for deterministic self-stabilizing leader election algorithms. In *25th International Conference on Principles of Distributed Systems, OPODIS 2021*, 2021. arxiv: 1905.08563.

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.

Feu20b Laurent Feuilloley. How long it takes for an ordinary node with an ordinary id to output? *Theor. Comput. Sci.*, 811:42–55, 2020. doi: 10.1016/j.tcs.2019.01.023. Article à SIROCCO 2017 invité à la special issue.

Feu21 Laurent Feuilloley. Introduction to local certification. *Discret. Math. Theor. Comput. Sci.*, 23(3), 2021. arxiv: 1910.12747.

FF16 Laurent Feuilloley and Pierre Fraigniaud. Survey of distributed decision. *Bulletin of the EATCS*, 119, 2016. url: bulletin.eatcs.org link, arXiv: 1606.04434.

FF17 Laurent Feuilloley and Pierre Fraigniaud. Error-sensitive proof-labeling schemes. In *31st International Symposium on Distributed Computing, DISC 2017*, pages 16:1–16:15, 2017. doi:LIPIcs.DISC.2017.16.

FF21 Laurent Feuilloley and Pierre Fraigniaud. Randomized local network computing: Derandomization beyond locally checkable labelings. *ACM Transactions on Parallel Computing*, 8, 2021. doi:10.1145/3470640. Voir aussi la version conférence à SPAA 2015.

FFH21a Laurent Feuilloley, Pierre Fraigniaud, and Juho Hirvonen. A hierarchy of local decision. *Theor. Comput. Sci.*, 856:51–67, 2021. 10.1016/j.tcs.2020.12.017. Voir aussi la version conférence à ICALP 2016.

FFH+21b Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Ami Paz, and Mor Perry. Redundancy in distributed proofs. *Distributed Comput.*, 34(2):113–132, 2021. doi: 10.1007/s00446-020-00386-z. Voir aussi la version conférence à DISC 2018.

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.

FH18 Laurent Feuilloley and Juho Hirvonen. Local verification of global proofs. In *32nd International Symposium on Distributed Computing, DISC 2018*, volume 121 of *LIPIcs*, pages 25:1–25:17, 2018. doi:10.4230/LIPIcs.DISC.2018.25.

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.

Autres références

AKY90 Yehuda Afek, Shay Kutten, and Moti Yung. Memory-efficient self stabilizing protocols for general networks. In Jan van Leeuwen and Nicola Santoro, editors, *Distributed Algorithms, 4th International Workshop, WDAG '90*, volume 486, pages 15–28, 1990. doi:10.1007/3-540-54099-7_2.

BT19 Leonid Barenboim and Yaniv Tzur. Distributed symmetry-breaking with improved vertex-averaged complexity. In *Proceedings of the 20th International Conference on Distributed Computing and Networking, ICDCN 2019*, pages 31–40, 2019. doi:10.1145/3288599.3288601.

BFP14 Lélia Blin, Pierre Fraigniaud, and Boaz Patt-Shamir. On proof-labeling schemes versus silent self-stabilizing algorithms. In *16th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 18–32, 2014. doi:10.1007/978-3-319-11764-5_2.

BGJ99 Joffroy Beauquier, Maria Gradinariu, and Colette Johnen. Memory space requirements for self-stabilizing leader election protocols. In Brian A. Coan and Jennifer L. Welch, editors, *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing, PODC, '99*, pages 199–207, 1999. doi:10.1145/301308.301358.

CGP20 Soumyottam Chatterjee, Robert Gmyr, and Gopal Pandurangan. Sleeping is efficient: MIS in $O(1)$-rounds node-averaged awake complexity. In *PODC '20: ACM Symposium on Principles of Distributed Computing*, pages 99–108, 2020. doi:10.1145/3382734.3405718.

CPP20 Keren Censor-Hillel, Ami Paz, and Mor Perry. Approximate proof-labeling schemes. *Theor. Comput. Sci.*, 811:112–124, 2020. doi:10.1016/j.tcs.2018.08.020.

Dam90 Peter Damaschke. Forbidden ordered subgraphs. *Topics in Combinatorics and Graph Theory, R. Bodendiek and R. Hennm Eds*, pages 219–229, 1990.

FOS21 Orr Fischer, Rotem Oshman, and Dana Shamir. Explicit space-time tradeoffs for proof labeling schemes in graphs with small separators. In *25th International Conference on Principles of Distributed Systems, OPODIS 2021*, 2021.

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

Fre83 P.R. Freeman. The secretary problem and its extensions: A review. *International Statistical Review/Revue Internationale de Statistique*, pages 189–206, 1983.

Gol85 Martin Charles Golumbic. Interval graphs and related topics. *Discret. Math.*, 55(2):113–121, 1985. doi: 10.1016/0012-365X(85)90039-1.

Gol10 Oded Goldreich. Introduction to testing graph properties. In *Property Testing - Current Research and Surveys*, volume 6390 of *Lecture Notes in Computer Science*, pages 105–141. Springer, 2010. doi:10.1007/978-3-642-16367-8_7.

GR21 Jan Grebík and Václav Rozhoň. Local problems on grids from the perspective of distributed algorithms, finitary factors, and descriptive combinatorics, 2021. arxiv: 2103.08394.

GS16 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(19):1–33, 2016.

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.

KOS18 Gillat Kol, Rotem Oshman, and Raghuvansh R. Saxena. Interactive distributed proofs. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018*, pages 255–264. ACM, 2018. acm:3212771.

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

KKP10 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010.

KNR20 Haim Kaplan, David Naori, and Danny Raz. Competitive analysis with a sample and the secretary problem. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 2082–2095, 2020. doi:10.1137/1.9781611975994.128.

Lin92 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

MMNS11 Ross M. McConnell, Kurt Mehlhorn, Stefan Näher, and Pascal Schweitzer. Certifying algorithms. *Comput. Sci. Rev.*, 5(2):119–161, 2011. doi:10.1016/j.cosrev.2010.09.009.

NPY20 Moni Naor, Merav Parter, and Eylon Yogev. The power of distributed verifiers in interactive proofs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1096–115. SIAM, 2020. doi:10.1137/1.9781611975994.67.

NS95 Moni Naor and Larry J. Stockmeyer. What can be computed locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995.

OPR17 Rafail Ostrovsky, Mor Perry, and Will Rosenbaum. Space-time tradeoffs for distributed verification. In *Structural Information and Communication Complexity - 24th International Colloquium, SIROCCO 2017 Revised Selected Papers*, pages 53–70, 2017.

Pel00 David Peleg. *Distributed Computing: A Locality-Sensitive Approach.* SIAM, 2000.