The computational landscape of permutation patterns

Marie-Louise Bruner

Joint work with Martin Lackner

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Permutation Patterns 2013, July 4th

Types of permutation patterns

- classical patterns
- vincular patterns
- bivincular patterns
- consecutive patterns
- mesh patterns
- boxed mesh patterns

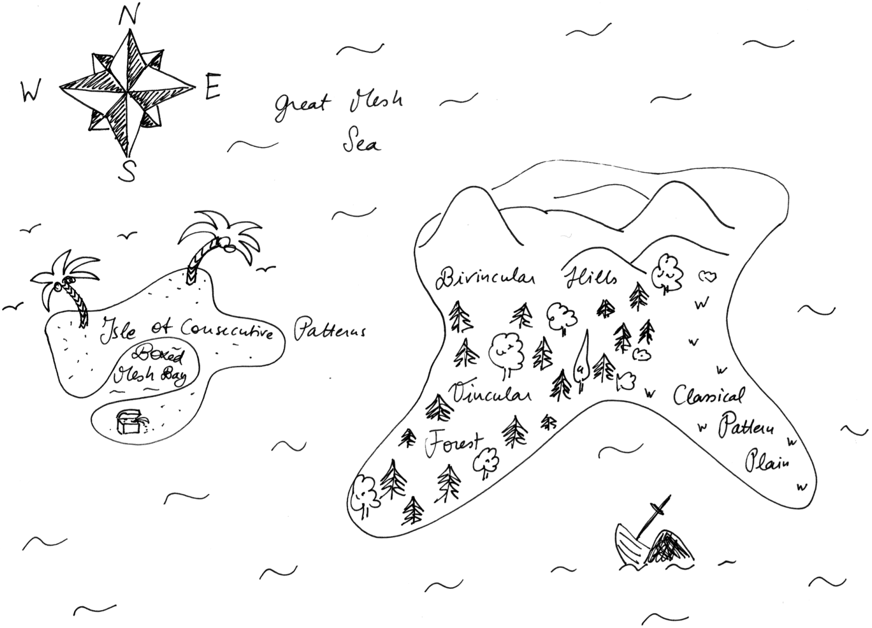
Permutation Pattern Matching Problems

Every type of permutation pattern naturally defines a corresponding computational problem. Let $\mathcal{C}$ denote any type of permutation pattern, i.e., let $\mathcal{C} \in \{$classical, vincular, bivincular, mesh, boxed mesh, consecutive$\}$.

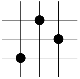
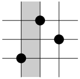
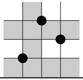
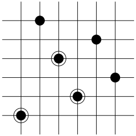
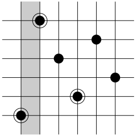
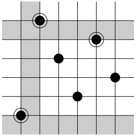
$\mathcal{C}$ PERMUTATION PATTERN MATCHING ($\mathcal{C}$ PPM)

Instance: A permutation T (the text) and a $\mathcal{C}$ pattern P

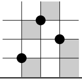
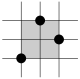
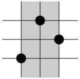
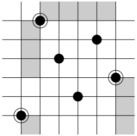
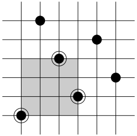
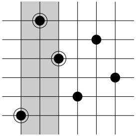
Question: Does the $\mathcal{C}$ pattern P occur in T?


N
W
E
S
Great Mesh Sea
Isle of Consecutive Patterns
~~Boxed~~ Mesh Bay
Bivincular Hills
Vincular Forest
Classical Pattern Plain

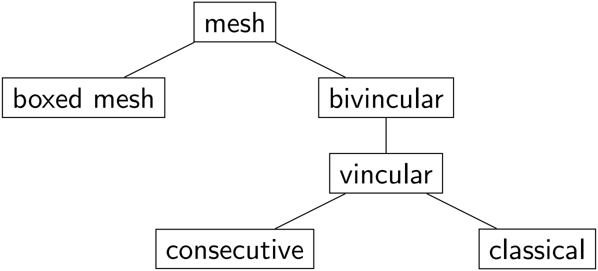
Definitions and Examples

	Classical	Vincular	Bivincular
Pattern	$P = 132 =$ 	$P = \underline{13}2 =$ 	$P = \begin{matrix} \lceil 1\,\overline{2\,3} \\ \underline{1\,3}\,2 \end{matrix} =$ 
Text			

Definitions and Examples 2

	Mesh	Boxed mesh	Consecutive
Pattern	$P = (\pi, R) =$ 	$P = \boxed{132} =$ 	$P = \underline{132} =$ 
Text			

Hierarchy of pattern types



Classical complexity analysis

Theorem [Bose, Buss, and Lubiw, 1993]

Classical permutation Pattern Matching is NP-complete.

