



**HAL**  
open science

## Localiser une cible dans un graphe

Julien Bensmail, Dorian Mazauric, Fionn Mc Inerney, Nicolas Nisse, Stéphane Pérennes

► **To cite this version:**

Julien Bensmail, Dorian Mazauric, Fionn Mc Inerney, Nicolas Nisse, Stéphane Pérennes. Localiser une cible dans un graphe. ALGOTEL 2018 - 20èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2018, Roscoff, France. hal-01774827

**HAL Id: hal-01774827**

**<https://inria.hal.science/hal-01774827v1>**

Submitted on 24 Apr 2018

**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.

# Localiser une cible dans un graphe<sup>†</sup>

Julien Bensmail<sup>1</sup> et Dorian Mazauric<sup>2</sup> et Fionn Mc Inerney<sup>1</sup> et Nicolas Nisse<sup>1</sup> et Stéphane Pérennes<sup>1</sup>

<sup>1</sup> Université Côte d'Azur, Inria, CNRS, I3S, France

<sup>2</sup> Université Côte d'Azur, Inria Sophia Antipolis - Méditerranée, France

---

Le jeu de la localisation d'une cible (invisible et immobile) dans un graphe a été introduit par Seager en 2013. Dans ce jeu, une cible est placée secrètement sur un sommet et, à chaque tour, il est possible d'interroger un sommet et recevoir, comme réponse, la distance exacte entre ce sommet et la cible. L'objectif est de localiser la cible en minimisant le nombre de tours, et ce, quelle que soit sa position.

Nous considérons une généralisation de ce jeu où  $k$  sommets peuvent être interrogés à chaque tour. Celle-ci est notamment liée à la notion de dimension métrique d'un graphe. Nous étudions aussi la variante où les distances relatives sont données comme réponses, qui généralise la dimension centroïdale des graphes. Pour les deux variantes, nous montrons que localiser la cible en un nombre minimum de tours est NP-complet en général, lorsque  $k$  est fixé.

Dans le cas des arbres (à  $n$  nœuds) et des distances exactes, nous montrons que le problème est NP-complet lorsque  $k$  fait partie de l'entrée. Nous donnons cependant une (+1)-approximation de ce problème : nous présentons un algorithme qui, en temps  $O(n \log n)$  (indépendant de  $k$ ), calcule une stratégie pour localiser la cible en au plus un tour de plus que l'optimal. Cet algorithme peut aussi être utilisé pour résoudre exactement le problème en temps  $n^{O(k)}$ .

**Mots-clefs :** Graphes, Dimension Métrique, Dimension Centroïdale, Jeux dans les graphes.

---

## 1 Introduction

L'essor récent des téléphones portables dans nos vies a favorisé l'émergence de plusieurs problématiques sociétales, comme celle de la localisation, qui est considérée dans ce papier. Supposons que l'on veuille localiser un émetteur au moyen de détecteurs. Si ces derniers déterminent parfaitement la distance (euclidienne) d'où un signal est émis, il suffit de trois détecteurs pour localiser exactement la source (par triangulation). Cependant, des facteurs peuvent rendre la localisation beaucoup plus compliquée : par exemple lorsque la métrique est celle d'un graphe (*i.e.*, les distances sont les longueurs de plus courts chemins dans un graphe) et/ou si le signal de l'émetteur est altéré par des obstacles, ce qui empêche les détecteurs de déterminer la distance exacte à l'émetteur. Nous considérons ce type de problème en étudiant le jeu suivant.

Imaginons un randonneur grièvement blessé et immobile, perdu et isolé dans un environnement modélisé par un graphe dont chaque sommet contient un détecteur. Pour le retrouver, des secouristes peuvent interroger à chaque tour du jeu un nombre limité de détecteurs. Dans le cas d'une communication parfaite, un détecteur retourne sa distance (dans le graphe) au randonneur. Sinon, nous ne pouvons que comparer la puissance des signaux reçus par les détecteurs interrogés (et en déduire lequel est le plus près du randonneur, lequel est le deuxième plus près, etc.). Le temps presse, et, sous toutes ces conditions, le problème est de retrouver le randonneur le plus vite possible (*i.e.*, en minimisant le nombre de tours).

