

Satisfaire un internaute impatient est difficile

Fedor V. Fomin, Frédéric Giroire, Alain Jean-Marie, Dorian Mazauric, Nicolas Nisse

▶ **To cite this version:**

Fedor V. Fomin, Frédéric Giroire, Alain Jean-Marie, Dorian Mazauric, Nicolas Nisse. Satisfaire un internaute impatient est difficile. 14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel), 2012, La Grande Motte, France. hal-00687102

HAL Id: hal-00687102
https://hal.science/hal-00687102

Submitted on 12 Apr 2012

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.

Satisfaire un internaute impatient est difficile

F. V. Fomin[1] and F. Giroire[2] and A. Jean-Marie[3,4] and D. Mazauric[2,3] and N. Nisse[2]

[1] *University of Bergen, Norway*
[2] *Mascotte, INRIA, I3S(CNRS/UNS), Sophia Antipolis, France*
[3] *Maestro, INRIA, Sophia Antipolis, France*
[4] *LIRMM, Montpellier, France*

Considérons un internaute qui va d'une page Web à une autre en suivant les liens qu'il rencontre. Pour éviter que l'internaute ne (s'im)patiente, il est important d'essayer de télécharger les documents avant que l'internaute ne les atteigne. Cependant, le coût d'un tel pré-téléchargement ne doit pas excéder le gain en temps qu'il génère. Ainsi, il faut minimiser la bande passante utilisée pour le pré-téléchargement tout en s'assurant que l'internaute impatient n'attende jamais. Nous modélisons ce problème sous forme d'un jeu de type *Cops and Robber* dans les graphes. En particulier, étant donnés un graphe G qui représente le graphe du Web et une page Web de départ $v_0 \in V(G)$, nous définissons l'*indice de contrôle* de G, $ic(G,v_0) \in \mathbb{N}$, qui modélise la vitesse minimum de téléchargement suffisante pour que l'internaute partant de v_0 n'attende jamais quoi qu'il fasse.

Nous considérons le problème de décider si $ic(G,v_0) \leq k$ et démontrons plusieurs résultats de complexité. En particulier, décider si $ic(G,v_0) \leq 2$ est NP-difficile si G est cordal, et décider si $ic(G,v_0) \leq 4$ est PSPACE-complet si G est un graphe orienté acyclique. Nous donnons un algorithme exponentiel exact qui calcule $ic(G,v_0)$ en temps $O^*(2^n)$ dans un graphe de n sommets quelconque. Puis, nous montrons que le problème est polynomial dans le cas des arbres et des graphes d'intervalles. Enfin, nous donnons une caractérisation combinatoire de l'indice de contrôle. Pour tout graphe G et $v_0 \in V(G)$, $ic(G,v_0) \geq \max_S \lceil \frac{|N[S]|-1}{|S|} \rceil$ avec $v_0 \in S \subseteq V$, S induit un sous-graphe connexe et $N[S]$ l'ensemble des sommets de S ou voisins d'un sommet de S. Il y a de plus égalité dans le cas des arbres.

Keywords: Pré-charchement, jeux de "Cops and Robber", PSPACE-complet, arbres, graphes d'intervalles

1 Naviguer vite sur le Web sans gaspiller de ressources

En informatique, le *pré-téléchargement* est une technique classique qui exploite le parallélisme entre l'exécution d'une tâche et le transfert des informations nécessaires aux tâches suivantes pour réduire les temps d'attente. Par exemple, dans un processeur, les instructions et les données sont chargées dans la mémoire simultanément à l'exécution des instructions précédentes. Actuellement, cette technique peut être utilisée dans le contexte du Web où les navigateurs peuvent télécharger les documents accessibles depuis le document en train d'être visioné (page Web, video, etc.). L'accès au document suivant paraît instantané à l'utilisateur et donne l'impression d'une navigation rapide [phd11]. C'est pourquoi le pré-téléchargement a été proposé par Mozilla comme un draft de standard Internet [Inc99]. Cependant, pré-télécharger tous les documents accessibles pourrait résulter en un gaspillage des ressources du réseau (e.g., bande passante, mémoire) puisque tous les documents téléchargés ne seront pas forcément visionnés ou utilisés. Il est donc nécessaire d'établir un compromis entre le gain de temps et la perte des ressources du réseau qui en résulte.