Classical complexity analysis

Theorem [Bose, Buss, and Lubiw, 1993]

CLASSICAL PERMUTATION PATTERN MATCHING is NP-complete.

$\Longrightarrow$ The following problems are also NP-hard:

- VINCULAR PPM
- BIVINCULAR PPM
- MESH PPM

Classical complexity analysis

Theorem [Bose, Buss, and Lubiw, 1993]

Classical permutation Pattern Matching is NP-complete.

$\Longrightarrow$ The following problems are also NP-hard:

- Vincular PPM
- Bivincular PPM
- Mesh PPM

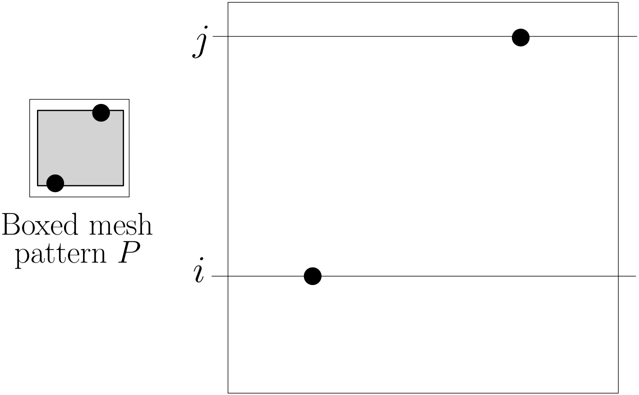
How about Boxed Mesh PPM and Consecutive PPM?

Boxed Mesh PPM is in P

It is easy to see that Boxed Mesh PPM can be solved in $\mathcal{O}(n^3)$-time.

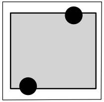
BOXED MESH PPM is in P

It is easy to see that BOXED MESH PPM can be solved in $\mathcal{O}(n^3)$-time.

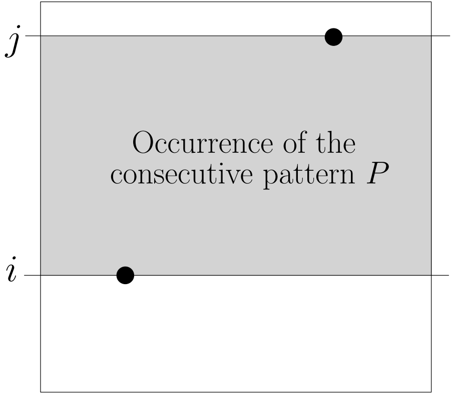


Boxed Mesh PPM is in P

It is easy to see that Boxed Mesh PPM can be solved in $\mathcal{O}(n^3)$-time.

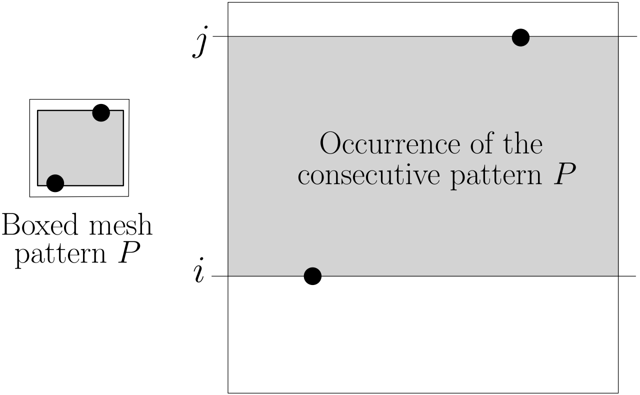


Boxed mesh pattern P



Boxed Mesh PPM is in P

It is easy to see that Boxed Mesh PPM can be solved in $\mathcal{O}(n^3)$-time.



There are $\mathcal{O}(n^2)$ pairs (i, j) that have to be checked.

Consecutive PPM is in P

Consecutive PPM can be solved in linear time in the length of the text:

