

Parameterized Complexity of Asynchronous Border Minimization*

Robert Ganian¹, Martin Kronegger¹, Andreas Pfandler^{1,2},
and Alexandru Popa³

¹Vienna University of Technology, Vienna, Austria
firstname.lastname@tuwien.ac.at

²University of Siegen, Siegen, Germany

³Nazarbayev University, Astana, Kazakhstan
alexandru.popa@nu.edu.kz

March, 2015

Abstract. Microarrays are research tools used in gene discovery as well as disease and cancer diagnostics. Two prominent but challenging problems related to microarrays are the *Border Minimization Problem* (BMP) and the *Border Minimization Problem with given placement* (P-BMP).

In this paper we investigate the parameterized complexity of natural variants of BMP and P-BMP, termed BMP^e and P-BMP^e respectively, under several natural parameters. We show that BMP^e and P-BMP^e are in FPT under the following two combinations of parameters: 1) the size of the alphabet (c), the maximum length of a sequence (string) in the input (ℓ) and the number of rows of the microarray (r); and, 2) the size of the alphabet and the size of the border length (o). Furthermore, P-BMP^e is in FPT when parameterized by c and ℓ . We complement our tractability results with corresponding hardness results.

1 Introduction

DNA and peptide microarrays [3, 12] are important research tools used in gene discovery, multi-virus discovery as well as disease and cancer diagnosis. Apart from measuring the amount of gene expression [18], microarrays are an efficient tool for making a qualitative

*Supported by the Austrian Science Fund (FWF): P25518-N23 and P26696, and the German Research Foundation (DFG) under grant ER 738/2-1.

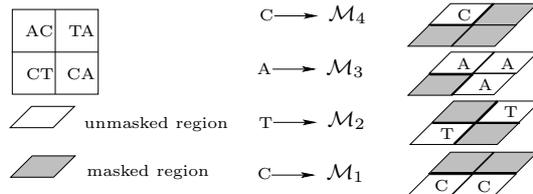


Figure 1: Asynchronous synthesis of a 2×2 microarray. The deposition sequence $\mathcal{D} = \text{CTAC}$ corresponds to four masks $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3,$ and \mathcal{M}_4 . The masked regions are shaded and the border between the masked and unmasked regions is represented by bold lines.

statement about the presence or absence of biological target sequences in a sample. For example, peptide microarrays are used for detecting tumor biomarkers [2, 16, 19].

A microarray is a plastic or glass slide consisting of thousands of sequences of nucleotides called *probes* that are assigned to one cell in the array. The synthesis process [10] consists of two components: *probe placement* and *probe embedding*. In the probe placement, the goal is to determine an assignment of each probe to a unique cell of the array. If the placement is given one has to create the sequences at their respective cells (probe embedding). This can be achieved with help of the following two operations: It is possible to *mask* a certain set of cells. Furthermore, one can *append* a certain nucleotide to the probes in all those cells which are currently unmasked. Essentially, the nucleotides are represented as characters and the probes as strings. In probe embedding we want to find a common supersequence of all probes, called the *deposition sequence*, and a sequence of 2D arrays describing the masks. The cells of a mask can be either masked (opaque) or unmasked (transparent) allowing the deposition of the nucleotide associated with the mask. For any cell, the concatenation of the nucleotides for which the cell is transparent has to match the probe in that cell of the microarray. See Figure 1 for an example [15].

Due to diffraction, the cells on the *border* between the masked and the unmasked regions are often subject to unintended illumination [10], and can compromise experimental results. Therefore, unintended illumination should be minimized. The magnitude of unintended illumination can be measured by the *border length* of the masks used, which is the number of borders shared between masked and unmasked regions, e.g., in Figure 1, the border length of $\mathcal{M}_1, \mathcal{M}_3, \mathcal{M}_4$ is 2 and \mathcal{M}_2 is 4 which yields a total border length of 10.

The problem of finding both the placement and the embedding is termed the Border Minimization Problem (BMP). If the placement is given and the task is to find only the embedding, we speak of P-BMP. We refer the reader to Section 2 for formal definitions of BMP and P-BMP.

VARIANTS OF BORDER MINIMIZATION. In this paper we consider the exhaustive variants of BMP and P-BMP, termed BMP^e and P-BMP^e respectively. The difference is that in P-BMP^e (and, consequently, in BMP^e) we assume that a mask is always applied

	c or c, r	c, ℓ	c, ℓ, r	c, o
P-BMP ^e	paraNP-h (Prop. 2)	FPT (Prop. 4)	FPT (Prop. 4)	FPT (Thm. 4)
BMP ^e	paraNP-h (Thm. 1)	open	FPT (Thm. 3)	FPT (Thm. 5)

Table 1: Overview of results.

exhaustively (we call this the *exhaustive rule*). More precisely, when a mask that synthesizes a character c is applied, the mask has a transparent cell wherever the corresponding sequence begins with the character c .

Without this assumption it is possible to artificially increase the length of the deposition sequence which, as a consequence, also increases the length of the sequence of masks. In most application scenarios this is undesirable, since applying a mask requires an additional cycle of work that causes a waste of material and can also introduce new errors. A second advantage of these exhaustive variants is that they allow the concise description of solutions: a solution to P-BMP^e is fully characterized by the deposition sequence, while for P-BMP it is also necessary to explicitly describe each mask in the sequence. To clarify, we remark that an optimal exhaustive solution need not always be an optimal solution for P-BMP (or BMP): there are cases where the border length can increase.