Les modèles développés jusqu'à présent pour l'étude du pré-téléchargement sont essentiellement basés sur la notion de *graphe d'exécution* où les sommets représentent les tâches et les arcs représentent les ordonnancements possibles entre tâches. Par exemple, les sommets représentent les pages Web et les arcs modélisent les liens qu'il faut suivre pour accéder d'une page Web à une autre. L'exécution du programme ou la navigation d'un internaute sur le Web correspondent alors à une marche dans le graphe d'exécution. Il s'agit alors d'optimiser une certaine fonction de coût correspondant à la quantité de ressources consommées par le pré-téléchargement. Cette fonction dépend de la marche suivie dans le graphe d'exécution et de la gêne occasionnée par l'attente d'informations au cours de l'exécution de la tâche ou de la navigation sur le Web. En d'autres termes, il s'agit d'optimiser la bande passante nécessaire au pré-téléchargement, tout

en satisfaisant la qualité de service requise. Généralement, il est difficile de résoudre de tels problèmes. Par exemple, dans les modèles Markoviens [JG97] où les arcs du graphe d'exécution sont associés à des transitions de probabilités (modélisant un internaute qui navigue plus ou moins aléatoirement sur le Web), le problème de pré-téléchargement s'inscrit dans le contexte de la programmation dynamique stochastique [GCD02, MJM10]. Sa résolution exacte requiert un effort de calcul exponentiel en le nombre de nœuds du graphe d'exécution (taille de l'espace d'états de ces modèles de Markov).

Dans cet article, nous considérons le problème du pré-téléchargement *parfait* où l'internaute "surfant" sur le Web ne doit jamais pouvoir accéder à une page Web qui n'a pas été pré-téléchargée auparavant. En d'autres termes, l'internaute est impatient et ne tolère aucune attente. La limite de bande passante se traduit par le fait qu'un nombre limité de pages Web peuvent être pré-téléchargées à chaque étape. Notre problème consiste à déterminer le nombre constant minimum de pages Web qui doivent être pré-téléchargées à chaque étape. Un avantage de notre approche est qu'elle ne nécessite aucune supposition quant au comportement de l'internaute, contrairement à [GCD02, MJM10]. Pour étudier ce problème nous en proposons une modélisation sous forme de jeu de type *Cops and Robber* dans les graphes [FT08, BN11].

Nous formalisons le *jeu de surveillance* d'un graphe dans la Section 2 et décrivons sa relation avec le problème de pré-téléchargement. En particulier, nous discutons des hypothèses utilisées. Nos résultats sont présentés dans la Section 3. Les preuves sont seulement esquissées voire omises mais sont disponibles dans [FGJM$^+$11]. Nous concluons par quelques perspectives et de nombreuses questions non résolues.

2 Graphe sous haute surveillance.

Au cours du *jeu de surveillance*, deux joueurs, le *fugitif* et le *surveillant*, s'affrontent sur un graphe (orienté ou non) connexe $G = (V,E)$ à partir d'un sommet $v_0 \in V$ qui est le seul sommet initialement *marqué* dans G et la position initiale du fugitif. Soit $k \in \mathbb{N}$ un entier fixé. Le jeu se déroule tour-à-tour, en commençant par le surveillant. À son tour, le surveillant marque au plus k sommets de G. Un sommet marqué le reste jusqu'à la fin du jeu. Puis, le fugitif peut se déplacer le long d'une arête (ou d'un arc) de G. Le fugitif gagne si, à une étape du jeu, il atteint un sommet non-marqué. Le surveillant gagne dans le cas contraire. Le surveillant a, bien sûr, intérêt à marquer le plus de sommets possible à chaque étape. Notons que le jeu termine au bout d'au plus $\lceil |V|/k \rceil$ étapes lorsque tous les sommets sont marqués.

L'*indice de contrôle*, $ic(G,v_0)$, de $G = (V,E)$ est le plus petit $k \geq 1$ qui permet au surveillant de gagner quels que soient les déplacements du fugitif à partir de $v_0 \in V$. Soit Δ le degré maximum de G et $deg(v_0)$ le degré de v_0 : $deg(v_0) \leq ic(G,v_0) \leq \Delta$.

