Finding small patterns in permutations in linear time*

Sylvain Guillemot † Dániel Marx ‡

Abstract

Given two permutations σ and π, the Permutation Pattern problem asks if σ is a subpattern of π. We show that the problem can be solved in time $2^{O(\ell^2 \log \ell)} \cdot n$, where $\ell = |\sigma|$ and $n = |\pi|$. In other words, the problem is fixed-parameter tractable parameterized by the size of the subpattern to be found.

We introduce a novel type of decompositions for permutations and a corresponding width measure. We present a linear-time algorithm that either finds σ as a subpattern of π, or finds a decomposition of π whose width is bounded by a function of $|\sigma|$. Then we show how to solve the Permutation Pattern problem in linear time if a bounded-width decomposition is given in the input.

1 Introduction

A permutation of length n is a bijective mapping $\pi : [n] \to [n]$; one way to represent it is as the sequence of numbers $\pi(1)\pi(2)\dots\pi(n)$. We say that a permutation π written in this notation *contains* permutation σ if π has a (not necessarily consecutive) subsequence where the relative ordering of the elements is the same as in σ. In this case, we say that σ is a *subpattern* of π; otherwise, π *avoids* σ. For example, 3215674 contains the pattern 132, since the subsequence 154 is ordered the same way as 132. On the other hand, the permutation avoids 4321: it does not contain a descending subsequence of 4 elements.

Counting the number of permutations avoiding a fixed pattern σ has been a very actively investigated topic of enumerative combinatorics. It was shown that for every length n, the number of permutations avoiding the pattern 123 and the number of permutations avoiding the pattern 231 are the same, namely the nth Catalan number [22, 23, 29, 9], which is asymptotically $4^{n+o(1)}$. Around 1990, Stanley and Wilf conjectured that for every fixed pattern σ, the number of permutations of length n avoiding σ can be bounded by c^n for some constant c depending on σ (whereas the total number of permutations is $n! = 2^{\Theta(n \log n)}$). This conjecture has been proved by Marcus and Tardos [24] in 2004.

The algorithmic study of permutations avoiding fixed patterns was motivated first by the observation that permutations sortable by stacks and deques can be characterized by certain forbidden patterns and such permutations can be recognized in linear time [22, 27, 28]. In the Permutation Pattern problem, two permutations σ and π are given and the task is to decide if π contains σ. In general, the problem is NP-hard [5]. There are known polynomial-time solvable special cases of the problem: for example, when σ is the identity permutation $12\cdots k$, then the problem is a special case of Longest Increasing Subsequence, whose polynomial-time solvability is a standard textbook exercise [10]. Other polynomial cases include the cases when σ and π are separable [5], or when both σ and π avoids 321 [20]. For more background, the reader is referred to the survey of Bruner and Lackner [7].

The Permutation Pattern problem can be solved by brute force in time $O(n^\ell)$, where $\ell = |\sigma|$ and $n = |\pi|$. This has been improved to $O(n^{0.47\ell+o(\ell)})$ by Ahal and Rabinovich [1]. These results imply that the problem is polynomial-time solvable for fixed pattern σ, but as the size of σ appears in the exponent of the running time, this fact is mostly of theoretical interest only. Our main result is an algorithm where the running time is linear for fixed σ and the size of σ appears only in the multiplicative factor of the running time.

Theorem 1.1. Permutation Pattern *can be solved in time $2^{O(\ell^2 \log \ell)} \cdot n$, where $\ell = |\sigma|$ is the length of the pattern and $n = |\pi|$.*

In other words, Permutation Pattern is fixed-parameter tractable parameterized by the size of the pattern: recall that a problem is fixed-parameter tractable with a parameter ℓ if it can be solved in time $f(\ell) \cdot n^{O(1)}$, where f is an arbitrary computable function depending only on ℓ; see [12, 15]. The fixed-parameter tractability of Permutation Pattern has been an

*Research supported by the European Research Council (ERC) grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results," reference 280152 and OTKA grant NK105645.

†Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) sguillem@sztaki.hu.

‡Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) dmarx@cs.bme.hu.

open question implicit in previous work.

The main technical concept in the proof of Theorem 1.1 is a novel form of decomposition for permutations. The decomposition can be explained most intuitively using a geometric language. Given a permutation π of length n, one can represent it as the set of points $(1, \pi(1))$, $(2, \pi(2))$, $\ldots$, $(n, \pi(n))$ in the 2-dimensional plane. We can view these points as a family of degenerate rectangles, each having width and height 0. Starting with this family of n degenerate rectangles, our decomposition consists of a sequence of families of rectangles, where the next family is created from the previous one by a *merge* operation. The merge operation removes two rectangles R_1, R_2 from the family and replaces them with their bounding box, that is, the smallest rectangle containing both (see Figure 1). The decomposition is a sequence of $n-1$ merges that eventually replaces the whole family with a single rectangle. Note that the rectangles created by the merges are not necessarily disjoint. We define a notion of *width* for a family of rectangles, which roughly corresponds to the maximum number of other rectangles a rectangle can "see" either horizontally or vertically. The decomposition has width at most d if the rectangle family has width at most d at every step of the decomposition. Let us observe that the merge operation can increase the width (by creating a large rectangle that sees many other rectangles) or it can decrease width (since if a rectangle sees both of the merged rectangles, then it sees one less rectangle after the merge operation). Therefore, whether it is possible to maintain bounded width during a sequence of merge operations is a very subtle and highly nontrivial question.

The proof of Theorem 1.1 follows from the following two results on bounded-width decompositions:

(1) For every fixed pattern σ, there is a linear-time algorithm that, given a permutation π, either shows that σ appears in π or outputs a bounded-width decomposition of π (Theorem 4.1).

(2) Permutation Pattern can be solved in linear time if a bounded-width decomposition for π is given in the input (Theorem 5.1).

The proof of (1) needs to show how to find the next mergeable pair in the decomposition. We argue that there have to be two rectangles that are "close" in a certain sense (ensuring that the width is still bounded after the merge), otherwise the rectangles are so much spread out that a result of Marcus and Tardos [24] guarantees that every permutation of length ℓ (in particular, σ) appears in π. The implementation of this idea needs careful control of the global structure of the rectangles. Therefore, the algorithm is not based on simply merging pairs in a greedy way: instead of showing that a mergeable pair always exist, what we show is that if the global property holds, then it is possible to merge two rectangles such that the global property still holds after the merge.

The algorithm in (2) uses dynamic programming the following way. Given a family of rectangles, we can define a *visibility graph* where the vertex set is the set of rectangles and two rectangles are adjacent if their horizontal or vertical projections intersect. For our purposes, the meaning of this graph is that if two rectangles R_1 and R_2 are nonadjacent, then the relative position of $x \in R_1$ and $y \in R_2$ follows from the relative position of R_1 and R_2; on the other hand, if R_1 and R_2 are adjacent, then the relative position of x and y can depend on exactly where they appear in R_1 and R_2. The fact that the decomposition has bounded width implies that the visibility graph of the rectangle family at every step has bounded degree. We enumerate every set K of at most ℓ rectangles that induces a connected graph in the visibility graph; as the visibility graph has bounded degree, the number of such sets is linear in n. The subproblems of the dynamic programming are as follows: at each step of the decomposition, for each set K of size at most ℓ that is connected in the visibility graph, we have to enumerate every pattern that appears in the points contained in the rectangles (and the possible distribution of the elements of the pattern among the rectangles). In each step of the decomposition, only those subproblems have to be updated that involve the merged rectangles, and this can be done efficiently using the information at hand. At the last step, there is only a single rectangle containing every element of π; thus the subproblems for this single rectangle tell us if σ is contained in π.

The fixed-parameter tractability of the Permutation Pattern problem seems to be very fragile: every reasonable extension or generalization of the problem (e.g., introducing colors or introducing additional constraints such as certain elements of the pattern having to appear consecutively) turned out to be W[1]-hard [7, 20, 18]. In Section 6, we prove the W[1]-hardness of another colored variant of the problem and then infer that the natural 3-dimensional generalization of Permutation Pattern is also W[1]-hard.

The reason for the $2^{O(\ell^2 \log \ell)}$ dependence on ℓ in Theorem 1.1 is the following. Recall that Marcus and Tardos [24] proved that for every fixed permutation σ, there is a constant c such that the number of permutations of length n that avoid σ can be bounded by c^n. In their proof, the constant c is exponential in the length of σ, but it might be true that the result holds with polynomially bounded c. Our algorithm for finding a decomposition relies on the proof of Marcus

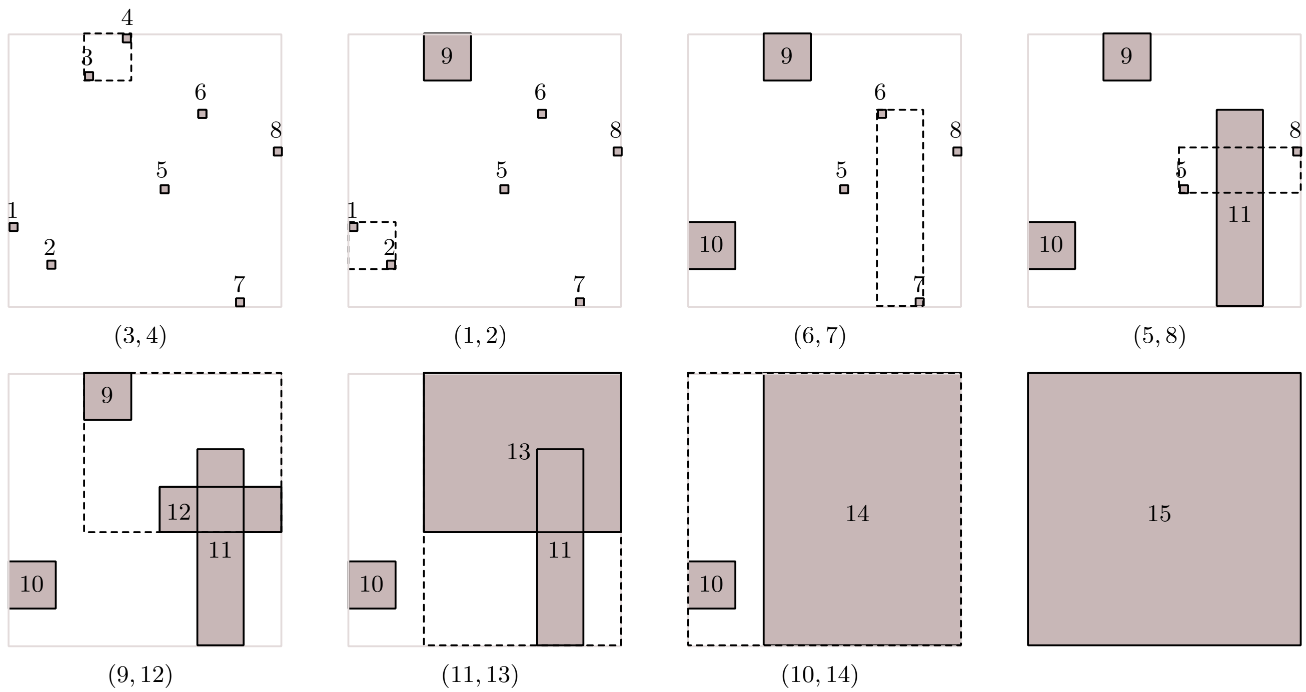


Figure 1: A possible decomposition of the permutation 32784615. The dashed rectangles and the pairs of numbers below the figures show the next two rectangles to be merged. We follow the convention that the rectangle created in the i-th merge is labeled $n + i$.

and Tardos, and the bound we get on the width is exponential in the length ℓ of σ, which implies that the algorithm using this decomposition has running time $2^{O(\ell^2 \log \ell)} \cdot n$. Improving the exponential bound in the result of Marcus and Tardos to a polynomial would immediately imply that we can find a decomposition of polynomially bounded width, and then the running time would be $2^{O(\ell \log \ell)} \cdot n$.

We investigate a specific class of permutations for which we can give improved bounds. A sequence is *monotone* if it is either increasing or decreasing; we say that a permutation is *t-monotone* if it can be partitioned into t monotone (not necessarily consecutive) subsequences. For the special case when π is t-monotone we show that it is possible to find a decomposition of width polynomially bounded by t. Moreover, we show that there is a very simple way of solving PERMUTATION PATTERN when π is t-monotone. The crucial observation is that if, given t monotone sequences, the task is to select some elements from each sequence such that they form a specific pattern, then this can be encoded as a constraint satisfaction problem (CSP) having a majority polymorphism. It is well-known that CSPs with a majority polymorphism is polynomial-time solvable, which gives us a polynomial-time solution for PERMUTATION PATTERN for a specific distribution of the elements of σ among the t monotone subsequences of π. Finally, we can try all possible way of distributing the elements of the pattern among the monotone sequences, yielding a very compact proof for the fixed-parameter tractability of PERMUTATION PATTERN on t-monotone permutations.

On a high level, our algorithm can be described by the following scheme: either the permutation π has no bounded-width decomposition, in which case we can answer the problem immediately, or we can find a bounded-width decomposition, in which case we can use an algorithm working on the decomposition. This win/win scenario is very similar to how the notion of treewidth is used for many graph problems: high treewidth implies an immediate answer and bounded-treewidth graphs can be handled using standard techniques. This idea was used, for example, in the classical work of Bodlaender [4, 3] and more recently in the framework of bidimensionality for planar graphs [11]. However, it has to be pointed out that our notion of decomposition is very different from tree decompositions. The main property of tree decompositions is that the graph is broken down into parts that interact with each other only via a small boundary. Nothing similar happens in our decomposition: when we merge two rectangles, then the points appearing in the two rectangles can have very complicated relations. Perhaps it is even misleading to call our notion a "decomposition": it would be more properly described as a construction scheme that maintains a notion of bounded-degreeness throughout the process. It would be interesting to see if there is a corresponding graph-theoretic analog for this scheme, which might be useful for solving some graph-theoretical problem.

The paper is organized as follows. Section 2 introduces notation, including a somewhat nonstandard way of looking at permutations as a labeled point set. Section 3 defines our notion of decomposition and width measure, and observes some properties. Section 4 presents our algorithm for finding a decomposition. Section 5 shows how to solve the Permutation Pattern problem given a decomposition. Section 6 proves hardness results for some natural generalizations of the problem. Section 7 investigates the special case when π is a t-monotone permutation.

2 Definitions

A *permutation* of length n is a bijection $\pi : [n] \to [n]$. It will be convenient for us to look at permutations from a more geometric viewpoint by considering them as point sets, as our decomposition can be explained conveniently in terms of families of points and rectangles. A *point* is an element $p = (x, y) \in \mathbb{N}^2$; we denote $\mathrm{pr}_1(p) = x$ and $\mathrm{pr}_2(p) = y$ (these are called the x-coordinate and y-coordinate of p). A *point set* is a finite set of points; it is in *general position* if no two points have the same x-coordinate or the same y-coordinate. We define a *permutation* as a pair $\pi = (S, P)$, where S is a subset of positive integers and $P : S \to \mathbb{N}^2$ is an injection such that $P(S)$ is a point set in general position. For a permutation $\pi = (S, P)$, we use $S(\pi)$ to refer to the set S, and define the *length* of π as $|\pi| = |S(\pi)|$. Given $S' \subseteq S$, we define the permutation $\pi|S' = (S', P|S')$.