Plus formellement, soient  $G = (V, E)$  un graphe et  $k \geq 1$  un entier. Nous considérons le jeu tour à tour suivant. Initialement, une *cible* invisible et immobile est placée secrètement sur un sommet  $c$  de  $G$ . À chaque tour, il est possible d'*interroger*  $k$  sommets  $v_1, \dots, v_k$  de  $G$ , pour recevoir, en réponse, la distance exacte de  $c$  à ceux-ci, c'est-à-dire  $(d_G(c, v_1), \dots, d_G(c, v_k))$  avec  $d_G(u, v)$  la longueur d'un plus court chemin de  $u$  à  $v$  dans  $G$ . Le but du jeu est de déterminer  $c$  en minimisant le nombre de tours. Nous notons  $\lambda_k^{ex}(G)$  ce nombre minimum de tours pour  $G$  et  $k$ . Par exemple, si  $P$  est un chemin, il suffit d'interroger une extrémité, *i.e.*,

---

<sup>†</sup>Par manque de place, nous omettons les preuves de nos résultats. Celles-ci se trouvent dans la version complète de ce papier : <https://hal.archives-ouvertes.fr/hal-01717629>

$\lambda_k^{ex}(P) = \lambda_1^{ex}(P) = 1$ . Dans le cas d'une étoile  $S_n$  à  $n$  feuilles, il faut et il suffit d'interroger toutes les feuilles sauf une, i.e.,  $\lambda_k^{ex}(S_n) = \lceil \frac{n-1}{k} \rceil$ .

Nous étudions également la variante de ce jeu dans laquelle l'interrogation de  $k$  sommets  $v_1, \dots, v_k$  donne, en réponse, les distances relatives de ceux-ci à  $c$ . Plus précisément, la réponse est une partition  $(V_1, \dots, V_p)$  où chaque  $V_j$  contient les sommets de  $\{v_1, \dots, v_k\} \setminus (V_1 \cup \dots \cup V_{j-1})$  les plus proches de  $c$ . Autrement dit, pour tous  $1 \leq i < j \leq k$ , nous apprenons qui de  $v_i$  ou  $v_j$  est le plus proche de  $c$  (ou s'ils sont à égale distance), sans connaître leur distance exacte à  $c$ . Soit  $\lambda_k^{rel}(G)$  le nombre minimum de tours nécessaires pour localiser la cible pour  $G$  et  $k$  dans ce cas. Notons que dans cette variante, il est nécessaire que  $k \geq 2$  dès que  $G$  a au moins deux sommets (en particulier, interroger le sommet abritant la cible permet de déterminer que ce sommet est le plus proche de la cible mais sans savoir *a priori* que la cible est là).

Pour ces deux problèmes, nous nous intéressons aussi aux paramètres duaux, dénotés  $\kappa_\ell^{ex}(G)$  et  $\kappa_\ell^{rel}(G)$ , qui, pour un  $\ell \geq 1$  fixé, représentent le plus petit nombre  $k$  tel que la cible peut être localisée en au plus  $\ell$  tours en interrogeant au plus  $k$  sommets par tour, lorsque ceux-ci fournissent leur distance exacte et relative, respectivement, à la cible. Notons que  $\lambda_k^{ex}(G) \leq \ell$  si et seulement si  $\kappa_\ell^{ex}(G) \leq k$ , et que  $\lambda_k^{rel}(G) \leq \ell$  si et seulement si  $\kappa_\ell^{rel}(G) \leq k$ . De plus,  $\lambda_k^{ex}(G) \leq \lambda_k^{rel}(G)$  et  $\kappa_\ell^{ex}(G) \leq \kappa_\ell^{rel}(G)$ .

**État de l'art.** De nombreux jeux liés à la localisation d'une cible ou d'un agent mobile dans un graphe ont été étudiés. Les deux problèmes considérés ici généralisent certains d'entre eux. Notre jeu avec les distances exactes est une généralisation de la notion de *dimension métrique* introduite indépendamment par Slater [Sla75] et Harary et Melter [HM76]. Plus précisément, la dimension métrique  $DM(G)$  d'un graphe  $G$  est la taille minimum d'un *ensemble résolvant*, i.e., un ensemble de sommets qu'il suffit d'interroger pour localiser la cible immédiatement (en un tour). Ainsi,  $DM(G)$  est exactement  $\kappa_1^{ex}(G)$ . Par ailleurs,  $\lambda_k^{ex}(G) \leq \lceil DM(G)/k \rceil$  puisqu'il est possible d'interroger séquentiellement les sommets d'un ensemble résolvant minimum. Cependant, il est possible de montrer que l'écart entre  $\lambda_k^{ex}(G)$  et  $\lceil DM(G)/k \rceil$  peut être arbitrairement grand (par exemple la famille de graphes dont est issu le graphe  $G$  de la Figure 1).

