HAL
open science

Complexité en moyenne pour le modèle LOCAL

Laurent Feuilloley

To cite this version:

Laurent Feuilloley. Complexité en moyenne pour le modèle LOCAL. ALGOTEL 2015 - 17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2015, Beaune, France. hal-01148513

HAL Id: hal-01148513

https://hal.science/hal-01148513v1

Submitted on 4 May 2015

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

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

Complexité en moyenne pour le modèle LOCAL

Laurent Feuilloley[1†]

[1] *Département informatique de l'École Normale Supérieure de Cachan et LIAFA à l'université Paris Diderot.*

Un modèle classique en calcul distribué synchrone est le modèle LOCAL. Dans ce modèle, les processeurs travaillent par rondes successives. Pour la plupart des algorithmes, on suppose que les processeurs ont accès au nombre de sommets du réseau, noté n, ce qui leur permet de calculer le nombre de rondes après lequel ils doivent tous stopper, et donner une sortie. Il a été montré récemment que, pour nombre de problèmes classiques, il est possible de se passer de la connaissance de n [2, 4]. Dans ce contexte, les sommets peuvent choisir leurs sorties à des rondes différentes, mais continuent à transmettre des messages. Avec ou sans connaissance de n, la mesure du temps de calcul est toujours le nombre de rondes avant que le dernier nœud ne donne une sortie. Dans cet article, on s'intéresse à une nouvelle mesure : la moyenne (sur les nœuds) du nombre de rondes avant d'annoncer une sortie. On montre que la complexité d'un problème peut être exponentiellement plus petite avec cette nouvelle mesure, mais que la borne inférieure de Linial pour la coloration [3] est encore valide.

Keywords: Modèle LOCAL, Coloration, Complexité en moyenne

1 Introduction

1.1 Modèle LOCAL et mesures de complexité

Dans le modèle LOCAL, les processeurs sont les nœuds du réseau. Ils ont des identifiant,s distincts deux à deux, et fonctionnent par rondes successives. À chaque ronde, chaque processeur envoie des messages à ses voisins directs, reçoit des messages, et calcule son nouvel état. La puissance de calcul des processeurs et la taille des messages ne sont pas limitées. Dans la version classique de ce modèle, tous les sommets stoppent et choisissent une sortie à la même ronde. La mesure du temps de calcul est alors le nombre de rondes avant l'arrêt global du calcul. Dans certaines variantes, par exemple quand le nombre de sommets n n'est pas connu, les nœuds peuvent choisir leurs sorties à des rondes différentes, mais ils continuent à transmettre des messages [2, 4]. On peut définir le temps de calcul comme le numéro de la première ronde à laquelle tous les nœuds ont choisi une sortie. Dans ce texte, on s'intéresse à une autre mesure : la moyenne, sur les nœuds, du nombre de rondes avant de choisir une sortie.

1.2 Formalisation

Pour formaliser les notions en jeu, on utilise un point de vue alternatif sur le modèle LOCAL, qui consiste à parler de « vues » plutôt que de rondes. Le modèle peut être défini de manière équivalente, en disant que chaque nœud récupère d'abord une « vue » à une certaine distance (c'est-à-dire qu'il rassemble toute l'information contenue dans un certain voisinage : la structure de ce voisinages et les identifiants), puis calcule une sortie en fonction de ce voisinage.

On commence par formaliser la notion de vue. Une vue est un triplet (H,v,k), où H est un graphe connexe pourvu d'identifiants distincts, v un sommet de H, et k un entier. Une vue doit satisfaire la condition suivante : pour toute arête (u,w) de H, $\min(d(v,u),d(v,w)) \leq k-1$. Ceci indique que le graphe centré sur v est de rayon au plus k, et qu'il n'y a pas d'arêtes entre les sommets à distance k de v. On parle de « vue à distance k ». Par exemple, les vues à distance 0 sont réduites à un sommet unique, et les vues à distance 1

†Supporté par le projet ANR DISPLEXITY

sont des étoiles. On note $\mathcal{T}$, l'ensemble des vues, et on définit une relation d'ordre partielle sur cet ensemble, notée $\preceq$. Soit deux vues $T_1 = (H_1, v_1, k_1)$ et $T_2 = (H_2, v_2, k_2)$, on note $T_1 \preceq T_2$, si $k_1 \leq k_2$, et si (H_2, v_2) est une extension de (H_1, v_1), c'est-à-dire qu'il est possible d'obtenir H_2 à partir de H_1, en identifiant v_1 à v_2, et en rajoutant des sommets à distance au moins $k_1 + 1$ de v, et des arêtes entre sommets à distance au moins k_1 de v. On considère en particulier les suites de vues du type $T_i = (H_i, v, k_i = i)$, telles que pour tout i, $T_i \preceq T_{i+1}$, et telles qu'il existe un entier j avec pour tout $i > j$, $H_i = H_j$. Cette dernière condition exprime le fait que l'on se place dans un graphe fini, et qu'à partir d'un certain rayon, la vue contient tout le graphe.