Let us discuss how permutations are represented in algorithms. We say that a permutation $\pi = (S, P)$ of length n is *reduced* if $S = [n]$ and $P(S) \subseteq [n] \times [n]$. A reduced permutation can be represented naturally as an array of n points in $[n] \times [n]$. We require that the permutation given as an input of an algorithm is reduced and has this representation; we mainly use this assumption to ensure that we can sort the points by x- or y-coordinate in linear time. Note that if we consider a permutation to be a bijection $\pi : [n] \to [n]$, then it is straightforward to obtain such a representation.

Given $p, p' \in S$ and $\alpha \in \{1, 2\}$, we denote $p <^{\pi}_{\alpha} p'$ iff $\mathrm{pr}_\alpha(P(p)) < \mathrm{pr}_\alpha(P(p'))$. Given two permutations σ and π, a mapping $\phi : S(\sigma) \to S(\pi)$ is an *embedding of σ into π* iff for every $p, p' \in S(\sigma)$, for each $\alpha \in \{1, 2\}$, $p <^{\sigma}_{\alpha} p'$ iff $\phi(p) <^{\pi}_{\alpha} \phi(p')$. We say that σ is a *subpattern* of π, or that π *contains* σ, if there is an embedding of σ into π. (Intuitively, we can represent a permutation as a 0-1 matrix where every column and every row contains at most one cell with 1 in it; then σ is a subpattern of π if it corresponds to a submatrix of the matrix representation of π). We define the following decision problem:

Permutation Pattern
Input: Two reduced permutations σ and π.
Question: Is σ a subpattern of π?

For a given instance (σ, π) of the problem, we will denote $\ell = |\sigma|$ and $n = |\pi|$. Besides points and sets of points, we will be dealing with rectangles and sets of rectangles as well. Given two positive integers p, q with $p \le q$, we define the *interval* $[p, q] = \{p, p+1, \ldots, q\}$; note that we only consider discrete intervals. Given two intervals $I = [p, q]$ and $I' = [p', q']$, we denote $I < I'$ iff $q < p'$. A (axis-parallel) *rectangle* is a set $R = I \times J$ where I, J are two intervals; we denote $I_1(R) = I$ and $I_2(R) = J$.

3 Decompositions

The purpose of this section is to introduce the decomposition used by the main algorithm and observe some of its properties. A *rectangle family* is a set of rectangles indexed by a subset of natural numbers; formally, a rectangle family is a pair $\mathcal{R} = (S, R)$, where $S \subseteq \mathbb{N}$ is a set and R maps each element $i \in S$ to a rectangle $R(i)$. For a rectangle family $\mathcal{R} = (S, R)$, we use $S(\mathcal{R})$ to refer to the set S and we define the size of $\mathcal{R}$ as $|\mathcal{R}| = |S(\mathcal{R})|$. Note that a point is a degenerate rectangle, and thus a permutation can also be viewed as a rectangle family. We define the operation of *merging* two rectangles in a family as follows. Given two elements $i, j \in S$ and $k \notin S$, we denote by $\mathcal{R}[i, j \to k]$ the rectangle family $\mathcal{R}' = (S', R')$ where $S' = S - \{i, j\} + \{k\}$, $R'(p) = R(p)$ for every $p \in S - \{i, j\}$, and $R'(k)$ is the smallest rectangle enclosing $R(i) \cup R(j)$. That is, we replace rectangles $R(i)$ and $R(j)$ by their bounding box, and assign the index k to the new rectangle.

Our notion of decomposition is defined as follows.

Definition 3.1. *Let $\pi = (S, P)$ be a permutation of length n. A* decomposition *of π is a sequence $\mathcal{D} = (\mathcal{R}_0, \ldots, \mathcal{R}_s)$ of rectangle families such that:*

(i) $\mathcal{R}_0 = \pi$;
(ii) there exists a sequence of integers $k_1 < k_2 < \ldots < k_s$ such that $\max S < k_1$ and for every

$1 \le p \le s$, there exist $i, j \in S(\mathcal{R}_{p-1})$ such that $\mathcal{R}_p = \mathcal{R}_{p-1}[i, j \to k_p]$;

(iii) $|\mathcal{R}_s| = 1$.

That is, in each step we are merging two rectangles to create a new rectangle. Observe that by Point (iii) we have $s = n-1$, i.e. the decomposition contains n rectangle families. This means that the obvious representation of the decomposition can have size $\Omega(n^2)$. However, let us observe that it is sufficient to list the pairs of rectangles that are merged in each step. Therefore, we can compactly represent the decomposition in space $O(n)$ by the *merge sequence* $\Sigma = \sigma_1 \dots \sigma_s$, where for each $1 \le p \le s$ we have $\sigma_p = (i, j, k_p)$ if $\mathcal{R}_p = \mathcal{R}_{p-1}[i, j \to k_p]$.

Next we define a notion of width for permutations. For $\alpha \in \{1, 2\}$, we say that two rectangles R, R' *α-view* each other if $I_\alpha(R)$ intersects $I_\alpha(R')$. Let $\mathcal{R} = (S, R)$ be a rectangle family. Given $i \in S$ and $\alpha \in \{1, 2\}$, we define $\mathrm{view}_\alpha(\mathcal{R}, i)$ as the set of elements $j \in S - \{i\}$ such that $R(i)$ and $R(j)$ α-view each other. Given $i \in S$, we define $\mathrm{view}(\mathcal{R}, i) = \max_{\alpha \in \{1,2\}} |\mathrm{view}_\alpha(\mathcal{R}, i)|$. Note that we define the number $\mathrm{view}(\mathcal{R}, i)$ as the maximum of the two cardinalities rather as the cardinality of the union, for reasons that will become clear later. Let d be an integer. We say that a rectangle family $\mathcal{R}$ is *d-wide* if $\mathrm{view}(\mathcal{R}, i) < d$ holds for every $i \in S(\mathcal{R})$. We say that a decomposition $\mathcal{D} = (\mathcal{R}_0, \dots, \mathcal{R}_s)$ of π is d-wide if each rectangle family $\mathcal{R}_p$ is d-wide. Observe that it is enough to ask whenever $\mathcal{R}_p$ merges rectangles i and j to produce rectangle k, then $\mathrm{view}(\mathcal{R}_p, k) < d$: indeed, the view number of a rectangle can increase only if it views k but not i, j, in which case it is upper bounded by the view number of k. We define the *width* of a permutation π, denoted by $w(\pi)$, as the minimum d such that π has a d-wide decomposition.

3.1 Basic properties We observe that width is monotone for subpatterns:

Lemma 3.2. *If σ is a subpattern of π, then $w(\sigma) \le w(\pi)$.*

Proof. It is sufficient to show that if S' is a subset of $S(\pi)$, then $w(\pi|S') \le w(\pi)$; in fact, by induction, it is sufficient to show this for the case when $S' = S - \{j_1\}$ for some $j_1 \in S(\pi)$. Consider a d-wide decomposition $\mathcal{D} = (\mathcal{R}_0, \dots, \mathcal{R}_{n-1})$ of π. We modify the decomposition as follows. There is a unique step i_1 in $\mathcal{D}$ when j_1 is merged with some rectangle j_2 and they are replaced by the bounding box j_3. If this is the last step of the decomposition, then it is clear that removing this step and removing j_1 from each of $\mathcal{R}_0, \dots, \mathcal{R}_{n-2}$ results in a d-wide decomposition of $\pi|S'$.

Otherwise, suppose that this is not the last step. Then there is a unique step $i_2 > i_1$ when j_3 is merged with some rectangle j_4. We remove step i_1 and modify step i_2 such that j_4 is merged with j_2 instead of j_3. Therefore, we obtain a decomposition $\mathcal{D}' = (\mathcal{R}'_0, \dots, \mathcal{R}'_{n-2})$, where

- for $0 \le i < i_1$, rectangle family $\mathcal{R}'_i$ is obtained from $\mathcal{R}_i$ by removing element j_1.
- for $i_1 \le i \le i_2 - 2$, rectangle family $\mathcal{R}'_i$ is obtained from $\mathcal{R}_{i+1}$ by replacing j_3 with j_2,
- for $i_2 - 1 \le i \le n-2$, rectangle family $\mathcal{R}'_i$ is obtained from $\mathcal{R}_{i+1}$ by modifying only a single rectangle, namely the one whose construction involved j_1.

In the last case, the single modified rectangle cannot become larger: it is constructed as the merge of one fewer points than in $\mathcal{D}$. Therefore, in all cases, the fact that every $\mathcal{R}_i$ is d-wide implies that every $\mathcal{R}'_i$ is d-wide. Thus $\mathcal{D}'$ is a d-wide decomposition of $\pi|S'$ and $w(\pi|S') \le w(\pi)$ follows. $\square$

Next, we observe a relation between the width and the existence of close pairs of points. Let π be a permutation. Given $p, p' \in S(\pi)$ and $\alpha \in \{1, 2\}$, if $p <^\pi_\alpha p'$ then we denote $\mathrm{Int}_\alpha(\pi, p, p') = \{p'' \in S(\pi) \mid p <^\pi_\alpha p'' <^\pi_\alpha p'\}$; if $p' <^\pi_\alpha p$ then we let $\mathrm{Int}_\alpha(\pi, p, p') := \mathrm{Int}_\alpha(\pi, p', p)$. For an integer d, we say that $\{p, p'\} \subseteq S(\pi)$ is a *d-close pair* of π if for each $\alpha \in \{1, 2\}$ it holds that $|\mathrm{Int}_\alpha(\pi, p, p')| < d$. Let us observe that the existence of a d-close pair is a necessary condition for having a d-wide decomposition: the first pair $\{j_1, j_2\}$ of points merged in the decomposition should be d-close: otherwise the rectangle family obtained by replacing j_1 and j_2 with their bounding box would have a view number greater than d.

Proposition 3.1. *If $w(\pi) \le d$, then π has a d-close pair.*

Note that by Lemma 3.2, in fact every subpermutation of π has a d-close pair. As we shall see in Section 4, the existence of d-close pairs in the subpermutations approximately characterizes the width of the permutation.

3.2 Separable permutations In this section, we relate our width measure to the well-known notion of separable permutations (note that this connection is not needed for the main algorithmic results of the paper). The separable permutations are the permutations that are totally decomposable under the *substitution decomposition* [25, 2], and we show in Proposition 3.3 below that they correspond to permutations of width at most 1.

We first define the operation of *substitution* for permutations. Let $\pi = (S, P)$ and $\pi' = (S', P')$ be two permutations with $S \cap S' = \emptyset$. Given $x \in S$, we define the permutation $\pi[x \leftarrow \pi']$ as follows. This is a permutation $\pi'' = (S'', P'')$, where $S'' = S - \{x\} + S'$, and such that two elements $p, p' \in S''$ have the following relations: (i) if $p, p' \in S$ then $p <_\alpha^{\pi''} p'$ iff $p <_\alpha^{\pi} p'$; (ii) if $p, p' \in S'$, then $p <_\alpha^{\pi''} p'$ iff $p <_\alpha^{\pi'} p'$; (iii) if $p \in S, p' \in S'$, then $p <_\alpha^{\pi''} p'$ iff $p <_\alpha^{\pi} x$.

PROPOSITION 3.2. *Given two permutations π and π', and given $x \in S(\pi)$, it holds that $w(\pi[x \leftarrow \pi']) = \max(w(\pi), w(\pi'))$.*

Proof. Let $d = w(\pi)$ and $d' = w(\pi')$, and let $\pi'' = \pi[x \leftarrow \pi']$. As π and π' are subpatterns of π'', it follows that $w(\pi'') \geq \max(d, d')$ by Lemma 3.2. Let us show that $w(\pi'') \leq \max(d, d')$. Let $\mathcal{D} = (\mathcal{R}_0, \ldots, \mathcal{R}_r)$ be a d-wide decomposition of π and let $\mathcal{D}' = (\mathcal{R}'_0, \ldots, \mathcal{R}'_s)$ be a d'-wide decomposition of π'. We assume w.l.o.g. that $\mathcal{D}'$ produces a sequence of indices $k'_1 < \ldots < k'_s$ and $\mathcal{D}$ produces a sequence of indices $k_1 < \ldots < k_r$ such that $\max S(\pi'') < k'_1$ and $k'_s < k_1$. We construct a decomposition $\mathcal{D}'' = (\mathcal{R}''_0, \ldots, \mathcal{R}''_t)$ of π'' as follows. We first simulate the merges of $\mathcal{D}'$, then once the points of π' have been merged into a single rectangle we simulate the merges of $\mathcal{D}$. More precisely, we start with $\mathcal{R}''_0 = \pi''$, and:

- for $1 \leq p \leq s$, if $\mathcal{R}'_p = \mathcal{R}'_{p-1}[i, j \to k]$ then $\mathcal{R}''_p = \mathcal{R}''_{p-1}[i, j \to k]$;
- for $1 \leq p \leq r$, if $\mathcal{R}_p = \mathcal{R}_{p-1}[i, j \to k]$ then $\mathcal{R}''_{s+p} = \mathcal{R}''_{s+p-1}[i', j' \to k]$, where i', j' are obtained from i, j by replacing x with k'_s.

Observe that a rectangle created in the first step views the same rectangles as in $\mathcal{D}'$, while a rectangle created in the second step views the same rectangles as in $\mathcal{D}$. We conclude that $\mathcal{D}''$ is a d''-wide decomposition of π'' with $d' = \max(d, d')$. ☐

We recall that the separable permutations can be defined as follows [5]. A permutation π is *increasing* (resp. *decreasing*) if for each $p, p' \in S(\pi)$ it holds that $p <_1^\pi p'$ iff $p <_2^\pi p'$ (resp. $p' <_2^\pi p$). A permutation π is *monotone* if it is increasing or decreasing. The *separable* permutations is the smallest class of permutations that contains the monotone permutations and is closed under substitution; alternatively, they are the permutations that do not contain 2 4 1 3 or 3 1 4 2.

PROPOSITION 3.3. *A permutation π is separable iff $w(\pi) \leq 1$.*

Proof. As the monotone permutations have width at most 1, it follows from Proposition 3.2 that the separable permutations have width at most 1. Conversely, if a permutation π is not separable, then π contains 2 4 1 3 or 3 1 4 2; as these two permutations have no 1-close pair, they have width at least 2 by Proposition 3.1, which implies that $w(\pi) \geq 2$ by Lemma 3.2. ☐

3.3 Grids In this section, we define certain permutations with a grid-like structure, and we characterize their widths. The main interest of these permutations is that they serve as obstruction patterns to small width; moreover, we will see in Section 4 that they are the only obstructions in an approximate sense.