We illustrate the usefulness of the assumption by a simple example. In the P-BMP^e instance $a|b|a$, this assumption indeed helps to reduce the number of masks without increasing the border length. A non-exhaustive optimal solution might work on the left a first, while an exhaustive optimal solution works on both a concurrently. Even though the border length is in both cases 4, the non-exhaustive case could require an additional mask.

OUR RESULTS. Our results are summarized in Table 1. In this paper we investigate the parameterized complexity of the BMP^e and P-BMP^e problems under several natural parameters. First of all, throughout this work we consider the number of available nucleotides c (i.e., the alphabet size) as a parameter. Notice that this assumption does not impose a serious restriction, since in practice the number of available nucleotides is very limited (or even constant). Orthogonal to this assumption we explore the parameterized complexity of the BMP^e and P-BMP^e problem with respect to three natural parameters, i.e., the maximum length of a sequence in the array (ℓ), the maximum border length cost (o), and the maximum number of rows in the array (r). Since errors become more likely as the length of the sequence grows, the length of the constructed probes will be rather limited. Notice that the parameter o models the cost of a solution and hence is also a natural parameter. Finally, with the maximum number of rows r the shape of the array is restricted in the sense that the one dimension does not grow arbitrarily. This is, in particular, interesting because it allows to generalize from the one-dimensional case studied in [17].

More precisely, we show fpt-algorithms for BMP^e and P-BMP^e if we are given either c, ℓ, r or c, o as parameters. We complement these results with parameterized intractabil-

ity results, i.e., by showing **paraNP**-hardness. We use a polynomial time reduction from P-BMP^e to BMP^e to build upon the result that P-BMP^e parameterized by c and r is **paraNP**-hard¹ and obtain hereby **paraNP**-hardness for BMP^e parameterized by c and r . Notice that with the exception of BMP^e parameterized by c and ℓ , we obtain a full parameterized complexity map of the two considered problems with respect to all additional parameters considered in this paper. We furthermore provide a reduction relating the complexity of BMP^e parameterized by c and ℓ to k -BALANCED PARTITION on grids, a well-studied problem whose parameterized complexity on grids is open (Proposition 3).

The rest of the paper is organized as follows. In Section 2 we introduce the problems formally and give preliminaries. Then, in Section 3 we show the reduction from P-BMP^e to BMP^e . Section 4 introduces the fpt-algorithms and, finally, in Section 5 we present conclusions and open problems.

2 Preliminaries

For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \dots, n\}$. For two sequences s_1, s_2 , we use $s_1 \cdot s_2$ to mark their concatenation.

The microarray has size $r \times m$, where r is the number of rows and m is the number of columns. The multiset of input sequences (also called *probes*) is denoted by $\mathcal{S} = \{s_1, s_2, \dots, s_{r \cdot m}\}$ and the input alphabet by Σ . Moreover, let $c = |\Sigma|$. For any sequence s_i , we denote the length of the sequence by ℓ_i and the t -th character of a sequence s_i by $s_i[t]$. We use ℓ for the maximum length of the probes, i.e., $\ell = \max_{i \in [r \cdot m]} \ell_i$. Two cells of the array $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$ are said to be *neighbors* if $|x_1 - x_2| + |y_1 - y_2| = 1$. For each cell v , we denote the set of neighbors of v by $\mathcal{N}(v)$.

In order to give the formal definition of BMP , we introduce several notions related to the synthesis process.

Definition 1. A *placement* of the probe sequences is a bijective function φ that maps each probe sequence to a unique cell in the array.

Definition 2. A *deposition sequence* D for a set of sequences \mathcal{S} is a sequence of characters which is a common supersequence of all sequences in \mathcal{S} .

Definition 3. An *embedding* of a sequence s_i into a deposition sequence D is a length- $|D|$ sequence ε_i over alphabet $\Sigma \cup \{-\}$ such that:

1. ε_i contains precisely $|s_i|$ characters other than “-” occurring at positions $\varepsilon_i[u_1], \varepsilon_i[u_2], \dots, \varepsilon_i[u_{|s_i|}]$,
2. u_1 is the minimum position such that $\varepsilon_i[u_1] = s_i[1]$,
3. for $2 \leq j \leq |s_i|$, u_j is the minimum position such that $\varepsilon_i[u_j] = s_i[j]$ and $u_{j-1} < u_j$.

¹Although in [17] only NP-hardness is proven for P-BMP , the reduction can also be used to show **paraNP**-hardness for P-BMP^e when parameterized by c and r .

Informally, ε_i captures how a sequence is built (or, equivalently, deleted) by the deposition sequence; notice that due to the exhaustive rule, the embedding is uniquely determined by the deposition sequence. An *embedding* of a set of probes \mathcal{S} into a deposition sequence D is then denoted by $\varepsilon_D = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{|\mathcal{S}|}\}$. Note that we will drop the subscript when the associated deposition sequence is clear from the context. The final key notion we need are masks.

Definition 4. A mask \mathcal{M} (for some character c) is a 2D-array such that $\mathcal{M}(i, j)$ is either c or a space “-” (here the space means that the character is not deposited into this cell).

The sequence of masks associated with a deposition sequence D and a placement φ is $\omega = \mathcal{M}_1, \dots, \mathcal{M}_{|D|}$ where $\mathcal{M}_i(a, b) = \varepsilon_{\varphi^{-1}(a,b)}[i]$ for $i \in [|D|]$. Notice that due to the exhaustive rule, a mask for character c is always maximal with respect to c , i.e., there is no “-” in the mask that could be replaced by c . We introduce now the *border length* of a given placement of the probes in the array, which is the value we aim to optimize.

