Parameterized algorithms for constraint satisfaction problems
Exercises

Marcin Pilipczuk

Day 1: simple and above guarantee parameterizations

1. Design a *deterministic* polynomial-time algorithms that, given a CNF formula ϕ, finds an assignment that satisfies at least $\sum_{i \geq 1}(1 - 2^{-i})m_i$ clauses, where m_i is the number of clauses with i literals.

2. Prove that CNF MaxSAT is FPT when parameterized by the number of clauses.

   Note: the difficulty here is that clauses have unbounded (in parameter) length.

3. In the Not-All-Equal MaxSAT, the input is a multiset of clauses, and a clause is satisfied if not all literals in the clause evaluate to the same value (i.e., not all are true and not all are false). Prove that Not-All-Equal MaxSAT is FPT when parameterized by the number of clauses.

4. Prove that CNF MaxSAT is FPT when we ask to satisfy at least $m/2+k$ clauses and we parameterize by k, where m is the number of clasues.

5. Consider now a MaxSAT problem, where the domain is $\{0, 1\}$ and the only available constraint is a binary inequality $x \neq y$.

   - Represent the MaxSAT problem as a graph problem. What do you obtain?
   - Prove that there is always an assigment satisfying at least $m/2$ constraints, where m is the number of constraints on input.
   - Prove that MaxSAT is FPT when we ask to satisfy at least $m/2 + k$ constraints and we parameterize by k.

     Hint: consider a maximum matching in the graph problem you obtained. Prove that on one hand, MaxSAT *is FPT when parameterized by the size of the maximum matching, and on the other hand the answer is always positive if the matching is large (compared to the parameter k).*

6. Consider now a MaxSAT problem, where the domain is $\mathbb{Q}$ and the only available constraint is $x < y$.

   - Represent the MaxSAT problem as a (directed) graph problem. What do you obtain?
   - Prove that there is always an assigment satisfying at least $m/2$ constraints, where m is the number of constraints on input.
   - Prove that MaxSAT is FPT when we ask to satisfy at least $m/2 + k$ constraints and we parameterize by k.

Day 2: parameterization by the number of unsatisfied constraints

1. At the lecture, we have discussed the Vertex Cover problem, parameterized by the excess above the LP lower bound, i.e., by $k := p - LP$, where p is the budget and LP is the value of the linear programming relaxation.

We used a result of Nemhauser and Trotter that asserts that (a) there exists an optimal value to the LP relaxation that uses only values 0, 1, and 1/2; (b) for every optimal solution to the LP relaxation, there exists a minimum-size vertex cover that contains all vertices assigned 1 and no vertices assigned 0. This allowed us to reduce to the case, where there an LP solution that assigns 1/2 to every vertex is the unique optimal LP solution.

We argued that then a branching on an arbitrary vertex reduces k by at least 1/2 in every branch. This have $4^k \cdot \mathrm{poly}(n)$-time algorithm. In this exercise we analyse further the branch where the pivot v is not taken to the solution (and $N(v)$ is taken to the solution).

- Prove that, if all-1/2 is the unique optimal solution to the LP relaxation, then the following is true: for every independent set $I \subseteq V(G)$, we have $|N(I)| > |I|$ and, furthermore, for every $u \in N(I)$, there exists a matching of size $|I|$ in the bipartite subgraph of G between I and $N(I) \setminus \{v\}$.
- Prove that, if in this branch the gap k decreased by only 1/2, then there exists an independent set $I \subseteq V(G)$ with $|N(I)| = |I| + 1$.
- Design a reduction rule reducing (without branching) such an independent set as in the previous point.
- Design an algorithm with running time bound $c^k \cdot \mathrm{poly}(n)$ for some $c < 3$.

2. Consider the following two problems:

   - In the Odd Cycle Transversal problem, we are given an undirected graph G and an integer k, and we ask for a set $X \subseteq V(G)$ of size at most k such that $G - X$ is bipartite.
   - In the Split Vertex Deletion problem, we are given an undirected graph G and an integer k, and we ask for a set $X \subseteq V(G)$ of size at mot k such that $G - X$ is a split graph. (A graph H is a split graph if $V(H)$ can be partitioned into $K \uplus I$ such that $G[K]$ is complete and $G[I]$ is edgeless.)

   For each of these problems, parameterized by k, design a parameterized reduction to the Vertex Cover problem, parameterized as in the previous exercise. Your reduction should preserve the value of the parameter k. Deduce that these problems admit a parameterized algorithm with running time bound $c^k \cdot \mathrm{poly}(n)$ for some constant $c < 3$.