Given an interval I, we say that a sequence $P = (I_1, \ldots, I_s)$ of intervals is a *partition* of I if (i) the I_j's are disjoint and their union is I, and (ii) $I_1 < I_2 < \ldots < I_s$. Consider the rectangle $R = I \times J$, and fix two integers r, s. An *$r \times s$-gridding* of R is a pair $G = (P_1, P_2)$, where $P_1 = (I_1, \ldots, I_r)$ is a partition of I, and $P_2 = (J_1, \ldots, J_s)$ is a partition of J. Fix $x \in [r], y \in [s]$. We call I_x the *xth column* of G, and J_y the *yth row* of G; the rectangle $G(x, y) := I_x \times J_y$ is called the *(x, y)th-cell* of G. If M is a point set, we say that M *contains an $r \times s$-grid* if there exists an $r \times s$-gridding G such that for every $x \in [r], y \in [s]$, $G(x, y)$ intersects M. By extension, if $\pi = (S, P)$ is a permutation, we say that π *contains an $r \times s$-grid* if $P(S)$ does.

An *$r \times s$-grid permutation* is a permutation of length rs that contains an $r \times s$-grid. Observe that a permutation contains an $r \times s$-grid if and only if it contains an $r \times s$-grid permutation. Furthermore, observe that if a permutation contains an $r \times r$-grid, then it contains every permutation of length r; this fact will be crucial for our algorithm. The *canonical $r \times s$-grid permutation* is the permutation π corresponding to the point set $\{((j-1)s + (s-i+1), (i-1)r + j) | 1 \leq i \leq s, 1 \leq j \leq r\}$; let us denote by $p_{i,j}$ the element of $S(\pi)$ corresponding to point $((j-1)s + (s-i+1), (i-1)r + j)$. Intuitively, $p_{i,j}$ is the point in row i and column j, where rows are numbered from bottom to top and columns are numbered from left to right (see Figure 2(a)). Note that the indexing of points $p_{i,j}$ departs from the convention used for points in cartesian coordinates, i.e. point $p_{i,j}$ is inside the (j, i)th cell of the gridding of π.

The following result shows that $r \times r$-grid permutations have width $\Omega(r)$.

PROPOSITION 3.4. *If π is a $(2r + 4) \times (2r + 4)$-grid permutation, then $w(\pi) \geq r$.*

Proof. Consider a decomposition $(\mathcal{R}_0, \ldots, \mathcal{R}_s)$ of π. Let $\mathcal{R}_t$ be the first family in this sequence that includes a rectangle R containing points from two nonadjacent rows or from two nonadjacent columns. Suppose without loss of generality that R contains points from rows

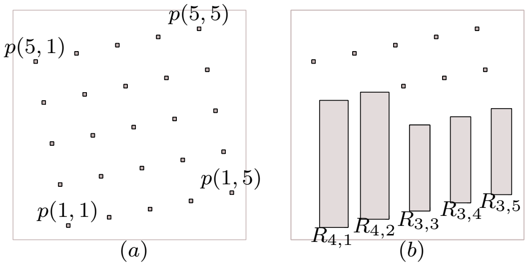


Figure 2: (a) A 5×5 canonical grid. (b) A step of the decomposition in the proof of Proposition 3.5.

y_1 and y_2 with $y_2 - y_1 > 1$. Consider the set X of $2r+4$ points of π in row y_1+1. In family $\mathcal{R}_{t-1}$, no rectangle contains points from two nonadjacent columns, thus at most two points of X can be contained in each rectangle of $\mathcal{R}_{t-1}$, i.e., points of X are contained in at least $r+2$ rectangles. At most two of these rectangles can participate in the merge that created rectangle R in $\mathcal{R}_t$. Therefore, at least r of these rectangles survive in $\mathcal{R}_t$ and are distinct from R. All of these rectangles 2-view R, hence $\mathcal{R}_t$ (and therefore the decomposition) cannot be r-wide. □

Proposition 3.5. *If π is the canonical $r \times r$-grid permutation, then $w(\pi) = r$.*

Proof. We first show that $w(\pi) \geq r$. Consider two distinct elements $p = p_{i,j}$ and $p' = p_{i',j'}$ in $S(\pi)$. We have $|\mathrm{Int}_1(\pi,p,p')| = |(j'-j)r + (i-i')| - 1$ and $|\mathrm{Int}_2(\pi,p,p')| = |(i'-i)r + (j'-j)| - 1$. Observe that if $|j'-j| \geq 2$ then $|\mathrm{Int}_1(\pi,p,p')| \geq r-1$, and likewise if $|i'-i| \geq 2$ then $|\mathrm{Int}_2(\pi,p,p')| \geq r-1$. Suppose that $\{p,p'\}$ is a $(r-1)$-close pair of π. We then have $0 \leq |i'-i| \leq 1$ and $0 \leq |j'-j| \leq 1$, and one of them is equal to 1; we suppose w.l.o.g. that $i'-i = 1$. Then $|\mathrm{Int}_2(\pi,p,p')| < r-1$ implies that $j-j' = 1$, and thus $|\mathrm{Int}_1(\pi,p,p')| = r > r-1$, contradiction. It follows that π has no $(r-1)$-close pair, and thus $w(\pi) \geq r$ by Proposition 3.1.

For $1 \leq j \leq r$, let $R_{1,j}$ be the rectangle containing only point $p_{1,j}$. We define a decomposition that first merges $R_{1,1}$ and $p_{2,1}$ to obtain $R_{2,1}$; then $R_{1,2}$ and $p_{2,2}$ to obtain $R_{2,2}$; …; then $R_{1,r}$ and $p_{2,r}$ to obtain $R_{2,r}$. We continue in a similar way with the next row: we merge $R_{2,1}$ and $p_{3,1}$ to obtain $R_{3,1}$; then $R_{2,2}$ and $p_{3,2}$ to obtain $R_{3,2}$; …; then $R_{2,r}$ and $p_{3,r}$ to obtain $R_{3,r}$ (see Figure 2(b)). After repeating this process for each row, only r rectangles $R_{r,1}$, …, $R_{r,r}$ remain. What needs to be observed is that when we merge $R_{i,j}$ and $p_{i+1,j}$ to obtain $R_{i+1,j}$, then $R_{i+1,j}$ 2-views only $R_{i+1,1}$, …, $R_{i+1,j-1}$, $R_{i,j+1}$, …, $R_{i,r}$ (i.e., $r-1$ rectangles) and does not 1-view any other rectangle. Therefore, the rectangle family is always r-wide. When there are only r remaining rectangles, we can merge them in any order. We get an r-wide decomposition of π, showing that $w(\pi) \leq r$. □

3.4 Tree representation

Although it is not used explicitly in the paper, we can give an alternative representation of a decomposition by a labeled tree. We state in Proposition 3.6 below a characterization of d-wide decompositions in terms of the associated tree.

A *numbered tree* is a (directed) tree $T = (V, A)$, where (i) $V \subseteq \mathbb{N}$, (ii) the leaves of T precede the internal nodes in the natural ordering, (iii) for each arc (i,j) it holds that $j < i$. We denote $L(T)$ the set of leaves of T, $I(T)$ the set of internal nodes of T and $N(T)$ the set of nodes of T. Suppose that $\mathcal{D} = (\mathcal{R}_0, \ldots, \mathcal{R}_s)$ is a decomposition of a permutation π, we represent it by a binary numbered tree T constructed as follows: (i) start with one vertex per element of $S(\pi)$; (ii) for p going from 1 to s, if $\mathcal{R}_p = \mathcal{R}_{p-1}[i,j \to k]$ then add a vertex k with arcs (k,i) and (k,j). Conversely, if T is a binary numbered tree with $L(T) = S(\pi)$, then there exists a decomposition $\mathcal{D}(\pi,T)$ whose associated tree is T.

We need the following additional definitions. Let T be a numbered tree. Given two nodes i,j of T, we denote $i <_T j$ (resp. $i \leq_T j$) if j is a proper ancestor (resp. ancestor) of i. Fix a node $i \in N(T)$. We denote by $T(i)$ the subtree of T rooted at i. We let $R(\pi,T,i)$ denote the bounding box of $\pi|L(T(i))$, and for each $\alpha \in \{1,2\}$ we let $I_\alpha(\pi,T,i) := I_\alpha(R(\pi,T,i))$. We let $S(T,i)$ denote the set of elements $j \in N(T)$ such that $j \leq i$ and that are maximal for $<_T$ with this property. We let $\mathcal{R}(\pi,T,i)$ denote the rectangle family $\mathcal{R}' = (S', R')$ with $S' = S(T,i)$ and for each $j \in S(T,i)$, $R(j) = R(\pi,T,j)$. Observe that if $\mathcal{D}(\pi,T) = (\mathcal{R}_0,\ldots,\mathcal{R}_s)$ and step p produces index j, then $\mathcal{R}_p = \mathcal{R}(\pi,T,j)$. Finally, we define the *restriction* of a numbered tree: if T is a numbered tree and $X \subseteq L(T)$, then $T|X$ is the minimum homeomorphic subtree of T containing the leaves of X.

Proposition 3.6. *Let π be a permutation, and let T be a binary numbered tree with $L(T) = S(\pi)$. The following statements are equivalent:*

(i) *$\mathcal{D}(\pi,T)$ is a d-wide decomposition of π;*

(ii) *for every $X \subseteq S(\pi)$ with $|X| \geq 2$, if i is the minimum internal node of $T|X$ then $L(T|X(i))$ is a d-close pair of $\pi|X$.*

Proof. *(ii)* $\Rightarrow$ *(i)*: We need to show that $\mathcal{D}(\pi,T)$ is a d-wide decomposition of π. Let $i \in I(T)$, and let

$S = S(T,i)$ and $\mathcal{R} = \mathcal{R}(\pi,T,i)$. Fix $i' \in S$, we need to show that $\text{view}(\mathcal{R},i') < d$. Fix $\alpha \in \{1,2\}$. Consider $j \in S$ such that $I_\alpha(\pi,T,i') \subseteq I_\alpha(\pi,T,j)$ and $I_\alpha(\pi,T,j)$ is maximal with this property. As $\text{view}_\alpha(\mathcal{R},i') - \{j\} + \{i'\} \subseteq \text{view}_\alpha(\mathcal{R},j)$, we have $|\text{view}_\alpha(\mathcal{R},j)| \geq |\text{view}_\alpha(\mathcal{R},i')|$. We will thus show that $|\text{view}_\alpha(\mathcal{R},j)| < d$, which will imply that $|\text{view}_\alpha(\mathcal{R},i')| < d$ as needed. Let $V = \text{view}_\alpha(\mathcal{R},j)$, let p (resp. p') be the minimal (resp. maximal) element of $L(T(j))$ in the order $<^\pi_\alpha$, and let j' denote the least common ancestor of p and p' in T, with $j' \leq_T j$. For each $x \in V$, choose an element $p_x \in L(T(x))$ such that $p <^\pi_\alpha p_x <^\pi_\alpha p'$; this is possible as $R(\pi,T,x)$ and $R(\pi,T,j)$ α-view each other, and as we cannot have $I_\alpha(\pi,T,j) \subseteq I_\alpha(\pi,T,x)$ by definition of j. Let $Y = \{p_x \mid x \in V\}$. As the nodes of V form an antichain for the relation $<_T$, the elements p_x are distinct, implying that $|Y| = |V|$. Consider the set $X = \{p,p'\} \cup Y$, and let $T' = T|X$. By definition of j', it is still a node of T'. Furthermore, we have that each p_x is not in $L(T(j))$ and thus not in $L(T'(j'))$, implying that $L(T'(j')) = \{p,p'\}$. We claim that j' is the internal node of T' with minimum index. By way of contradiction, suppose that j' is preceded by another internal node k, such that $k \leq j' \leq j \leq i$. Consider $k' \in S$ such that $k \leq_T k'$. Then $L(T(k'))$ intersects Y and thus $R(\pi,T,k')$ and $R(\pi,T,j)$ α-view each other, implying that $k' \in V$. It follows that $L(T(k'))$ contains two elements p_x, p_y, a contradiction. We obtain that j' is the internal node of T' with minimum index, and thus $L(T'(j')) = \{p,p'\}$ is a d-close pair of $\pi|X$. As $\text{Int}_\alpha(\pi|X,p,p') = Y$ and $|V| = |Y|$, we conclude that $|V| < d$.

$(i) \Rightarrow (ii)$: The fact that T verifies property (ii) is a consequence of the following points.

Point 1: let i be the internal node of T with minimum index, then $L(T(i))$ is a d-close pair in π. Suppose that $L(T(i)) = \{x,y\}$. Since the leaves of T precede the internal nodes, $S(T,i) - \{i\}$ contains exactly the leaves of T distinct from x,y. It follows that for each $\alpha \in \{1,2\}$, $\text{view}_\alpha(\mathcal{R}(\pi,T,i),i)$ corresponds to the elements of $\text{Int}_\alpha(\pi,x,y)$, which has thus cardinality less than d.

Point 2: for every $X \subseteq S(\pi)$, $\mathcal{D}(\pi|X, T|X)$ is a d-wide decomposition of $\pi|X$. It is enough to show this for X of the form $S(\pi) - \{j\}$ with $|X| \geq 2$. Suppose that X has this form, let $\pi' = \pi|X$ and let $T' = T|X$. Let u denote the parent of leaf j in T, let v denote the other child of u, and let w denote the parent of u in T (possibly undefined if u is the root of T). Note that T' is obtained from T by suppressing the nodes j and u, and attaching v as a child of w. Fix $i \in I(T')$, let the associated sets be $S = S(T,i)$ and $S' = S(T',i)$, and let the associated rectangle families be $\mathcal{R} = \mathcal{R}(\pi,T,i)$ and $\mathcal{R}' = \mathcal{R}(\pi',T',i)$. We need to show that $\text{view}(\mathcal{R}',i) < d$. Fix $\alpha \in \{1,2\}$, let $V = \text{view}_\alpha(\mathcal{R},i)$ and $V' = \text{view}_\alpha(\mathcal{R}',i)$. Observe that for $k \in I(T')$, if the intervals $I_\alpha(\pi',\mathcal{R}',k)$ and $I_\alpha(\pi',\mathcal{R}',i)$ intersect, then in π the corresponding intervals $I_\alpha(\pi,R,k)$ and $I_\alpha(\pi,\mathcal{R},i)$ also intersect. We consider three cases:

- Case 1: $j \in S$. In this case, it holds that $S' = S - \{j\}$, which implies that $V' \subseteq V$ and thus $|V'| \leq |V|$.
- Case 2: $u \in S$. In this case, it holds that $S' = S - \{u\} + \{v\}$. Thus, we have either $V' = V$ (if $u \notin V$) or $V' \subseteq V - \{u\} + \{v\}$ (if $u \in V$), and thus $|V'| \leq |V|$.
- Case 3: $u, j \notin S$. In this case, it holds that $S' = S$, which implies that $V' \subseteq V$ and thus $|V'| \leq |V|$.

In all cases, we obtain that $|V'| \leq |V| < d$, which concludes the proof. ☐

4 Finding decompositions

We present in this section a linear-time algorithm that either finds a large grid or gives a decomposition of bounded width:

THEOREM 4.1. *There exists an algorithm that, given a reduced permutation π of length n, runs in $O(n)$ time, and either finds an $r \times r$-grid of π, or returns the merge sequence of a $g(r)$-wide decomposition of π, where $g(r) = 2^{O(r \log r)}$.*

On one hand, Theorem 4.1 proves that grids are the only obstructions for having a bounded-width decomposition. On the other hand, this decomposition algorithm together with the algorithm of Section 5 working on bounded-width decompositions show that PERMUTATION PATTERN is linear-time solvable for fixed ℓ.