Le problème avec les distances relatives est, lui, une généralisation de la notion de *dimension centroïdale* introduite par Foucaud, Klasing et Slater [FKS14]. Plus précisément, la dimension centroïdale de  $G$  correspond à  $\kappa_1^{rel}(G)$ . Calculer la dimension métrique ou centroïdale d'un graphe est NP-complet en général [GJ79, FKS14], ce qui indique que les deux problèmes de calculer  $\kappa_1^{ex}(G)$  et  $\kappa_1^{rel}(G)$  sont difficiles.

Le problème de déterminer  $\lambda_1^{ex}(G)$  a aussi été considéré par Seager [Sea13], qui a donné la valeur exacte de ce paramètre pour des familles restreintes d'arbres. D'autres jeux proches sont étudiés dans la littérature, dans lesquels la cible peut se déplacer le long d'une arête à chaque tour. La minimisation du nombre de tours dans le cas d'une cible mobile avec  $k = 1$  et distances exactes a été considérée dans certaines topologies (e.g., arbres) [Sea12, CCD<sup>+</sup>12, Sea14, BDE<sup>+</sup>17]. La minimisation du nombre de sommets interrogés par tour pour que la localisation d'une cible mobile soit possible est étudiée dans [BGG<sup>+</sup>b] (distances exactes) et [BGG<sup>+</sup>a] (distances relatives).

**Résultats.** Nous initions l'étude de nos deux problèmes en montrant que calculer la valeur de  $\kappa_\ell^{ex}(G)$ ,  $\kappa_\ell^{rel}(G)$ ,  $\lambda_k^{ex}(G)$  ou  $\lambda_k^{rel}(G)$  est NP-complet pour tout  $k$  ou  $\ell$  fixé. Nous nous intéressons ensuite au problème avec les distances exactes, *a priori* plus simple. Dans le cas des arbres, nous résolvons complètement le problème de Seager [Sea13]. Plus précisément, dans le cas d'un arbre  $T$ , nous montrons, de manière surprenante, que la difficulté est de choisir les nœuds à interroger au premier tour. Plus précisément, lorsque  $k$  fait partie de l'entrée, déterminer  $\lambda_k^{ex}(T)$  demeure NP-complet même quand restreint à un arbre  $T$ . En revanche, nous donnons un algorithme opérant en temps polynomial qui calcule une stratégie "optimale à partir du deuxième tour d'interrogation". Plus précisément, notre algorithme constitue une  $(+1)$ -approximation au problème. Une autre conséquence de notre algorithme est que  $\lambda_k^{ex}(T)$  peut être déterminé en temps polynomial (en  $|V(T)|$ ) lorsque  $k$  est fixé. Nos résultats positifs restent valides même pour les arbres pondérés (lorsque les arêtes peuvent avoir des longueurs).

## 2 Difficulté générale des deux problèmes

La complexité de nos deux problèmes peut être attribuée à plusieurs facteurs. Tout d'abord, bien que plusieurs interrogations successives permettent de restreindre l'espace  $S$  auquel la cible appartient, il est

## Localiser une cible dans un graphe

possible de construire des situations dans lesquelles la prochaine interrogation optimale inclura des sommets hors de  $S$ . Ainsi, pour des familles de graphes closes par sous-graphes, une stratégie récursive naïve n'est pas forcément optimale. Un autre facteur rendant compte de la complexité du problème est que savoir jouer dans un graphe  $G$  n'implique pas de savoir jouer dans un sous-graphe isométrique  $H$  de  $G$ . Autrement dit, les deux problèmes ne sont pas clos par sous-graphes isométriques. Ainsi, réduire un graphe à une instance connue dans laquelle les distances entre sommets sont préservées n'aide pas toujours.

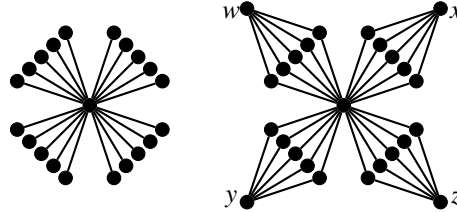


FIGURE 1: Exemple d'un sous-graphe isométrique  $H$  (gauche) d'un graphe  $G$  (droite).

Par exemple, la Figure 1 représente un graphe  $G$  (droite) contenant un sous-graphe isométrique  $H$  (gauche) vérifiant  $\lambda_4^{ex}(G) < \lambda_4^{ex}(H)$  et  $\lambda_4^{rel}(G) < \lambda_4^{rel}(H)$ . Nous considérons ici le cas  $k = 4$  par simplicité, mais la construction se généralise facilement à tout  $k$  (l'écart entre les paramètres peut être arbitrairement grand). Nous montrons que  $\lambda_4^{ex}(G) < \lambda_4^{ex}(H)$ . Notons que  $H$  est l'étoile  $S_{20}$  à 20 feuilles et donc  $\lambda_4^{ex}(H) = 5$ . Considérons maintenant  $G$ . Au premier tour, si nous interrogeons les sommets  $w, x, y$  et  $z$ , dans le pire des cas les positions restantes pour  $c$  forment un stable de taille 5 (cas où l'un des quatre sommets indique être à distance 1), impliquant la localisation de  $c$  au prochain tour. Ainsi,  $\lambda_4^{ex}(G) \leq 2$ . La preuve dans le cas des distances relatives est similaire.