En effet, si $k < deg(v_0)$, le surveillant ne peut pas marquer tous les voisins de v_0 à la première étape et le fugitif peut atteindre un sommet non-marqué dès son premier déplacement. Au contraire, si $k \geq \Delta$, le surveillant est certain de gagner en marquant tous les voisins (non déjà marqués) de la position courante du fugitif à chaque étape. Pour G non orienté, nous prouvons : $ic(G,v_0) = \Delta$ si et seulement si $deg(v_0) = \Delta$.

Il est clair que le graphe G joue ici le rôle du graphe d'exécution où les nœuds sont les pages Web et les arcs sont les liens entre pages Web. Le fugitif est l'internaute surfant sur le Web et sa position courante dans G correspond à la page Web en train d'être visualisée. Enfin, marquer un sommet correspond à pré-télécharger la page Web correspondante. Ainsi, l'indice de contrôle représente la vitesse minimum qui permet un pré-téléchargement parfait, c'est-à-dire qui permet de satisfaire l'internaute.

Avant de présenter nos résultats (Section 3), nous discutons des hypothèses de ce modèle.

Tout d'abord, notre modèle laisse supposer que toutes les pages Web peuvent être téléchargées à la même vitesse. En fait, notre modèle convient aussi lorsque chaque page Web v à une certaine taille w_v (dont dépend le temps de téléchargement). En effet, dans ce cas, il suffit de considérer le graphe G^p obtenu en remplaçant chaque sommet v de G par une clique K_v de taille w_v et chaque arête $\{u,v\}$ par un biparti complet entre K_v et K_u. Le problème de pré-téléchargement dans G est alors équivalent au problème de surveillance dans G^p.

Une autre hypothèse est la discrétisation du temps et le fait que l'internaute passe un temps identique sur chaque page. Comme nous considérons un pire cas, la durée d'une étape de notre modèle correspond simplement au temps minimum qu'un internaute peut passer sur une page Web. Le cas où le temps minimum que l'internaute passe sur une page Web diffère d'une page Web à une autre est laissé pour de futurs travaux.

L'hypothèse (probablement) la plus forte que nous faisons et celle d'une mémoire infinie : une page

téléchargée (un sommet marqué) le reste jusqu'à la fin. Il s'agit d'un premier modèle où, même avec cette hypothèse simplificatrice, le problème est très dur à résoudre. Comme travaux futurs, la gestion du cache doit être considérée : deux modèles plus réalistes peuvent être étudiés : soit un sommet marqué devient non-marqué après un certain nombre (fixé) d'étapes, soit le surveillant peut "dé-marquer" des sommets, en respectant que le nombre total de sommets marqués ne dépasse jamais une certaine limite fixée.

3 Un problème bien difficile

Dans cette section, nous nous intéressons à la complexité algorithmique du calcul de l'indice de contrôle d'un graphe. Nous prouvons que le problème est difficile même restreint à certaines classes de graphes "simples" comme les graphes cordaux (généralisation des arbres) et les graphes orientés acycliques (DAG). Tout n'est pourtant pas perdu puisque nous proposons des algorithmes polynomiaux pour résoudre le problème dans les arbres et les graphes d'intervalle (généralisation des chemins).

La plupart des preuves des résultats que nous avons obtenus tirent parti du fait que l'on peut se restreindre au cas où le fugitif ne peut suivre que des chemins induits de G. Plus précisément, soit $mic(G,v_0)$ le plus petit $k \geq 1$ qui permet au surveillant de gagner dans le graphe G lorsque le fugitif se déplace à chaque étape et ne peut suivre que des chemins induits débutant en $v_0 \in V(G)$. Évidemment, $mic(G,v_0) \leq ic(G,v_0)$.

Théorème 1. *Pour tout graphe G et $v_0 \in V(G)$, $mic(G,v_0) = ic(G,v_0)$.* *[FGJM+11]*

Un *graphe cordal* est un graphe dont tout cycle de longueur au moins 4 contient au moins une corde. Un graphe est dit *séparant* si son ensemble de sommets peut être partitionné en une clique et un stable. Un *graphe d'intervalle* est le graphe d'intersection d'intervalles de $\mathbb{R}$. Notons que les graphes d'intervalle et les graphes séparants sont deux sous-classes des graphes cordaux.