The proof of Theorem 4.1 relies on the following statement, which is a variation of the main technical result of Marcus and Tardos [24] in the proof of the Stanley-Wilf conjecture.

THEOREM 4.2. *Let $f(r) = r^4\binom{r^2}{r}$. For every $p, q, r \in \mathbb{N}$ with $p+q > 2$, if M is a point set included in $[p]\times[q]$ with $|M| > f(r)(p+q-2)$, then M contains an $r \times r$-grid. Moreover, such a grid can be found in time $O(|M|)$.*

As the result in [24] is not stated algorithmically and it finds a permutation pattern rather than a grid, we reproduce the proof in Appendix A with appropriate modifications. The proof of Theorem 4.1 below yields $g(r) = 4f(r)$; therefore, any improvement to Theorem 4.2 would immediately improve Theorem 4.1.

Proof. [Proof (of Theorem 4.1)] Let $d = 4f(r)$. The algorithm (see Algorithm 1) maintains an integer k, a rectangle family $\mathcal{R}$, a merge sequence Σ and a gridding G. Given a column x of G (resp. a row y of G), we denote by $d_1(x)$ (resp. $d_2(y)$) the number of rectangles of $\mathcal{R}$ included in column x (resp. row y).

Initially: $k = n+1$; $\mathcal{R} = \pi$; Σ is empty; the gridding G consists of rows $r_1, \dots, r_s$ and columns $c_1, \dots, c_s$, such that each row r_i and each column c_i ($1 \leq i < s$) contains exactly d points of π. The algorithm ensures that the following invariant conditions hold at each step:

(C1) each rectangle of $\mathcal{R}$ is included in a cell of G;
(C2) for any column x of G, $d_1(x) \leq d$, and for any row y of G, we have $d_2(y) \leq d$;
(C3) for any two consecutive columns x, x' of G, we have $d_1(x) + d_1(x') > d$;
(C4) for any two consecutive rows y, y' of G, we have $d_2(y) + d_2(y') > d$;
(C5) $\mathcal{R}$ is d-wide.

Clearly, these conditions hold initially.

The algorithm performs the following main step repeatedly. As long as $\mathcal{R}$ contains at least two rectangles, it does the following: (i) it looks for a cell (x, y) of G which contains at least two rectangles of $\mathcal{R}$; (ii) if there is no such cell, then it constructs a point set M corresponding to the nonempty cells and invokes the algorithm of Theorem 4.2 to find an $r \times r$-grid; (iii) otherwise, let i, j be two rectangles of $\mathcal{R}$ inside $G(x, y)$. The algorithm merges them in a new rectangle numbered by k, i.e. it updates $\mathcal{R} \leftarrow \mathcal{R}[i, j \to k]$, and it appends the pair (i, j, k) to Σ. After this merge, the algorithm can update the gridding G as follows: (i) if there is a column x' of G consecutive to x such that $d_1(x) + d_1(x') \leq d$, then merge columns x and x'; (ii) if there is a row y' of G consecutive to y such that $d_2(y) + d_2(y') \leq d$, then merge rows y and y'. Finally, the algorithm increments k, and moves to the next step of the loop.

Correctness. To prove the correctness of the algorithm, we first observe that the invariant conditions (C1)–(C5) hold every time Step 6 of Algorithm 1 is reached. Indeed, (C1) remains true, since we are modifying G by merging rows and columns; (C2) holds, since we merge two rows or columns only if they together contain at most d rectangles; and (C3)–(C4) hold, since we immediately merge any pair of rows or columns that would violate it. Invariant (C5) is a consequence of (C1) and (C2): a rectangle can only view other rectangles in the same row or column.

Suppose that G is a $p \times q$-gridding when Step 14 is reached, and let us construct the point set $M = \{(x, y) \in [p] \times [q] \mid G(x, y) \text{ contains a rectangle of } \mathcal{R}\}$. As the condition in Step 6 did not hold, each point $(x, y) \in M$ corresponds to a single rectangle of $\mathcal{R}$. It follows that $|M| > d\lfloor \frac{p}{2} \rfloor \geq d\frac{p-1}{2}$ by Invariant (C3), and $|M| > d\lfloor \frac{q}{2} \rfloor \geq d\frac{q-1}{2}$ by Invariant (C4). Thus, $|M| > d\frac{p+q-2}{4} = f(r)(p+q-2)$: we obtain by Theorem 4.2 that M contains an $r \times r$-grid, which yields an $r \times r$-grid in π. Therefore, Step 15 indeed finds an $r \times r$-grid in π, which we return.

Finally, we observe that the sequence Σ returned in Step 18 is the merge sequence of a d-wide decomposition. Indeed, these merges produce a sequence of rectangle families, with the last one containing only a single rectangle. By invariant (C5), each rectangle family is d-wide.

Details of implementing Algorithm 1 and achieving the claimed $O(n)$ running time appear in the full version of the paper [19]. ☐

We close this section by stating a corollary of Theorem 4.1. Given a permutation π, we can define three values that measure the "complexity" of π. The first measure is the largest integer r such that π contains an $r \times r$-grid; we denote this measure as $g(\pi)$. The second measure is the smallest integer d such that every subpattern of π has a d-close pair; we denote this measure as $d(\pi)$. The third measure is the width of π defined earlier, denoted by $w(\pi)$. We observe that these three measures are equivalent, in the following sense: we say that two functions m, m' mapping permutations to integers are *equivalent* if there exist increasing functions $f, g : \mathbb{N} \to \mathbb{N}$ such that $m(\pi) \leq f(m'(\pi))$ and $m'(\pi) \leq g(m(\pi))$ hold for any permutation π.

COROLLARY 4.1. *The measures g, d and w are equivalent.*

Proof. The equivalence between g and w follows from Proposition 3.4 and Theorem 4.1. We now argue that d is equivalent to the other two, by showing that $d(\pi) \leq w(\pi)$ and $g(\pi) < 5d(\pi)$. On one hand, $d(\pi) \leq w(\pi)$ follows from Proposition 3.6. On the other hand, $g(\pi) < 5d(\pi)$ follows by showing that any $5r \times 5r$-grid permutation π has a subpattern containing no r-close pair. Suppose that π is such a permutation, let G be the corresponding $5r \times 5r$-gridding, and let $p(x, y)$ denote the point of π in column x, row y of G. We define the subset $S' \subseteq S(\pi)$ containing the points $p(x, y)$ such that $y \equiv 2x \pmod 5$, and we let $\pi' = \pi|S'$. Observe that S' contains exactly r points in each row and column of G. Furthermore, we cannot have two distinct points $p(x, y), p(x', y') \in S'$ with $|x' - x| \leq 1$ and $|y' - y| \leq 1$. It follows that for any two elements $p, p' \in S'$, either $|\mathrm{Int}_1(\pi', p, p')| \geq r$ (if p, p' belong to non-consecutive columns) or $|\mathrm{Int}_2(\pi', p, p')| \geq r$ (if p, p' belong to non-consecutive rows). We conclude that π' is a subpattern of π with no r-close pair. ☐

Algorithm 1 BuildDecomposition(π)

Input:
π: a permutation of length n

$\mathcal{R}$:= the rectangle family representing π
$\Sigma := ()$
$k := n + 1$
initialize gridding G such that every row and column (except the last ones) contains exactly d points
while $|\mathcal{R}| > 1$
  if there are two rectangles $R(i), R(j)$ in some cell $G(x, y)$
    $\mathcal{R} := \mathcal{R}[i, j \to k]$
    append (i, j, k) to Σ
    if $d_1(x) + d_1(x') \leq d$ for some $x' \in \{x - 1, x + 1\}$
      merge columns x and x' in G
    if $d_2(y) + d_1(y') \leq d$ for some $y' \in \{y - 1, y + 1\}$
      merge rows y and y' in G
  else
    construct the point set M /* We have $|M| > f(r)(p + q - 2)$ */
    use the algorithm of Theorem 4.2 to find an $r \times r$ grid in M
    return the grid
  $k := k + 1$
return Σ

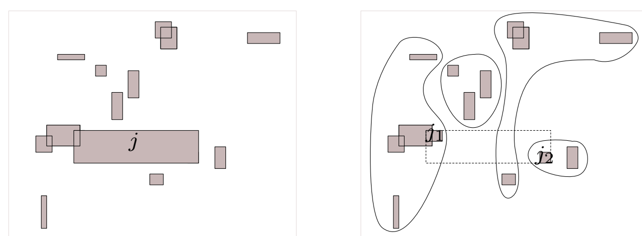
5 Solving the Permutation Pattern problem

This section is devoted to showing that the Permutation Pattern problem can be solved in linear time if a decomposition of bounded width is given in the input:

Theorem 5.1. *The* Permutation Pattern *problem can be solved in time $(d\ell)^{O(\ell)} \cdot n$, where $\ell = |\sigma|$ and $n = |\pi|$, if the merge sequence of a d-wide decomposition of π is given in the input.*

To prove Theorem 1.1, we run first the algorithm of Theorem 4.1 on permutation π with $r = \ell$, which takes $O(n)$ time. If the algorithm concludes that π has an $\ell \times \ell$-grid, we conclude that σ is a subpattern of π and we answer "yes". Otherwise, we obtain the merge sequence of a $g(\ell)$-wide decomposition of π, where $g(\ell) = 2^{O(\ell \log \ell)}$. Using Theorem 5.1, we can then decide if σ is a subpattern of π in time $(g(\ell)\ell)^{O(\ell)} \cdot n = 2^{O(\ell^2 \log \ell)} \cdot n$.

For the proof of Theorem 5.1, let $\mathcal{D} = (\mathcal{R}_0, \ldots, \mathcal{R}_{n-1})$ be the decomposition of π given in the input, with $\mathcal{R}_i = (S_i, R_i)$. Recall that each rectangle in $\mathcal{R}_i$ was created by a sequences of merges (possibly 0) from the rectangle family $\mathcal{R}_0$ representing the permutation π. We denote by $L(j)$ the set of points (more precisely, indices) taking part in the merges creating rectangle indexed by j. For example, in Figure 1, we have $L(13) = \{3, 4, 5, 8\}$. Note that, even if a point p is covered by rectangle j, it is not necessarily in $L(j)$: for example, in Figure 1, point 6 is covered by rectangle 13, but 6 is not in $L(13)$, as it did not take part in any of the 3 merges creating 13 (point 6 appears only in $L(6)$, $L(11)$, $L(14)$, and $L(15)$).



(a) (b)

Figure 3: Step $i+1$ of the decomposition merges j_1 and j_2 to j. (a) A connected set K of G_{i+1} containing j. (b) Replacing j with j_1 and j_2 gives a set K^{Π} inducing 4 connected components in G_i.

For $0 \leq i \leq n - 1$, we define the *visibility graph* G_i at step i of the decomposition the following way: the vertex set of G_i is S_i and $x, y \in S_i$ are adjacent if and only if the rectangles $R_i(x)$ and $R_i(y)$ α-view each other for some $\alpha \in \{1, 2\}$. As $\mathcal{R}_i$ is a d-wide rectangle family, it follows that G_i has maximum degree at most $2d$. Figure 3(a) shows a connected set of rectangles in the visibility graph (by "connected set", we mean that they induce a connected subgraph of the visibility graph).

We solve the Permutation Pattern problem using dynamic programming. For each step i, we define a set of subproblems. Informally, a subproblem asks for a subset of σ to be embedded into elements of π that appear in a set K of rectangles inducing a connected subgraph of the visibility graph G_i, with the elements of σ distributed among the rectangles of K in a specified way.

For the formal definition of the subproblems, we need the following definition first. Given two sets X, Y, a *distribution of X into Y* is a function $F : Y \to 2^X$ such that for $i, j \in Y$ distinct, $F(i) \cap F(j) = \emptyset$; the *range* of F is $\mathrm{Rng}(F) = \cup_{i \in Y} F(i)$. An *admissible subproblem* is a triple $t = (i, K, F)$, where

- $0 \le i \le n-1$,
- K is a connected subset of G_i, and
- F is a distribution of $S(\sigma)$ into K such that $F(i) \neq \emptyset$ for each $i \in K$.

Note that the last condition implies that K can have at most ℓ vertices. We define the *range* of t as $\mathrm{Rng}(F)$. The number of possible distributions is $(|K|+1)^{|S(\sigma)|} = \ell^{O(\ell)}$. The following simple fact bounds the number of possible connected sets, and it follows that the number of subproblems is $(d\ell)^{O(\ell)} \cdot n$ for a given i.

PROPOSITION 5.1. *If G is a graph with maximum degree Δ and v is a vertex of G, then the number of sets K of size at most ℓ such that $v \in K$ and $G[K]$ is connected is $\Delta^{O(\ell)}$. Moreover, all these sets can be enumerated in time $\Delta^{O(\ell)}$.*

Proof. The vertices of $G[K]$ can be visited by a walk of length at most $2\ell - 1$ starting at v. In each step of the walk, we move to one of the at most Δ neighbors. Thus there are at most $\Delta^{2\ell-1}$ such walks, which is an upper bound on the number of sets K. Enumerating all these walks gives $\Delta^{O(\ell)}$ sets that are not necessarily distinct. However, we may sort these sets in time $\Delta^{O(\ell)}$ and remove the duplicates. □

We say that t is *satisfiable* iff there exists a mapping $\phi : \mathrm{Rng}(F) \to S(\pi)$ such that

(i) for each $p \in \mathrm{Rng}(F)$, if $p \in F(j)$, then $\phi(p) \in L(j)$, and
(ii) ϕ is an embedding of $\sigma|\mathrm{Rng}(F)$ into π.

In this case, we say that ϕ is a *solution* of t. Recall that $\mathcal{R}_{n-1}$ contains only a single rectangle j and $L(j) = S(\pi)$. Therefore, there is an embedding from σ to π if and only if the subproblem $(n-1, \{j\}, F)$ is satisfiable, where F is the distribution of $S(\sigma)$ into $\{j\}$ such that $F(j) = S(\sigma)$.

Lemma 5.1 below gives a recurrence relation that allows us to decide if a subproblem $t = (i+1, K, F)$ is satisfiable, assuming that we have computed the satisfiable subproblems at step i. Our goal is to show that a solution ϕ for t can be constructed by putting together solutions for particular subproblems at step i.