Definition 5. Let $\text{border}_D(s_i, s_j)$ be the Hamming distance between ε_i and ε_j (with respect to deposition sequence D). The *border length* of a placement φ and a deposition sequence D is then defined as the sum of borders over all pairs of neighboring probe sequences

$$\text{BL}(\varphi, D) = \sum_{\substack{\forall i, j \in \mathbb{N} : i < j < |\mathcal{S}| \\ \wedge \varphi(s_j) \in \mathcal{N}(\varphi(s_i))}} \text{border}_D(s_i, s_j). \quad (1)$$

We can also equivalently define border length in terms of the border length of all the masks.

Definition 6. For any mask \mathcal{M} of deposition character x , the border length of \mathcal{M} , denoted by $\text{BL}(\mathcal{M})$, is defined as the number of pairs of neighboring cells (i_1, j_1) and (i_2, j_2) such that $\mathcal{M}(i_1, j_1) = x$ and $\mathcal{M}(i_2, j_2) \neq x$. For a placement and deposition sequence that corresponds to a sequence of masks $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{|D|}$, we let

$$\text{BL}(\varphi, D) = \sum_{h=1}^{|D|} \text{BL}(\mathcal{M}_h) \quad (2)$$

The BMP^e and the P-BMP^e problem are defined as follows.

Problem 1. In the BMP^e problem, we are given $r, m \in \mathbb{N}$ and a multiset of $r \cdot m$ sequences \mathcal{S} . The objective is to find a placement φ and a deposition sequence D so that $\text{BL}(\varphi, D)$ is minimized.

Problem 2. In the P-BMP^e problem, we are given $r, m \in \mathbb{N}$ and a multiset of $r \cdot m$ sequences \mathcal{S} and a placement φ . The objective is to find a deposition sequence D so that $\text{BL}(\varphi, D)$ is minimized.

For a set $\pi \subseteq \{c, r, \ell, o\}$, we denote by BMP_π^e (P- BMP_π^e) the BMP^e (P- BMP^e) problem parameterized by π . For a problem BMP_π^e (P- BMP_π^e) where $o \in \pi$, we assume that an upper bound on the border length o is additionally given in the input and only solutions with minimum border length $\leq o$ are admitted.

We conclude this section with some useful observations. A deposition sequence D is called *redundant* if it contains a character $D[i]$ such that $\varepsilon_j[i] = \text{“-”}$ for each $\varepsilon_j \in \varepsilon$. Note that for any redundant deposition sequence D and any placement φ , it holds that $\text{BL}(\varphi, D) = \text{BL}(\varphi, D')$, where D' is obtained by deleting the redundant character $D[i]$. We say that a deposition sequence D is *good* if it is not redundant.

Observation 1. *Let (φ, D) be such that $\text{BL}(\varphi, D)$ is minimized for some (\mathcal{S}, r, m) . If D is redundant, then there exists a subsequence D' of D such that $\text{BL}(\varphi, D') = \text{BL}(\varphi, D)$ and D' is good.*

As a consequence, when searching for optimal solutions of these problems it suffices to consider only good deposition sequences. Aside from the trivial (quadratic) algorithm for computing the border length for a fixed deposition sequence and placement, we will utilize another algorithm which will in some cases yield better running times:

Proposition 1. *For any given $(\varphi, D, \mathcal{S}, r, m)$, there exists an algorithm which computes $\text{BL}(\varphi, D)$ in time $O(|\mathcal{S}| + p^2 \cdot |D|)$, where p is the number of distinct sequences in \mathcal{S} .*

Proof. The algorithm proceeds in four steps. First, in time $O(|\mathcal{S}|)$ it finds all unique sequences in \mathcal{S} and stores them in a set Q along with a mapping $\eta : \mathcal{S} \rightarrow Q$ which maps sequences from \mathcal{S} to their representative in Q . Second, in time $O(p^2 \cdot |D|)$ it computes and stores $\text{border}_D(q_1, q_2)$ for each $q_1, q_2 \in Q$. Third, in time $O(|\mathcal{S}|)$ for each sequence $s \in \mathcal{S}$ it computes the set $R_s = \varphi^{-1}(\mathcal{N}(\varphi(s)))$ of neighboring sequences. Finally, in time $O(|\mathcal{S}|)$ it computes $\frac{1}{2} \sum_{\forall s \in \mathcal{S}, r \in R_s} \text{border}_D(\eta(s), \eta(r))$ which is easily seen to be equal to $\text{BL}(\varphi, D)$. □

2.1 Parameterized Complexity

Parameterized algorithmics is a promising approach to obtain efficient algorithms for fragments of computationally hard problems. The aim is to find a parameter that describes the structure of the instance such that the combinatorial explosion can be confined to this parameter. In a parameterized complexity analysis the runtime of an algorithm is studied with respect to the input size n and a parameter $k \in \mathbb{N}$ (or a combination of parameters). For a more detailed introduction we refer to the literature [4, 9].

Formally, a *parameterized problem* is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is the input alphabet. If a combination of parameters k_1, \dots, k_l is considered, the second component of an instance (x, k) is given by $k = \sum_{1 \leq i \leq l} k_i$. The class **FPT** (*fixed-parameter tractable*) contains all problems that can be decided by an algorithm running in $f(k) \cdot n^{\mathcal{O}(1)}$ time, where f is a computable function and n is the input size. Such algorithms are often called fixed-parameter tractable (fpt).