On formalise maintenant la notion d'algorithme. Soit un problème dans lequel les nœuds doivent choisir un label dans un alphabet Σ. Une fonction $f : \mathcal{T} \mapsto \Sigma \cup \varepsilon$, représente un algorithme si, pour toute suite de configurations du type décrit ci-dessus, $T_0 \preceq T_1 \preceq T_2 ...$, il existe un entier p, et un élément σ de Σ, tels que la condition suivante est satisfaite. Pour tout entier i, strictement inférieur à p, $f(T_i) = \varepsilon$, et pour tout i supérieur ou égal à p, $f(T_i) = \sigma$. L'algorithme se déduit simplement de f : commencer avec la vue à distance 0, et tant que f renvoie ε pour la vue actuelle, augmenter le rayon ; puis, arrivé au p correspondant à cette suite de vues, renvoyer σ. L'entier p est appelé rayon de v, et est noté $r(v)$. La complexité classique (*i.e.* la complexité du pire nœud) et la complexité en moyenne, pour une taille n, sont alors respectivement :

$$\max_{G:|G|=n} \left(\max_{v \in G} r(v) \right) \qquad \text{et} \qquad \max_{G:|G|=n} \left(\sum_{v \in G} r(v)/n \right).$$

1.3 Résultats

La complexité en moyenne est toujours inférieure à la complexité du pire nœud, et l'on montre qu'il existe un problème naturel pour lequel la première est exponentiellement plus petite que la seconde. Dans ce problème, appelé *largest ID*, chaque nœud d'un cycle doit décider s'il a le plus grand identifiant du graphe ou non, ce qui est une technique classique pour élire un leader. A contrario, on montre que la borne inférieure de Linial sur la coloration des cycles [3] tient toujours avec la mesure moyenne, c'est-à-dire que le rayon moyen nécessaire pour calculer une 3-coloration est $\Omega(\log^* n)$. Notons que cette borne inférieure coïncide asymptotiquement avec la borne supérieure puisqu'il est possible de colorer le cycle de n nœuds en temps $O(\log^* n)$, en se passant de la connaissance de n [2][4].

Le but principal de ce travail est de poursuivre l'étude de la localité, en proposant une mesure légèrement différente de la mesure étudiée habituellement. On donne de plus deux exemples d'applications envisageables. Premièrement, dans le contexte des graphes dynamiques, la complexité en moyenne permet de mieux estimer le nombre de nœuds à mettre à jour après le changement d'un nœud aléatoire, que la mesure du pire nœud. Deuxièmement, dans le contexte du calcul parallèle simulant un calcul distribué, on peut tirer avantage du fait qu'une tâche soit terminée en avance pour en commencer une autre plus tôt ; le temps de calcul final est alors relié à la complexité en moyenne.

2 Algorithme pour le problème *largest ID* dans un cycle

Dans le problème *Largest ID*, le graphe est un cycle, et tout sommet doit sortir : *Oui* s'il a le plus grand identifiant du graphe, et *Non* si c'est n'est pas le cas. La complexité classique du problème est de l'ordre de n, car, quelque soit l'algorithme, le sommet d'identifiant maximum doit voir tout le cycle. Un algorithme naturel pour ce problème est le suivant : chaque sommet augmente son rayon jusqu'à trouver un identifiant plus grand que le sien (il sort alors *Non*), ou bien jusqu'à avoir vu tout le graphe sans voir de plus grand identifiant (il sort *Oui*). On montre que la complexité en moyenne de cet algorithme est logarithmique en n.

Le sommet ayant le plus grand identifiant a besoin d'environ $n/2$ rondes pour découvrir qu'il doit sortir *Oui*. On peut considérer que les autres sommets sont sur un chemin, et que « voir » un bout du chemin est suffisant pour stopper et sortir *Non*. Le rayon d'un sommet de ce chemin est alors la distance minimum à un bout ou à un sommet d'identifiant supérieur. On divise le chemin en trois : le sommet d'identifiant maximum du chemin (qui doit atteindre un bout), et les identifiants de chaque coté de ce sommet. Notons $a(p)$, la somme maximum des rayons dans un chemin de p sommets. La relation de récurrence suivante est vérifiée :

$$a(p) = \max_{1 \leq k \leq \lceil p/2 \rceil} \{k + a(k-1) + a(p-k)\}.$$

En particulier, on peut se restreindre à $1 \leq k \leq \lceil p/2 \rceil$ par symétrie. La suite $a(p)$, avec valeurs initiales $a(0) = 0$ et $a(1) = 1$, est dans $\theta(n \ln(n))$[5]. En conséquence, la complexité en moyenne de l'algorithme est logarithmique en n, ce qui est exponentiellement plus petit que la complexité classique du problème.

