

Exploration des graphes dynamiques T -intervalle-connexes : le cas de l’anneau[†]

David Ilcinkas et Ahmed Mouhamadou Wade

LaBRI, CNRS & Université de Bordeaux, France

Dans cet article, nous étudions les graphes dynamiques T -intervalle-connexes du point de vue du temps nécessaire à leur exploration par une entité mobile (agent). Un graphe dynamique est T -intervalle-connexe ($T \geq 1$) si pour chaque fenêtre de T unités de temps, il existe un sous-graphe couvrant connexe stable. Cette propriété de stabilité de connexion au cours du temps a été introduite par Kuhn, Lynch et Oshman [6] (STOC 2010). Nous nous concentrons sur le cas où le graphe sous-jacent est un anneau de taille n et nous montrons que la complexité en temps en pire cas est de $2n - T - \Theta(1)$ unités de temps si l’agent connaît la dynamique du graphe, et $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ unités de temps sinon, où δ est le temps maximum entre deux apparitions successives d’une arête.

Keywords: Graphe dynamique, exploration, T -intervalle-connexité, agent mobile

1 Introduction

Entre autres dû à la très forte augmentation du nombre d’objets communicants que l’on observe aujourd’hui, les systèmes de calcul distribué deviennent de plus en plus dynamiques. Les modèles de calcul pour les réseaux statiques ne suffisent clairement plus à capturer le fonctionnement de ces nouveaux réseaux de communication. En fait, même les modèles de calcul prenant en compte une certaine tolérance aux fautes deviennent insuffisants pour certains réseaux très dynamiques. En effet, les modèles classiques de tolérance aux fautes supposent soit que la fréquence des fautes est faible, ce qui donne le temps à l’algorithme de s’adapter aux changements, soit que le système se stabilise au bout d’un certain temps (comme dans les systèmes auto-stabilisants par exemple). De ce fait, depuis une décennie environ, ont été développés de nombreux modèles plus ou moins équivalents qui prennent en compte l’extrême dynamisme de certains réseaux de communication. Le lecteur intéressé trouvera dans [1] un survol très complet des différents modèles et études de graphes dynamiques (voir aussi [7]).

L’un des premiers modèles développés, et aussi l’un des plus classiques, est le modèle des graphes évolutifs [3]. Pour simplifier, étant donné un graphe statique G , appelé graphe sous-jacent, un graphe évolutif basé sur G est une suite (potentiellement infinie) de sous-graphes (couvrants mais non nécessairement connexes) de G (voir Section 2 pour les définitions précises). Ce modèle est particulièrement adapté pour modéliser les réseaux dynamiques synchrones.

Dans toute sa généralité, le modèle des graphes évolutifs permet de considérer un ensemble extrêmement riche de réseaux dynamiques. Par conséquent, pour obtenir des résultats intéressants, il est quasiment toujours nécessaire de formuler des hypothèses permettant de réduire les possibilités de graphes dynamiques engendrés par le modèle. Citons par exemple l’hypothèse de connexité au cours du temps, qui stipule qu’il existe un chemin temporel de tout sommet vers tout autre sommet, et l’hypothèse de constante connexité, pour laquelle le graphe doit être connexe à chaque instant. Cette dernière hypothèse, très classique, a été généralisée dans un récent article de Kuhn, Lynch et Oshman [6] par la notion de T -intervalle-connexité. Grossièrement, étant donné un entier $T \geq 1$, un graphe dynamique est T -intervalle-connexe si, pour toute fenêtre de T unités de temps, il existe un sous-graphe connexe couvrant stable pendant toute la période. (La notion de constante connexité est donc équivalente à la notion de 1-intervalle-connexité.) Cette nouvelle

[†]Ce travail a été réalisé avec le soutien du projet ANR DISPLEXITY, du projet INRIA CEPAGE, et du projet européen EULER.

notion, qui capture la stabilité de connexion au cours du temps, permet de dériver des résultats intéressants : la T -intervalle connexité permet une économie d'un facteur environ $\Theta(T)$ sur le nombre de messages nécessaire et suffisant pour réaliser un échange d'information entre tous les sommets [2, 6].

