

Dessin de graphes : suppression de la superposition des sommets avec PRISM

Encadrant : Bruno Pinaud <bruno.pinaud@labri.fr>

Contexte

La représentation automatique de graphes est un domaine de recherche à part entière. Son origine se situe dans la construction de circuit imprimés. De nos jours, on dessine des graphes dans de très nombreuses applications dont la visualisation d'informations relationnelles (réseaux d'interactions, réseaux sociaux, réseaux internet/informatiques,...) qui est une des activités du thème visualisation du LaBRI.

Plus formellement :

Entrée : un graphe $G=(V,E)$ avec un ensemble V de sommets et un ensemble E d'arêtes.

Sortie : un tracé de G lisible et compréhensible par l'utilisateur (bien souvent non expert des méthodes employées pour dessiner le graphe mais expert des données).

Les notions de lisibilité et compréhensibilité peuvent être définies formellement [1] notamment par la définition de critères esthétiques à respecter (minimisation du nombre de croisements d'arêtes pour le critère le plus efficace sur la lisibilité [2]. A noter que ce problème est NP-Complet [3]).

Travail à effectuer

La taille des sommets est un autre critère esthétiques à prendre en compte notamment si les sommets du graphe contiennent du texte ou des informations spécifiques à afficher. Néanmoins, la plupart des algorithmes de dessins de graphes réduisent les sommets à des points. Lorsque les sommets retrouvent leur taille normale à la fin de l'algorithme, plusieurs sommets peuvent se superposer rendant le graphe difficile à lire.

L'objectif de ce projet est d'étudier un algorithme de suppression de superposition de sommets appelé « PRISM » et de l'implémenter sous Tulip (<http://www.tulip-software.org>) en C++ (de préférence) ou en Python afin de pouvoir refaire les comparaisons présentées dans l'article. Les références de l'article sont :

Emden Gansner and Yifan Hu, Efficient, Proximity-Preserving Node Overlap Removal. Journal of Graph Algorithms and Applications, Vol. 14, no. 1, pp. 53-74, 2010.

Note sur l'implémentation : l'algorithme concurrent VPSC présenté dans l'article est disponible sous Tulip. Les parties de l'algorithme les plus difficiles à implémenter sont déjà présentes dans Tulip (triangulation de Delaunay et calcul des boîtes englobantes (bounding box) notamment).

Bibliographie

[1] Di-Battista, G.; Eades, P.; Tamassia, R. & Tollis, I.-G. Graph drawing -- Algorithms for the visualization of graphs. *Prentice-Hall*, 1999

[2] Purchase, H. Which aesthetic has the greatest effect on human understanding? *Proc. Graph Drawing'97*, Springer Verlag, 1997, 1353, 248-261

[3] Garey, M. & Johnson, D. Crossing number is NP-Complete. *J. Algebraic Discrete Methods*, 1983, 4, 312-316