Let L_1 and L_2 be parameterized problems, with $L_1 \subseteq \Sigma_1^* \times \mathbb{N}$ and $L_2 \subseteq \Sigma_2^* \times \mathbb{N}$. A *parameterized reduction* (or *fpt-reduction*) from L_1 to L_2 is a mapping $P : \Sigma_1^* \times \mathbb{N} \rightarrow \Sigma_2^* \times \mathbb{N}$ such that (1) $(x, k) \in L_1$ iff $P(x, k) \in L_2$; (2) the mapping can be computed by an fpt-algorithm with respect to parameter k ; (3) there is a computable function g such that $k' \leq g(k)$, where $(x', k') = P(x, k)$.

There is a variety of classes capturing *parameterized intractability*. For our results, we require only the class **paraNP** [8], which is defined as the class of problems that are solvable by a nondeterministic Turing-machine in fpt-time. We will make use of the characterization of **paraNP**-hardness given by Flum and Grohe [9], Theorem 2.14: any parameterized problem that remains NP-hard when the parameter is set to some constant is **paraNP**-hard. Showing **paraNP**-hardness for a problem rules out the existence of an fpt-algorithm under the usual complexity theoretic assumptions.

3 Hardness

In this section we overview and present new (parameterized) intractability results for BMP^e and P-BMP^e with respect to several combinations of parameters. As our starting point, we notice that the NP-hardness proof for P-BMP of Popa, Wong and Yung [17] can be straightforwardly adapted to $\text{P-BMP}_{c,r}^e$.

Proposition 2 (cf. [17, Theorem 1]). *$\text{P-BMP}_{c,r}^e$ is paraNP-hard.*

Proof. Observe that the reduction used in the proof of Theorem 1 in [17] constructs instances of BMP which only contain 3 characters. Furthermore, while the instances are formally defined as square arrays, all rows below the 5-th contain only a dummy character $\$$ and hence can be omitted without loss of generality. Finally, by Lemma 2 in [17] it follows that optimal exhaustive solutions for these BMP instances are also optimal solutions (in fact, it is these exhaustive solutions that are used to prove Theorem 1 in [17]). \square

The hardness result for BMP^e relies on a new polynomial-time reduction from P-BMP^e to BMP^e . We believe that this reduction is an interesting result on its own, as it is one of the first results that relates the complexity of these two problems in a general setting. We begin by showcasing a tool for forcibly “separating” any optimal deposition sequence.

Lemma 1. *Let $\mathcal{I} = (\mathcal{S}, r, m)$ be an instance of BMP^e such that each $s \in \mathcal{S}$ consists of a prefix $s_{pre} \in \Sigma_{pre}^*$, a fixed separator $sep \in (x^*y^*)^*$ and a suffix $s_{suf} \in \Sigma_{suf}^*$, where $\Sigma_{pre}, \Sigma_{suf}, \{x, y\}$ form a partition of Σ . Let $u \geq 8 \cdot \max_{s \in \mathcal{S}} (|s_{pre}|) + 8 \cdot \max_{s \in \mathcal{S}} (|s_{suf}|) + 1$. If $sep = (x^{r \cdot m \cdot u} \cdot y^{r \cdot m \cdot u})^{r \cdot m \cdot u}$ then every optimal good deposition sequence has the form $D_{pre} \cdot sep \cdot D_{suf}$ where $D_{pre} \in \Sigma_{pre}^*$ and $D_{suf} \in \Sigma_{suf}^*$.*

Proof. Notice that $r \cdot m \cdot u - 1$ forms a trivial upper-bound on the border length of \mathcal{I} , as witnessed by any deposition sequence of the form $D_{pre} \cdot sep \cdot D_{suf}$ (regardless of placement). Indeed, there are at most $4r \cdot m$ pairs of neighboring cells in the array, and for each such pair the border length is bounded by the hamming distance between the

embeddings placed on these cells, where any deposition sequence of this form yields a bound of $2 \cdot \max_{s \in \mathcal{S}}(|s_{pre}|) + 2 \cdot \max_{s \in \mathcal{S}}(|s_{suf}|)$.

Consider any optimal good deposition sequence D and let $p \in \Sigma_{pre}, q \in \Sigma_{suf}$. Consider for a contradiction that qp is a subsequence of D . Then $pre \cdot qp$ would also be a subsequence of D ; however, each mask for a character in pre would yield an increase of the border length by at least 1, since the array contains a cell in the array where this mask cannot be applied (specifically, this is the cell containing the sequence beginning with p). This would already break the upper-bound provided above. hence qp cannot be subsequence of D .

Next, consider for a contradiction that qy is a subsequence of D . Then $sep \cdot qy = (x^{r \cdot m \cdot u} y^{r \cdot m \cdot u})^{r \cdot m \cdot u} \cdot qy$ would also be a subsequence of D . This means that there exist two embeddings $\varepsilon_1, \varepsilon_2$ which differ in the positions of their first, second, third, $\dots, (r \cdot m \cdot u)^2$ -th y characters. Let $offset_x$ be the number of masks for x which occur between the position of the first y character in ε_1 and the first y character in ε_2 ; notice that $0 < offset_x \leq r \cdot m \cdot u$. Each mask for x in the offset has a border length of at least 1, since there is a sequence s_2 in the array which begins with y . If $offset_x < r \cdot m \cdot u$ then the upper-bound on the border length of D is broken by the fact that that $x^{r \cdot m \cdot u} y^{r \cdot m \cdot u}$ occurs $(r \cdot m \cdot u)$ -many times in succession in the deposition sequence, and each occurrence would necessarily increase the border length by at least 1. On the other hand, if $offset_x = r \cdot m \cdot u$ then the upper-bound on the border length would be broken already by all the masks for x which occur in the offset.

