HAL
open science

Tester efficacement la T-intervalle connexité dans les graphes dynamiques

Arnaud Casteigts, Ralf Klasing, Yessin M. Neggaz, Joseph G. Peters

To cite this version:

Arnaud Casteigts, Ralf Klasing, Yessin M. Neggaz, Joseph G. Peters. Tester efficacement la T-intervalle connexité dans les graphes dynamiques. ALGOTEL 2015 - 17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2015, Beaune, France. hal-01147110

HAL Id: hal-01147110

https://hal.science/hal-01147110

Submitted on 29 Apr 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.

Tester efficacement la T-intervalle connexité dans les graphes dynamiques. ‡

Arnaud Casteigts[1], Ralf Klasing[1], Yessin M. Neggaz[1] et Joseph G. Peters[2]

[1] *LaBRI, CNRS, Université de Bordeaux, France*
[2] *School of Computing Science, Simon Fraser University, Burnaby, BC, Canada*

Beaucoup de réseaux dynamiques sont constitués d'entités durables dont les liens évoluent à travers le temps. D'un point de vue global et discret, ces réseaux se modélisent bien sous forme d'un graphe évolutif, c'est à dire une suite de graphes $\mathcal{G} = \{G_1, G_2, ..., G_\delta\}$ telle que $G_i = (V, E_i)$ représente la topologie du réseau pendant la période i. Une telle suite est dite T-intervalle connexe si pour tout $t \in [1, \delta - T + 1]$ les graphes $G_t, G_{t+1}, ..., G_{t+T-1}$ partagent un même sous-graphe couvrant connexe. Dans cet article, nous étudions le problème de décider si une suite $\mathcal{G}$ donnée est T-intervalle connexe pour un T donné. Nous étudions également la variante où T n'est pas donné et il s'agit de trouver le plus grand T tel que $\mathcal{G}$ est T-intervalle connexe. Nous supposons que les changements entre deux graphes consécutifs sont arbitraires, et que deux opérations (*intersection binaire* et *test de connexité*) sont utilisées comme briques de base. Nous montrons que $\Omega(\delta)$ de ces opérations sont nécessaires pour les deux variantes et présentons des solutions qui en utilisent $O(\delta)$ également. Ces solutions sont donc optimales à une constante multiplicative près.

1 Introduction

La plupart des réseaux dynamiques se composent d'entités durables qui interagissent au fil du temps. Les hypothèses sur la dynamique sont très variables, allant d'un changement occasionnel de la topologie à des changements fréquents plus ou moins contraints (e.g. garantie de connexité classique ou temporelle). Chacun de ces contextes peut être vu comme une *famille* de graphes dynamiques, la plupart de ces dernières pouvant être reliées par inclusion au sein d'une classification [3]. Récemment, la famille des graphes *T-intervalle connexes* a été identifiée dans [7] comme jouant un rôle important en algorithmique distribuée, p.ex. pour le calcul d'une fonction quelconque de l'état des nœuds. Elle a aussi été étudiée dans le cadre du problème de l'exploration [6]. La T-intervalle connexité se définit naturellement dans le cadre des graphes évolutifs (introduits dans [5]), où la dynamique du graphe est représentée par une suite de graphes $\mathcal{G} = \{G_1, G_2, ..., G_\delta\}$. Un tel graphe est dit T-intervalle connexe si pour tout $t \in [1, \delta - T + 1]$ les graphes $G_t, G_{t+1}, ..., G_{t+T-1}$ partagent un même sous-graphe couvrant connexe.

Le test automatique de propriétés sur les graphes dynamiques est une problématique récente, qui se justifie par le besoin de comprendre des cas réels de mobilité via l'analyse de traces topologiques qui en sont issues. Dans ce cadre, nous nous intéressons à la conception d'algorithmes permettant de vérifier automatiquement l'appartenance d'un graphe dynamique aux différentes familles de graphes identifiées dans la classification de [3]. Peu d'algorithmes existent en ce sens, exception faite de tests relatifs à la connexité temporelle [1, 2, 8].