Suppose that $t = (i+1, K, F)$ is an admissible subproblem, and suppose that step $i+1$ of the decomposition merges j_1, j_2 into j. Clearly, this means that $L(j)$ is the disjoint union of $L(j_1)$ and $L(j_2)$. Any solution ϕ of t maps $F(j)$ to $L(j) = L(j_1) \cup L(j_2)$, hence it defines a bipartition of the elements of $L(j)$. As a first step of solving t, we guess this bipartition, that is, which elements of $F(j)$ are mapped to $L(j_1)$ and to $L(j_2)$ (there are $2^{|L(j)|} \le 2^\ell$ such bipartitions). Let $X = F(j)$, and fix a bipartition $\Pi = (X_1, X_2)$ of X. Mapping ϕ maps $\mathrm{Rng}(F)$ to the rectangles $K - \{j\} + \{j_1, j_2\}$ of G_i. However, there is a technical detail here: if X_1 or X_2 is empty, then ϕ does not map any element of $\mathrm{Rng}(F)$ to $L(j_1)$ or $L(j_2)$, respectively. Therefore, we define the set K^Π as follows:

$$K^\Pi = \begin{cases} K - \{j\} + \{j_1, j_2\} & \text{if } X_1, X_2 \neq \emptyset, \\ K - \{j\} + \{j_1\} & \text{if } X_1 \neq \emptyset,\ X_2 = \emptyset, \\ K - \{j\} + \{j_2\} & \text{if } X_1 = \emptyset,\ X_2 \neq \emptyset. \end{cases}$$

We define the distribution $F^\Pi : K^\Pi \to 2^{S(\sigma)}$ that describes how mapping ϕ maps the elements of K to the rectangles in K^Π:

$$F^\Pi(k) = \begin{cases} F(k) & \text{if } k \notin \{j_1, j_2\}, \\ X_1 & \text{if } k = j_1, \\ X_2 & \text{if } k = j_2. \end{cases}$$

Assuming that we have already computed the satisfiable subproblems at step i, we would like to use this information to decide whether there is a solution ϕ satisfying $t = (i+1, K, F)$ that corresponds to the bipartition Π. Let us observe that if (i, K^Π, F^Π) happens to be an admissible and satisfiable subproblem, then it immediately implies the existence of such a solution ϕ. However, in general, K^Π is not necessarily connected. In that case, we would like to put together the solution ϕ from the solutions of the subproblems corresponding to the connected components of $G_i[K^\Pi]$. Formally, if C is a connected component of $G_i[K^\Pi]$, we define the subproblem $t^{\Pi,C} = (i, C, F^\Pi|C)$. If $C_1, \ldots, C_m$ are the connected components of $G_i[K^\Pi]$, then we let $T^\Pi = \{t^{\Pi,C_1}, \ldots, t^{\Pi,C_m}\}$ denote the set of subproblems corresponding to these components; observe that they are admissible subproblems. Note that $G_i[K^\Pi]$ can have more than two connected components: it is not true that every connected component contains either j_1 or j_2. As Figure 3(b) shows, rectangle j can view rectangles that neither j_1 nor j_2 view, thus there can be connected components not containing either j_1 or j_2.

We would like to combine solutions of the subproblems in T^Π to obtain a solution for subproblem t. However, the subproblems have to satisfy a certain condition in order for this to be possible. Consider two admissible subproblems $t_1 = (i, K_1, F_1)$ and $t_2 = (i, K_2, F_2)$. We say that t_1 and t_2 are *independent* if $\mathrm{Rng}(F_1), \mathrm{Rng}(F_2)$ are disjoint, K_1 and K_2 are disjoint, and G_i has no edge between K_1 and K_2. Observe that the subproblems in T^Π are pairwise independent. Suppose that t_1 and t_2 are independent with solutions $\phi_1 : \mathrm{Rng}(F_1) \to S(\pi)$ and $\phi_2 : \mathrm{Rng}(F_2) \to S(\pi)$, respectively. Then the mapping $\phi : \mathrm{Rng}(F_1) \cup \mathrm{Rng}(F_2) \to S(\pi)$ defined the obvious way from ϕ_1 and ϕ_2 is not necessarily a correct embedding of $\sigma|\mathrm{Rng}(F_1) \cup \mathrm{Rng}(F_2)$. The problem is that if $r_1 \in K_1$ and $r_2 \in K_2$, then for some $s_1 \in F_1(r_1)$ and $s_2 \in F_2(r_2)$, the relative position of the points $\phi(s_1)$ and $\phi(s_2)$ is not necessarily the same as the relative position of s_1 and s_2. However, the crucial observation here is that the rectangles r_1 and r_2 do not view each other, hence the relative position of $\phi(s_1)$ and $\phi(s_2)$ depend only on the relative position of r_1 and r_2, and *not* on the actual selection of points in r_1 and r_2. Therefore, the sets K_1 and K_2 and the distributions F_1 and F_2 already determine if the two solutions can be combined. Formally, two independent subproblems $t_1 = (i, K_1, F_1)$ and $t_2 = (i, K_2, F_2)$, are said to be *compatible* if for each $r_1 \in K_1, r_2 \in K_2$ and $\alpha \in \{1, 2\}$:

- if $I_\alpha(R_i(r_1)) < I_\alpha(R_i(r_2))$ then for each $p_1 \in F_1(r_1), p_2 \in F_2(r_2)$ it holds that $p_1 <^\sigma_\alpha p_2$, and
- if $I_\alpha(R_i(r_2)) < I_\alpha(R_i(r_2))$ then for each $p_1 \in F_1(r_1), p_2 \in F_2(r_2)$ it holds that $p_2 <^\sigma_\alpha p_1$.

Note that by the independence assumption for t_1 and t_2, one of the above two conditions must hold. We are now able to state the recurrence relation as follows:

LEMMA 5.1. *Suppose that $t = (i + 1, K, F)$ is an admissible subproblem. Then t is satisfiable if and only if*

- *$j \notin K$ and $t' = (i, K, F)$ is satisfiable, or*
- *$j \in K$ and there exists a bipartition Π of X such that the subproblems of T^Π are satisfiable and pairwise compatible.*

Proof. The case when $j \notin K$ is clear, by observing that if $j \notin K$ then t' is also an admissible subproblem. Let us consider the case when $j \in K$.

Suppose that t is satisfiable through a mapping $\phi : Z \to S(\pi)$ with Z the range of t. Let $\Pi = (X_1, X_2)$ with $X_r = \phi^{-1}(L(j_r))$, we show that the subproblems of T^Π satisfy the requirement. Suppose that the connected components of $G_i[S^\Pi]$ are $C_1, \ldots, C_m$, and let $t_r = t^{\Pi, C_r}$ for $r \in [m]$. As the C_r's are the connected components of $G_i[S^\Pi]$, each t_r is an admissible subproblem, and they are pairwise independent. Let Z_r denote the range of t_r, then $Z_1, \ldots, Z_m$ form a partition of Z. We first show that each subproblem t_r is satisfiable through $\phi|Z_r$. Point (ii) of the definition is verified as it holds for ϕ, and for point (i) observe that if $p \in F^\Pi(k)$ and $k \in C_r$ then: either $k \neq j_1, j_2$, hence $p \in F(k)$ and thus $\phi(p) \in L(k)$ (by definition of ϕ), or $k = j_s$ in which case $p \in X_s$ and thus $\phi(p) \in L(j_s)$ (by definition of X_s). We now show that two triples $t_r = (i, K_1, F_1), t_s = (i, K_2, F_2)$ are compatible. Indeed, suppose that $k \in K_1$ and $k' \in K_2$ are such that $I_\alpha(R_i(k)) < I_\alpha(R_i(k'))$, then for $p \in F_1(k)$ and $p' \in F_2(k')$ we have $\phi(p) \in L(k)$, $\phi(p') \in L(k')$ (by definition of ϕ), and thus $\phi(p) <^\pi_\alpha \phi(p')$, which implies that $p <^\sigma_\alpha p'$ as ϕ is an embedding.

Conversely, suppose that there exists $\Pi = (X_1, X_2)$ bipartition of X such that the subproblems of T^Π are satisfiable and pairwise compatible. Suppose that the connected components of $G_i[S^\Pi]$ are $C_1, \ldots, C_m$, and let $t_r = t^{\Pi, C_r}$ for $r \in [m]$. Let Z_r denote the range of t_r, and let Z denote the range of t, then $Z_1, \ldots, Z_m$ form a partition of Z. Suppose that t_r is satisfiable through a mapping $\phi_r : Z_r \to S(\pi)$. We can then define $\phi : Z \to S(\pi)$ which coincides with ϕ_r on Z_r. We show that t is satisfiable through ϕ. We first show Point (i) of the definition. Suppose that $p \in F(k)$. If $k \neq j$, we have that $p \in (F^\Pi|C_r)(k)$ for some r, which implies that $\phi(p) = \phi_r(p) \in L(k)$ (by definition of ϕ_r). If $k = j$, then we have $p \in X_s$ for some $s \in \{1, 2\}$, and thus $p \in (F^\Pi|C_r)(j_s)$ for the component C_r containing j_s, which implies that $\phi(p) = \phi_r(p) \in L(j_s)$ (by definition of ϕ_r) and thus in $L(j)$. We now show Point (ii) of the definition. Let p, p' be two elements of Z and $\alpha \in \{1, 2\}$, and suppose that $p <^\sigma_\alpha p'$, we need to show that $\phi(p) <^\pi_\alpha \phi(p')$. If p, p' belong to a same set Z_r, then $\phi(p) = \phi_r(p) <^\pi_\alpha \phi_r(p') = \phi(p')$ (by definition of ϕ_r). If $p \in Z_r, p' \in Z_s$ with $r \neq s$, suppose that $p \in (F^\Pi|C_r)(k)$ and $p' \in (F^\Pi|C_s)(k')$. As t_r and t_s are compatible and as $p <^\sigma_\alpha p'$, we have $I_\alpha(R_i(k)) < I_\alpha(R_i(k'))$. As $\phi(p) = \phi_r(p) \in L(k)$ and $\phi(p') = \phi_s(p') \in L(k')$, we conclude that $\phi(p) <^\pi_\alpha \phi(p')$. ☐

Lemma 5.1 allows us to determine if $t = (i + 1, K, F)$ is satisfiable, assuming that we have solved all subproblems at step i (see Algorithm 2). Let us briefly sketch how to implement this algorithm in time $(d\ell)^{O(\ell)} \cdot n$; more details and a more precise time bound can be found in the full version of the paper [19].

We maintain a representation of the graph G_i and for every $v \in S_i$, a linked list of every subset K of size at most ℓ containing v such that $G_i[K]$ is connected; note that each such list is of size $d^{O(\ell)}$ by Lemma 5.1. For each such set K, there is a linked list of all the satisfiable subproblems (i, K, F) for every distribution F; the length of this list is $\ell^{O(\ell)}$. In order to get G_{i+1} from G_i efficiently, we maintain a sorted linked list of

Algorithm 2 FINDPATTERN$(\pi, \sigma, \mathcal{D})$

Input:
π: a permutation of length n
σ: a permutation of length ℓ
$\mathcal{D}$: a decomposition of π

for $i := 0$ to $n-2$
  suppose that step $i+1$ merges j_1, j_2 to obtain j.
  update G_i to obtain G_{i+1}.
  for every subproblem $t = (i+1, K, F)$
    if $j \notin K$
      add $(i+1, K, F)$ to the list of satisfiable subproblems if (i, K, F) is satisfiable.
    else
      for every bipartition Π of $F(j)$
        compute V^Π
        compute the connected components $C_1, \ldots, C_q$
        compute the set T^Π of subproblems.
        if the subproblems in T^Π are pairwise compatible
          add $(i+1, K, F)$ to the list of satisfiable subproblems.
let j be the unique rectangle in $\mathcal{R}_{n-1}$
let F be the distribution of $S(\sigma)$ into $\{j\}$ with $F(j) = S(\sigma)$.
if $(n-1, j, F)$ is satisfiable
  return "yes"
else
  return "no"

the horizontal endpoints of all the rectangles appearing in $\mathcal{R}_i$ (containing two entries for each rectangle) and a similar list for the vertical endpoints. When j_1 and j_2 are merged to obtain j, we can use these lists of endpoints to efficiently find all those rectangles that j views, but neither j_1 not j_2 views. Therefore, we can efficiently update G_i to obtain G_{i+1}. Then for every vertex of G_{i+1} at distance at most ℓ, we have to recompute the list of connected subsets; there are $d^{O(\ell)}$ such vertices and enumeration of the subsets takes $d^{O(\ell)}$ time at each vertex.

To solve the subproblems at step $i+1$, we have to update only those subproblems $t = (i+1, K, F)$ for which $j \in K$ holds, where j is the new rectangle created at this step. For each such subproblem, we enumerate every bipartition Π of $L(j)$ and compute the set K^Π and distribution F^Π. We compute the connected components $C_1, \ldots, C_m$ of $G_i[K^\Pi]$ and the set of subproblems T^Π. By Lemma 5.1, we need to check if these problems are pairwise compatible, and if so, we have to add this satisfiable subproblem to the list of every $v \in K$. Note that all these steps involve only vertices at distance at most ℓ from j, j_1, or j_2, whose number is $d^{O(\ell)}$, and each operation can be performed in time $d^{O(\ell)}$. As there are $(d\ell)^{O(\ell)}$ subproblems $(i+1, K, F)$ with $j \in K$, all the updates for step $i+1$ can be done in time $(d\ell)^{O(\ell)}$.

6 Hardness results

In this section, we establish the W[1]-hardness of some variants of the PERMUTATION PATTERN problem. We first consider the following constrained version of the problem.

> PARTITIONED PERMUTATION PATTERN
> **Input:** Two permutations σ and π, a partition of $S(\pi)$ in sets S_i ($i \in S(\sigma)$)
> **Question:** Does there exist an embedding ϕ of σ into π such that $\phi(i) \in S_i$ for each $i \in S(\sigma)$?

THEOREM 6.1. PARTITIONED PERMUTATION PATTERN *is* W[1]*-hard for parameter* $|\sigma|$*, even when* σ *is a canonical* $r \times r$*-grid.*

Proof. We give a reduction from PARTITIONED CLIQUE [26]. Let $\mathcal{I}$ be an instance of PARTITIONED CLIQUE, consisting of a graph $H = (V, E)$, an integer k, and a partition of V into sets $V_1, \ldots, V_k$. We let σ be the canonical $(3k+1) \times (3k+1)$-grid. We now describe the construction of π. For each $i \in [k]$, let $v_i^1, \ldots, v_i^{n_i}$ be an enumeration of V_i. We start with a rectangle $R = I \times J$, and we subdivide I into intervals $I_0, \ldots, I_k$ and J into intervals $J_0, \ldots, J_k$. Then we subdivide each interval I_i ($i \in [k]$) into $3n_i$ consecutive intervals $I^1_{i,j}, I^2_{i,j}, I^3_{i,j}$

$(1 \le j \le n_i)$, and we subdivide each interval J_i ($i \in [k]$) into $3n_i$ consecutive intervals $J^1_{i,j}, J^2_{i,j}, J^3_{i,j}$ $(1 \le j \le n_i)$. Let G be the resulting $(3n+1) \times (3n+1)$-gridding of R. Each cell C of G will contain at most one point of π according to the following criterion: (i) if $C = I^2_{i,x} \times J^2_{j,y}$, then C contains a point iff ($i = j$ and $x = y$) or ($i \neq j$ and $\{v^i_x, v^j_y\} \in E$); (ii) every other cell contains one point. It remains to describe the horizontal ordering of the points inside a column of G, and the vertical ordering of the points inside a row of G. Inside a column x of G corresponding to an interval $I^r_{i,j}$, we order the points such that if $p \in G(x,y)$ and $p' \in G(x,y')$, then $\mathrm{pr}_1(p) < \mathrm{pr}_1(p')$ iff $y' < y$. Inside a row y of G corresponding to an interval $J^r_{i,j}$, we order the points such that if $p \in G(x,y)$ and $p' \in G(x',y)$ then $\mathrm{pr}_2(p) < \mathrm{pr}_2(p')$ iff $x < x'$. Inside column 1 of G corresponding to the interval I_0, we order the points such that if $p \in G(1,y)$ and $p' \in G(1,y')$, then: (i) if y corresponds to $J^r_{i,j}$ and y' corresponds to $J^s_{i,j'}$ with $j < j'$, then $\mathrm{pr}_1(p) < \mathrm{pr}_1(p')$; (ii) in all other cases, $\mathrm{pr}_1(p) < \mathrm{pr}_1(p')$ iff $y' < y$. Inside row 1 of G corresponding to the interval J_0, we order the points such that if $p \in G(x,1)$ and $p' \in G(x',1)$, then: (i) if x corresponds to $I^r_{i,j}$ and x' corresponds to $I^s_{i,j'}$ with $j' < j$, then $\mathrm{pr}_2(p) < \mathrm{pr}_2(p')$; (ii) in all other cases, $\mathrm{pr}_2(p) < \mathrm{pr}_2(p')$ iff $x < x'$. We let π be the resulting permutation. Finally, we define the sets S_i ($i \in S(\sigma)$) as follows. Let us denote by $p(x,y)$ the element of $S(\sigma)$ corresponding to the point in the (x,y)th cell of the gridding of σ. For a point p of π, we define $f_1(p)$ such that if $p \in I_0 \times J$ then $f_1(p) = 1$, and if $p \in I^r_{i,j} \times J$ then $f_1(p) = 3(i-1) + r + 1$; we define $f_2(p)$ symmetrically, and we put the point p in $S_{p(f_1(p), f_2(p))}$. Let $\mathcal{I}'$ be the resulting instance of Partitioned Permutation Pattern, then $\mathcal{I}'$ can clearly be constructed in polynomial time.