3 Borne inférieure pour la coloration

Un approche naïve pour la 3-coloration d'un cycle, consiste à essayer d'établir peu à peu des règles de coloration cohérentes. Par exemple, si l'identifiant du sommet considéré est pair, et que ses deux voisins sont impairs, alors prendre la couleur 1 ; si le sommet est impair et que tous ses voisins à distance au plus deux sont impairs, alors couleur 2, et ainsi de suite. Le tout en essayant de traiter tous les cas, en augmentant le moins possible le rayon, et en conservant une bonne coloration pour toute distribution des identifiants. Il se trouve que l'algorithme le plus connu pour la coloration d'un cycle, l'algorithme de Cole et Vishkin [1], utilise une toute autre approche, et fait prendre le même rayon pour tous les nœuds. Cet algorithme est optimal pour la mesure classique. En effet, il a une complexité de $O(\log^* n)$, et la borne inférieure de Linial dit que ce problème a une complexité de $\Omega(\log^* n)$[3]. Cependant, rien n'empêche, a priori, de reprendre l'approche naïve pour optimiser la complexité en moyenne, en espérant obtenir une constante. On montre dans cette section que cela est vain : la borne inférieure de Linial sur la coloration [3] est encore vérifiée si l'on considère la complexité en moyenne au lieu du pire nœud.

Theorem 1. *La complexité en moyenne de la 3-coloration d'un cycle de n nœuds est* $\Omega(\log^* n)$.

On dira qu'un algorithme A pour la 3-coloration est minimal, s'il n'existe pas d'autre algorithme A' strictement plus efficace. Plus précisément, A est minimal s'il n'existe pas A', utilisant au plus le même rayon sur tous les voisinages, et un rayon strictement plus petit pour au moins un voisinage. Sans perte de généralité, on prouve le théorème seulement pour les algorithmes minimaux. On commence par montrer deux lemmes sur la régularité de la distribution des rayons. Pour des raisons techniques, le premier lemme porte sur la 4-coloration, mais on revient à une 3-coloration dans le second.

Lemma 2. *Dans un cycle G, pourvu d'identifiants, étant donné deux sommets x et y séparés par k sommets, si un algorithme A est minimal pour la 4-coloration, alors les rayons des sommets situés entre x et y valent au plus* $\max\{r(x), r(y)\} + k$.

Démonstration. Par l'absurde, supposons qu'il existe G, x, y, k et A, comme dans le lemme, et un sommet entre x et y, dont le rayons est strictement plus grand que le seuil $\max\{r(x), r(y)\} + k$. Soit N le chemin comprenant les sommets entre x et y, le $r(x)$-voisinage de x, et le $r(y)$ voisinage de y. On montre comment transformer A en un algorithme strictement plus petit A', en réduisant les rayons de certains sommets de N.

Pour tout sommet v, si, en suivant l'algorithme A, le sommet découvre qu'il n'est pas dans N entre x et y, ou bien s'il stoppe avant d'arriver au seuil $\max\{r(x), r(y)\} + k$, alors A' lui fait faire exactement la même chose (même rayon et même sortie). Sinon, A' le fait stopper à $\max\{r(x), r(y)\} + k$, et lui fait choisir une sortie qui satisfait les deux règles suivantes (sans perte de généralité on suppose que $ID(x) > ID(y)$, et on note d la distance de x à v). Premièrement, si un voisin direct u, a stoppé strictement avant le seuil, alors v peut calculer la couleur de u, et doit en choisir une différente. Deuxièmement, si un voisin n'a pas stoppé avant le seuil, alors : si d est paire, v doit sortir une couleur de $\{1, 2\}$, sinon v doit sortir une couleur dans $\{3, 4\}$.

On peut vérifier que l'on peut toujours appliquer les règles précédentes, et qu'elles permettent d'obtenir une 4-coloration valide pour tout cycle et toute distribution des identifiants. De plus, ce nouvel algorithme A' est strictement plus petit que A puisque aucun rayon n'a été augmenté et que les rayons qui dépassaient le seuil entre x et y ont été tronqués. Il y a contradiction avec la minimalité de A, ce qui montre le lemme. □

On utilise le lemme 2 pour montrer le résultat suivant, plus pratique pour la suite de la preuve. Il indique qu'en moyenne, au voisinage d'un sommet de rayon r, les rayons sont au moins de l'ordre de r.