Dans cet article, nous considérons le problème de décider si une suite $\mathcal{G}$ donnée est T-intervalle connexe pour un T donné, ainsi que la variante qui consiste à trouver le T maximum pour lequel $\mathcal{G}$ est T-intervalle connexe. Nous supposons que les changements entre deux graphes consécutifs sont arbitraires et qu'une approche opérant à l'échelle globale de la suite est donc appropriée. Précisément, nous considérons deux opérations élémentaires qui sont *l'intersection binaire* (étant donné deux graphes, calculer leur intersection) et *le test de connexité* (étant donné un graphe, déterminer s'il est connexe). Nous montrons que $\Omega(\delta)$ de

† Ce travail a été réalisé pendant que Joseph G. Peters était en visite au LaBRI en tant que professeur invité.
‡ La version originale de ce papier est à paraître dans les actes de CIAC 2015 (version longue accessible sur arXiv [4])

ces opérations sont nécessaires pour les deux variantes du problème et présentons des solutions qui n'en requièrent que $O(\delta)$. Ces solutions sont donc optimales à une constante multiplicative près. Les preuves complètes des lemmes et théorèmes de ce papier peuvent être consultées dans la version longue [4].

2 Définitions et observations

Étant donné un ensemble (fini) S de graphes $\{G' = (V,E'), G'' = (V,E''), \ldots\}$, on appelle le graphe $(V, \cap\{E', E'', \ldots\})$ le *graphe d'intersection* de S noté $\cap\{G', G'', \ldots\}$. Lorsque l'ensemble comporte seulement deux graphes, on parle d'une intersection binaire et on utilise la notation infixe $G' \cap G''$. Si l'intersection concerne une sous-suite consécutive $\{G_i, G_{i+1}, \ldots, G_j\}$ d'un graphe dynamique $\mathcal{G}$, on s'autorise à noter $G_{(i,j)}$ le graphe d'intersection $\cap\{G_i, G_{i+1}, \ldots, G_j\}$. Soit $\mathcal{G}^T = \{G_{(1,T)}, G_{(2,T+1)}, ..., G_{(\delta-T+1,\delta)}\}$. Nous appelons $\mathcal{G}^T$ la $T^{\text{ème}}$ ligne dans la *hiérarchie d'intersection* de $\mathcal{G}$ (illustrée à la Figure 1). Un cas particulier est $\mathcal{G}^1 = \mathcal{G}$. Pour tout $1 \leq i \leq \delta - T + 1$, nous définissons $\mathcal{G}^T[i] = G_{(i,i+T-1)}$ et notons $\mathcal{G}^T[i]$ le $i^{ème}$ élément de la ligne $\mathcal{G}^T$.

Un graphe dynamique $\mathcal{G}$ est T*-intervalle connexe* si et seulement si tous les graphes de $\{G_t, ..., G_{t+T-1}\}$ partagent un sous-graphe couvrant connexe commun, autrement dit si le graphe d'intersection $G_{(t,t+T-1)}$ est connexe pour tout $t \in [1, \delta - T + 1]$, ou encore mieux : si tous les graphes de la ligne $\mathcal{G}^T$ sont connexes.

Nous appelons T-INTERVAL-CONNECTIVITY le problème de décider si un graphe dynamique $\mathcal{G}$ est T-intervalle connexe pour un T donné, et INTERVAL-CONNECTIVITY celui de trouver $\max\{T : \mathcal{G}$ est T-intervalle connexe$\}$.