We now argue for the correctness of the reduction. Suppose that H has a clique $C = \{v^{p_1}_1, \ldots, v^{p_k}_k\}$. Given $2 \le i \le 3k+1$, let $f(i) = ((i-2) \text{ div } 3 + 1, (i-2) \bmod 3 + 1)$. We define $\phi : S(\sigma) \to S(\pi)$ as follows:

- We map $p(1,1)$ to the unique point of π in $I_0 \times J_0$;
- for $2 \le x \le 3k+1$, let $(i,r) = f(x)$, then we map $p(x,1)$ to the unique point of π in $I^r_{i,p_i} \times J_0$;
- for $2 \le y \le 3k+1$, let $(j,s) = f(y)$, then we map $p(1,y)$ to the unique point of π in $I_0 \times J^s_{j,p_j}$;
- for $2 \le x, y \le 3k+1$, let $(i,r) = f(x)$ and $(j,s) = f(y)$, then we map $p(x,y)$ to the unique point of π in $I^r_{i,p_i} \times J^s_{j,p_j}$.

Note that in the last case, the existence of the point follows from the fact that $\{v^i_{p_i}, v^j_{p_j}\} \in E$. We then have $\phi(p) \in S_p$ for each $p \in S(\sigma)$, and it can be checked that ϕ is an embedding of σ into π. Conversely, suppose that ϕ is an embedding of σ into π such that $\phi(p) \in S_p$ for each $p \in S(\sigma)$. Given $i \in [k]$, consider the elements $q^1_i = p(3(i-1)+2, 1), q^2_i = p(3(i-1)+3, 1), q^3_i = p(3(i-1)+4, 1)$ in $S(\sigma)$. We have $q^1_i <^\sigma_2 q^2_i <^\sigma_2 q^3_i$, and by the arrangement of the points it means that there exists $1 \le p_i \le n_i$ such that $\phi(q^a_i)$ is the unique point of $I^a_{i,p_i} \times J_0$. Likewise, given $j \in [k]$, by considering the points $r^1_j = p(1, 3(j-1)+2), r^2_j = p(1, 3(j-1)+3), r^3_j = p(1, 3(j-1)+4)$, we obtain that there exists $1 \le p'_j \le n_j$ such that $\phi(r^a_j)$ is the unique point of $I_0 \times J^a_{j,p'_j}$. Now, for $i, j \in [k]$, consider $s_{i,j} = (3(i-1)+3, 3(j-1)+3)$. Since $q^1_i <^\sigma_1 s_{i,j} <^\sigma_1 q^3_i$, it follows that $\phi(s_{i,j})$ is in $I^2_{i,p_i} \times J$; since $r^1_j <^\sigma_2 s_{i,j} <^\sigma_2 r^3_j$, it follows that $\phi(s_{i,j})$ is in $I \times J^2_{j,p'_j}$; thus $\phi(s_{i,j})$ is the unique point of $I^2_{i,p_i} \times J^2_{j,p'_j}$. In particular, since for $i \in [k]$ the point $\phi(r_{i,i})$ is present we obtain that $p_i = p'_i$, and since for $i, j \in [k]$ distinct the point $\phi(r_{i,j})$ is present we obtain that $\{v^{p_i}_i, v^{p_j}_j\} \in E$. We conclude that $C = \{v^{p_1}_1, \ldots, v^{p_k}_k\}$ is a clique of H. □

We now consider the generalization of the Permutation Pattern problem to d-dimensional permutations. Given an integer d, a *d-dimensional point* is a tuple $p = (x_1, \ldots, x_d) \in \mathbb{N}^d$, and for $\alpha \in [d]$ we define $\mathrm{pr}_\alpha(p) = x_\alpha$. A *$d$-dimensional permutation* is defined as a tuple $\pi = (S, P)$ with S a set and $P : S \to \mathbb{N}^d$ an injection such that $P(S)$ is a set of d-dimensional points in general position (i.e. for each $\alpha \in [d]$ it holds that pr_α is injective on $P(S)$); we let $S(\pi) = S$. Given $p, p' \in S$ and $\alpha \in [d]$, we denote $p <^\pi_\alpha p'$ iff $\mathrm{pr}_\alpha(P(p)) < \mathrm{pr}_\alpha(P(p'))$. Given two d-dimensional permutations σ and π, an *embedding* of σ into π is a function $\phi : S(\sigma) \to S(\pi)$ such that for every $p, p' \in S(\sigma)$, for every $\alpha \in [d]$, $p <^\sigma_\alpha p'$ iff $\phi(p) <^\pi_\alpha \phi(p')$. We consider the following problem.

d-Dimensional Permutation Pattern	
Input:	Two d-dimensional permutations σ and π.
Question:	Does there exist an embedding of σ into π?

Theorem 6.2. *For every $d \ge 3$, d-Dimensional Permutation Pattern is $\mathsf{W}[1]$-hard for parameter $|\sigma|$.*

Proof. We prove the result for $d = 3$, since the extension to any $d \ge 3$ is straightforward. We give the following reduction from Partitioned Permutation Pattern. Let $\mathcal{I}$ be an instance of Partitioned Permutation Pattern, consisting of a permutation σ with $S(\sigma) = [\ell]$, a permutation π, and a partition of $S(\pi)$ in sets

$S_1, \dots, S_\ell$. We assume w.l.o.g. that for $i, j \in [\ell]$, $i < j$ iff $i <_1^\sigma j$. Suppose that $\sigma = (S_\sigma, P_\sigma)$ and $\pi = (S_\pi, P_\pi)$. We define two 3-dimensional permutations $\sigma' = (S_\sigma, P_{\sigma'})$ and $\pi' = (S_\pi, P_{\pi'})$ as follows. For each $i \in S_\sigma$, if $P_\sigma(i) = (x, y)$ then we set $P_{\sigma'}(i) = (x, y, i)$. Now, we define a numbering $f(i)$ of the points of π as follows: we first number the points of S_1 by decreasing x-coordinate, then the points of S_2 by decreasing x-coordinate, etc. For each $i \in S_\pi$, if $P_\pi(i) = (x, y)$ then we set $P_{\pi'}(i) = (x, y, f(i))$. Let $\mathcal{I}' = (\sigma', \pi')$ be the resulting instance, observe that $\mathcal{I}'$ can be constructed in polynomial time. We show that $\mathcal{I}$ is a positive instance of Partitioned Permutation Pattern iff $\mathcal{I}'$ is a positive instance of 3-Dimensional Permutation Pattern.

Suppose that $\mathcal{I}$ is a positive instance via an embedding ϕ of σ into π. Given $i, j \in S_\sigma$ distinct, for each $\alpha \in \{1, 2\}$ we have that $i <_\alpha^{\sigma'} j$ iff $\phi(i) <_\alpha^{\pi'} \phi(j)$ (by definition of ϕ), and for $\alpha = 3$ we have that $i <_3^{\sigma'} j$ iff $i < j$ iff $f(\phi(i)) < f(\phi(j))$ (since $\phi(i) \in S_i$ and $\phi(j) \in S_j$, and as i, j are distinct) iff $\phi(i) <_3^{\pi'} \phi(j)$. Conversely, suppose that $\mathcal{I}'$ is a positive instance via an embedding ϕ of σ' into π'. Clearly, ϕ is also an embedding of σ into π, and there is a function $\psi : [\ell] \to [\ell]$ such that $\phi(i) \in S_{\psi(i)}$ for each $i \in [\ell]$. Suppose by contradiction that ψ is not the identity function, then there exist $i, j \in [\ell]$ such that $i < j$ and $\psi(j) \leq \psi(i)$. If $\psi(i) = \psi(j)$, we obtain that $i <_1^\sigma j$ and thus $\phi(i) <_1^\pi \phi(j)$, but then $f(\phi(j)) < f(\phi(i))$; if $\psi(j) < \psi(i)$, we also obtain that $f(\phi(j)) < f(\phi(i))$. In both cases, we obtain that $\phi(j) <_3^{\pi'} \phi(i)$ and $i <_3^{\sigma'} j$, contradicting the assumption that ϕ is an embedding. $\square$

7 The case of t-monotone permutations

The notions of *increasing* and *decreasing* permutations are defined the obvious way and we will use *monotone* for a permutation that is either increasing or decreasing. Formally, let π be a permutation, we say that π is *increasing* (resp., *decreasing*) if for any $p, p' \in S(\pi)$, it holds that $p <_1^\pi p'$ iff $p <_2^\pi p'$ (resp., $p' <_2^P p$); we say that π is *monotone* if it is either increasing or decreasing. Given an integer t, we say that π is *t-increasing* (resp., *t-monotone*) if there is a partition $\Pi = (S_1, \dots, S_t)$ of $S(\pi)$ such that $\pi|S_i$ is increasing (resp., monotone) for each $i \in [t]$. The partition Π will be called a *t-increasing (resp., t-monotone) partition.*

Let us briefly discuss the recognition problem for these classes. While t-increasing permutations can be recognized in polynomial time, recognizing t-monotone permutations is NP-hard for unbounded t [6]. For a fixed t, recoginizing t-monotone permutations is fixed-parameter tractable: the algorithm of Heggernes et al. solves the problem in time $2^{O(t^2 \log t)} \cdot n^{O(1)}$ [21]. We can also give a constant-factor approximation for the problem in the sense that, given a permutation π of length n, in time $O(n^2)$ we can either find a ct-monotone partition of π, or conclude that π is not t-monotone. This easily follows from Greene theorem with $c = 2$ [17], and there exists a better algorithm that yields $c = 1.71$ [16].

7.1 Width of t-monotone permutations It can be seen that a t-increasing permutation cannot have a $(t+1) \times (t+1)$-grid, and that a t-monotone permutation cannot have a $(2t+1) \times (2t+1)$-grid. It follows that these permutations have bounded width (at most $4f(2t+1)$) by Theorem 4.1. The following result gives a better bound.

Proposition 7.1. *If π is a t-monotone permutation, then $w(\pi) \leq 6t - 5$.*

Proof. We first need some additional definitions on rectangle families. Let $\mathcal{R} = (S, R)$ be a rectangle family. Given $S' \subseteq S$, we let $\mathcal{R}|S' = (S', R|S')$. We say that $\mathcal{R}$ is *increasing* if there is an enumeration $i_1, \dots, i_n$ of S such that for each $p < q$, $I_1(R(i_p)) < I_1(R(i_q))$ and $I_2(R(i_p)) < I_2(R(i_q))$. Likewise, we say that $\mathcal{R}$ is *decreasing* if there is an enumeration $i_1, \dots, i_n$ of S such that for each $p < q$, $I_1(R(i_p)) < I_1(R(i_q))$ and $I_2(R(i_p)) > I_2(R(i_q))$. In each case, we call *consecutive* two indices of the form i_p, i_{p+1}. We say that $\mathcal{R}$ is *monotone* if it is increasing or decreasing; we say that $\mathcal{R}$ is *t-monotone* if it admits a t-monotone partition, i.e. a partition of S in sets $S_1, \dots, S_t$ such that each $\mathcal{R}|S_r$ is monotone.

We are now ready to prove the proposition. Suppose that π has a t-monotone partition $\Pi = (S_1, \dots, S_t)$. Starting with $\mathcal{R} = \pi$, we will do a sequence of merges maintaining the following invariants: (i) $\mathcal{R}$ is a $(6t-5)$-wide rectangle family; (ii) Π is a t-monotone partition of $\mathcal{R}$. At each step, we proceed as follows. Suppose that $\mathcal{R} = (S, R)$ and $\Pi = (S_1, \dots, S_t)$. If each set S_r is a singleton, then $\mathcal{R}$ has at most t rectangles and we can easily complete the sequence of merges. Suppose now that some set S_r is not a singleton. We define the set $\mathcal{M}$ of *mergeable pairs* as the set of pairs (i, j) coming from a same set S_r and that are consecutive in $\mathcal{R}|S_r$; observe that $\mathcal{M}$ is not empty. Given a pair $m = (i, j) \in \mathcal{M}$ coming from a set S_r, by *merging* the pair m, we mean the following: (i) replace $\mathcal{R}$ by $\mathcal{R}' = \mathcal{R}[i, j \to k]$ where k is a new index; (ii) replace Π by $\Pi' = (S_1', \dots, S_t')$ where $S_r' = S_r - \{i, j\} + \{k\}$, and $S_s' = S_s$ for $s \neq r$. Observe that after this operation, Π' is still a t-monotone partition of $\mathcal{R}'$. We will show that we can find a mergeable pair in $\mathcal{M}$ whose merging results in a new rectangle k with $\mathrm{view}(\mathcal{R}', k) < 6t - 5$.

Consider a pair $m = (i,j) \in \mathcal{M}$, and let R be the smallest rectangle enclosing $R(i) \cup R(j)$. For each $\alpha \in \{1,2\}$, we define $\mathrm{pin}_\alpha(m)$ as the set of elements $i' \in S - \{i,j\}$ such that $I_\alpha(R(i')) \subseteq I_\alpha(R)$, we define the sum $\Sigma_\alpha := \sum_{m\in\mathcal{M}} |\mathrm{pin}_\alpha(m)|$, and we define $\Sigma := \sum_{m\in\mathcal{M}} \max(|\mathrm{pin}_1(m)|, |\mathrm{pin}_2(m)|)$.