3. Design a parameterized reduction from Joker 2-SAT to Almost 2-SAT that keeps the parameter unchanged.

   Note that in the lecture we have used a reduction in the other direction.

4. In the Bundled Cut problem, we are given a directed graph G with distinguished vertices $s, t \in V(G)$, an integer k, and a family $\mathcal{B}$ of pairwise disjoint subsets of $E(G)$ called *bundles*. The goal is to find an st-cut $Z \subseteq \bigcup \mathcal{B}$ (i.e., the edges not in a bundle are undeletable) that intersects at most k bundles.

   Consider a restriction of Bundled Cut, where every bundle is a path of length at most ℓ.

   - Prove that for $\ell = 2$, the problem is equivalend to the Minimum Cut problem, and thus solvable in polynomial time.
   - For $\ell = 3$, provide an example where every solution is not a minimum cut.
   - Design an NP-hardness reduction for the case $\ell = 3$.

Day 3: applications of flow-augmentation

1. In the Bicriteria MinCut problem, we are given a directed graph G, vertices $s, t \in V(G)$, weights $w : E(G) \to \mathbb{Z}_+$, and integers $k, W \in \mathbb{Z}_{\geq 0}$; the goal is to find an st-cut $Z \subseteq E(G)$ with $|Z| \leq k$ and $w(Z) \leq W$.

- Prove that Bicriteria MinCut is polynomial-time solvable in the special case when k is the minimum possible cardinality of an st-cut.
- Prove that Bicriteria MinCut is FPT when parameterized by k.

2. Recall the Multicut problem: the input consists of an undirected graph G, integer k, and a set $\mathcal{T} \subseteq \binom{V(G)}{2}$ of terminal pairs; the goal is to find a set $Z \subseteq E(G)$ of size at most k such that for every $st \in \mathcal{T}$, the vertices s and t are in different connected components of $G - Z$.

   By a standard iterative compression trick, you can assume that you are also given on input a set $M \subseteq V(G)$ of size at most $k+1$ such that for every $st \in \mathcal{T}$, either $s \in M$, or $t \in M$, or s and t are in different connected components of $G - M$.

   Prove that Multicut is FPT by reducing it to the tractable case of MinUnSAT in ID_2.

   Hints: The first step should be a standard branching step, where you branch how M is split among the connected components of $G - Z$; you can merge vertices that land in the same connected component of Z, reducing to the case when every connected component of $G - Z$ contains at most one vertex of M. After this step, your MinUnSAT *instance should have a variable $x(v,p)$ for every $v \in V(G)$ and $p \in M$ with the intended meaning "v is in the connected component of $G - Z$ that contains p".*

3. Recall the Directed Subset Feedback Arc Set problem: the input consists of a directed graph G with a set $F \subseteq E(G)$ of *red edges* and an integer k; the goal is to find a set $Z \subseteq E(G)$ of size at most k such that in $G - Z$ there every arc of F has endpoints in different strong components.

   By a standard iterative compression trick, you can assume that you are also given on input a set $M \subseteq V(G)$ of size at most $k+1$ such that $G - M$ has no cycles containing red edges.

   Prove that DSFAS is FPT by reducing it to Bundled Cut with Pairwise Linked Deletable Edges.

   Hints: The first step should be a standard branching step, where you branch how M is split among the strong components of $G - Z$ and how they ordered in a topological order of strong components of $G - Z$; you can merge vertices that land in the same strong component of $G - Z$, reducing to the case when every strong component of $G - Z$ contains at most one vertex of M, and the topological order of the vertices of M is known. After this step, you should follow as in the reduction for Directed Feedback Arc Set *in the lecture, but you will need $2|M| - 1$ copies of G: one for each vertex of M and one between each two consecutive vertices of M.*

4. Consider a temporal CSP with a single relation $R(x,y,z) = (x > y) \vee (x > z)$ (which is equivalent to $x > \min(y,z)$). Prove that for this language, MinUnSAT is W[2]-hard.

   Hint: reduce from Hitting Set. *In this W[2]-hard problem, you are given a universe U, a family $\mathcal{F}$ of subsets of U, and an integer k; the goal is to find $X \subseteq U$ of size at most k such that $X \cap A \neq \emptyset$ for every $A \in \mathcal{F}$.*

Day 4: parameterization by the number of variables

1. For every $n \geq k \geq 1$, show that there is an embedding of an n-vertex clique into a $k \times k$ grid of congestion $2\lceil n/k \rceil$.