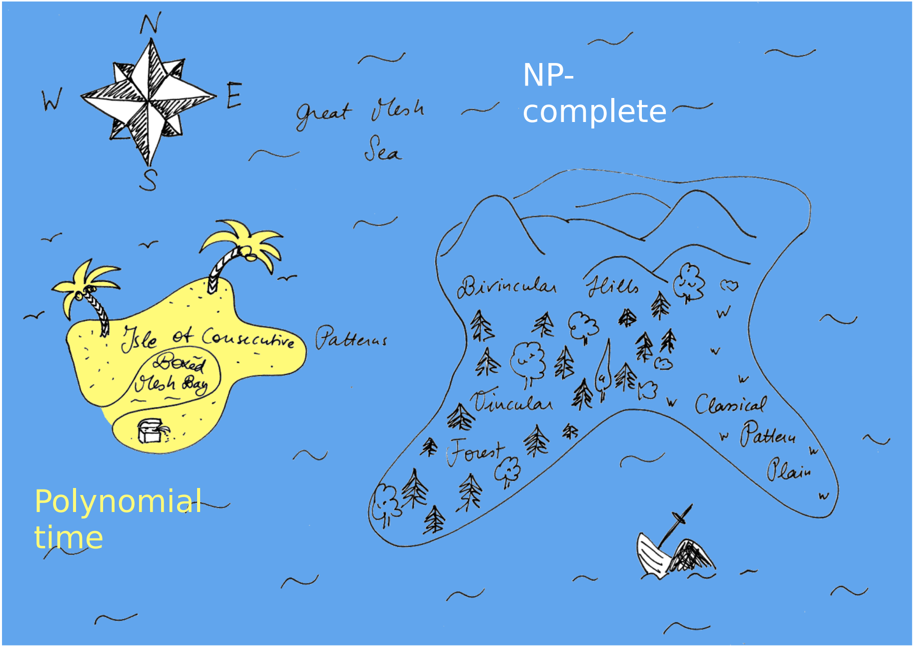
- M. Kubica, T. Kulczyski, J. Radoszewski, W. Rytter, and T. Wale, *A linear time algorithm for consecutive permutation pattern matching*, Information Processing Letters, 2013.
- D. Belazzougui, A. Pierrot, M. Raffinot and S. Vialette, *Single and multiple consecutive permutation motif search*, CoRR, 2013.

Consecutive PPM is in P

Consecutive PPM can be solved in linear time in the length of the text:

- M. Kubica, T. Kulczyski, J. Radoszewski, W. Rytter, and T. Wale, *A linear time algorithm for consecutive permutation pattern matching*, Information Processing Letters, 2013.
- D. Belazzougui, A. Pierrot, M. Raffinot and S. Vialette, *Single and multiple consecutive permutation motif search*, CoRR, 2013.

It is very easy to see that Consecutive PPM can be solved in $\mathcal{O}((n-k)\cdot k)$-time.


N
W
E
S
Great Mesh Sea
NP-complete
Birincular Hills
Vincular Forest
Classical Pattern Plain
Isle of Consecutive Patterns
Boxed Mesh Bay
Polynomial time

A parameterized point of view

Idea: Which *parameter* a PPM instance makes this problem computationally hard?

A parameterized point of view

Idea: Which *parameter* a PPM instance makes this problem computationally hard?

Parameterized Problem: $L \subseteq \Sigma^* \times \mathbb{N}$

A parameterized point of view

Idea: Which *parameter* a PPM instance makes this problem computationally hard?

Parameterized Problem: $L \subseteq \Sigma^* \times \mathbb{N}$

Fixed-parameter tractability

A parameterized problem L is *fixed-parameter tractable* if there is a computable function f and an integer c such that there is an algorithm solving L in time $\mathcal{O}(f(p) \cdot |I|^c)$.

Proving intractability

Definition

The class W[1] is defined as the class of all problems that are fpt-reducible to the following problem.

CLIQUE

Instance:	A graph $G = (V, E)$ and a positive integer k.
Parameter:	k
Question:	Is there a subset of vertices $S \subseteq V$ of size k such that S forms a clique, i.e., the induced subgraph $G[S]$ is complete?

Proving intractability

Definition

The class W[1] is defined as the class of all problems that are fpt-reducible to the following problem.

CLIQUE

Instance:	A graph $G = (V, E)$ and a positive integer k.
Parameter:	k
Question:	Is there a subset of vertices $S \subseteq V$ of size k such that S forms a clique, i.e., the induced subgraph $G[S]$ is complete?

If we can prove W[1]-hardness, this implies, under standard complexity theoretic assumptions, that no fpt-alogrithm exists.

Fpt-reductions

Definition

Let $L_1, L_2 \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. An *fpt-reduction* from L_1 to L_2 is a mapping $R : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ such that

- $(I, k) \in L_1$ iff $R(I, k) \in L_2$.
- R is computable by an fpt-algorithm.
- There is a computable function g such that for $R(I, k) = (I', k')$, $k' \leq g(k)$ holds.

Segregated PPM

Segregated Permutation Pattern Matching (SPPM)

Instance: A permutation T (the text) of length n, a permutation P (the pattern) of length $k \leq n$ and two positive integers $p \in [k]$, $t \in [n]$.

Parameter: k

Question: Is there a matching μ of P into T such that $\mu(i) \leq t$ iff $i \leq p$?