**Complexité.** Nous montrons que déterminer  $\lambda_k^{ex}(G)$  et  $\lambda_k^{rel}(G)$  pour tout  $k$  (au moins 1 pour le premier paramètre, au moins 2 pour le second) fixé est NP-complet.

**Théorème 1.** Soient  $k \geq 1$  et  $k' > 1$  fixés. Les problèmes de déterminer  $\lambda_k^{ex}(G)$  et  $\lambda_{k'}^{rel}(G)$  sont NP-complets dans la classe des graphes  $G$  de diamètre au plus 2.

Soit  $\ell \geq 1$  fixé. Les problèmes de déterminer  $\kappa_\ell^{ex}(G)$  et  $\kappa_\ell^{rel}(G)$  sont NP-complets.

La première partie du théorème est prouvée par une réduction de 3-DIMENSIONAL MATCHING. La seconde est prouvée via deux réductions de METRIC DIMENSION et CENTROIDAL DIMENSION.

## 3 Distances exactes et arbres

Pour un arbre  $T$ , la principale difficulté du problème de déterminer  $\lambda_k^{ex}(T)$  provient exclusivement du premier tour, *i.e.*, du choix des  $k$  premiers nœuds à interroger. Précisément, nous montrons que trouver un bon ensemble de  $k$  nœuds à interroger au premier tour est NP-complet, par une réduction de HITTING SET.

**Théorème 2.** Étant donné un arbre  $T$  et un entier  $k \geq 1$ , le problème de déterminer  $\lambda_k^{ex}(T)$  est NP-complet.

De manière surprenante, nous donnons un algorithme polynomial qui calcule une stratégie "optimale à partir du deuxième tour d'interrogation". Cela vient du fait qu'avec les informations obtenues au premier tour, les positions potentielles restantes pour la cible se situent toutes à même distance d'un certain nœud  $r$ . Après une série de réductions, nous pouvons ensuite supposer, pour les tours suivants, que  $T$  est enraciné en un nœud  $r$ , que toutes les feuilles de  $T$  sont à même distance de  $r$  et que la cible se trouve sur une feuille.

**Lemme 1.** Soient  $k \geq 1$  et  $T$  un arbre (possiblement arête-pondéré) à  $n$  nœuds enraciné en  $r$ , dont les feuilles sont toutes à même distance de  $r$  et abritent la cible. Il existe un algorithme qui calcule en temps  $O(n \log n)$  (indépendant de  $k$ ) une stratégie optimale pour localiser la cible.

*Esquisse de preuve.* Notons  $v_1, \dots, v_d$  les voisins de  $r$ ,  $T_{v_i}$  le sous-arbre de  $T$  enraciné en  $v_i$  et  $S$  l'ensemble des feuilles. Étant donné que la cible occupe un nœud de  $S$  (dont la distance à  $r$  est connue), une propriété importante est qu'il est possible de savoir si la cible est dans  $T_{v_i}$  (ou non) simplement en interrogeant un unique (et arbitraire) nœud de  $T_{v_i}$ . Localiser la cible revient donc à identifier le sous-arbre  $T_{v_i}$  la contenant,

puis à répéter récursivement ce processus dans le sous-arbre où la cible a été détectée. Notons que lorsque nous testons si la cible est dans un sous-arbre  $T_{v_i}$ , il est parfois avantageux d’interroger plusieurs nœuds de  $T_{v_i}$  car cela permet de jouer *pleinement* le premier tour du jeu dans  $T_{v_i}$ , dans l’éventualité de la présence de la cible dans ce sous-arbre. Nous prouvons que si nous testons les sous-arbres de manière gloutonne, de ceux qui demandent le plus de tours (e.g., une étoile avec de nombreuses feuilles) à ceux qui en demandent le moins (e.g., un chemin), il en résulte une stratégie optimale.