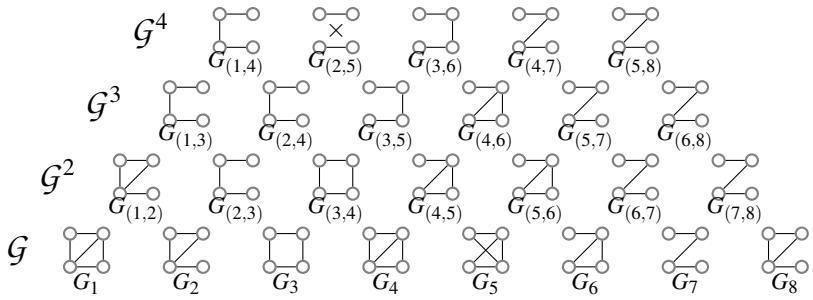

FIGURE 1: Exemple d'une hiérarchie d'intersection pour un graphe dynamique $\mathcal{G}$ donné. Ici, $\mathcal{G}$ est 3-intervalle connexe, mais pas 4-intervalle connexe car $\mathcal{G}^4$ contient un graphe déconnecté, $G_{(2,5)}$, qui implique que G_2, G_3, G_4, G_5 ne partagent pas de sous-graphe couvrant connexe.

Observation 1. En utilisant des listes d'adjacences pour représenter les graphes, l'*intersection binaire* peut être effectuée en temps linéaire en le nombre d'arêtes. Le *test de connexité* peut également être effectuée en temps linéaire en le nombre d'arêtes, en utilisant l'algorithme de Tarjan pour les composantes connexes (si le graphe est orienté) ou un simple DFS (s'il ne l'est pas). Ces deux opérations ont donc des coûts similaires et nous les comptabilisons ensemble.

Borne inférieure. La borne inférieure suivante est valable pour n'importe quel algorithme utilisant uniquement les deux opérations élémentaires que nous considérons.

Lemme 1. $\Omega(\delta)$ *opérations élémentaires sont nécessaires pour résoudre* T-INTERVAL-CONNECTIVITY.

La démonstration procède par contradiction, en montrant que si tous les éléments ne sont pas examinés, alors deux suites différentes peuvent engendrer des traitements identiques alors même que l'une est T-interval connexe et l'autre non. Un argument similaire peut être utilisé pour INTERVAL-CONNECTIVITY.

Borne supérieure triviale. On peut voir que $O(\delta^2)$ opérations suffisent pour résoudre les deux versions du problème. Il suffit pour cela de construire les lignes de la hiérarchie d'intersection de manière incrémentale, chaque graphe de la ligne k étant obtenu à partir de deux graphes de la ligne $k-1$, par exemple ici $G_{(3,6)} = G_{(3,5)} \cap G_{(4,6)}$. Dans le cas de T-INTERVAL-CONNECTIVITY, l'algorithme s'arrête à la ligne T et

répond `vrai` si et seulement si tous les graphes de cette ligne sont connexes. Dans le cas de INTERVAL-CONNECTIVITY, on procède de manière similaire, mais en vérifiant la connexité des graphes au fur et à mesure. Si un graphe non connexe est trouvé à la ligne k, la réponse $k-1$ est renvoyée. Si tous les graphes sont connexes jusqu'à la ligne δ, alors δ est renvoyé.

Amélioration. Cette stratégie peut être améliorée pour obtenir une complexité de $O(\delta \log \delta)$ opérations, en calculant seulement certaines lignes en utilisant la propriété suivante. (Cette variante mérite d'être mentionnée car elle nous permet de classer les deux problèmes dans la classe de complexité **NC** des problèmes qui admettent une solution parallélisable sur PRAM avec un temps d'exécution polylogarithmique.)

Lemme 2. *Etant donné une ligne $\mathcal{G}^k$, toute ligne $\mathcal{G}^\ell$ pour $k+1 \leq l \leq 2k$ peut être calculée en utilisant $O(\delta)$ opérations élémentaires.*

Dans le cas de T-INTERVAL-CONNECTIVITY, cette propriété permet de calculer les lignes $\mathcal{G}^{2^i}$ pour tout i de 1 à $\lceil \log_2 T \rceil - 1$ sans calculer les lignes intermédiaires. Il suffit ensuite de calculer la ligne $\mathcal{G}^T$ directement à partir de la ligne $\mathcal{G}^{2^{\lceil \log_2 T \rceil - 1}}$ et de tester si tous les graphes de cette ligne sont connexes. Dans le cas de INTERVAL-CONNECTIVITY, on calcule les lignes $\mathcal{G}^{2^i}$ jusqu'à ce que la ligne contienne un graphe non connexe (les tests sont effectués au fur et à mesure). Soit $\mathcal{G}^{2^{j+1}}$ cette ligne et $\mathcal{G}^{2^j}$ la ligne précédemment calculée, l'algorithme démarre une recherche dichotomique entre ces deux lignes pour trouver la première ligne ayant un graphe non connexe. Au final, on calcule donc $2\lceil \log_2 T \rceil = O(\log \delta)$ lignes.

