

Projet Honor : Limites du calcul catalytique

Pierre McKenzie

Janvier 2025

1 Background

Burhman *et al* [BCK⁺14] ont introduit en 2014 un modèle de calcul qui formalise l'idée suivante : est-ce qu'un ordinateur peut réduire son usage de mémoire interne si cet ordinateur peut accéder à une mémoire externe (dite “catalytique”) gigantesque mais déjà remplie de données qu'il faut remettre en l'état à la fin d'un calcul ? Les auteurs ont démontré étonnamment que, conditionnellement à la véracité de conjectures de longue date en théorie de la complexité, la réponse est affirmative.

De nombreux travaux théoriques ont suivi, explorant la puissance du calcul *catalytique* :

1. Cook et Mertz [CM20] en 2020 ont déboulonné une conjecture de (Stephen) Cook *et al* [CMW⁺12] concernant la complexité en espace du problème d'évaluation d'arbre.
2. Cook et Mertz ont amélioré leur technique presque optimalement [CM22, CM24].
3. Près d'une vingtaine de publications (voir [FMST24, Page 2]) faisant état d'applications à la théorie de la complexité ont vu le jour.

En parallèle, James Cook [Coo21] implantait en langage C, en guise de preuve de concept, un algorithme catalytique réduisant l'espace-mémoire requis pour déterminer l'accessibilité d'un sommet à un autre dans un graphe orienté.

2 Description du projet

2.1 Composante théorique

Jusqu'où peut aller l'économie d'espace rendue possible par le calcul catalytique ? Voilà bien sûr la question à laquelle se sont intéressés tous les travaux mentionnés ci-dessus. Ici, il s'agit d'investiguer un cas particulier d'amélioration possible : celui des programmes de branchement résolvant un problème P-complet appelé GEN. Comme tout problème P-complet, GEN peut être résolu en petit espace (lire : espace logarithmique) si et seulement si tous les problèmes de la classe P le peuvent. *Petit espace* pour une machine de Turing correspond à *taille polynomiale* pour le modèle de calcul du programme de branchement. Or des bornes inférieures exponentielles sur la taille de programmes de branchement résolvant

GEN sont connues, en autant que ces programmes soient restreints à posséder une propriété que l'on a nommée l'*incrémentalité* [GKM08].

La composante théorique du projet explorera la possibilité d'améliorer les bornes connues sur la taille de programmes de branchement résolvant GEN à l'aide du calcul catalytique. Une amélioration asymptotique est peu envisageable puisqu'elle aurait comme conséquence l'égalité de classes de complexité que l'on croit distinctes depuis 50 ans. Le but est plutôt de chercher à améliorer, même par un facteur constant, les bornes supérieures connues : est-ce que le calcul catalytique est d'une quelconque aide pour faire mieux à l'aide de programmes de branchements sans restriction que nous ne savons le faire à l'aide de programmes de branchement incrémentaux ? Si une amélioration ne semble pas possible, quelles sont précisément les obstacles rencontrés ?

Le projet nécessitera un survol des articles mentionnés et une étude plus approfondie de certains d'entre eux, dont certainement [CM24].

2.2 Composante appliquée

Faisant suite à l'implantation en langage C par James Cook [Coo21] de la méthode catalytique d'origine [CM20] appliquée au problème de l'accessibilité dans un graphe, il s'agira d'implanter la méthode récente plus élaborée [CM24] à l'encontre du problème de l'évaluation d'un arbre, dont la complexité d'espace connue est $O((\log n)^2)$. Selon les difficultés rencontrées, l'implantation pourra être étendue à la résolution du problème GEN, dont on pense la complexité d'espace être d'ordre polynomial.

Références

- [BCK⁺14] Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory : catalytic space. In *Symposium on Theory of Computing, STOC 2014*, pages 857–866, 2014.
- [CM20] James Cook and Ian Mertz. Catalytic approaches to the tree evaluation problem. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 752–760. ACM, 2020.
- [CM22] James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 8 :1–8 :21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [CM24] James Cook and Ian Mertz. Tree evaluation is in space $o(\log n \cdot \log \log n)$. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1268–1278. ACM, 2024.
- [CMW⁺12] S. Cook, P. McKenzie, D. Wehr, M. Braverman, and R. Santhanam. Pebbles and branching programs for tree evaluation. *ACM TOCT*, 3(2) :4, 2012.

- [Coo74] S. Cook. An observation on time-storage trade off. *J. Comput. Syst. Sci.*, 9(3) :308–316, 1974.
- [Coo21] James Cook. How to borrow memory. <https://www.falsifian.org/blog/2021/06/04/catalytic/>, janvier 2021.
- [FMST24] Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker. Fully characterizing lossy catalytic computation. *CoRR*, abs/2409.05046, 2024.
- [GKM08] A. Gál, M. Koucký, and P. McKenzie. Incremental branching programs. *Theory Comput. Syst.*, 43(2) :159–184, 2008.