Dans cet article, nous poursuivons l'étude de ces graphes dynamiques T -intervalle-connexes en considérant le problème de l'exploration. Une entité mobile (appelée agent) se déplaçant dans un graphe dynamique doit traverser/visiter au moins une fois chacun de ses sommets. (Le temps de traversée d'une arête est unitaire.) Ce problème fondamental en algorithmique par agents mobiles a été très étudié dans les graphes statiques depuis l'article originel de Claude Shannon [8]. Concernant les graphes fortement dynamiques, seul le cas des graphes dynamiques périodiques a été étudié [4, 5]. Nous nous intéressons ici à la complexité en temps (en pire cas) de ce problème, à savoir le nombre d'unités de temps utilisées par l'agent pour résoudre le problème. Le problème de l'exploration, outre ses intérêts théoriques, peut être appliqué pour la maintenance de réseaux, où un agent mobile doit contrôler le bon fonctionnement de chacun des sommets du graphe.

Nous considérons le problème dans deux scénarios. Dans le premier, l'agent connaît entièrement et exactement le graphe dynamique à explorer. Cette situation correspond aux réseaux dynamiques prévisibles, tels que les réseaux de transport par exemple. Dans le second scénario, l'agent ne connaît pas la dynamique du graphe, c'est-à-dire les temps d'apparition et de disparition des arêtes. Ce cas correspond typiquement aux réseaux dont les changements sont liés à des pannes fréquentes et imprévisibles. Dans ce second scénario, Kuhn, Lynch et Oshman [6] ont noté que le problème de l'exploration est impossible à résoudre sous la seule hypothèse de 1-intervalle-connexité. En fait, il est assez facile de se convaincre qu'en rajoutant l'hypothèse que chaque arête du graphe sous-jacent apparaît infiniment souvent, le problème de l'exploration devient possible mais la complexité en temps est non bornée. Dans cet article, et seulement pour le second scénario, nous rajoutons donc l'hypothèse de δ -récurrence : chaque arête du graphe sous-jacent apparaît au moins une fois toutes les δ unités de temps.

Il s'avère que le problème de l'exploration est beaucoup plus complexe dans les graphes dynamiques que dans les graphes statiques. En effet, considérons par exemple le premier scénario (graphe connu). Le temps d'exploration des graphes statiques à n sommets est clairement en $\Theta(n)$ (pire cas $2n - 3$). Par contre la complexité en temps en pire cas de l'exploration des graphes dynamiques (1-intervalle-connexes) à n sommets reste largement inconnue. Aucune borne inférieure meilleure que la borne statique n'est connue, tandis que la meilleure borne supérieure connue est quadratique, et découle directement du fait que le diamètre temporel de tels graphes est borné par n . De ce fait, nous nous concentrons ici sur l'étude des graphes dynamiques T -intervalle-connexes dont le graphe sous-jacent est un anneau.

Nos résultats. Nous déterminons dans cet article la complexité exacte en temps de l'exploration des graphes dynamiques T -intervalle-connexes à n sommets basés sur l'anneau lorsque l'agent connaît la dynamique du graphe. Celle-ci est essentiellement de $2n - T - 1$ unités de temps, deux paliers étant atteints en-dessous de 2 et au-dessus de $(n + 1)/2$ (voir Section 3 pour les détails). Dans le cas où l'agent ne connaît pas la dynamique du graphe, nous rajoutons l'hypothèse de δ -récurrence, et nous montrons que la complexité augmente pour atteindre $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ unités de temps.

2 Définitions et modèles

Cette section fournit les définitions précises des concepts et modèles évoqués informellement dans la section précédente. Certaines définitions sont similaires voire identiques aux définitions présentes dans [6].

Définition 1 (Graphe dynamique) *Un graphe dynamique est une paire $\mathcal{G} = (V, \mathcal{E})$, où V est un ensemble statique de n sommets, et \mathcal{E} est une fonction qui associe à tout nombre entier i un ensemble d'arêtes non orientées $\mathcal{E}(i)$.*

Définition 2 (Graphe sous-jacent) *Étant donné un graphe dynamique $\mathcal{G} = (V, \mathcal{E})$, le graphe statique $G = (V, \bigcup_{i=0}^{\infty} \mathcal{E}(i))$ est appelé graphe sous-jacent de \mathcal{G} . Inversement le graphe dynamique \mathcal{G} est dit basé sur le graphe statique G .*

Dans cet article, nous considérons les graphes dynamiques basés sur un anneau de taille n , noté A_n .

Définition 3 (T -intervalle-connexité) Un graphe dynamique est T -intervalle-connexe, pour un entier $T \geq 1$, si pour tout entier i , le graphe statique $G_{[i, i+T[} = (V, \bigcap_{j=i}^{i+T-1} E(j))$ est connexe.