By a symmetric argument, we obtain that xp also cannot occur as a subsequence of D . Hence the deposition sequence must have the form $D_{pre} \cdot sep \cdot D_{suf}$. \square

Observe that “flipping” the array horizontally or vertically preserves the optimal border length but formally changes the placement φ . The purpose of the following key lemma is to provide a tool to fix the optimal positions of probes in the array; to this end, we will be considering placements which are unique up to these simple symmetries.

Lemma 2. *Let $a, b, x, y \in \Sigma$ and $r, m, t \in \mathbb{N}$. Consider an $r \times m$ array, and probes $\mathcal{S} = \{a^{i \cdot t} \cdot sep \cdot b^{j \cdot t} \mid i \in [r] \text{ and } j \in [m]\}$. Then:*

1. *the unique optimal placement φ_0 (up to simple symmetries) places each probe $a^{i \cdot t} \cdot sep \cdot b^{j \cdot t}$ in cell (i, j) ,*
2. *the unique optimal good deposition sequence is $D_0 = a^{r \cdot t} \cdot sep \cdot b^{m \cdot t}$, and*
3. *for any placement $\varphi \neq \varphi_0$ (except for symmetries of φ_0) and any deposition sequence D , it holds that $BL(\varphi, D) \geq BL(\varphi_0, D_0) + t$.*

Proof. We proceed in two steps. First, we compute the border length of (φ_0, D_0) . Then, we establish that φ_0 is the only optimal placement up to the above-mentioned simple symmetries, and that other placements yield a border length which is lower-bounded by $t + BL(\varphi_0, D_0)$. Notice that D_0 is the only optimal good deposition sequence regardless of placement by Lemma 1.

Claim. $BL(\varphi_0, D_0) = ((r - 1) \cdot m + r \cdot (m - 1)) \cdot t$.

$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	\cdots	$s_{1,m-2}$	$s_{1,m-1}$	$s_{1,m}$
$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	\cdots	$s_{2,m-2}$	$s_{2,m-1}$	$s_{2,m}$
\vdots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots
$s_{r-1,1}$	$s_{r-1,2}$	$s_{r-1,3}$	\cdots	$s_{r-1,m-2}$	$s_{r-1,m-1}$	$s_{r-1,m}$
$s_{r,1}$	$s_{r,2}$	$s_{r,3}$	\cdots	$s_{r,m-2}$	$s_{r,m-1}$	$s_{r,m}$

Figure 2: An $r \times m$ array. The corners and the perimeter are highlighted in gray.

Proof of Claim: For character a , we start with t -many masks that contain character a in each cell. Notice that these masks have border length zero. Then we continue with t -many masks that have character “-” in the first row and character a everywhere else. Each of these masks has border length m . Next we use t -many masks, where the first two rows contain character “-”, and so on. In total, we obtain a border length of $(r-1) \cdot m \cdot t$ for character a . For character x and y , all masks contain character x or y in each cell and hence all have a border length of zero. Finally, for character b the procedure is analogous – we simply swap columns and rows. This gives a border length of $r \cdot (m-1) \cdot t$ for character b . ■

Now consider any optimal solution (φ, D) . The fact that $D = D_0$ follows from Lemma 1. We now proceed to the core of our proof. Notice that for each pair of probes $s_1, s_2 \in \mathcal{S}$ it holds that $\text{border}_{D_0}(s_1, s_2) \geq t$. We say that s_1, s_2 are *similar* if $\text{border}_{D_0}(s_1, s_2) = t$. Since the number of pairs of cells which are neighbors in an $r \times m$ array is exactly $(r-1) \cdot m + r \cdot (m-1)$ and $\text{BL}(\varphi_0, D_0) = ((r-1) \cdot m + r \cdot (m-1)) \cdot t$, any optimal placement φ may only place probes which are similar into neighboring cells. Furthermore, if a placement φ is not optimal, then $\text{BL}(\varphi, D) \geq t + \text{BL}(\varphi_0, D_0)$ since for any s_1, s_2 which are not similar it holds that $\text{border}_{D_0}(s_1, s_2) \geq 2t$.

Let us denote the cells which have at most 3 neighbors in the array the *perimeter* and the cells which have at most 2 neighbors the *corners*. For the final part of the proof, we use the inductive assumption that φ_0 is the unique optimal placement for all $r' \times m'$ arrays such that $r' < r$ and $m' < m$ as long as the placement of at least two corners is fixed. Furthermore, we assume that $\min(r, m) > 1$; the lemma trivially holds for $\min(r, m) = 0$, and is easily seen that $\min(r, m) = 1$ the optimal placement must be an ascending sequence, which is unique if its corners/endpoints are fixed.

For each $s \in \mathcal{S}$, let $\text{sim}(s)$ denote the set of probes which are similar to s . Notice that there are precisely four probes such that $|\text{sim}(s)| = 2$ and precisely $2r + 2m - 4$ probes such that $|\text{sim}(s)| = 3$, and there is a unique (up to symmetry) placement of these probes in the corners and perimeter so that similar probes are placed on neighboring cells (see Fig. 2). Let \mathcal{S}_0 contain all the probes placed into the perimeter.

Notice that the placement of these probes on the perimeter precisely matches φ_0 , and the placement of probes such that $|\text{sim}(s)| \leq 2$ in $\mathcal{S}' = \mathcal{S} - \mathcal{S}_0$ is fixed by the placement of \mathcal{S}_0 in the perimeter.