3 Stratégie optimale

Nous présentons maintenant une stratégie qui utilise $O(\delta)$ opérations élémentaires, rejoignant ainsi la borne inférieure (à une constante multiplicative près). Cette stratégie s'appuie sur le concept d'*échelle*, à savoir une suite de graphes qui "grimpe" la hiérarchie d'intersection vers la droite ou vers la gauche.

Définition 1. *Une* échelle droite *de longueur l à l'indice i, notée $\mathcal{R}^l[i]$, est la suite de graphes d'intersection $\{\mathcal{G}^k[i], k = 1,2,\ldots,l\}$. Une* échelle gauche *de longueur l à l'indice i, notée $\mathcal{L}^l[i]$, est la suite $\{\mathcal{G}^k[i-k+1], k = 1,2,\ldots,l\}$. Une échelle droite (resp. gauche) de longueur $l-1$ à l'indice i est dite* incrémentée *lorsque le graphe $\mathcal{G}^l[i]$ (resp. $\mathcal{G}^l[i-l+1]$) y est ajouté, ce graphe étant alors qualifié d'*incrément.

Les échelles sont illustrées à la Figure 2. On peut prouver les propriétés suivantes sur les échelles.

Lemme 3. *Une échelle de longueur l peut être calculée en utilisant $l-1$ intersections binaires.*

Lemme 4. *Étant donné une échelle gauche de longueur l_ℓ à l'indice i_ℓ et une échelle droite de longueur l_r à l'indice $i_r = i_\ell + 1$. Pour n'importe quelle paire (i,k) telle que $i_r - l_\ell \leq i < i_r$ et $i_r - i < k \leq i_r - i + l_r$, le graphe $\mathcal{G}^k[i]$ peut être calculé en utilisant seulement une intersection binaire, à savoir $\mathcal{G}^{i_r - i}[i] \cap \mathcal{G}^{k - i_r + i}[i_r]$. (Ces intersections sont illustrées par des traits fins courbés dans la Figure 2.)*

T-INTERVAL-CONNECTIVITY (Figure 2). L'idée est de parcourir la $T^{\text{ème}}$ ligne dans la hiérarchie d'intersection de gauche à droite, en commençant par $\mathcal{G}^T[1]$. Si un graphe non connexe est trouvé, l'algorithme renvoie `faux` et termine. Si l'algorithme atteint le dernier graphe de la ligne et aucun graphe non connexe n'a été trouvé, alors il renvoie `vrai`. Les échelles sont construites de la manière suivante : chaque échelle gauche est entièrement construite (de bas en haut) lorsque le parcours arrive à l'emplacement correspondant à son plus haut élément dans la ligne T (i.e. où le dernier incrément doit avoir lieu). C'est le cas, par exemple, de la première échelle à gauche dont le dernier incrément est le graphe de départ $\mathcal{G}^T[1]$. Si un graphe non connexe est trouvé pendant ce processus, l'exécution se termine et renvoie `faux`. Les échelles droites, quant à elles, sont construites à mesure que le parcours progresse. L'échelle droite courante est ainsi incrémentée de sorte que son dernier incrément est immédiatement utilisé pour calculer le graphe à l'indice courant dans la ligne T (en utilisant le Lemme 4). Cela continue jusqu'à ce que l'échelle droite atteigne la ligne $T-1$, puis, une nouvelle échelle gauche est construite *(cercles gris sur la Figure)*.