Définition 4 (δ -récurrence) Un graphe dynamique est δ -récurent si toute arête du graphe sous-jacent apparaît au moins une fois toutes les δ unités de temps.

Une entité mobile, appelée *agent*, opère sur ces graphes dynamiques. L'agent peut, en une unité de temps, traverser au plus une arête. Nous disons qu'un agent explore le graphe dynamique si et seulement si il visite l'ensemble de ses sommets.

3 L'agent connaît la dynamique du graphe

Dans cette section, nous supposons que l'agent connaît parfaitement le graphe à explorer. Nous montrons que le temps d'exploration est faible, borné par $2n$, lorsque le graphe sous-jacent est un anneau. De plus, nous montrons que l'agent peut profiter de la T -intervalle-connexité pour gagner un facteur additif T . Notons que notre borne supérieure est constructive.

Théorème 1 Pour tout $n \geq 3$ et $T \geq 1$ et pour tout graphe dynamique T -intervalle-connexe basé sur A_n , il existe un agent (algorithme) capable d'explorer ce graphe dynamique en un temps au plus

$$\begin{array}{ll} 2n - 3 & \text{si } T = 1 \\ 2n - T - 1 & \text{si } 2 \leq T < (n + 1)/2 \\ \lfloor \frac{3(n-1)}{2} \rfloor & \text{si } T \geq (n + 1)/2 \end{array} \quad \text{unités de temps.}$$

Éléments de preuve. Notons tout d'abord que, le graphe dynamique étant au moins 1-intervalle-connexe, au plus une arête est absente à chaque unité de temps. Considérons deux algorithmes (agents), l'un allant dans le sens horaire et l'autre dans le sens trigonométrique, chacun des deux progressant dès que le graphe dynamique le permet. Dans un premier temps, au moins l'un des deux agents est capable de progresser à chaque unité de temps. Cependant, au moment où les agents allaient se rejoindre pour la première fois (donc après un temps au plus n), leur progression peut être interrompue par l'absence d'une même arête e .

Si cette arête e est absente pendant au moins $n - 1$ unités de temps, l'un ou l'autre des agents a le temps de faire demi-tour et d'explorer tous les sommets de l'anneau dans l'autre sens.

Si l'arête e ne reste pas absente suffisamment longtemps et réapparaît au temps t , nous modifions les deux algorithmes de la façon suivante. L'agent progressant précédemment dans le sens horaire, respectivement trigonométrique, commence maintenant l'exploration de l'anneau dans le sens trigonométrique, respectivement horaire, et fait demi-tour le plus tard possible de façon à pouvoir atteindre l'arête e au plus tard au temps $t - 1$. Au temps t , les deux algorithmes modifiés se croisent et continuent à progresser dans leur direction habituelle jusqu'à ce que l'un d'eux termine son exploration. Notons qu'après le croisement des agents, nous avons de nouveau la propriété qu'au moins l'un des deux progresse à chaque unité de temps.

Globalement, la vitesse (en traversées d'arête par unité de temps) de l'un des deux agents est donc d'au moins $1/2$ en moyenne en-dehors de la période d'absence de e . Par ailleurs, la modification des algorithmes permet à chacun des agents d'explorer une partie d'anneau supplémentaire. Malheureusement, cette portion d'anneau est parcourue deux fois au lieu d'une. Cependant, intuitivement, la vitesse des deux agents modifiés passe à 1 en l'absence de e et permet de compenser la perte de temps liée à l'aller-retour. Une analyse beaucoup plus précise des algorithmes modifiés permet d'obtenir les bornes annoncées. \square

Nous montrons maintenant que les bornes supérieures présentées ci-dessus sont atteintes exactement.

Théorème 2 Pour tout $n \geq 3$ et $T \geq 1$, il existe un graphe dynamique T -intervalle-connexe basé sur A_n tel que tout agent doit faire au moins

$$\begin{array}{ll} 2n - 3 & \text{si } T = 1 \\ 2n - T - 1 & \text{si } 2 \leq T < (n + 1)/2 \\ \lfloor \frac{3(n-1)}{2} \rfloor & \text{si } T \geq (n + 1)/2 \end{array} \quad \text{unités de temps pour l'explorer.}$$

Éléments de preuve. Pour $T \geq (n + 1)/2$, le graphe dynamique difficile à explorer est celui pour lequel l'arête opposée au sommet de départ reste déconnectée pendant $3n/2$ unités de temps.