Claim 7.1. *For each $\alpha \in \{1,2\}$, $\Sigma_\alpha \le 2(t-1)|\mathcal{M}|$.*

Proof. Fix $\alpha \in \{1,2\}$. We say that an element $i \in S$ *contributes* to a pair $m \in \mathcal{M}$ if $i \in \mathrm{pin}_\alpha(m)$; we let $\mathrm{cont}(i)$ denote the number of pairs $m \in \mathcal{M}$ to which i contributes. Then clearly $\Sigma_\alpha = \sum_{i\in S} \mathrm{cont}(i)$. Observe that an element $i \in S_r$ contributes to no pair in S_r, and to at most two pairs in each set S_s ($s \ne r$), hence $\mathrm{cont}(i) \le 2(t-1)$. This yields that $\Sigma_\alpha \le 2(t-1)|S|$. We can slightly improve the bound to $2(t-1)|\mathcal{M}|$ as follows. For each $s \in [t]$, let i_s, i'_s be the first and last indices in the natural enumeration of S_s. Let us sort the elements $s \in [t]$ by increasing order of the left endpoint of $I_\alpha(R(i_s))$; this gives an enumeration E_1 of $[t]$. Likewise, let us sort the elements $s \in [t]$ by decreasing order of the right endpoint of $I_\alpha(R(i'_s))$; this gives an enumeration E_2 of $[t]$. Now, if s is the pth element of E_1 (resp., E_2), we have that $\mathrm{cont}(i_s) \le 2(p-1)$ (resp., $\mathrm{cont}(i'_s) \le 2(p-1)$). It follows that $\Sigma_\alpha \le 2(t-1)(|S|-2t) + 2\sum_{p=1}^{t} 2(p-1) = 2(t-1)|S| - 4t(t-1) + 2t(t-1) = 2(t-1)(|S|-t) = 2(t-1)|\mathcal{M}|$. ⌟

Now, for each $m \in \mathcal{M}$, we have $\max(|\mathrm{pin}_1(m)|, |\mathrm{pin}_2(m)|) \le |\mathrm{pin}_1(m)| + |\mathrm{pin}_2(m)|$, and thus $\Sigma \le \sum_{m\in\mathcal{M}}(|\mathrm{pin}_1(m)| + |\mathrm{pin}_2(m)|) = \Sigma_1 + \Sigma_2 \le 4(t-1)|\mathcal{M}|$. Hence, we can find a pair $m \in \mathcal{M}$ coming from a set S_r such that for each $\alpha \in \{1,2\}$ it holds that $|\mathrm{pin}_\alpha(m)| \le 4(t-1)$. Consider the result of merging pair m into a new rectangle k, thus yielding the rectangle family $\mathcal{R}'$ and the t-monotone partition $\Pi' = (S'_1, \ldots, S'_t)$.

Claim 7.2. $\mathrm{view}(\mathcal{R}', k) \le 6(t-1)$.

Proof. We show that $|\mathrm{view}_\alpha(\mathcal{R}', k)| \le 6(t-1)$ holds for each $\alpha \in \{1,2\}$. Let $V = \mathrm{view}_\alpha(\mathcal{R}', k)$, we partition V in two sets $V_1 := \mathrm{pin}_\alpha(m)$ and $V_2 := V \backslash V_1$. Observe that S'_r contains no element from V, and that for $s \ne r$ the set S'_s can contain at most two elements from V_2 (for if S'_s contains three elements $u, v, w \in V$ with $I_\alpha(R(u)) < I_\alpha(R(v)) < I_\alpha(R(w))$, then $v \in V_1$). It follows that $|V_2| \le 2(t-1)$, and as we also have $|V_1| = |\mathrm{pin}_\alpha(m)| \le 4(t-1)$, we conclude that $|V| \le 6(t-1)$ as claimed. ⌟

That is, $\mathcal{R}'$ is also $(6t-5)$-wide, as required. □

Observe that the proof of Proposition 7.1 can be turned into an algorithm that takes a permutation π of length n together with a t-monotone partition, and produces in time $O(n^2)$ a $(6t-5)$-wide decomposition of π. Combining this with Theorem 5.1, this yields a $t^{O(\ell)}n^2$ algorithm for the Permutation Pattern problem on t-monotone permutations. However, it turns out that this problem admits a very simple algorithm using the theory of constraint satisfaction problems and completely independent of our decomposition and width measure; we present this algorithm in the next section.

7.2 CSPs and t-monotone permutations We present an algorithm for solving Permutation Pattern on t-monotone instances by reducing it to a constraint satisfaction problem. The algorithm relies on the known fact that a CSP instance with a majority polymorphism can be solved in polynomial time.

As our use of CSP techniques is standard and what makes it surprising is the observation is that these techniques solve the problem immediately, we recall only briefly the most important notions related to CSPs. For more background, the reader is referred to, e.g., the survey [8].

Definition 7.3. *An instance of a* constraint satisfaction problem *is a triple (V, D, C), where:*

- *V is a set of variables,*
- *D is a domain of values,*
- *C is a set of constraints, $\{c_1, c_2, \ldots, c_q\}$. Each constraint $c_i \in C$ is a pair $\langle s_i, R_i \rangle$, where:*
  - *s_i is a tuple of variables of length m_i, called the* constraint scope, *and*
  - *R_i is an m_i-ary relation over D, called the* constraint relation.

For each constraint $\langle s_i, R_i \rangle$ the tuples of R_i indicate the allowed combinations of simultaneous values for the variables in s_i. The length m_i of the tuple s_i is called the *arity* of the constraint. A *solution* to a constraint satisfaction problem instance is a function f from the set of variables V to the domain of values D such that for each constraint $\langle s_i, R_i \rangle$ with $s_i = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_m} \rangle$, the tuple $\langle f(v_{i_1}), f(v_{i_2}), \ldots, f(v_{i_m}) \rangle$ is a member of R_i.

A *polymorphism* of a (say, n-ary) relation R on D is a mapping $f : D^k \to D$ for some k such that for any tuples $\mathbf{a}_1, \ldots, \mathbf{a}_k \in R$ the tuple

$$f(\mathbf{a}_1, \ldots, \mathbf{a}_k) = (f(\mathbf{a}_1[1], \ldots, \mathbf{a}_k[1]), \ldots, f(\mathbf{a}_1[n], \ldots, \mathbf{a}_k[n]))$$

belongs to R. A *majority polymorphism* is a ternary polymorphism f with the property that $f(x,x,y) = f(x,y,x) = f(y,x,x) = x$ for any $x, y \in D$. It is known that if there is a function f that is a majority polymorphism for *every* constraint of the instance, then the instance can be solved in polynomial time [14].

We solve a constrained version of the Permutation Pattern problem, where the image of each element of σ has to be in a prespecified monotone sequence. That is, given two permutations σ and π, given a t-monotone partition $\Sigma = (S_1, \dots, S_t)$ of σ and a t-monotone partition $\Pi = (S'_1, \dots, S'_t)$ of π, the task is to find an embedding ϕ of σ into π such that $\phi(S_i) \subseteq S'_i$ holds for each $i \in [t]$. Such an embedding ϕ will be called a (Σ, Π)-embedding. We show that the (Σ, Π)-embedding problem is polynomial-time solvable; then by trying all possible partitions Σ, we get an algorithm for the original Permutation Pattern problem on t-monotone permutations.

We define a CSP instance $I = (V, D, C)$ with $V = S(\sigma)$ and $D = S(\pi)$. The intended meaning of the value of variable $x \in S(\sigma)$ is the image of x in the embedding, or in other words, we want to introduce constraints such that there is a one-to-one correspondence between the solutions of I and the (Σ, Π)-embeddings. The constraints are defined as follows. For each $x, y \in S(\sigma)$ and $\alpha \in \{1, 2\}$, if $x <^\sigma_\alpha y$ holds, then we introduce the constraint $\langle (x, y), R_{x,y,\alpha} \rangle$, where $R_{x,y,\alpha}$ is defined as follows. Suppose that $x \in S_i$ and $y \in S_j$ (possibly $i = j$); we let

$$R_{x,y,\alpha} = \{(x', y') \mid x' \in S'_i, y' \in S'_j, x' <^\pi_\alpha y'\}.$$

That is, the images of x and y have to appear in the prespecified classes of the partition and have to respect the same ordering relation in π as in σ. It is easy to see that indeed there is a correspondence between solutions and embeddings.

Our goal is to show that there is a function f that is majority polymorphism for every constraint in I. Given three elements $x'_1, x'_2, x'_3 \in S(\pi)$ and an $\alpha \in \{1, 2\}$, we define $\mathrm{mid}_\alpha(x'_1, x'_2, x'_3)$ as the median value x' of these three elements with respect to the ordering $\leq^\pi_\alpha$, that is, at most one of $\{x'_1, x'_2, x'_3\}$ is strictly larger than x' and at most one element is strictly smaller than x'; note that this value x' is well defined. The crucial observation where monotone sequences come into play is that if $x'_1, x'_2, x'_3 \in S'_i$, i.e., they come from the same monotone sequence, then $\mathrm{mid}_1(x'_1, x'_2, x'_3) = \mathrm{mid}_2(x'_1, x'_2, x'_3)$. This allows us to show that both of these functions are polymorphisms of every constraint:

Proposition 7.2. *Both* mid_1 *and* mid_2 *are polymorphisms of every constraint in* I.

Proof. Suppose that $(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3) \in R_{x,y,\alpha}$; we need to show that

$$(\mathrm{mid}_1(x'_1, x'_2, x'_3), \mathrm{mid}_1(y'_1, y'_2, y'_3)) \in R_{x,y,\alpha}$$
$$(\mathrm{mid}_2(x'_1, x'_2, x'_3), \mathrm{mid}_2(y'_1, y'_2, y'_3)) \in R_{x,y,\alpha}$$

Suppose that $x \in S_i$ and $y \in S_j$ hold; it follows that $x'_1, x'_2, x'_3 \in S'_i$ and $y'_1, y'_2, y'_3 \in S'_j$. Therefore, as observed above, mid_1 and mid_2 coincide on these values, thus it is sufficent to prove the statement for mid_α.

It is clear that $\mathrm{mid}_\alpha(x'_1, x'_2, x'_3) \in S'_i$ and $\mathrm{mid}_\alpha(y'_1, y'_2, y'_3) \in S'_j$. Thus we need to show only $\mathrm{mid}_\alpha(x'_1, x'_2, x'_3) \leq^\pi_\alpha \mathrm{mid}_\alpha(y'_1, y'_2, y'_3)$. This is simply the well-known fact that the median function is a polymorphism of a linear ordering. For completeness, we provide a simple proof. Without loss of generality, suppose that $x'_1 \leq^\pi_\alpha x'_2 \leq^\pi_\alpha x'_3$, that is, $\mathrm{mid}_\alpha(x'_1, x'_2, x'_3) = x'_2$. We consider the following cases:

- If $\mathrm{mid}_\alpha(y'_1, y'_2, y'_3) = y'_1$, then either $y'_2 \leq^\pi_\alpha y'_1$ (implying $x'_2 \leq^\pi_\alpha y'_2 \leq^\pi_\alpha y'_1$) or $y'_3 \leq^\pi_\alpha y'_1$ (implying $x'_2 \leq^\pi_\alpha x'_3 \leq^\pi_\alpha y'_3 \leq^\pi_\alpha y'_1$).
- If $\mathrm{mid}_\alpha(y'_1, y'_2, y'_3) = y'_2$, then $x'_2 \leq^\pi_\alpha y'_2$ holds.
- If $\mathrm{mid}_\alpha(y'_1, y'_2, y'_3) = y'_3$, then $x'_2 \leq^\pi_\alpha x'_3 \leq^\pi_\alpha y'_3$ holds.

In all cases, we have shown that $\mathrm{mid}_\alpha(x'_1, x'_2, x'_3) \leq^\pi_\alpha \mathrm{mid}_\alpha(y'_1, y'_2, y'_3)$, completing the proof. $\square$

Combining Proposition 7.2 with the result of [14], we obtain a polynomial-time algorithm for the above CSP instance. Actually, we may observe that this particular CSP can be directly reduced to a 2SAT instance and can be solved in time $O(\ell^2 n^2)$. Note that this immediately implies a fixed-parameter algorithm for the Permutation Pattern problem on t-monotone permutations: given a pattern σ, and a t-monotone target π with a t-monotone partition $\Pi = (S_1, \dots, S_t)$, we enumerate each possible partition Σ of σ into t classes, test whether Σ is a t-monotone partition of σ, and if so test in $O(\ell^2 n^2)$ the existence of a (Σ, Π)-embedding. Thus, we obtain:

Theorem 7.1. *Given an instance* (σ, π) *of the* Permutation Pattern *problem, and a* t*-monotone partition* Π *of* π*, we can solve* (σ, π) *in time* $O(t^\ell \ell^2 n^2)$ *and polynomial space.*

We make two remarks about this result. First, it extends a result of [18] that solves the problem in $O(t^\ell \ell n)$ time for t-increasing permutations. Second, note that it assumes that a t-monotone partition of π is given as input. However, if we have a promise that π is t-monotone without knowing the explicit partition, then we can first obtain a $2t$-monotone partition in

$O(n^2)$ time as mentioned above, and thus we can solve the problem in $O((2t)^\ell \ell^2 n^2)$ time for t-monotone permutations.

The previous theorem has an interesting consequence. Observe that a permutation of length n is always t-monotone for $t = 2\lceil\sqrt{n}\rceil$ (this can be deduced from Greene's theorem [17] or from Erdős-Szekeres theorem [13]). Plugging this into Theorem 7.1 yields a nontrivial $n^{\frac{\ell}{2}+o(\ell)}$ time algorithm for PERMUTATION PATTERN using polynomial space. This has to be compared with the algorithm of [1] that uses $n^{0.47\ell+o(\ell)}$ time and exponential space. Note also that our FPT algorithm for PERMUTATION PATTERN uses exponential space, due to the dynamic-programming step.

THEOREM 7.2. *We can solve the* PERMUTATION PATTERN *problem in time* $n^{\frac{\ell}{2}+o(\ell)}$ *and polynomial space.*

References

[1] S. Ahal and Y. Rabinovich. On Complexity of the Subpattern Problem. *SIAM J. Discrete Math.*, 22(2):629–649, 2008.

[2] M. Albert and M. Atkinson. Simple permutations and pattern restricted permutations. *Discrete Math.*, 300(1–3):1–15, 2005.

[3] H. L. Bodlaender. On Linear Time Minor Tests with Depth-First Search. *J. Algorithms*, 14(1):1–23, 1993.

[4] H. L. Bodlaender. On Disjoint Cycles. *Int. J. Found. Comput. Sci.*, 5(1):59–68, 1994.

[5] P. Bose, J. F. Buss, and A. Lubiw. Pattern Matching for Permutations. *Inf. Process. Lett.*, 65(5):277–283, 1998.

[6] A. Brandstädt and D. Kratsch. On Partitions of Permutations into Increasing and Decreasing Subsequences. *Elektronische Informationsverarbeitung und Kybernetik*, 22(5/6):263–273, 1986.

[7] M.-L. Bruner and M. Lackner. The computational landscape of permutation patterns. *CoRR*, abs/1301.0340, 2013.

[8] H. Chen. A rendezvous of logic, complexity, and algebra. *ACM Comput. Surv.*, 42(1), 2009.

[9] A. Claesson and S. Kitaev. Classification of bijections between 321- and 132-avoiding permutations. *Sém. Lothar. Combin.*, 60:Art. B60d, 30, 2008/09.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.).* MIT Press, 2009.

