Problème du voyageur de commerce - approche dynamique

Benjamin Hellouin

Définition 1 (Problème du voyageur de commerce (PVC)).
Entrée : un graphe complet (V, E) *et une fonction de pondération* $w : E \to \mathbb{N}$ *;*
Sortie : un cycle <u>*hamiltonien*</u> *de poids minimal.*

1 Difficulté de PVC

Théorème 1. *Le problème PVC est NP-complet. Plus généralement, toute* ρ*-approximation de PVC pour* $\rho \in [1, \infty[$ *est NP-complète.*

Preuve : On réduit ce problème à celui de trouver un chemin hamiltonien dans un graphe quelconque.

Soit $G = (V, E)$ un graphe quelconque. On construit $G' = (V', E')$ le graphe complet sur le même ensemble de sommets : $V' = V$ et $E' = V \times V$. On définit

$$w : e \mapsto \begin{cases} 1 & \text{si } e \in E \\ \rho \cdot |G| + 1 & \text{sinon} \end{cases}$$

On remarque alors que :
- si G admet un cycle hamiltonien, alors G' admet un cycle hamiltonien de poids $|G|$;
- si G n'admet pas de cycle hamiltonien, alors G' admet un cycle hamiltonien de poids $\geq \rho \cdot |G| + 1$;

Ainsi, G admet un cycle hamiltonien si et seulement si il existe une ρ-approximation du cycle hamiltonien de poids minimal dans G' de poids $\leq \rho \cdot |G|$.

2 Programmation dynamique appliquée à PVC

L'approche par force brute consiste à énumérer l'ensemble des cycles hamiltoniens possibles, ce qui correspond à l'ensemble des permutations. On peut obtenir un algorithme en temps $O(n!)$.

On présente un algorithme basé sur la programmation dynamique permettant de réduire significativement cette complexité. On choisit arbitrairement un sommet s et on note $G(v, I)$ le chemin de poids minimal d'extrémités v et s, passant exactement par les sommets de I. Ainsi, $G(s, E)$ est la solution du problème.

Calcul de $G[v, I]$: Par argument de sous-structure optimale, on a $G[v, I] = \min_{v' \in I \setminus \{v,s\}} w(v, v') + G[v', I \setminus \{v\}]$ pour $I \neq \{v, s\}$. On a ainsi des dépendances décroissantes vis-à-vis de $|I|$. Pour $I = \{v, s\}$, on a évidemment $G[v, I] = w(v, s)$.

Chaque calcul requiert $O(n)$ opérations (où $n = |G|$) et on doit calculer une table de taille $n \cdot 2^n$. Ainsi, la complexité globale sera en $O(n^2 \cdot 2^n)$, ce qui est bien plus faible que $O(n!) = O(\sqrt{n} \cdot (\frac{n}{e})^n)$