Exact Algorithms via Monotone Local Search

Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh

In this paper we study relationship between parameterized algorithms and exact exponential algorithms. We will prove that if we have a good parameterized algorithms for some problem $\mathcal{Q}$ then we have also good exact exponential algorithms for $\mathcal{Q}$.

Definition 1. An *implicit set system* is a function Φ that takes a string $I \in \{0,1\}^*$ as an input and it outputs a set system $(U_I, \mathcal{F}_I)$ where U_I is a universe and $\mathcal{F}_I$ is a collection of subsets of U_I.

We consider only polynomial computable implicit set systems, i.e., given I we can compute U_I in polynomial time (in $|I|$) and given $S \subseteq U_I$ we can decide if $S \in \mathcal{F}_I$ in polynomial time. For an implicit set system Φ we define the following problem.

PROBLEM:	Φ-Extension
Input:	An instance $I \in \{0,1\}^*$, a set $X \subseteq U_I$ and $k \in \mathbb{N}$.
Question:	Is there a subset $S \subseteq U_I \setminus X$ such that $S \cup X \in \mathcal{F}_I$ and $|S| \leq k$.

By $N = |I|$ we denote the size of the instance and by $n = |U_I|$ we denote the size of the universe. The main result is summarized by the following theorem.

Theorem 2. *If there exists an algorithm for* Φ-Extension *with running time* $c^k N^{\mathcal{O}(1)}$ *then there exists an algorithm for* Φ-Extension *with running time* $(2 - \frac{1}{c})^{n+o(n)} N^{\mathcal{O}(1)}$.

We prove Theorem 2 in two steps. First, we give a randomized algorithm $\mathcal{A}$ such that it solves Φ-Extension for an instance (I, X, k) which runs in time $(2-\frac{1}{c})^{n-|X|} N^{\mathcal{O}(1)}$ and is always correct on *no*-instances and is correct on *yes*-instances with a probability greater than $\frac{1}{2}$. Then we discuss a derandomization of the algorithm $\mathcal{A}$ at cost of a subexponential factor $(2 - \frac{1}{c})^{o(n)}$.

Randomized Algorithm

Let $\mathcal{B}$ be a paremterized algorithm for Φ-Extension given by assumptions of Theorem 2. Let (I, X, k) be an instance of Φ-Extension and $k' \leq k$. The main procedure $\mathcal{P}(k')$ (which will be repeated many times) of the exact algorithm $\mathcal{A}$ consists of two steps.

1. Choose an integer $t = t(c, n, k', |X|)$. Select a random subset $Y \subseteq U_I \setminus X$ of size t.
2. Run algorithm $\mathcal{B}$ on the instance $(I, X \cup Y, k' - t)$ and return the answer.

Let $\kappa(m,p,q) = \binom{m}{q}/\binom{p}{q}$ for $0 \leq q \leq p \leq m$. For each $k' \leq k$, the algorithm $\mathcal{A}$ repeat the procedure $\mathcal{P}(k')$ $\kappa(n - |X|, k', t)$-times and returns yes if some run of $\mathcal{P}$ returned yes. By the choice of the repetition number we get the correct bound for the probability of success. The parameter t determines a trade-off when is cheaper to add random vertices to the solution and repeat the subroutine $\mathcal{P}$ and when is cheaper to compute the solution exactly by the algorithms $\mathcal{B}$. By the right choice of t we can put down the running time below the required bound.

Lemma 3. *Let $c \geq 1$ be a constant and $m \in \mathbb{N}$. Then,*

$$\max_{0 \leq p \leq m} \min_{0 \leq q \leq p} \kappa(m,p,q) c^{p-q} \leq \left(2 - \frac{1}{c}\right)^m m^{\mathcal{O}(1)}.$$

Corollary 4. *The running time of the algorithm $\mathcal{A}$ is bounded by* $(2 - \frac{1}{c})^{n-|X|} N^{\mathcal{O}(1)}$.

Derandomization

For the derandomization of the algorithm $\mathcal{A}$ we need to enumerate all subsets Y of $U_I \setminus X$ of size t such that for every subset $S \subseteq U_I \setminus X$ of size k' there exists at least one Y such that $Y \subseteq S$.

Definition 5. Let U be a universe of size m and let $0 \leq q \leq p \leq m$. A family $\mathcal{C} \subseteq \binom{U}{q}$ is an (m, p, q)-*inclusion family* if for every $S \in \binom{U}{p}$ there is a set $Y \in \mathcal{C}$ such that $Y \subseteq S$.

Theorem 6. *There is an algorithm that given m, p and q outputs an (m, p, q)-inclusion family $\mathcal{C}$ of size $\kappa(m, p, q)2^{o(m)}$ in time $\kappa(m, p, q)\ 2^{o(m)}$.*

Instead of $\kappa(n - |X|, k', t)$ repetitions of $\mathcal{P}(k')$ the algorithm $\mathcal{A}$ loops over all $Y \in \mathcal{C}$. Thus, the running time of the algorithm $\mathcal{A}$ is bounded by $(2 - \frac{1}{c})^{n+o(n)} N^{\mathcal{O}(1)}$.

The algorithm from Theorem 6 is constructed in two steps. First, we give an inefficient algorithm for the inclusion family of a small size using an approximation algorithm for Set Cover. Then, we decrease the size of the universe using hash functions and decrease the time of the construction of (m, p, q)-inclusion family.

PROBLEM:	Set Cover
Input:	A universe $\mathcal{V}$, a collection $\mathcal{T}$ of subsets of $\mathcal{V}$.
Output:	Minimum sized sub-collection $\mathcal{T}' \subseteq \mathcal{T}$ which covers $\mathcal{V}$, i.e., $\bigcup_{T \in \mathcal{T}'} T = \mathcal{V}$.

Theorem 7. *There is an $\mathcal{O}(\log |\mathcal{V}|)$-factor approximation algorithm for* Set Cover *which runs in time $\mathcal{O}(|\mathcal{V}| + \sum_{T \in \mathcal{T}} |T|)$.*

Definition 8. Let U be a set and $b \in \mathbb{N}$. Let $\mathcal{H}$ be a collection of hash functions $U \to [b]$. The collection $\mathcal{H}$ is *pair-wise independent* if for every $i, j \in [b]$ and every distinct $u, v \in U$ holds that

$$\Pr_{f \in \mathcal{X}}[f(u) = i, f(v) = j] = \frac{1}{b^2}.$$

Theorem 9. *There is a polynomial time algorithm that given a universe U and a prime b constructs a pair-wise independent collection $\mathcal{H}$ of hash functions $U \to [b]$ such that $|\mathcal{H}| = b^2$.*