[11] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, New York, 1999.

[13] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[14] T. Feder and M. Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[15] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, Berlin, 2006.

[16] F. V. Fomin, D. Kratsch, and J.-C. Novelli. Approximating minimum cocolorings. *Inf. Process. Lett.*, 84(5):285–290, 2002.

[17] C. Greene. An Extension of Schensted's Theorem. *Advances in Mathematics*, 14:254–265, 1974.

[18] S. Guillemot. Parameterized Algorithms for Inclusion of Linear Matchings. In *ISAAC*, pages 354–363, 2011.

[19] S. Guillemot and D. Marx. Finding small patterns in permutations in linear time. *CoRR*, abs/1307.3073, 2013.

[20] S. Guillemot and S. Vialette. Pattern Matching for 321-Avoiding Permutations. In *ISAAC*, pages 1064–1073, 2009.

[21] P. Heggernes, D. Kratsch, D. Lokshtanov, V. Raman, and S. Saurabh. Fixed-Parameter Algorithms for Cochromatic Number and Disjoint Rectangle Stabbing. In *SWAT*, pages 334–345, 2010.

[22] D. E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms.* Addison-Wesley, 1968.

[23] P. A. MacMahon. *Combinatory Analysis.* London: Cambridge University Press, 1915.

[24] A. Marcus and G. Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Comb. Theory, Ser. A*, 107(1):153–160, 2004.

[25] R. Möhring and F. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. In *Algebraic and combinatorial methods in operations research*, volume 95 of *North-Holland Math. Stud.*, pages 257–355, 1984.

[26] K. Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.*, 67(4):757–771, 2003.

[27] V. R. Pratt. Computing Permutations with Double-Ended Queues, Parallel Stacks and Parallel Queues. In *STOC*, pages 268–277, 1973.

[28] P. Rosenstiehl and R. E. Tarjan. Gauss codes, planar hamiltonian graphs, and stack-sortable permutations. *J. Algorithms*, 5(3):375–390, 1984.

[29] R. Simion and F. W. Schmidt. Restricted permutations. *European J. Combin.*, 6(4):383–406, 1985.

A Proof of Theorem 4.2

We follow the proof technique of [24]. We show by induction on $p+q$ that: if $M \subseteq [p] \times [q]$ is a point set with no $r \times r$-grid, then $|M| \leq f(r)(p+q-2)$. Clearly, we can assume that $p, q, r \geq 2$. The base case of the induction is when $p+q \leq 2r^2(r+1)$. In this case, observe that $\binom{r^2}{r}(p+q-2) \geq (r+1)^2$ as $p,q,r \geq 2$. As $|M| \leq \frac{(p+q)^2}{4}$, we thus have $|M| \leq r^4(r+1)^2 \leq r^4\binom{r^2}{r}(p+q-2) = f(r)(p+q-2)$. For the general case, we now suppose that $p+q > 2r^2(r+1)$.

Let $p' = \lceil \frac{p}{r^2} \rceil$ and $q' = \lceil \frac{q}{r^2} \rceil$. We partition $[p]$ into intervals $I_1, \dots, I_{p'}$ such that each I_x $(1 \leq x < p')$ has length r^2, and we partition $[q]$ into intervals $J_1, \dots, J_{q'}$ such that each J_y $(1 \leq y < q')$ has length r^2. For each $x \in [p'], y \in [q']$, we define the block $B_{x,y} = I_x \times J_y$. From M, we define a point set $M' \subseteq [p'] \times [q']$ which contains a point (x,y) iff the block $B_{x,y}$ contains a point of M. We say that a block $B_{x,y}$ is *wide* (respectively *tall*) if it contains points of M in at least r different columns (respectively rows).

LEMMA A.1. *M' contains no $r \times r$-grid.*

Proof. Towards a contradiction, suppose that M' contains an $r \times r$-grid, via a gridding G consisting of intervals $I'_1, \dots, I'_r$ and $J'_1, \dots, J'_r$. Define the gridding G' of M consisting of intervals $I''_1, \dots, I''_r$ with $I''_x = \cup_{j \in I'_x} I_j$, and of intervals $J''_1, \dots, J''_r$ with $J''_y = \cup_{j \in J'_y} J_j$. For every $i,j \in [r]$, we have that M' contains a point $(x',y') \in I'_x \times J'_y$, and thus $B_{x',y'} = I_{x'} \times J_{y'}$ contains a point of M. It follows that M contains a point of $I''_x \times J''_y \supseteq B_{x',y'}$, and as this holds for every $x,y \in [r]$ we conclude that M contains an $r \times r$-grid, contradiction. □

LEMMA A.2. *For every $x \in [p']$, the number of blocks in column x that are wide is less than $r\binom{r^2}{r}$.*

Proof. Suppose the contrary. For each wide block $B_{x,y}$, suppose that it contains points of M in r different columns $x_1, \dots, x_r$, and associate to $B_{x,y}$ the set $\{x_1, \dots, x_r\} \subseteq I_x$. There are at most $\binom{r^2}{r}$ possible such sets, and thus there are r blocks $B_{x,y_1}, \dots, B_{x,y_r}$ $(y_1 < \dots < y_r)$ that are assigned the same subset $S = \{x_1, \dots, x_r\}$. Set $x_{r+1} = r^2x+1$, and define the intervals $I'_1, \dots, I'_r$ by $I'_i = [x_i, x_{i+1}-1]$ for $i \in [r]$. Next, set $y_{r+1} = q'+1$, and define the intervals $J'_1, \dots, J'_r$ by $J'_j = \cup_{y_j \leq y < y_{j+1}} J_y$ for $j \in [r]$. These two families of intervals define a $r \times r$-gridding G. Observe that for every $i,j \in [r]$, $G(i,j)$ intersects M, as B_{x,y_j} contains a point in column x_i. We conclude that M contains an $r \times r$-grid, a contradiction. □

LEMMA A.3. *For every $y \in [q']$, the number of blocks in row y that are tall is less than $r\binom{r^2}{r}$.*

Proof. Follows by the same proof as Lemma A.2. □

We are now ready to finish the proof. Let X_1 denote the set of wide blocks, let X_2 denote the set of tall blocks, and let X_3 denote the set of nonempty blocks that are neither wide nor tall. We obtain $|X_1| \leq p'r\binom{r^2}{r}$ by Lemma A.2, $|X_2| \leq q'r\binom{r^2}{r}$ by Lemma A.3, and $|X_3| \leq f(r)(p'+q'-2)$ by Lemma A.1. As each block contains at most r^4 points of M, and as each block of X_3 contains at most $(r-1)^2$ points of M, it follows that:

$$\begin{aligned} |M| &\leq r^4|X_1| + r^4|X_2| + (r-1)^2|X_3| \\ &\leq r^5\binom{r^2}{r}(p'+q') + (r-1)^2 f(r)(p'+q'-2) \\ &\leq f(r)(r^2-r+1)(p'+q'-2) + 2rf(r) \end{aligned}$$

Now, observe that $p'+q'-2 \leq \frac{p+q}{r^2}$, and thus:

$$\begin{aligned} |M| &\leq f(r)\frac{r^2-r+1}{r^2}(p+q) + 2rf(r) \\ &\leq f(r)(p+q) - f(r)\frac{p+q}{r^2} + 2rf(r) \\ &\leq f(r)(p+q) - 2f(r)(r+1) + 2rf(r) = f(r)(p+q-2) \end{aligned}$$

Here, we have used that $r \geq 2$ in the second inequality, and we have used that $p+q \geq 2r^2(r+1)$ in the third inequality. We obtain that $|M| \leq f(r)(p+q-2)$, concluding the proof.

Implementation. Following the above proof, we describe a recursive algorithm FINDGRID(p,q,r,M) that takes a point set $M \subseteq [p] \times [q]$ with $|M| > f(r)(p+q-2)$, and finds in $O(|M|)$ time an $r \times r$-grid in M. Note that by the assumption on $|M|$ we have $p,q = O(|M|)$. The algorithm assumes that M is described as a list of points, and the resulting grid is described by listing the endpoints of the horizontal and vertical intervals.

We first describe a subroutine FINDBLOCKS(p,q,r,M) that collects the non-empty blocks of M'. The result will be represented by a list Blocks, where each entry $b \in \mathsf{Blocks}$ represents a non-empty block $B_{x,y}$ and holds two fields: $\mathsf{point}(b)$ equal to (x,y); $\mathsf{cols}(b)$ equal to the list of non-empty columns of the block, sorted by increasing order. The subroutine proceeds as follows. First, it arranges the points of M in columns, constructing for each $x \in [p]$ the set $L(x) = \{z \in M : pr_1(z) = x\}$; this can be performed in $O(|M|)$ time. Second, it scans the columns from left to right, collecting the blocks. For each row $y \in [q']$, we maintain a variable $\mathsf{block}[y]$ pointing to the last created block in row y. We initialize all variables $\mathsf{block}[y]$ to $\perp$. When processing column $x \in [p]$, we examine each point $(x,y) \in L(x)$, and in each case: (i) we compute the block $B_{x',y'}$ containing (x,y); (ii) if $\mathsf{block}[y'] = \perp$ or $\mathsf{block}[y']$ is a block b such that $pr_1(\mathsf{point}(b)) < x'$, then we allocate a new block b, we set $\mathsf{block}[y']$ to b, and we initialize $\mathsf{point}(b)$ to (x', y') and $\mathsf{cols}(b)$ to $\{x\}$; (iii) otherwise, if $b = \mathsf{block}[y']$ then we append x to $\mathsf{cols}(b)$ if it was not already present. The list Blocks is then returned; it is clear that it holds the desired information, and that its construction takes $O(|M|)$ time.

We now describe a second subroutine $\textsc{FindGridOrReduce}(p, q, r, M)$. The subroutine first calls $\textsc{FindBlocks}(p, q, r, M)$ to obtain the list Blocks. Now, for each column $x \in [p']$, it constructs a list $\mathsf{BigSets}[x]$ as follows. Initially each such list is empty. Then, we examine each block b of Blocks, compute $(x, y) = \mathsf{point}(b)$, test if $|\mathsf{cols}(b)| \geq r$, and if so we obtain S an arbitrary r-subset of $\mathsf{cols}(b)$, and add (y, S) to $\mathsf{BigSets}[x]$. For each $x \in [p']$, we determine if there are r entries of $\mathsf{BigSets}[x]$ that have the same second component; if so, we find an $r \times r$-grid G as in the proof of Lemma A.2, and we return (yes, G). If we find no such grid, then we construct the matrix M' containing the points $\mathsf{point}(b)$ for each block $b \in \mathsf{Blocks}$, and we return (no, M'). We claim that this algorithm can be implemented to run in $O(|M|)$ time. First, the construction of the lists $\mathsf{BigSets}$ can be done in time $O(\sum_{b \in \mathsf{Blocks}} |\mathsf{cols}(b)|) = O(|M|)$. Second, for a given $x \in [p']$, consider the time needed to determine if there are r entries of $\mathsf{BigSets}[x]$ that have the same second component. We can do this in time $O(r|\mathsf{BigSets}[x]|)$, by constructing a trie of height r where each leaf is labeled by an r-set S together with the set I of indices $y \in [q']$ such that $\mathsf{BigSets}[x]$ contains (y, S); note that the insertion of a new set in the trie takes $O(r)$, and that at the end of the construction we need to look for a leaf whose set of indices I contains at least r elements. Thus, the total time needed for this second step is at most $O(\sum_{b \in \mathsf{Blocks}} |\mathsf{cols}(b)|) = O(|M|)$. Finally, we can construct the reduced matrix M' in $O(|M|)$ time.

To conclude the description of the algorithm, we implement $\textsc{FindGrid}(p, q, r, M)$ as follows. First, we call $\textsc{FindGridOrReduce}$ (q, p, r, M^t), where $M^t = \{(y, x) : (x, y) \in M\}$. If this call returns (yes, G'), we conclude that G' is an $r \times r$-grid in M^t, and we return the corresponding $r \times r$-grid in M. Otherwise, we call $\textsc{FindGridOrReduce}(p, q, r, M)$. If this call returns (yes, G''), we conclude that G'' is an $r \times r$-grid in M, and we return it. Otherwise, we obtain (no, M'), where M' is the set of points $(x, y) \in [p'] \times [q']$ such that $B_{x,y}$ intersects M. As the two calls to $\textsc{FindGridOrReduce}$ answered negatively, we have obtained that $|X_1| \leq p'r\binom{r^2}{r}$ and that $|X_2| \leq q'r\binom{r^2}{r}$, and by the above proof we conclude that $|M'| > f(r)(p' + q' - 2)$. We remove some points of M' to obtain $M'' \subseteq M'$ such that $f(r)(p' + q' - 2) < |M''| \leq 1.1f(r)(p' + q' - 2)$. This is possible: since $f(r)(p' + q' - 2) \geq f(2) \geq 10$, we have $f(r)(p' + q' - 2) + 1 \leq 1.1f(r)(p' + q' - 2)$. Finally, we call recursively $\textsc{FindGrid}(p', q', r, M'')$.

The correctness of the algorithm follows from the above proof, so let us argue about the running time. Consider a call to $\textsc{FindGrid}(p, q, r, M)$ with $|M| > f(r)(p + q - 2)$. Assume that the two resulting calls to $\textsc{FindGridOrReduce}$ take time at most $c_1|M|$, and that the instructions executed inside the call to $\textsc{FindGrid}$ (excluding the function calls) take time at most $c_2|M|$. Let $c_0 = c_1 + c_2$, let c be the solution of $c = c_0 + \frac{3.3c}{r^2}$, and let $c' = \frac{2.2cf(r)}{r^2}$. As $r \geq 2$, it holds that c is positive and that $c \geq c_0$. We show by induction on $p+q$ that the call to $\textsc{FindGrid}(p, q, r, M)$ takes time at most $T(M) \leq c|M|$. If the call issues no recursive call, then it takes time at most $c_0|M| \leq c|M|$. Suppose now that it issues a recursive call $\textsc{FindGrid}(p', q', r, M'')$ with $f(r)(p' + q' - 2) < |M''| \leq 1.1f(r)(p' + q' - 2)$, and that this recursive call takes time at most $c|M''|$ by induction hypothesis. Considering the time taken by the initial call, we obtain:

$$\begin{aligned}
T(M) &\leq c_0|M| + c|M''| \\
&\leq c_0|M| + 1.1cf(r)(p' + q' - 2) \\
&\leq c_0|M| + 1.1c\frac{f(r)}{r^2}(p + q - 2) + c' \\
&\leq c_0|M| + 3.3c\frac{|M|}{r^2} \\
&\leq c|M|
\end{aligned}$$

Here we used that $|M''| \leq 1.1f(r)(p' + q' - 2)$ in the second inequality, that $p' + q' - 2 \leq \frac{p+q}{r^2}$ in the third inequality, and that $|M| > f(r)(p + q - 2)$ and $c' \leq \frac{2.2c|M|}{r^2}$ in the fourth inequality. As c is bounded by a constant independent of r ($c \leq 5.72c_0$), we conclude that the running time of $\textsc{FindGrid}(p, q, r, M)$ is $O(|M|)$.