Théorème 2. *Le problème, avec un graphe G et $v_0 \in V$ en entrées, décidant si $ic(G,v_0) \leq k$ est : [FGJM+11]*
- *NP-difficile dans la classe des graphes cordaux pour $k = 2$ paramètre fixé ;*
- *PSPACE-complet dans la classe des DAGs pour $k = 4$ paramètre fixé ;*
- *NP-difficile dans la classe des graphes séparants si k fait partie de l'entrée.*
  De plus, dans les graphes séparant, le jeu dure au plus 2 étapes.

Idée de la preuve. Les assertions 1 et 3 découlent d'une réduction du problème 3-*transverse minimum* †. Le fait que le jeu dure au plus 2 étapes dans les graphes séparant vient de ce qu'un plus long chemin induit dans ces graphes a longueur au plus 2. L'assertion 2 découle d'une réduction de 3-QSAT ‡. □

D'après le théorème précédent, le problème de surveillance de graphe n'est pas soluble à paramètre fixé (FPT). De plus, ce problème reste difficile même si le jeu est limité à un nombre fixé de tours.

Théorème 3. *Le problème de surveillance peut être résolu en temps :* *[FGJM+11]*
- *linéaire dans la classe des graphes de degré au plus 3 ;*
- *$O(n \log n)$ dans la classe des arbres de n sommets ;*
- *$O(n\Delta^3)$ dans la classe des graphes d'intervalle de n sommets et de degré maximum Δ ;*
- *$O^*(2^n)$ dans la classe des graphes de n sommets.*

Idée de la preuve.
- La première assertion est presque triviale puisque $ic(G,v_0) = 1$ si et seulement si G est un chemin avec v_0 comme extrémité et que $ic(G,v_0) = \Delta$ si et seulement si v_0 est de degré Δ.
- Dans le cas d'un arbre T enraciné en $v_0 \in V(T)$, le problème peut être résolu par programmation dynamique. Soit $f_k : V(T) \to \mathbb{N}$ la fonction qui associe 0 aux feuilles et $\max\{0, d-k+\sum_{w\in C} f_k(w)\}$ à tout sommet avec C l'ensemble de ses d fils. Intuitivement, $f_k(v)$ est le nombre minimum de sommets de T_v (différents de v) qui doivent être marqués avant le début du jeu pour gagner avec k dans le sous-arbre T_v enraciné en v, en commençant de v déjà marqué. Nous prouvons que $ic(T,v_0) \leq k$ si et seulement si $f_k(v_0) = 0$.

†. 3-transverse minimum : étant donnés un ensemble E et $\mathcal{T}$ un ensemble de triplets de E, trouver $H \subseteq E$ de cardinalité minimum tel que $H \cap S \neq \emptyset$ pour tout $S \in \mathcal{T}$.

‡. 3-QSAT : étant donnée une formule F avec $2n$ variables bouléennes $(x_i, y_i)_{i\leq n}$, existe-t-il une assignation de ces variables telle que $\forall x_1, \exists y_1, \cdots, \forall x_n, \exists y_n F$ soit satisfaite ?

– Dans les cas des graphes d'intervalle, nous montrons que, pour tout $k \leq \Delta$, il existe un ensemble de Δ^2 chemins induits tels que si k suffit à les surveiller, alors $ic(G, v_0) \leq k$. Tester un chemin demandant un temps linéaire, le résultat en découle.
– Nous prouvons que décider si $ic(G, v_0) \leq k$ revient à trouver, par programmation dynamique, un chemin dans le graphe dont les sommets sont les configurations (ensemble des sommets marqués et la position du fugitif à une étape) et il y a un arc entre deux configurations si on peut passer de l'une à l'autre en un tour du jeu avec k. □

Les algorithmes polynomiaux ci-dessus calculent également des stratégies de surveillance optimales.

