



Projet individuel ou en binôme: arbre de Steiner dans un graphe, avec et sans restriction.

BM Bui-Xuan

Graphe géométrique : Un graphe géométrique dans un plan 2D est défini par un ensemble de points dans le plan appelés sommets, et un seuil sur la distance entre les points : il existe une arête entre deux sommets si et seulement si la distance Euclidienne entre les deux sommets est inférieure à ce seuil. Quand une arête existe entre deux sommets, le poids de l'arête est la distance Euclidienne entre les deux sommets dans le plan.

Problème de l'arbre de Steiner dans un graphe : Etant donné un graphe $G = (V, E)$ et un sous ensemble $S \subseteq V$ de sommets, le problème de l'arbre de Steiner couvrant S consiste à calculer un sous graphe de G qui est un arbre et qui passe par tous les points de S , tel que la longueur totale des arêtes de l'arbre est la plus petite possible.

Problème de l'arbre de Steiner avec restriction budgétaire, dans un graphe : Soient $G = (V, E)$ un graphe, $S \subseteq V$ un sous ensemble de sommets, $s \in S$ un point appelé maison-mère, et B un réel appelé budget. Le problème de l'arbre de Steiner couvrant S de budget B passant par s consiste à calculer un sous graphe de G qui est un arbre, qui passe par s , de longueur totale inférieure à B , passant par le plus grand nombre possible de points de S .

1 L'énoncé du projet

Il s'agit de proposer une heuristique pour le problème de l'arbre de Steiner dans un graphe géométrique, avec et sans restriction budgétaire. Dans le fichier canevas, V est la liste `points`, S est la liste `hitPoints`, le seuil sur la distance entre les sommets est `edgeThreshold`. Pour la version avec restriction budgétaire, le point "maison-mère" est le premier point de `hitPoints`, et le budget est le nombre mythique $B = 1664$. Il est obligatoire, pour les captures d'écran de rendu, d'utiliser exactement le fichier `input.points` disponible sur le site de l'énoncé du projet.

ATTENTION : l'évaluation du rendu de projet sera portée sur la distance totale de l'arbre (version sans budget) et le nombre total de points de S couverts par l'arbre (version avec budget), ainsi que la clarté du code. Il sera bon de bien commenter son code : expliquer pourquoi on écrit une ligne de code ; surtout ne pas répéter la ligne de code dans les commentaires sous une autre formulation en langue humaine...

Contraintes :

- A réaliser en individuel ou en binôme. Le plagiat est strictement interdit.
- Archiver la totalité du rendu en un seul fichier compressé contenant une capture d'écran et le code de chaque version (avec et sans budget), et tout ce dont on juge utile à la lecture du projet sans toute fois dépasser la dizaine de Méga-octet.
- Envoyer ce fichier à `buixuan@lip6.fr`, 3 emails maximum par rendu. L'utilisation des hébergeurs en ligne (drive et compagnies) est interdite. La nomination de préférence est `cpa-miniprojet-2021-NOM1-NOM2.piki`, où `piki` peut être un élément de `{tgz, zip, rar, 7z, etc}`.
- Deadline : 06 Mars 2021, 23h59, cachet de serveur de messagerie faisant foi. Pénalité de retard : malus de $2^k - 1$ points pour k minutes de retard.