Exact Resolution of NP-hard Problems

Homework 3

Lecturer: Michael Lampis

The number of ★s in front of each exercise is an indicator of difficulty: (1=easy, 3=hard, $>$ 3=research level). Each student should submit his/her own solutions by email to (michail.lampis@dauphine.fr) by 28/2. Discussions and collaborations are allowed (and encouraged), but in your solutions you should include a short acknowledgement mentioning the names of people with whom you collaborated.

1. (★) Consider the following problem: Given a SAT formula ϕ where no variable appears negated (that is, a formula that does not contain $\neg$), decide if there exists a satisfying assignment which sets at most k variables to value 1. This problem is known as MONOTONE WEIGHTED SAT. (Monotone = no negations, Weighted = we care about the "weight", that is, number of true variables of the satisfying assignment).

- Show that this problem can be solved in roughly n^k time, where n is the number of variables.
- Show an FPT reduction from k-Dominating Set to this problem.
- Conclude that under the SETH this problem cannot be solved in $n^{0.99k}$ time.

2. (★) Consider the following problem: Given a 3-SAT formula ϕ, decide if there exists a satisfying assignment that sets at most k variables to value 1. Show that this problem is FPT, and there exists an algorithm to decide it running in time roughly 3^k. This problem is known as WEIGHTED 3-SAT. (Note that now we are not talking about the monotone version of the problem, so negations are allowed).

3. (★) Consider again the WEIGHTED 3-SAT problem of the previous question, but with the difference that now we are asking if there exists a satisfying assignment that sets **exactly** k variables to 1. Observe that there exists an n^k algorithm for this problem. Give an FPT reduction from k-Clique to this problem, establishing that, if the ETH is true, there is no $n^{o(k)}$ algorithm that solves it.

4. (★★) In the k-INDUCED PATH problem we are given a graph G and we seek a set of k vertices S such that $G[S]$ induces a path. Show that this problem has no $n^{o(k)}$ algorithm, assuming the ETH.

5. (★★) Super Vertex Cover is the following problem: we are given a graph $G = (V, E)$ and for each edge $e \in E$ we are given a weight $w(e) \in \mathbb{N}$. For each $v \in V$ we need to select a power level, that is, a number $p(v)$ such that we have the following: all edges are covered, that is, for each $uv \in E$ we have $p(u) \geq w_{uv}$ or $p(v) \geq w_{uv}$; the sum $\sum_{v\in V} p(v)$ is minimized. Notice that this problem is exactly the standard vertex cover problem if all edge weights are equal to 1.

Give an algorithm that solves this problem in $n^{k+O(1)}$ time, where k is the feedback vertex set of the input graph. (★★★) Is there an FPT algorithm for the same problem and the same parameter k?