Théorème 4. *Pour tout graphe G et $v_0 \in V(G)$, $ic(G, v_0) \geq \max_S \lceil \frac{|N[S]|-1}{|S|} \rceil$ avec $v_0 \in S \subseteq V$, S induit un sous-graphe connexe et $N[S]$ le voisinage fermé de S. Il y a de plus égalité dans le cas des arbres. [FGJM+11]*

Une égalité pour tout graphe dans le théorème précédent impliquerait que le problème de surveillance est co-NP. Sous réserve que $P \neq NP$, cela contredirait le fait que le problème est PSPACE-complet. Cependant nous n'avons trouvé aucun exemple de graphe G et $v_0 \in V(G)$ tels que $ic(G, v_0) > \max_S \lceil \frac{|N[S]|-1}{|S|} \rceil$.

4 Conclusion

Plusieurs questions intéressantes restent à résoudre.

Questions. Le problème de surveillance peut-il être résolu en temps polynomial dans les graphes de degré borné ? de largeur arborescente (treewidth) bornée ? Existe-t-il un algorithme pour résoudre ce problème en temps $O^*(c^n)$, $c < 2$, dans les graphes de n sommets ?

Dans le contexte du pré-téléchargement, une variante naturelle du problème de surveillance de graphe est celle dans laquelle l'ensemble des sommets marqués est contraint à être connexe. En effet, cela revient à ne pouvoir télécharger que des pages Web liées à des pages Web déjà pré-téléchargées. Nous notons $cic(G, v_0)$ le plus petit $k \geq 1$ qui permet au surveillant de gagner dans le graphe G en satisfaisant la contrainte de connexité. Évidemment, $cic(G, v_0) \geq ic(G, v_0)$. Les résultats énoncés précédemment restent valides pour la variante connexe du problème de pré-téléchargement. En effet, les graphes considérés dans les réductions satisfont l'égalité des paramètres cic et ic. De plus, pour tout arbre, graphe d'intervalle, ou graphe de degré au plus 3, l'égalité est satisfaite. En ce qui concerne l'algorithme exponentiel exact, il suffit de ne considérer que les configurations connexes (l'ensemble des sommets marqués induit un sous-graphe connexe).

Lemma 1. *Pour tout $k \geq 2$, il y a un graphe G et $v_0 \in V(G)$ tels que $k = ic(G, v_0) < cic(G, v_0) = k + 1$.*

Idée de la preuve. Soit G le graphe avec $6k$ sommets : un chemin (v_0, v_1, v_2), $2k$ sommets $\{a_i, b_i\}_{i \leq k}$ tels que a_i est adjacent à v_2 et b_i pour tout $i \leq k$ et finalement, un ensemble K de $4k-3$ sommets adjacents à tous les b_i's. Alors, $k = ic(G, v_0) < cic(G, v_0) = k + 1$. □

Questions. Existe-t-il un graphe G et $v_0 \in V(G)$ tels que $ic(G, v_0) + 1 < cic(G, v_0)$? Existe-t-il f, une constante ou une fonction de n tel que $cic(G, v_0) \leq f \cdot ic(G, v_0)$ pour tout graphe G de n sommets et $v_0 \in V$?

Le Web n'étant pas connu *a priori*, il serait enfin intéressant d'étudier une variante online de ce problème.

Références

[BN11] A. Bonato and R. Nowakovski. *The game of Cops and Robber on Graphs*. American Math. Soc., 2011.

[FGJM+11] F.V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To satisfy impatient web surfers is hard. Technical Report INRIA-7740, INRIA, 2011. http://hal.inria.fr/inria-00625703/fr/.

[FT08] F.V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3) :236–245, 2008.

[GCD02] R. Grigoras, V. Charvillat, and M. Douze. Optimizing hypervideo navigation using a Markov decision process approach. In *ACM Multimedia*, pages 39–48, 2002.

[Inc99] Zona Research Inc. The economic impacts of unacceptable web-site download speeds. White paper, Redwood City, CA, April 1999. www.webperf.net/info/wp_downloadspeed.pdf.

[JG97] D. Joseph and D. Grunwald. Prefetching using Markov predictors. In *ISCA*, pages 252–263, 1997.

[MJM10] O. Morad and A. Jean-Marie. Optimisation en temps-réel du téléchargement de vidéos. In *Proc. of 11th Congress of the French Operations Research Soc.*, 2010.

[phd11] http://www.phdcomics.com/comics/archive.php?comicid=1456, 2011.