If $\min(r, m) = 2$ then this concludes the proof. If $\min(r, m) = 3$ then the remaining placement reduces to the placement of $\mathcal{S}' = \mathcal{S} - \mathcal{S}_0$ into a one-dimensional array, which is unique when the corners are fixed. Finally, if $\min(r, m) = 4$ then the remaining placement

reduces to the placement of \mathcal{S}' into an $(r-2) \times (m-2)$ array, which is again unique by our inductive hypothesis. \square

With Proposition 2 and Lemma 2, we can proceed to:

Theorem 1. $\text{BMP}_{c,r}^e$ is paraNP-hard.

Proof. We provide a reduction from P-BMP $_{c,r}^e$, which is paraNP-hard by Proposition 2. Let Σ' be the language of P-BMP $_{c,r}^e$, $x_1, y_1, x_2, y_2 \notin \Sigma'$ and $\Sigma = \Sigma' + \{x_1, y_1, x_2, y_2\}$. From any instance $\mathcal{I}' = (\mathcal{S}', \varphi', r, m)$ of P-BMP $_{c,r}^e$, we construct an instance $\mathcal{I} = (\mathcal{S}, r, m)$ of BMP $_{c,r}^e$ as follows. For each $s \in \mathcal{S}'$ such that $\varphi'(s) = (i, j)$ we put $a^{i \cdot t} \cdot \text{sep}_1 \cdot b^{j \cdot t} \cdot \text{sep}_2 \cdot s$ into \mathcal{S} , where:

- $t > (\max_{s \in \mathcal{S}'} |s| \cdot r \cdot m)^2$.
- $\text{sep}_1 = (x_1^{r \cdot m \cdot u_1} y_1^{r \cdot m \cdot u_1})^{r \cdot m \cdot u_1}$
- $\text{sep}_2 = (x_2^{r \cdot m \cdot u_2} y_2^{r \cdot m \cdot u_2})^{r \cdot m \cdot u_2}$
- the constants u_1, u_2 for sep_1 and sep_2 respectively are sufficiently large so as to satisfy the condition of Lemma 1; for instance, $u_2 > 100t^3$ and $u_1 > 1000t^4$.

By Lemma 1 we have that any optimal good deposition sequence for \mathcal{I} must have the form $a^{r \cdot u} \cdot \text{sep}_1 \cdot b^{m \cdot u} \cdot \text{sep}_2 \cdot D'$. Let us now compare an arbitrary solution (φ, D) to (φ', D) . By Lemma 2, either φ is equivalent to φ' by symmetry, or the border length of masks for a, x_1, y_1, b in (φ, D) will be at least t greater than the border length of these masks in (φ', D) . However, t was chosen to be sufficiently large to exceed the worst-case border length of all masks for Σ' . So we conclude that any optimal solution for \mathcal{I} must use a placement which is either the same as or symmetric to φ' .

Finally, observe that after the last mask of sep_2 is applied, the remainder of \mathcal{I} is equivalent to \mathcal{I}' , and hence D' is also a solution to \mathcal{I}' . \square

Theorem 1 and Proposition 2 show that one cannot hope to find an fpt-algorithm for BMP e or P-BMP e parameterized by any subset of $\{c, r\}$. These results complete the hardness part of our complexity map for BMP e or P-BMP e . For BMP $_{c,\ell}^e$ it remains open whether the problem is fixed parameter tractable. Still, we can relate this problem to k -BALANCED PARTITION, a problem studied well in the literature [1, 5, 6].

In a k -BALANCED PARTITION instance we are given a graph $G = (V, E)$ with $|V| = n$. The question is to find a partition of the vertices V into k sets V_1, \dots, V_k such that $|V_i| \leq \lceil \frac{n}{k} \rceil$ for all $1 \leq i \leq k$, and the cut size (i.e., the number of edges $\{x, y\}$ such that $x \in V_i, y \in V_j$, and $i \neq j$) is minimized. We remark that, to the best of our knowledge, the parameterized complexity of k -BALANCED PARTITION parameterized by k is open on solid rectangular grids [5]. Below we show that k -BALANCED PARTITION on solid rectangular grids can be reduced to BMP e and hence BMP e is at least as hard as k -BALANCED PARTITION.

Proposition 3. *There is a polynomial time reduction from k -BALANCED PARTITION on solid rectangular grids to BMP e .*

Proof. Let $G = (V, E)$ be a solid rectangular grid of size $r \times m$ with $|V| = n$. Further, let $l, x \in \mathbb{N}_0$ such that $n = l \cdot k + x$ and $\Sigma = \{c_1, \dots, c_k\}$. We construct a probe set $\mathcal{S} = \{f(i) \text{ copies of sequence } c_i \mid i \in [l]\}$, where function $f(i) = l + 1$ if $1 \leq i \leq x$ and $f(i) = l$ otherwise. It is easy to verify that the placement φ of a BMP^e solution gives also a solution to k -BALANCED PARTITION if the characters in Σ are seen as partition sets V_1, \dots, V_k . \square

4 Fpt-Algorithms

In the following sections we discuss fpt-algorithms for several parameters. The first group focuses on sequences of moderate length and an array whose size is primarily growing in one dimension, i.e., on the parameters c , ℓ , and r . In contrast, the second group parameterizes by c and the maximum admissible border length o .

4.1 Fpt-Algorithm for $\text{P-BMP}_{c,\ell}^e$

Our first algorithm provides a basic introduction to the techniques used later on.

