Optimisation Combinatoire pour la conception de circuits

Arbres de Steiner

Alix Munier Kordon
Alix.Munier@lip6.fr

http://www-asim.lip6.fr/~alix/

Laboratoire Lip6
Université Paris 6/ Paris 12

Plan

- Définition et complexité
- Algorithmes approchés
- Programmation dynamique

Arbre de Steiner

Instance: $G = (S, E)$ un graphe non orienté connexe, $c : E \longrightarrow \mathbb{Z}^+$, un ensemble R de sommets requis.

Question: peut on trouver un arbre de Steiner $H = (S', A)$ (*ie.* un sous graphe partiel de G qui est un arbre et qui connecte les sommets de R) tel que $\sum_{a \in A} c_a$ soit minimum ?

Complexité

- NP-complet (même pour une grille, un graphe biparti);
- Polynomial pour un graphe serie-parallèle;
- Polynomial si $|R| = 2$ (plus court chemin);
- Polynomial si $R = S$ (arbre de coût min);

Algorithme approché

Soient $G = (S, E)$ et R;

- Calculler le graphe complet $G_1 = (R, A)$. $\forall \{x, y\} \in R^2, c(x, y)$ est le poids d'un plus court chemin dans G de x à y;
- Calculler l'arbre de coût min H_1 de G_1;
- En déduire l'arbre de coût min H de G;

Performance de l'algorithme approché

Theorème 1. *Si H_{opt} est un arbre de Steiner optimal pour G,*

$$c(H) \leq 2(1 - \frac{1}{l})c(H_{opt})$$

l est le nombre minimum de feuilles d'un arbre de Steiner optimal.

Si l'on considère une grille et la distance rectilinéaire, on obtient le ratio $\frac{3}{2}$ (Hwang).

Algorithme de programmation dynamique

Soit $G = (S, E)$, $R \subset S$, $v \in S - R$;

- $S(G, R, v)$: instance de l'arbre de Steiner minimum de sommets requis $R \cup \{v\}$; $taille = 2|R|$;
- $RS(G, R, v)$: instance de l'arbre de Steiner minimum de sommets R qui passe par v qui n'est pas une feuille. $taille = 2|R| - 1$;

Lemme de décomposition

Lemme 2. *Soit H une solution optimale de $RS(G, R, v)$. En supprimant v de H, on obtient plusieurs composantes connexes. Soient R_1 l'ensemble des sommets de R d'une de ces composante[s]. Alors, H est l'union de deux arbres de Steiner minimaux de G avec pour sommets requis respectifs $R_1 \cup \{v\}$ et $R_2 \cup \{v\}$.*

Petites tailles et décomposition

- Pour $S(G, R, v)$ avec $taille = 2$, $R = \{w\}$ et la solution optimale est le plus court chemin de à w;
  Si $taille > 2$, R contient plus de 2 sommets.
- Pour $RS(G, R, v)$ avec $taille = 1$, le problème est non définie.
  Si $taille > 1$, R contient plus de 2 sommets.

Union de sous-arbres

Si H_1 et H_2 sont deux sous-arbres de G avec pour sommets requis respectifs R_1 et R_2 disjoints, alors $Union(H_1, H_2)$ est un arbre obtenue en

- effectuant l'union des arêtes de H_1 et H_2,
- puis en recherchant le sous arbre de coût min sur ce graphe.

Décomposition de $RS(G, R, v)$

Soient R_1 et R_2 une partition de R.

- Resoudre $S(G, R_1, v)$; soit $H(R_1)$ l'arbre obtenu
- Resoudre $S(G, R_2, v)$; soit $H(R_2)$ l'arbre obtenu.

Soit $H(R_1, R_2) = Union(H(R_1), H(R_2))$

Corollaire 3. *La solution de $RS(G, R, v)$ est alors $H(R_1^*, R - R_1^*)$ de coût minimum dans l'ensemble $\{H(R_1, R - R_1), R_1 \subset R, R_1 \neq R\}$.*

Décomposition de $S(G, R, v)$

Soit $w \in S$ et $p(v, w)$ le plus court chemin de v à w;

- Si $w \in R$, résoudre $S(G, R - \{w\}, w)$.
- Sinon, résoudre $RS(G, R, w)$.

Soit $H(w)$ l'arbre obtenu et $H(v, w) = Union(p(v, w), H(w))$.

Lemme 4. *La solution de $S(G, R, v)$ est alors $H(v, w^*)$ de coût minimum dans l'ensemble $\{H(v, w), w \in S\}$.*

Evolution de la taille

Lemme 5. *Pour les deux décompositions, la taille des sous-problèmes est strictement inférieure à celle du problème original.*

Structure de données

Pour $v \in S - R$

- $Steiner[R, v]$= coût de la sol. de $S(G, R, v)$;
  $BestS[R, v] = R_1^*$;
- $RSteiner[R, v]$= coût de la sol. de $RS(G, R, v)$;
  $BestS[R, v] = w^*$;

Algorithme de calcul des structures

Pour tout $(v, w) \in R \times S$
$Steiner[\{v\}, w] := p(v, w)$
Pour $i := 2$ a $|R|$
calculler $RSteiner[R', v']$ et $BestRS[R, v]$,
$R' \subset R, |R'| = i$ et $v' \in S - R'$;
calculler $Steiner[R', v']$ et $BestRS[R, v]$,
$R' \subset R, |R'| = i$ et $v' \in S - R'$;

Algorithme de Construction

fonction `Construire`(R, v): H

Si $R = \{w\}$ **alors** $H = p(v, w)$

Sinon

$w^* = BestS[R, v];$

Si $w^* \in R$ **alors**

$H := Union(Construire(R - \{w^*\}, w^*),$
$p(w^*, v))$

Sinon

$H := Union(Construire(BestRS[R, w^*], w^*$
$Construire(R - BestRS[R, w^*], w^*), p(w^*, v$

retourner H

Complexité

- Calcul de $BestRS$: $O(n.3^{|}R|)$;
- Calcule de $BestS$:$O(n^2 2^r)$;