Theorem 1. *T-INTERVAL-CONNECTIVITY peut être résolu en utilisant $O(\delta)$ opérations élémentaires.*

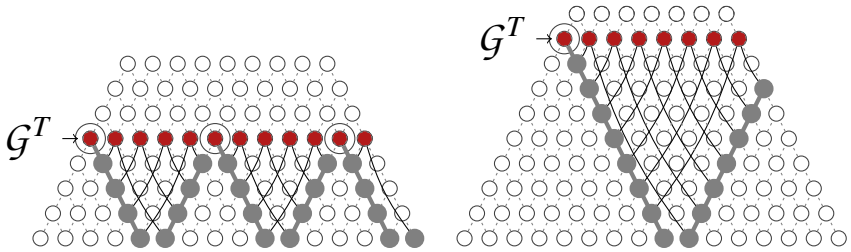

FIGURE 2: Exemple d'exécution de l'algorithme optimal pour T-INTERVAL-CONNECTIVITY avec $T < \delta/2$ (gauche) et $T \geq \delta/2$ (droite). $\mathcal{G}$ est T-intervalle connexe dans les deux exemples.

INTERVAL-CONNECTIVITY (Figure 3). La stratégie est légèrement plus complexe mais dans le même esprit. Il s'agit de faire un parcours bidimensionnel dans la hiérarchie d'intersection. Le parcours commence en bas à gauche sur $\mathcal{G}^1[1]$ et construit progressivement une échelle droite jusqu'à rencontrer un graphe non connexe. Si $\mathcal{G}^\delta[1]$ est atteint et qu'il est connexe, l'exécution termine en renvoyant δ. Sinon, supposons que le premier graphe non connexe soit $\mathcal{G}^{k+1}[1]$. Alors k est une borne supérieure sur la valeur recherchée et le parcours descend d'un niveau à $\mathcal{G}^k[2]$ qui est le graphe suivant dans la ligne k et qui doit être testé. Ceci induit la construction d'une échelle gauche se terminant en ce même graphe. Tant que le graphe courant est connexe, le parcours avance vers la droite sur la ligne k en utilisant une stratégie similaire à celle l'algorithme T-INTERVAL-CONNECTIVITY. Ici, cependant, chaque fois qu'un graphe non connexe est trouvé, le parcours descend d'une ligne. Si le parcours atteint le dernier graphe à droite d'une ligne et que ce graphe est connexe, le numéro de la ligne courante est renvoyée comme réponse. Sinon, le parcours se termine sur un graphe non connexe de la première ligne, et la valeur 0 est renvoyée.

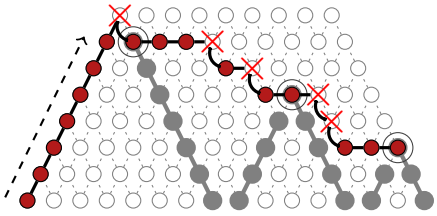

FIGURE 3: Exemple d'exécution de l'algorithme optimal pour INTERVAL-CONNECTIVITY.

Theorem 2. INTERVAL-CONNECTIVITY *peut être résolu en utilisant $O(\delta)$ opérations élémentaires.*

Références

[1] M. Barjon, A. Casteigts, S. Chaumette, C. Johnen, and Y. M. Neggaz. Un algorithme de test pour la connexité temporelle des graphes dynamiques de faible densité. In *Proc. of ALGOTEL*, 2014.

[2] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proc. of ADHOC-NOW*, 2003.

[3] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *Int. J. of Parallel, Emergent and Distributed Systems*, 27(5) :387–408, 2012.

[4] A. Casteigts, R. Klasing, Y. M. Neggaz, and J. G. Peters. Efficiently Testing T-Interval Connectivity in Dynamic Graphs. *ArXiv e-prints, No 1502.00089*, 2015.

[5] A. Ferreira. On models and algorithms for dynamic communication networks : The case for evolving graphs. In *Proc. of ALGOTEL*, 2002.

[6] D. Ilcinkas and A. Wade. Exploration des graphes dynamiques t-intervalle-connexes : le cas de l'anneau. In *Proc. of ALGOTEL*, 2013.

[7] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proc. of STOC*, 2010.

[8] J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume. Graphes d'accessibilité dynamiques. In *Proc. of ALGOTEL*, 2012.