Observation 2. *For any instance (\mathcal{S}, r, m) of $\text{BMP}_{c,\ell}^e$, there are at most c^ℓ unique sequences in \mathcal{S} .*

Lemma 3. *For any instance (\mathcal{S}, r, m) of $\text{BMP}_{c,\ell}^e$ or any instance $(\mathcal{S}, \varphi, r, m)$ of $\text{P-BMP}_{c,\ell}^e$ it holds that $|D| \leq c^\ell \cdot \ell$ for any good deposition sequence D .*

Proof. Assume towards contradiction that there is a good deposition sequence D which contains $|D| > c^\ell \cdot \ell$ characters. Since the total number of distinct sequences $s_i \in \mathcal{S}$ is bounded by c^ℓ , the total number of distinct embeddings ε_i is also bounded by c^ℓ . Each embedding ε_i contains at most ℓ characters in $\Sigma \setminus \{-\}$. Hence by the pigeon-hole principle there must exist some $j \in [|D|]$ such that $\varepsilon_i[j] = \text{“-”}$ for all $i \in [|\mathcal{S}|]$, which implies that D is not good (contradiction). \square

At this point we can already prove:

Proposition 4. *$\text{P-BMP}_{c,\ell}^e$ is fixed parameter tractable, and there exists an algorithm for $\text{P-BMP}_{c,\ell}^e$ which runs in time $c^{c^{O(\ell)}} |\mathcal{S}|$.*

Proof. By Lemma 3, it suffices to search for deposition sequences of length at most $c^\ell \cdot \ell$. We loop through all of the at most $c^{c^\ell \cdot \ell}$ such deposition sequences, and for each sequence D we compute $\text{BL}(\varphi, D)$ in time $O(|\mathcal{S}| + p^2 \cdot |D|)$ by Proposition 1. By Observation 2 and Lemma 3, we obtain that $O(|\mathcal{S}| + p^2 \cdot |D|) = O(|\mathcal{S}| + c^{3\ell} \ell)$, which altogether yields the runtime bound of $c^{c^{O(\ell)}} |\mathcal{S}|$. \square

4.2 Fpt-Algorithm for $\text{BMP}_{c,\ell,r}^e$

We first introduce some notation for our arrays. Given an $r \times m$ array A , a *column* is an $r \times 1$ sub-array of A . A *column placement* into a column of A is a mapping $\varphi : [r] \rightarrow \mathcal{S}$ from the cells of A to the multiset of probes.

Observation 3. *For any instance (\mathcal{S}, r, m) of BMP^e , it holds that there are at most $c^{\ell \cdot r}$ distinct column placements.*

Hence for any fixed r and \mathcal{S} , we can enumerate all possible column placements as $\varphi_1, \varphi_2, \dots, \varphi_{c^{\ell \cdot r}}$. Observe that, for any two column placements $\varphi_t, \varphi_{t'}$, it holds that either (i) $t = t'$ and $\varphi_t(x) = \varphi_{t'}(x)$ for all $x \in [r]$, or (ii) $t \neq t'$ and $\varphi_t(x) \neq \varphi_{t'}(x)$ for at least one $x \in [r]$.

Any placement $\varphi : s \in \mathcal{S} \mapsto (a \in \mathbb{N}, b \in \mathbb{N})$ into A can be uniquely decomposed into a sequence of column placements $(\varphi_{i(1)}, \varphi_{i(2)}, \dots, \varphi_{i(m)})$ where $\varphi_{i(x)}(y) = \varphi(x, y)$ and $i : [m] \rightarrow [c^{\ell \cdot r}]$. The column placement $\varphi_{i(j)}$ with $j \in [m]$ denotes that the j -th column of A is of placement $i(j)$. Furthermore, since φ is closed under permutation of non-distinct sequences in \mathcal{S} , each column placement can be uniquely identified by an r -tuple of sequences from \mathcal{S} , formally $\varphi_{i(x)} = (s_1, s_2, \dots, s_r) \iff \varphi_{i(x)}(y) = s_y$ for all $y \in [r]$.

Next, we prove that when searching for optimal solutions for BMP^e it suffices to restrict ourselves to placements such that identical column placements appear in ‘‘consecutive blocks’’.

Lemma 4. *Let (\mathcal{S}, r, m) be an instance of BMP^e , D be a deposition sequence and φ be a placement which decomposes into $(\varphi_{i(1)}, \varphi_{i(2)}, \dots, \varphi_{i(m)})$. Then if there exist $a, b \in [m]$, $a + 1 < b$, such that $\varphi_{i(a)} = \varphi_{i(b)}$ but $\varphi_{i(a+1)} \neq \varphi_{i(b)}$, then $\text{BL}(\varphi, D) \geq \text{BL}(\varphi', D)$, where φ' decomposes into*

$$(\varphi_{i(1)}, \dots, \varphi_{i(a)}, \varphi_{i(b)}, \varphi_{i(a+1)}, \varphi_{i(a+2)}, \dots, \varphi_{i(b-1)}, \varphi_{i(b+1)}, \dots, \varphi_{i(m)}).$$

Proof. Recall that by Equation 1, $\text{BL}(\varphi, D)$ is equal to the sum of Hamming distances of embeddings $\text{border}_D(s_p, s_q)$ between neighboring $s_p, s_q \in \mathcal{S}$. Since the embeddings, and hence also the Hamming distances, are the same for $\text{BL}(\varphi, D)$ as for $\text{BL}(\varphi', D)$, the only difference between these values may arise from which sequences are neighbors.

