

Un problème de positionnement d'étiquettes en temps réel sur une carte géographique

Entreprise

GIRO Inc.

Coordonnateur

Odile Marcotte
CRM et UQÀM

Référence

Un article intitulé *Fast Point-Feature Label Placement Algorithm for Real Time Screen Maps* (par M. Yamamoto, G. Camara et L. A. N. Lorena) est disponible sur la page des problèmes de l'atelier et peut être téléchargé.

Résumé

Sur une carte géographique, on affiche des étiquettes pour identifier des composantes de la carte (rues, clients, trajets, villes, quartiers, etc.). À peu près n'importe quelle composante de la carte peut posséder une étiquette. Le problème consiste à placer ces étiquettes de telle sorte que chaque étiquette soit proche de la composante qu'elle identifie, tout en chevauchant le moins possible les autres étiquettes. Le placement doit être calculé très rapidement puisqu'il doit être mis à jour chaque fois qu'un usager se déplace sur la carte ou effectue un zoom sur une portion de la carte. Un calcul prenant plus d'une demi-seconde est considéré comme lent. Nous nous intéressons à une généralisation de ce problème dans laquelle on spécifie des priorités pour les étiquettes, des préférences pour les positions, et peut-être un coût relatif pour les chevauchements.

Pour chaque étiquette à afficher, nous disposons d'une liste de positions permises. Chaque étiquette a un poids correspondant à son importance (relative à celle des autres étiquettes). Une étiquette d'importance élevée devra être moins chevauchée que les étiquettes de basse importance. Chaque position possède un poids correspondant à sa qualité. Le coût du chevauchement entre deux positions peut être absolu ou relatif; dans ce dernier cas, il est fonction du pourcentage de chevauchement par rapport à la plus petite étiquette. On peut alors construire un graphe dont les sommets sont les positions permises, et où deux sommets sont reliés par une arête si les positions correspondantes concernent la même étiquette ou se chevauchent.

Le poids d'un sommet est égal à la somme (ou au produit) de la priorité de l'étiquette et du poids de la position. Le coût d'une arête uv est égal à une très grande constante (dénotée M) si les positions correspondant aux sommets u et v concernent la même étiquette. Si non, le coût de l'arête uv est égal au produit d'une constante négative (dénotée $-C$), du maximum des poids des

sommets u et v , et de la somme des poids des sommets u et v (si l'on traite les chevauchements de manière absolue), ou bien au même produit multiplié par le pourcentage de chevauchement (si l'on traite les chevauchements de manière relative). Par exemple, si $-C$ égale -0.02 , les poids des sommets u et v sont respectivement de 10 et 8, et le pourcentage de chevauchement égale 50%, l'arête uv aura un coût de $-0.02 \cdot 0.5 \cdot 10 \cdot (10 + 8) = -1.80$. Si les poids des sommets u et v sont tous les deux égaux à 9, le coût de l'arête uv sera égal à -1.62 .

On cherchera donc à trouver, dans le graphe ainsi défini, le stable de poids maximal, où le poids d'un stable inclut les poids des sommets et des arêtes. Ce stable correspond au plus grand ensemble d'étiquettes placées de façon à produire un chevauchement de poids acceptable. On peut aussi ajouter la contrainte que toutes les étiquettes doivent être placées; cette contrainte peut avoir un impact important sur la stratégie de résolution.

Pour chaque étiquette, il faut au moins calculer sa position *préférée*. Toutefois, calculer des positions autres que celles-là est très coûteux en pratique, étant donné le court laps de temps que l'algorithme doit consommer. Il faut donc trouver une stratégie qui nous permette d'éviter de les calculer, dans toute la mesure du possible. L'approche que nous voulons mettre en oeuvre consiste des étapes suivantes:

1. générer les positions préférées de toutes les étiquettes,
2. s'il existe une position préférée dont les positions de rechange n'ont pas été calculées et qui est chevauchée par une autre position, alors calculer ses positions de rechange,
3. trouver un grand stable de poids maximal dans le graphe en appliquant l'algorithme glouton MIN (qui consiste à sélectionner itérativement le sommet de plus grand poids et plus petit chevauchement parmi les sommets potentiels, et à éliminer les sommets qui chevauchent celui sélectionné ou qui sont associés à la même étiquette),
4. insérer les étiquettes restantes aux positions de moindre coût, et
5. parcourir l'ensemble des étiquettes cinq fois et remplacer la position de l'étiquette courante par une position de la meilleure qualité possible.

Nous souhaitons explorer d'autres stratégies de résolution permettant d'éviter, autant que possible, de générer les positions de rechange. La stratégie décrite ci-dessus risque de provoquer la génération de positions de rechange pour la plupart des étiquettes.