Proposition de stage M2 Informatique ENS Lyon 2015
Décroisements de graphes dans le plan avec obstacles

Thématique : Géométrie et topologie algorithmiques

Lieu du stage : Laboratoire d'informatique de l'École normale supérieure, Paris 5ème

Directeur de stage : Éric Colin de Verdière, `eric.colin.de.verdiere(at)ens.fr`, `http://www.di.ens.fr/~colin/`.

Présentation générale du domaine

Le sujet proposé s'inscrit dans le domaine de la topologie algorithmique des courbes et graphes tracés sur les surfaces. Durant les dix dernières années, des avancées significatives ont été faites, avec des algorithmes efficaces permettant par exemple de :

- calculer un graphe décomposant topologiquement une surface [EW05, CdV10],
- décider si deux courbes sont homotopes (se déforment continûment l'une en l'autre) [LR12],
- décider si deux graphes sont isotopes (se déforment continûment l'un en l'autre sans introduire d'auto-intersections) [CdVdM14],
- raccourcir autant que possible une courbe en la déformant continûment sur la surface [CdVE10].

Ces questions intéressent les communautés de géométrie algorithmique (décomposition et modélisation d'objets géométriques), de théorie topologique des graphes (étude combinatoire des graphes plongés sur une surface), d'algorithmique des graphes (pour des algorithmes efficaces pour la classe des graphes plongeables sur une surface donnée) et d'infographie (plaquage de textures, morphing).

Description du sujet

Le but du stage est de trouver des algorithmes relatifs aux graphes dessinés sur une surface, et à leurs déformations. La programmation de certains de ces algorithmes est envisageable, en fonction des goûts, mais pas nécessaire pour le succès du stage.

Étant donné un nombre fini O de points dans le plan (des *obstacles*), et un graphe dessiné dans $\mathbb{R}^2 \setminus O$ avec des croisements, est-il possible de déformer continûment le graphe pour supprimer les auto-intersections, sans passer au-dessus des obstacles ? On cherchera un algorithme efficace pour ce problème.

Dans le cas où il n'y a pas d'obstacles, la réponse est facile : Il suffit de vérifier que le graphe est planaire. On s'attend au moins à ce que certains cas particuliers soient résolubles assez efficacement.

Deux types de prolongements semblent intéressants :

- Si le graphe s'avère décroisable, on pourra étudier le nombre de mouvements élémentaires pour le décroiser, où la notion de « mouvement élémentaire » peut recouvrir plusieurs notions intéressantes [AFPR13, GKO+09] ;
- le même problème se pose dans le cas où l'on remplace le plan privé d'obstacles par une surface ; on s'attend à ce qu'il soit plus difficile dans ce cas.

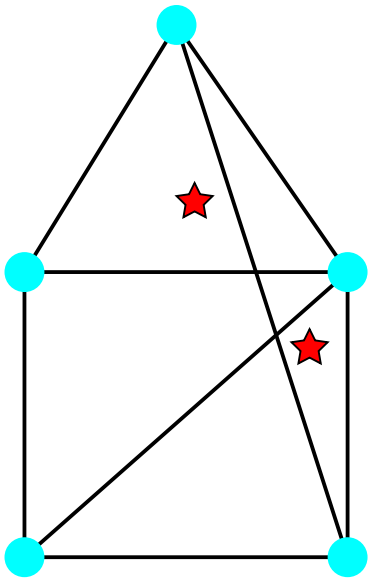

FIGURE 1 – Peut-on décroiser ce graphe sans passer au-dessus des obstacles (les étoiles) ?

Compétences espérées

De bonnes connaissances en algorithmique sont nécessaires. Des connaissances de base en topologie sont hautement souhaitables.

Références

[AFPR13] Patrizio Angelini, Fabrizio Frati, Maurizio Patrignani, and Vincenzo Roselli. Morphing planar graphs drawings efficiently. arXiv :1308.4291, 2013.

[CdV10] Éric Colin de Verdière. Shortest cut graph of a surface with prescribed vertex set. In *Proceedings of the 18th European Symposium on Algorithms (ESA), part 2*, number 6347 in Lecture Notes in Computer Science, pages 100–111, 2010.

[CdVdM14] Éric Colin de Verdière and Arnaud de Mesmay. Testing graph isotopy on surfaces. *Discrete & Computational Geometry*, 51(1) :171–206, 2014.

[CdVE10] Éric Colin de Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8) :3784–3813, 2010.

[EW05] Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1038–1046, 2005.

[GKO+09] Xavier Goaoc, Jan Kratochvíl, Yoshio Okamoto, Chan-Su Shin, Andreas Spillner, and Alexander Wolff. Untangling a planar graph. *Discrete & Computational Geometry*, 42(4) :542–569, 2009.

[LR12] Francis Lazarus and Julien Rivaud. On the homotopy test on surfaces. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 440–449, 2012.