TD n°12

2–3 mai 2018

Exercice 1 - hypergraph

A *hypergraph* H is a pair of sets (V, E) where V is the set of vertices and E is the set of hyperedges. Every hyperedge in E is a subset of V. In particular, an r-uniform hypergraph is one where the size of each edge is r. For example, a 2-uniform hypergraph is just a standard graph. A dominating set in a hypergraph H is a set of vertices $S \subset V$ such that $\ell \cap S \neq \emptyset$ for every edge $\ell \in E$. That is, S hits every edge of the hypergraph.
Let $H = (V, E)$ be an r-uniform hypergraph with n vertices and m edges. Show that there is a dominating set of size at most $np + (1-p)^r m$ for every real number $0 \leq p \leq 1$.

Exercice 2 - k-cut

We can generalize the problem of finding a large cut to finding a large k-cut. A k-cut is a partition of the vertices into k disjoint sets, and the value of a cut is the weight of all edges crossing from one of the k sets to another. Previously, we were considering 2-cut where all edges had the same weight 1. Generalize this argument to show that any graph G with m edges has a k-cut with value at least $(k-1)m/k$.

Exercice 3 - Chemins arête-disjoints

Supposons que n pairs d'utilisateurs sur un réseau (graphe) veulent communiquer sans congestion. Par là, on veut dire que le routage entre les paires doit emprunter des chemins qui deux à deux ne partagent pas une même arête (dits arête-disjoints). On met à leur disposition une collection F de m chemins empruntables.
Soit k le plus petit entier naturel tel qu'un chemin de F ne partage des arêtes qu'avec au plus k autres chemins de F
Donnez une condition reliant n, m et k pour qu'un tel routage soit possible.

Exercice 4 - LLL algorithmique

Soit $k \geqslant 40$ un entier pair. Le problème k-SAT est NP-complet (il l'est pour tout $k \geqslant 3$ fixé).
Si on suppose maintenant que chaque variable apparait dans au plus $T = 2^{k/4}$ clauses, on sait par LLL qu'il y a toujours une solution. Pour autant, on ne sait a priori pas comment en trouver une efficacement.
Proposez un algorithme en deux phases (c'est le premier indice) pour résoudre ces instances particulières en temps polynomial en espérance.