Précisément, l’algorithme fonctionne par programmation dynamique. Pour chaque sous-arbre  $T_{v_i}$ , il calcule une stratégie optimale (prenant un nombre de tours minimum  $\ell_i$ ) pour détecter la cible sur une feuille de  $T_{v_i}$  ainsi que le nombre minimum  $\pi_i \leq k$  de nœuds qui doivent être interrogés lors du premier tour d’une telle stratégie optimale. Les sous-arbres sont ordonnés du plus grand au plus petit selon l’ordre lexicographique sur  $(\ell_i, \pi_i)$ . Notre algorithme simule une stratégie qui teste les sous-arbres dans cet ordre. Nous prouvons qu’une telle stratégie est optimale et que  $(\ell_r, \pi_r)$  peut alors être calculé “facilement”.  $\diamond$

De l’algorithme précédent, nous déduisons une (+1)-approximation du problème de déterminer  $\lambda_k^{ex}(T)$  pour un arbre  $T$ , et un algorithme polynomial lorsque  $k$  est fixé.

**Théorème 3.** *Soient  $T$  un arbre (possiblement arête-pondéré) à  $n$  nœuds, et  $k \geq 1$  un entier. Il existe un algorithme qui calcule, en temps en  $O(n \log n)$ , une stratégie pour localiser la cible en au plus  $\lambda_k^{ex}(T) + 1$  tours. De plus, en temps  $O(n^{k+2} \log n)$ , cet algorithme détermine  $\lambda_k^{ex}(T)$  et une stratégie optimale.*

*Esquisse de preuve.* Pour localiser la cible en au plus  $\lambda_k^{ex}(T) + 1$  tours en temps  $O(n \log n)$ , il suffit d’interroger un nœud arbitraire  $r$  au premier tour. Les nœuds pouvant accueillir la cible sont alors tous à même distance (connue) de  $r$ . Il est donc possible d’appliquer l’algorithme du Lemme 1.

Pour calculer  $\lambda_k^{ex}(T)$  et une stratégie optimale en temps  $O(n^{k+2} \log n)$ , il suffit d’essayer les  $\binom{n}{k}$  premiers tours possibles puis, pour chaque réponse qu’il est possible d’obtenir (il y en a au plus  $n$ ), d’appliquer l’algorithme du Lemme 1. L’instance pour laquelle le nombre maximum de tours est minimum donne alors  $\lambda_k^{ex}(T)$  et une stratégie optimale.  $\diamond$

## 4 Futures pistes de recherche

Notre résultat dans les arbres laisse ouverte la question de savoir si  $\lambda_k^{ex}(T)$  peut être calculé en temps FPT (i.e., en temps  $f(k) \cdot poly(n)$ ) dans les arbres  $T$  à  $n$  nœuds. De manière générale, il serait intéressant d’étudier la localisation d’une cible, en distances exactes, dans d’autres familles de graphes comme les graphes d’intervalles ou les graphes planaires.

Le problème avec les distances relatives semble beaucoup plus compliqué que celui avec les distances exactes, même pour des topologies simples. Vers une meilleure compréhension de celui-ci, une première étape serait de pleinement comprendre le cas des chemins (i.e., déterminer  $\kappa_1^{rel}(P)$  pour tout chemin  $P$ ), partiellement résolu dans [FKS14], avant de considérer une généralisation aux arbres.

## Références

- [BDE<sup>+</sup>17] A. Brandt, J. Diemunsch, C. Erbes, J. Legrand, and C. Moffatt. A robber locating strategy for trees. *Discrete Applied Math.*, 232 :99–106, 2017.
- [BGG<sup>+</sup>a] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokol, and M. Sleszynska-Nowak. Centroidal localization game. arXiv :1711.08836 (2017).
- [BGG<sup>+</sup>b] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokol, and M. Sleszynska-Nowak. Localization game on geometric and planar graphs. CoRR abs/1709.05904 (2017).
- [CCD<sup>+</sup>12] J. Carraher, I. Choi, M. Delcourt, L. H. Erickson, and D. B. West. Locating a robber on a graph via distance queries. *Theor. Computer Science*, 463 :54–61, 2012.
- [FKS14] F. Foucaud, R. Klasing, and P. J. Slater. Centroidal bases in graphs. *Networks*, 64 :96–108, 2014.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [HM76] F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combin.*, 2 :191–195, 1976.
- [Sea12] S. Seager. Locating a robber on a graph. *Discrete Math.*, 312 :3265–3269, 2012.
- [Sea13] S. Seager. A sequential locating game on graphs. *Ars Combin.*, 110 :45–54, 2013.
- [Sea14] S. Seager. Locating a backtracking robber on a tree. *Theor. Computer Science*, 539 :28–37, 2014.
- [Sl75] P. J. Slater. Leaves of trees. In *Proc. 6th Southeastern Conf. Combin., Graph Theory, Computing in Congressus Numer.*, volume 14, pages 549–559, 1975.