Pour T compris entre 2 et $(n+1)/2$, le graphe dynamique difficile à explorer est construit de la manière suivante. Soit s_0, \dots, s_{n-1} les sommets de l'anneau dans le sens horaire, s_0 étant le sommet de départ de l'agent. L'arête $\{s_0, s_1\}$, respectivement $\{s_{T-1}, s_T\}$ est déconnectée aux temps $t \in [0, n-2T+1[$, respectivement $t \in [n-T, 2n[$. Toutes les autres arêtes de l'anneau sont toujours présentes.

Le graphe difficile à explorer pour $T = 1$ est le même que pour $T = 2$. \square

4 L'agent ne connaît pas la dynamique du graphe

Dans cette section, nous supposons que l'agent ne connaît pas la dynamique du graphe, c'est-à-dire ne connaît pas les temps d'apparition et de disparition des arêtes. Pour que le problème ait une solution, nous rajoutons l'hypothèse que les graphes dynamiques à explorer sont δ -récurrents, pour un certain entier $\delta \geq 1$. Nous prouvons tout d'abord qu'il existe un algorithme extrêmement simple qui permet d'explorer tous les graphes dynamiques T -intervalle-connexes δ -récurrents basés sur l'anneau. Cet algorithme consiste à se déplacer autant et dès que possible dans un sens arbitraire fixé.

Théorème 3 *Pour tout $n \geq 3$, $T \geq 1$ et $\delta \geq 1$, il existe un agent (algorithme) permettant d'explorer tout graphe dynamique T -intervalle-connexe δ -récurrent basé sur A_n en au plus $n-1 + \lceil \frac{n-1}{\max\{1, T-1\}} \rceil (\delta-1)$ unités de temps.*

Éléments de preuve. Considérons l'algorithme dictant à l'agent d'aller dans une direction arbitraire fixée à chaque fois que l'arête correspondante est présente. La complexité énoncée vient du fait, d'une part, que cet algorithme ne peut être bloqué plus de $\delta-1$ unités de temps d'affilée (δ -récence), et d'autre part qu'après chaque blocage l'agent peut traverser sans encombre $\max\{1, T-1\}$ arêtes (T -intervalle-connexité). \square

Le résultat suivant montre que l'algorithme très simple décrit précédemment est quasiment optimal.

Théorème 4 *Pour tout $n \geq 3$, $T \geq 1$, $\delta \geq 1$, et tout agent (algorithme), il existe un graphe dynamique T -intervalle-connexe δ -récurrent basé sur A_n tel que cet agent nécessite au moins $n-1 + \lfloor \frac{n-3}{\max\{1, T-1\}} \rfloor (\delta-1)$ unités de temps pour l'explorer. Ce résultat reste vrai même si l'agent connaît n , T et δ .*

Éléments de preuve. Fixons un agent, n , T et δ . Intuitivement, à chaque fois que l'agent effectue $\max\{1, T-1\}$ traversées d'arête dans une même direction, la prochaine arête est retirée pendant $\delta-1$ unités de temps ou jusqu'à ce que l'agent change de direction. L'arête est de nouveau retirée autant que possible à chaque fois que l'agent revient sur un sommet incident à cette arête. \square

Références

- [1] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, volume 27(5), 2012.
- [2] C. Dutta, G. Pandurangan, R. Rajaraman, and Z. Sun. Information spreading in dynamic networks. *CoRR*, abs/1112.0384, 2011.
- [3] A. Ferreira, Building a reference combinatorial model for MANETs. *Network*, IEEE, volume 18(5), pages 24–29, 2004.
- [4] P. Flocchini, B. Mans, and N. Santoro. On the exploration of time-varying networks. *Theoretical Computer science*, volume 469, pages 53–68, 2013.
- [5] D. Ilcinkas and A. M. Wade. On the Power of Waiting when Exploring Public Transportation Systems. In *15th International Conference On Principles Of Distributed Systems (OPODIS)*, LNCS 7109, pages 451–464, 2011.
- [6] F. Kuhn, N.A. Lynch, and R. Oshman, Distributed computation in dynamic networks. In *42nd ACM symposium on Theory of computing (STOC)*, pages 513–522, 2010.
- [7] F. Kuhn and R. Oshman, Dynamic networks : models and algorithms. *ACM SIGACT News*, volume 42(1), pages 82–96, 2011.
- [8] C. E. Shannon, Presentation of a maze-solving machine. 8th Conf. of the Josiah Macy Jr. Found. (Cybernetics), pages 173–180, 1951.