Parameterized Complexity of Superstring Problems

Ivan Bliznets Fedor V. Fomin Petr A. Golovach
Nikolay Karpov Alexander S. Kulikov Saket Saurabh

CPM 2015

Our problem

Shortest Superstring

Input:

Set of n strings $\mathcal{S} = \{s_1, \ldots, s_n\}$ over alphabet Σ and a non-negative integer ℓ.

Question:

Is there a string T of length at most ℓ containing all strings from $\mathcal{S}$ as substrings?

Bad News

Problem

- is NP-complete
- is inapproximable within factor $\frac{333}{332} - \epsilon$

Good News

Follows from MAX-ATSP

- Exact: 2^n
  - Dynamic programming, exp space
  - Inclusion-exclusion, poly space
  - $2^{n-\Omega(\sqrt{n/\log n})}$
- Approximation ratio: $2\frac{11}{30}$

Parameterized complexity

Fixed Parameter Tractable

An instance of parametrized problem is a pair $(\mathcal{I}, k)$ where $\mathcal{I}$ is an input and k is a parameter. It is said that a problem is fixed parameter tractable (FPT) if it can be solved in time $f(k) \cdot |\mathcal{I}|^{O(1)}$.

Parameterized complexity

Kernel

Kernelization is a polynomial time algorithm maps $(\mathcal{I}, k)$ to $(\mathcal{I}', k')$ such that:

- $(\mathcal{I}, k)$ and $(\mathcal{I}', k')$ is equivalent.
- The size of $\mathcal{I}'$ and k' is bounded by $\mathrm{f}(k)$

Problem admit polynomial *kernel* when f bounded by polynomial.

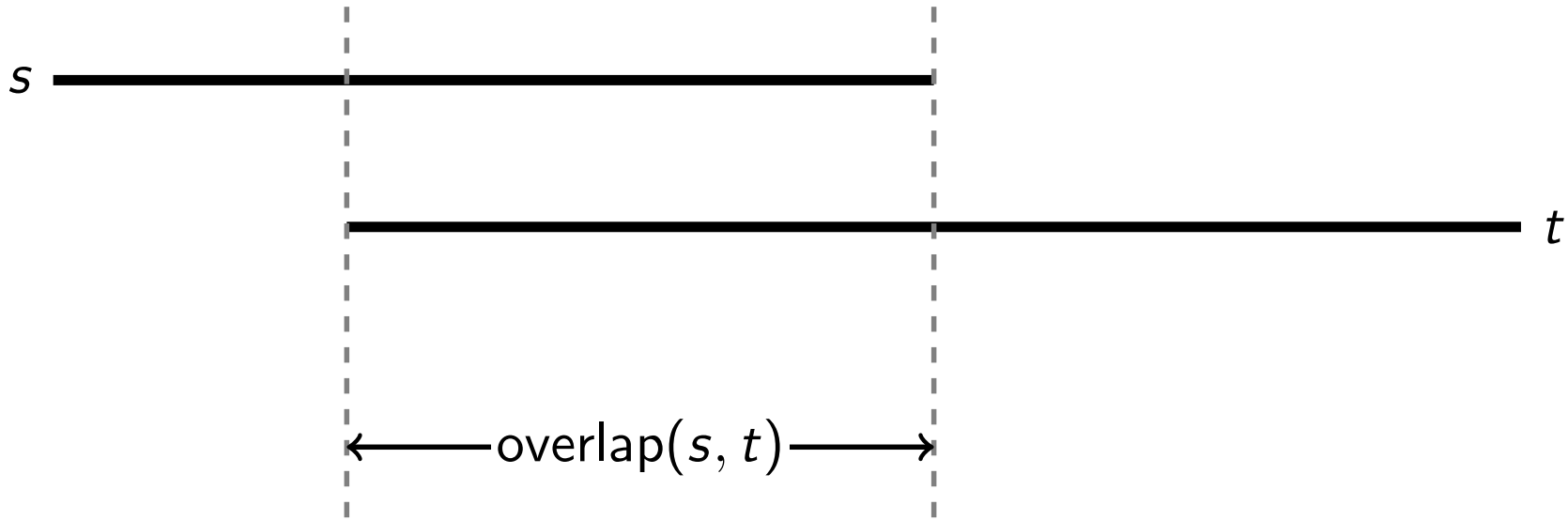
Our results

- FPT algorithms for variants of the problem.
- Kernelization results.

