Exact Algorithms Via Monotone Local Search

\[ c^k \Rightarrow (2 - \frac{1}{c})^n \]

Fedor Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh

Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary (MTA SZTAKI)

FPT seminar, July 4, 2016
Should someone mainly interested in exact algorithms care about parameterized complexity?

- Well yes, parameterized algorithms are exact algorithms...
- ... and a $c^k$ FPT algorithm gives a $c^n$ algorithm, if $k \leq n$
Should someone mainly interested in exact algorithms care about parameterized complexity?

- Well yes, parameterized algorithms are exact algorithms...
- ... and a $c^k$ FPT algorithm gives a $c^n$ algorithm, if $k \leq n$
- More interestingly, when $c < 4$,

$$\max_{0 \leq \alpha \leq 1} \left\{ \min\{c^\alpha, 2^{H(\alpha)}\} \right\} < 2$$
Say you want to solve a subset problem; i.e., select $k$ elements from $n$ to satisfy/optimize a property/objective value.

You may:

- use the best known **FPT algorithm in** $c^k$
- exhaustively try all $\binom{n}{k}$
Say you want to solve a subset problem; i.e., select $k$ elements from $n$ to satisfy/optimize a property/objective value.

You may:

- use the best known FPT algorithm in $c^k$
- exhaustively try all $\binom{n}{k}$

\[
\max_{0 \leq k \leq n} \left\{ \min \{ c^k, \binom{n}{k} \} \right\} = \max_{0 \leq \alpha \leq 1} \left\{ \min \{ c^{\alpha n}, \binom{n}{\alpha n} \} \right\}
\]
Say you want to solve a subset problem; i.e., select $k$ elements from $n$ to satisfy/optimize a property/objective value.

You may:

- use the best known FPT algorithm in $c^k$
- exhaustively try all $\binom{n}{k}$

$$\max_{0 \leq k \leq n} \{\min\{c^k, \binom{n}{k}\}\} = \max_{0 \leq \alpha \leq 1} \{\min\{2^{\alpha \log c}, 2^H(\alpha)\}\}^n$$

where $H(x) = -x \log x - (1 - x) \log(1 - x)$

since $\binom{n}{\alpha n} \leq 2^{H(\alpha)n}$
$\log \text{ of the basis}$

- $H(\alpha)$
- $\alpha \log c$

$3^k$ implies $1.953^n$
\[ \alpha \log c \implies 1.709^n \]
\[ \log c \left( \alpha \right) = \alpha \log c \]

1.27^k \text{ implies } 1.253^n
\[
\log \text{ of the basis } \quad H(\alpha) \\
= \alpha \log c
\]

\[3.5^k \text{ implies } 1.991^n\]
\[ \log \text{ of the basis} \]

\[ 4^k \text{ implies } 2^n \]
$\log$ of the basis

$H(\alpha)$

$\alpha \log c$

$4.5^k$ implies nothing
A very simple algorithm inspired by local search

Let Π be a *subset* problem.

- Guess the size of the optimal solution $k$.
- Select $t \leq k$ elements *uniformly at random*.
- Complete the solution with $k - t$ elements in **FPT time** $c^{k-t}$. 
A very simple algorithm inspired by local search

Let $\Pi$ be a \textit{subset} problem.

- Guess the size of the optimal solution $k$.
- Select $t \leq k$ elements \textit{uniformly at random}.
- Complete the solution with $k - t$ elements in \textit{FPT} time $c^{k-t}$.

Slight caveat: \textit{FPT algorithm} for the \textit{extension version} of $\Pi$. 
Analysis

Probablity of guessing $t$ good elements $\geq$
Analysis

Probability of guessing *t good* elements $\geq \binom{k}{t} \binom{n}{t}$
Analysis

Probability of guessing $t$ good elements $\geq \frac{k}{n}$

running time $\leq \max_{0 \leq k \leq n} \min_{0 \leq t \leq k} \frac{n}{t} \frac{k}{t} c^{k-t}$
Analysis

Probability of guessing $t$ good elements $\geq \frac{k}{n}$

running time $\leq \max_{0 \leq k \leq n} \min_{0 \leq t \leq k} \frac{n}{t} c^{k-t}$

So?
Analysis

Probablity of guessing $t$ good elements $\geq \frac{k\binom{n}{t}}{n\binom{k}{t}}$

running time $\leq \max_{0 \leq k \leq n} \min_{0 \leq t \leq k} \frac{n\binom{n}{t}}{k\binom{k}{t}} c^{k-t}$

So?