Lemma 3. *Soit un sommet v, utilisant un rayon r, et soit* $B(v, r/2)$*, l'ensemble des sommets à distance au plus* $\lfloor r/2 \rfloor$ *de v (y compris v). Dans un algorithme de 3-coloration minimal A, la moyenne des rayons des sommets de* $B(v, r/2)$ *est* $\Omega(r)$.

Démonstration. Remarquons d'abord que prouver le résultat pour la 4-coloration est suffisant, puisqu'il implique le résultat pour la 3-coloration. Soit v et r comme dans le lemme, et soit d un entier dans $\{1,2,...,\lfloor r/2 \rfloor\}$. Soit u_d et w_d, les deux sommets à distance d de v, sur le cycle. Le lemme 2 implique que $r \leq \max\{r(u_d), r(w_d)\} + 2d - 1$, c'est-à-dire, $r - 2d + 1 \leq r(u_d) + r(w_d)$. Alors en sommant sur d, on obtient :

$$\sum_{d=1}^{\lfloor r/2 \rfloor} (r - 2d + 1) \leq \sum_{d=1}^{\lfloor r/2 \rfloor} r(u_d) + r(w_d).$$

Si on ajoute r de chaque coté de l'équation, le terme de droite est la somme des rayons des sommets de $B(v, r/2)$, et le terme de gauche est quadratique en r. Le lemme est donc prouvé. □

Pour la fin de la preuve du théorème 1, on utilise le corollaire suivant de la borne de Linial. Pour tout algorithme qui 3-colore un cycle de longueur supérieure à $n/2$, il existe une distribution d'identifiants pour laquelle, au moins un sommet a un rayon supérieur à $\frac{1}{2}.\log^*(n/2)$. On montre qu'étant donné un algorithme minimal A, on peut construire une distribution d'identifiants π, donnant un rayon moyen de $\Omega(\log^* n)$. Premièrement, considérons un n-cycle et une distribution des identifiants, tels qu'un sommet v_1 a un rayon supérieur ou égal à $\frac{1}{2}.\log^*(n/2)$. On extrait les identifiants de la boule de rayon $\frac{1}{2}.\log^*(n/2)$ autour de v, et on les place au début de π. On considère désormais le reste des identifiants. On peut de nouveau trouver une distribution, avec un sommet v_2 de rayon supérieur à $\frac{1}{2}.\log^*(n/2)$. On extrait le $\frac{1}{2}.\log^*(n/2)$-voisinage autour de v_2, et on le concatène aux identifiants déjà présents dans π. On répète l'opération jusqu'à ce que le nombre d'identifiants dans π soit plus grand que $n/2$. On complète alors π, en rajoutant dans un ordre arbitraire les identifiants restants.

Les sommets v_i, qui avaient de grands rayons, ont le même $\frac{1}{2}\log^*(n/2)$-voisinage que dans les cycles dont ils ont été retirés. Avec le même algorithme A, ils ont donc des rayons au moins $\frac{1}{2}\log^*(n/2)$ dans π. D'après le lemme 3, la moyenne des rayons dans chaque voisinage est dans $\Omega(\log^*(n))$. Alors la moyenne des rayons de π est $\Omega(\log^*(n))$. Ceci conclue la preuve du théorème.

Conclusions et perspectives

Dans cet article, on a présenté une mesure alternative de la localité. Cette mesure diffère grandement de la mesure classique sur certains problèmes, mais est essentiellement égale pour d'autres. Il serait intéressant d'obtenir une caractérisation des problèmes du premier et du second type. De plus, on a seulement considéré des cycles, et le comportement sur des graphes quelconques reste encore à déterminer. Enfin, un autre axe de recherche serait de considérer des graphes dont les identifiants sont assignés selon une permutation aléatoire uniforme, et de voir comment se comportent les deux mesures dans ce cas.

Remerciements

Je remercie Pierre Fraigniaud, Juho Hirvonen, Tuomo Lempiäinen and Jukka Suomela pour des discussions intéressantes et stimulantes, et les relecteurs pour des remarques pertinentes.

Références

[1] Richard Cole and Uzi Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1) :32–53, 1986.

[2] Amos Korman, Jean-Sébastien Sereni, and Laurent Viennot. Toward more localized local algorithms : removing assumptions concerning global knowledge. *Distributed Computing*, 26(5-6) :289–308, 2013.

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

[4] Topi Musto. Knowledge of degree bounds in local algorithms. Master's thesis, University of Helsinki, 2011.

[5] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. A000788.