   Note: the CSP problem, obtained via this reduction, is often called Grid Tiling *and it is a common intermediate problem for proving hardness of problems in planar graphs.*

2. Consider the following problem: given a graph G embedded on a torus and integers r, k, check if G contains a set X of k vertices such that pairwise distance between the vertices of X is more than r. Prove that, assuming the Exponential Time Hypothesis, this problem does not admit an algorithm with running time $|V(G)|^{o(\sqrt{k})}$.

3. In the Odd Set problem, we are given a universe U, a family $\mathcal{F}$ of subsets of U, and an integer k; the goal is to find a set $X \subseteq U$ of size at most k such that for every $A \in \mathcal{F}$, $|A \cap X|$ is odd. Prove that this problem, parameterized by k, is W[1]-hard.

4. In the Strongly Connected Steiner Subgraph problem, we are given a directed graph G, an integer k, and a set $T \subseteq V(G)$; the goal is to find a strongly connected subgraph H of G that contains T and contains at most k vertices. Prove that this problem, parameterized by k, is W[1]-hard.

5. Show that if H is a k-vertex cubic graph, then the average distance between two vertices of H is $\Omega(\log k)$.

Day 5: exploring temporal CSPs

In all problems, the domain is $D = \mathbb{Q}$.

1. Consider a language $\Gamma = \{x < y\}$. Prove, that MaxSAT can be solved in time $2^n \cdot \mathrm{poly}(n)$, where n is the number of variables.

2. Consider a language $\Gamma = \{x < y \wedge y < z\}$. Prove, that MaxSAT can be solved in time $c^n \cdot \mathrm{poly}(n)$ for some constant c, where n is the number of variables.

3. For an integer $a \geq 1$, let R_a be a constraint of arity $2a$ defined as

$$R_a(x_1, y_1, \ldots, x_a, y_a) = \bigvee_{i=1}^{a} (x_i < y_i).$$

   Prove that, unless ETH fails, Satisfiability for the language $\{R_4\}$ cannot be solved in time $2^{o(n \log n)}$, where n is the number of variables.

   Hint: Reduce from $k \times k$ Clique. In $k \times k$ Clique, we are given a graph G with $V(G) = [k] \times [k]$ and we ask for a function $\beta : [k] \to [k]$ so that $\{(i, \beta(i)) \mid 1 \leq i \leq k\}$ induces a clique. It follows from our reductions discussed on Thursday that, unless ETH fails, this problem does not admit an algorithm with running time bound $2^{o(k \log k)}$.

   Remark: The existence of an algorithm with running time bound $2^{o(n \log n)}$ for Satisfiability is open for languages $\{R_2\}$ and $\{R_3\}$. Note that $\{R_1\}$ is the language discussed in Problem 1.

4. Consider a temporal CSP with a single relation $R(x, y, z) = (x > y) \vee (x > z)$ (which is equivalent to $x > \min(y, z)$). Prove that for this language, MinUnSAT is W[2]-hard.

   Hint: reduce from Hitting Set. In this W[2]-hard problem, you are given a universe U, a family $\mathcal{F}$ of subsets of U, and an integer k; the goal is to find $X \subseteq U$ of size at most k such that $X \cap A \neq \emptyset$ for every $A \in \mathcal{F}$.

5. Consider a temporal CSP with a relation $x = y$ and a relation $R(x, y, p, q) = (x = y) \to (p < q)$. Prove that for this language MinUnSAT is W[1]-hard.

6. Consider a temporal CSP with a relation $x = y$ and a relation $R(x, y, z) = (x = y) \to (x < z)$. Prove that for this language MinUnSAT is W[1]-hard.

7. Consider a temporal CSP with a single relation $R(x, y, p, q) = (x < y) \wedge (p < q) \wedge (x < q) \wedge (p < y)$. Prove that for this language MinUnSAT is FPT.

   Hint: reduce the problem to Directed Feedback Arc Set.

8. Consider a temporal CSP with a single relation $R(x, y, p, q) = (x < y) \wedge (p < q) \wedge (x \leq q) \wedge (p \leq y)$. Is MinUnSAT FPT or W[1]-hard?

   Note: I don't know the answer to the last problem. To the best of my knowledge, this is the smallest open case in the FPT vs W[1]-hard dichotomy for temporal CSPs, within the tractable fragment of the P vs NP-hard dichotomy of languages preserved by the "minimum" polymorphism.