Problème du voyageur de commerce euclidien

Clarence KINEIDER

Leçons : 925, 928

Référence(s) : CORMEN, LEISERSON, RIVEST, STEIN, *Introduction à l'algorithmique.*

On définit le problème du voyageur de commerce euclidien (problème d'optimisation) :

TSPE | Entrée : Un graphe complet $G = (V, E)$, un poids $w : E \to \mathbf{N}$ qui respecte l'inégalité triangulaire, i.e. $\forall x, y, z \in V, w(x, z) \leq w(x, y) + w(y, z)$.
Sortie : Un cycle hamiltonien de poids minimal.

On va montrer que l'algorithme suivant est un algorithme d'approximation pour **TSPE** :

Algorithme 1 : TSPE_Approx

TSPE_Approx$(G = (V, E), w)$:
- Choisir $v \in V$ #le choix de v n'est pas important, il sert juste pour appeler **Prim**
- T $\leftarrow$ **Prim**(G, w, v) #T est un arbre couvrant minimal
- **return Parcours_Profondeur**(T) #le cycle est renvoyé sous forme de liste de sommets

Les appels à **Prim** et à **Parcours_Profondeur** s'effectuent en temps polynomial (leurs complexités respectives dépendent des structure de données utilisées, mais elles restent polynomiales en $|V| + |E|$). La complexité de **TSPE_Approx** est donc polynomiale en $|V| + |E|$.

Théorème : L'algorithme **TSPE_Approx** est un algorithme de 2-approximation pour **TSPE**.

Démonstration : Soit $G = (V, E)$ un graphe complet et $w : E \to \mathbf{N}$ qui respecte l'inégalité triangulaire. Soit c^* un cycle hamiltonien de poids minimal. Soit T l'arbre couvrant minimal calculé lors de l'exécution de **TSPE_Approx**. Si on retire une arête de c^*, on obtient un arbre couvrant. Ainsi, on a $w(T) \leq w(c^*)$ par minimalité du poids de T. Soit $\widetilde{c}$ le cycle associé au parcours en profondeur de T (voir FIGURE 1). Il passe deux fois par chaque arête de T, donc $w(\widetilde{c}) = 2w(T)$. Enfin, soit c le cycle renvoyé par **TSPE_Approx** (voir FIGURE 1). Par inégalité triangulaire, on a $w(c) \leq w(\widetilde{c})$. En mettant toutes les inégalités bout à bout, on obtient $w(c) \leq w(\widetilde{c}) = 2w(T) \leq 2w(c^*)$. □

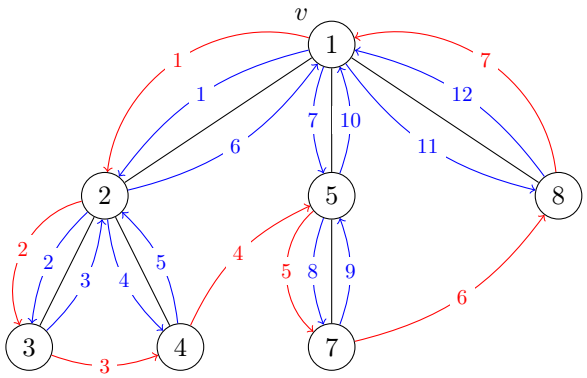

FIGURE 1 – Un exemple d'arbre couvrant T, avec en bleu le cycle $\widetilde{c}$ et en rouge le cycle hamiltonien c.

Moyennant une hypothèse forte sur l'entée, le problème du voyageur de commerce peut donc être résolu de manière approchée en temps polynomial. Étudions maintenant le cas général. On définit le problème du voyageur de commerce (problème de décision) :

TSP | Entrée : Un graphe $G = (V, E)$, un poids $w : E \to \mathbf{N}$.
Sortie : Un cycle hamiltonien de poids minimal.

Théorème : Si $\mathbf{P} \neq \mathbf{NP}$ alors il n'existe aucun algorithme de ρ-approximation en temps polynomial pour **TSP**, avec $\rho \geq 1$.

Démonstration : Supposons par l'absurde qu'il existe $\mathbf{A}$ un algorithme de ρ-approximation en temps polynomial pour **TSP**. On va utiliser $\mathbf{A}$ pour résoudre le problème du cycle hamiltonien (qui est **NP**-complet) en temps polynomial.

HAMCIRCUIT | Entrée : Un graphe $G = (V, E)$.
Sortie : Oui s'il existe un cycle hamiltonien dans G, non sinon.

Soit $G = (V, E)$ un graphe (une instance de **HAMCIRCUIT**). On pose $\widetilde{G} = (V, \widetilde{E})$ le graphe complet de sommets V. On pose également pour tout $u, v \in V, w(u, v) = \begin{cases} 1 \text{ si } (u, v) \in E \\ \rho|V| + 1 \text{ sinon} \end{cases}$.
Si G admet un cycle hamiltonien, alors il existe un cycle hamiltonien de poids $|V|$ dans $\widetilde{G}$. De plus, tous les cycles hamiltoniens de $\widetilde{G}$ qui passent par une arête qui n'est pas dans E sont de poids au moins $|V| - 1 + \rho|V| + 1 > \rho|V|$. L'algorithme $\mathbf{A}$ ne peut donc pas renvoyer un tel cycle s'il existe un cycle hamiltonien de poids $|V|$.
Donc G est une instance positive de **HAMCIRCUIT** si et seulement si $w(\mathbf{A}(\widetilde{G}, w)) \leq \rho|V|$. Le calcul de $\widetilde{G}$ et de w s'effectue en temps polynomial, de même que l'appel à $\mathbf{A}$ et le calcul du poids du chemin renvoyé. Donc **HAMCIRCUIT** $\in \mathbf{P}$, donc $\mathbf{P} = \mathbf{NP}$. □

Remarque :
En fait, on peut remplacer ρ par une fonction $\rho : \mathbf{N} \to [1, +\infty[$ calculable en temps polynomial. On a donc montré qu'il n'existe pas d'algorithme de $\rho(n)$-approximation en temps polynomial.