Segregated PPM

Segregated Permutation Pattern Matching (SPPM)

Instance: A permutation T (the text) of length n, a permutation P (the pattern) of length $k \leq n$ and two positive integers $p \in [k]$, $t \in [n]$.

Parameter: k

Question: Is there a matching μ of P into T such that $\mu(i) \leq t$ iff $i \leq p$?

Consider the pattern $P = 132$ and the text $T = 53142$. As shown by the matching $\mu(2) = 3$, $\mu(1) = 1$ and $\mu(3) = 4$, the instance $(P, T, 2, 3)$ is a yes-instance of the SPPM problem. However, $(P, T, 2, 4)$ is a NO-instance, since no matching of P into T can be found where $\mu(3) > 4$.

Segregated PPM is W[1]-hard.

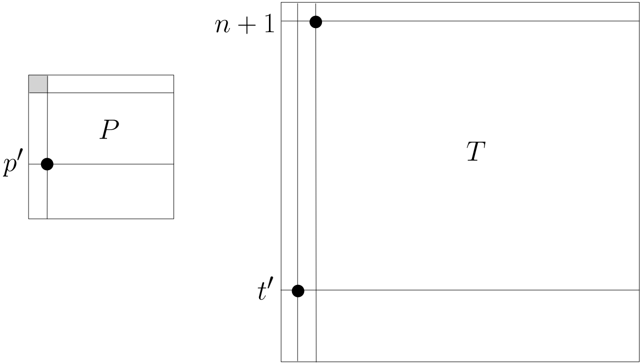
Proof idea: reduction from Clique.

MESH PPM is W[1]-hard.

Proof idea: Let (P, T, p, t) be a SEGREGATED PPM instance. We define $p' = p + 0.5$ and $t' = t + 0.5$.

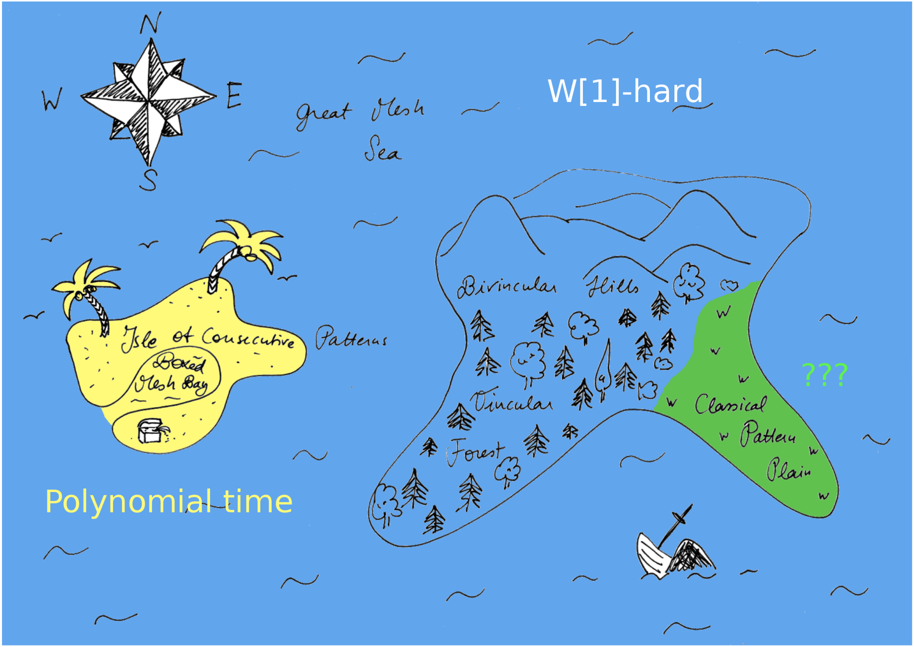
MESH PPM is W[1]-hard.

Proof idea: Let (P, T, p, t) be a SEGREGATED PPM instance. We define $p' = p + 0.5$ and $t' = t + 0.5$.
A MESH PPM instance (P', T') is constructed as follows:



BIVINCULAR AND VINCULAR PPM are also W[1]-hard.

Proof idea: reduction from SEGREGATED PPM.


N
W
E
S
Great Mesh Sea
W[1]-hard
Isle of Consecutive Patterns
Boxed Mesh Bay
Polynomial time
Bivincular Hills
Vincular Forest
Classical Pattern Plain
???

Further directions

- Is PPM fixed-parameter tractable with respect to the length of the pattern?
- Other parameters than the length of the pattern
- Find polynomial fragments
- Partially ordered patterns: a POPPM instance can be reduced to at most $k!$ many PPM instances
- Marked mesh and decorated patterns
- Patterns in words or in partitions