Tandem Duplication Parameterized by the Length Difference

Peter Damaschke

Chalmers, Göteborg, Sweden

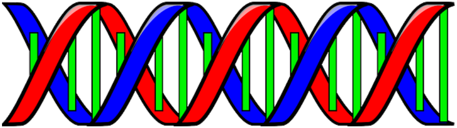
This is NOT Meant by Tandem Duplication

Tandem Duplication (TD) Problem

- **TD** turns ABC into ABBC (where A,B,C are substrings).
- **Contraction** of square BB turns ABBC into ABC.
- **Given:** strings S and T with $|S| < |T| = n$.
- **Problem**: turn S into T by a minimum number k of TDs.

- NP-complete, even with 5 distinct symbols (Cicalese, Pilati – IWOCA 2021).

Motivations



- Short TDs in DNA are indicators of certain genetic diseases.
- Recognize whether T could result from normal string S by TDs.
- Another variant of string editing.

Parameterization

- $O(n^{2k})$-time algorithm is obvious:
  Try to get S from T by all possible sequences of k contractions
  (Lafond, Zhu, Zou – SIAM J. Discr. Math. 2022).
- Idea for improved bounds:
  Many contractions yield the same string.
- Namely, many squares would overlap and imply periods.
  But in a run (periodic substring),
  it doesn't matter which square is contracted.
- Problem is in XP in parameter k.

STRINGINGINGOLOLOLOLOGY

Runs and Their Exponents

- **Run:** (sub)string with a period of at most half its length.
- **Exponent** of a run: length divided by shortest period ($R = P^e Q$).
- Hence, number of periods of a run is at most half exponent.
- Sum of exponents of all runs in a string is at most $4.1n$ (Crochemore, Kubica, Radoszewski, Rytter, Walen - J. Discr. Alg. 2012).
- This improves $O(n^{2k})$ to $O((2.05n)^k)$ time.
- A more elementary argument yields already $O((n \log n)^k)$ time.

Fixed-Parameter Tractability

- A problem with parameter k is in FPT
  if some algorithm can solve it in $O(f(k) \cdot p(n))$ time,
  where f is some computable function and p is some polynomial.
- Is the TD problem in FPT in parameter k? Open.
- We consider a weaker (but natural) parameter: $d = |T| - |S|$.
- Note that $k \leq d$.

TD Minimization is in FPT in the Length Difference

- Dynamic programming can take care of “windows” of length $O(d)$.
- In principle not too surprising. But details deserve some work.
- Counting arguments yield time bound $O(d^3(2d)^d n)$.

Kernelization

A **kernel** of a parameterized problem is, for every given instance:

- an equivalent instance of the problem
- which is computable in polynomial time
- and whose size is bounded by some function of the parameter only (not necessarily polynomial).

Existence of a kernel is equivalent to FPT.
Informally: preprocessing that cuts away the easy parts of a problem.

TD Problem has a Polynomial Kernel

Overall idea: The two strings consist of

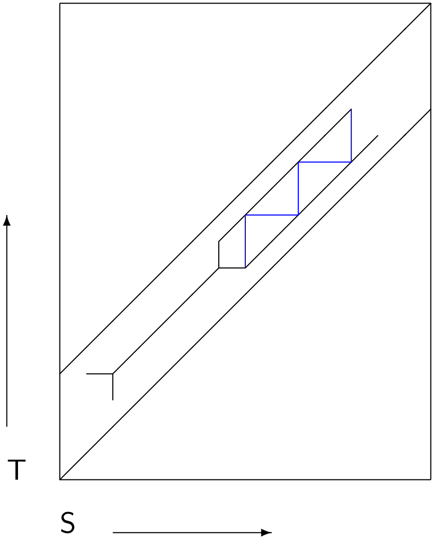
- nonperiodic substrings
  that must be aligned to each other uniquely,
- periodic substrings
  that can be shortened without changing the problem.

A geometric way to control these operations is to use
some "alignment graph" (similarly as in other string editing problems).

The Alignment Graph – in a Nutshell

- $|S| = m < n = |T|$, $d = n - m$.
- Vertices are, at most, the mn pairs of symbols in S and T.
- Every alignment induced by a sequence of TDs is a directed path (but not vice versa).
- Keep only vertices on such paths.
- Contained in some diagonal stripe of width d.
- Hence at most dn vertices remain.
- Long segements of left and right border are diagonal paths.
- They are either identical or represent strings with periods at most d.

Sketch



Kernelization Algorithm (on a high level)

- Construct the alignment graph.
- Identify the diagonal paths on its left and right border.
- Shorten the identical paths to single “fresh” symbols.
- Cut out periods.
- Can limit size to $O(d^3)$ and time to $O(dn)$.

Acknowledgments

- Former master's students: Belmin Dervisevic and Mateo Raspudic.
- Unknown person who made me aware of exponents.

Open Questions

- Better FPT time bound?
- FPT in stronger parameters?
- Smaller kernel?
- Kernelization without fresh symbols?