Kernelization

- Shortest Superstring admits a polynomial kernel being parameterized by "compression".

Overlap



$\mathrm{overlap}(\mathrm{ABC}, \mathrm{BCA}) = \mathrm{BC}$

$\mathrm{overlap}(\mathrm{BCA}, \mathrm{ABC}) = \mathrm{A}$

Compression

Compression is

- $\sum_{s \in \mathcal{I}} |s| - |t|$ where t is superstring.
- $\max_{\pi \in S_n} \sum_i |\mathrm{overlap}(s_{\pi_i}, s_{\pi_{i-1}})|$

Finding a shortest superstring is equivalent to finding an order of $s_1, \ldots, s_n$.

Kernelization

- Shortest Superstring admit kernel where parameter is $r = \sum_{s \in \mathcal{S}} |s| - \ell$ such that $|\mathcal{S}'| = \mathcal{O}(r^4)$.

Simple rules

- **Rule 1.** If x is a substring of y, then delete x and set $r = r - |x|$.
- **Rule 2.** If $\exists x$ such that $\forall y \in X \setminus \{x\}\ |\mathrm{overlap}(x, y)| = 0 \wedge |\mathrm{overlap}(y, x)| = 0$, then delete x and set $\ell = \ell - |x|$
- **Rule 3.** If there are distinct elements x and y of S such that $|\mathrm{overlap}(x, y)| \geq r$, then return a yes-answer.
- **Rule 4.** If there is $x \in S$ with $|x| > 2r$, then set $\ell = \ell - (|x| - 2r)$ and $x = \mathrm{prefix}_r(x)\mathrm{suffix}_r(x)$.

Rule of matching

Rule 5. Build auxillary graph,

- which vertices are strings, and
- edge connects string x with y iff $\mathrm{overlap}(x, y)$ or $\mathrm{overlap}(y, x)$ is not empty.

Find in this graph maximal matching M and if $|M| \geq r$ then return yes-answer.

Important strings

Rule 6. Construct a set of important strings which will be a part of the solution.

- Let X be the set of strings corresponding to endpoints of M.
- Let $Y = S \setminus X$. We take $R_{(i,j)}$ first $2|M|$ elements from Y sorted by the decrease $|\mathrm{overlap}(s_i, x)| + |\mathrm{overlap}(x, s_j)|$.
- We also take S_i first $2|M|$ elements from Y sorted by the decrease $|\mathrm{overlap}(s_i, x)|$.
- Take T_i first $2|M|$ elements from Y sorted by the decrease $|\mathrm{overlap}(x, s_i)|$.

Important strings

Set of strings

$$\mathcal{I} = X \cup \Big(\bigcup_{s_i \in X} \bigcup_{s_j \in X} R_{(i,j)} \Big) \cup \bigcup_{s_i \in X} S_i \cup \bigcup_{s_i \in X} T_i.$$

is *important*.

New equivalent instance is $(\mathcal{I}, \ell')$, and $\ell' = \ell - \sum_{x \in \mathcal{S} \setminus \mathcal{I}} |x|$

Size of set of important strings

$$\mathcal{I} = X \cup \Big(\bigcup_{s_i \in X} \bigcup_{s_j \in X} R_{(i,j)} \Big) \cup \bigcup_{s_i \in X} S_i \cup \bigcup_{s_i \in X} T_i.$$

$$|\mathcal{I}| \leq 2r + (2r)^2 \cdot 4r + 2r \cdot 4r + 2r \cdot 4r$$

Theorem

Shortest Superstring *is* NP*-complete for* $\ell = \sum_{x \in S} |x| - m - 1$ *even if restricted to the alphabet* $\Sigma = \{0, 1\}$ *where* m *is weight of maximal matching in auxillary graph.*

Auxillary graph

- vertices are strings
- weight of edge (u, v) is $\max(|\mathrm{overlap}(u, v)|, |\mathrm{overlap}(v, u)|)$

Further directions

- Improve algorithms.
- Upper bound for kernel.
- SUPERSEQUENCE

Thank you for your attention!