$\max_{0 \leq k \leq n} \min_{0 \leq t \leq k} \frac{n\binom{n}{t}}{k\binom{k}{t}} c^{k-t} \leq (2 - \frac{1}{c})^n$ (whiteboard)
## Breakthrough!

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>Parameterized</th>
<th>New bound</th>
<th>Previous Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Feedback Vertex Set</strong></td>
<td>$3^k (r)$</td>
<td>$1.6667^n (r)$</td>
<td></td>
</tr>
<tr>
<td><strong>Feedback Vertex Set</strong></td>
<td>$3.592^k$</td>
<td>$1.7217^n$</td>
<td>$1.7347^n$</td>
</tr>
<tr>
<td><strong>Subset Feedback Vertex Set</strong></td>
<td>$4^k$</td>
<td>$1.7500^n$</td>
<td>$1.8638^n$</td>
</tr>
<tr>
<td><strong>Feedback Vertex Set in Tournaments</strong></td>
<td>$1.6181^k$</td>
<td>$1.3820^n$</td>
<td>$1.4656^n$</td>
</tr>
<tr>
<td><strong>Group Feedback Vertex Set</strong></td>
<td>$4^k$</td>
<td>$1.7500^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Node Unique Label Cover</strong></td>
<td>$</td>
<td>\Sigma</td>
<td>^2_h$</td>
</tr>
<tr>
<td><strong>Vertex $(r, \ell)$-Partization $(r, \ell \leq 2)$</strong></td>
<td>$3.3146^k$</td>
<td>$1.6984^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Interval Vertex Deletion</strong></td>
<td>$8^k$</td>
<td>$1.8750^n$</td>
<td>$(2 - \varepsilon)^n$ for $\varepsilon &lt; 10^{-20}$ [4]</td>
</tr>
<tr>
<td><strong>Proper Interval Vertex Deletion</strong></td>
<td>$6^k$</td>
<td>$1.8334^n$</td>
<td>$(2 - \varepsilon)^n$ for $\varepsilon &lt; 10^{-20}$ [4]</td>
</tr>
<tr>
<td><strong>Block Graph Vertex Deletion</strong></td>
<td>$4^k$</td>
<td>$1.7500^n$</td>
<td>$(2 - \varepsilon)^n$ for $\varepsilon &lt; 10^{-20}$ [4]</td>
</tr>
<tr>
<td><strong>Cluster Vertex Deletion</strong></td>
<td>$1.9102^k$</td>
<td>$1.4765^n$</td>
<td>$1.6181^n$</td>
</tr>
<tr>
<td><strong>Thread Graph Vertex Deletion</strong></td>
<td>$8^k$</td>
<td>$1.8750^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Multicut on Trees</strong></td>
<td>$1.5538^k$</td>
<td>$1.3565^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>3-Hitting Set</strong></td>
<td>$2.0755^k$</td>
<td>$1.5182^n$</td>
<td>$1.6278^n$</td>
</tr>
<tr>
<td><strong>4-Hitting Set</strong></td>
<td>$3.0755^k$</td>
<td>$1.6750^n$</td>
<td>$1.8704^n$</td>
</tr>
<tr>
<td><strong>d-Hitting Set $(d \geq 3)$</strong></td>
<td>$(d - 0.9245)^k$</td>
<td>$(2 - \frac{1}{d-0.9245})^n$</td>
<td>$[17]$</td>
</tr>
<tr>
<td><strong>Min-Ones 3-SAT</strong></td>
<td>$2.562^k$</td>
<td>$1.6097^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Min-Ones d-SAT $(d \geq 4)$</strong></td>
<td>$d^k$</td>
<td>$(2 - \frac{1}{d})^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Weighted d-SAT $(d \geq 3)$</strong></td>
<td>$d^k$</td>
<td>$(2 - \frac{1}{d})^n$</td>
<td>NPR</td>
</tr>
<tr>
<td><strong>Weighted Feedback Vertex Set</strong></td>
<td>$3.6181^k$</td>
<td>$1.7237^n$</td>
<td>$1.8638^n$</td>
</tr>
<tr>
<td><strong>Weighted 3-Hitting Set</strong></td>
<td>$2.168^k$</td>
<td>$1.5388^n$</td>
<td>$1.6755^n$</td>
</tr>
<tr>
<td><strong>Weighted d-Hitting Set $(d \geq 4)$</strong></td>
<td>$(d - 0.832)^k$</td>
<td>$(2 - \frac{1}{d-0.832})^n$</td>
<td>$[17]$</td>
</tr>
</tbody>
</table>

What we did not talk about

- derandomization with so-called \textit{set-inclusion-families}.
- extension to permissive FPT algorithms.
- can be used for enumeration.