We say that two neighboring cells $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$ are x -neighbors if $|x_1 - x_2| = 1$ and y -neighbors otherwise, i.e., if $|y_1 - y_2| = 1$; let $\mathcal{N}_x(v)$ and $\mathcal{N}_y(v)$ contain the x -neighbors and y -neighbors of v , respectively. Notice that y -neighborhoods are identical between φ and φ' , since the latter is obtained by permuting whole columns of the former. On the other hand, consider the difference between x -neighboring sequences in φ and φ' . Notice that φ' is obtained by a simple permutation of the column placements of φ and in particular these differ only in the borders between $\{\varphi_{i(a)}, \varphi_{i(a+1)}, \varphi_{i(b-1)}, \varphi_{i(b)}, \varphi_{i(b+1)}\}$. For convenience, we use bd to denote the total ‘‘horizontal’’ border between two column placements; formally:

$$\text{bd}(u, t) = \sum_{\forall x \in [r]} \text{border}_D(\varphi_{i(u)}(x), \varphi_{i(t)}(x)).$$

Now we can express the difference between the border lengths of both placements as $\text{BL}(\varphi', D) = \text{BL}(\varphi, D) + \text{bd}(a, b) + \text{bd}(b, a+1) + \text{bd}(b-1, b+1) - \text{bd}(a, a+1) - \text{bd}(b-1, b) - \text{bd}(b, b+1)$. Since $\varphi_{i(a)} = \varphi_{i(b)}$, it holds that $\text{bd}(a, b) = 0$ and $\text{bd}(b, a+1) = \text{bd}(a, a+1)$. Furthermore, since the triangle inequality holds for Hamming distances (and border_D is defined as a Hamming distance between two sequences), we obtain $\text{bd}(b-1, b+1) - \text{bd}(b-1, b) - \text{bd}(b, b+1) \leq 0$. Hence we conclude that $\text{BL}(\varphi', D) \leq \text{BL}(\varphi, D)$. \square

We say that a placement φ is *consecutive* if it decomposes into column placements $(\varphi_{i(1)}, \varphi_{i(2)}, \dots, \varphi_{i(m)})$ where for each $\varphi_{i(a)}, \varphi_{i(b)}$ such that $\varphi_{i(a)} = \varphi_{i(b)}$ and $a < b$ it holds that $\varphi_{i(a)} = \varphi_{i(c)}$ for all $a < c < b$.

Corollary 1. *For any BMP^e instance (\mathcal{S}, r, m) , there exists an optimal solution (φ, D) such that φ is consecutive.*

Proof. Let (φ', D) be a solution for (\mathcal{S}, r, m) . We can repeatedly apply Lemma 4 until we obtain a consecutive placement—notice that the number of times Lemma 4 can be applied is bounded by m . \square

The next algorithm uses an Integer Linear Programming (ILP) subroutine. ILP is a well-known framework for formulating problems and a powerful tool for the development of fpt-algorithms for optimization problems. In following we only give a brief overview of the framework before we present the algorithm.

Definition 7 (*p*-Variable Integer Linear Programming Optimization). Let $A \in \mathbb{Z}^{q \times p}$, $b \in \mathbb{Z}^{q \times 1}$ and $c \in \mathbb{Z}^{1 \times p}$. The task is to find a vector $x \in \mathbb{Z}^{p \times 1}$ which minimizes the objective function $c \times \bar{x}$ and satisfies all q inequalities given by A and b , specifically satisfies $A \cdot \bar{x} \geq b$. The number of variables p is the parameter.

Lenstra [14] showed that *p*-ILP, together with its optimization variant *p*-OPT-ILP (defined above), are in FPT. His running time was subsequently improved by Kannan [13] and Frank and Tardos [11] (see also [7]).

Theorem 2 ([7, 11, 13, 14]). *p*-OPT-ILP can be solved using $\mathcal{O}(p^{2.5p+o(p)} \cdot L)$ arithmetic operations in space polynomial in L , L being the number of bits in the input.

We are now ready to prove the main theorem of this subsection.

Theorem 3. $\text{BMP}_{c,\ell,r}^e$ is fixed parameter tractable, and there exists an algorithm for $\text{BMP}_{c,\ell,r}^e$ which runs in time $c^{\mathcal{O}(\ell \cdot r)} \cdot |\mathcal{S}|$.

Proof. We give a multi-step algorithm for $\text{BMP}_{c,\ell,r}^e$:

1. We branch on the choice of deposition sequence D . By Observation 1, it suffices to consider only good deposition sequences, and by Lemma 3 the number of good deposition sequences is bounded by $c^{\mathcal{O}(\ell)}$.

2. In view of Corollary 1, we branch on which column placements appear in φ and the order in which they appear. Formally, we construct the set of all distinct column placements $\mathcal{T} = \{\varphi_1, \dots\}$, branch on all nonempty subsets $\mathcal{T}' \subseteq \mathcal{T}$. We then branch on all mappings $f : [t] \rightarrow [|\mathcal{T}'|]$ where $t = |\mathcal{T}'|$. Since $|\mathcal{T}| \leq c^{\ell \cdot r}$ by Observation 3, there are at most $O(c^{O(\ell \cdot r)})$ choices of f .

For each fixed f , we hence obtain a *template* $Q_f = (\varphi_{f(1)}, \varphi_{f(2)}, \dots, \varphi_{f(t)})$. A consecutive placement φ *matches* a template Q_f if there exists a *multiplicity function* $h : t \rightarrow \mathbb{N}$ such that φ decomposes into $(h(1) \cdot \varphi_{f(1)}, h(2) \cdot \varphi_{f(2)}, \dots, h(t) \cdot \varphi_{f(t)})$ where $x \cdot \varphi_z$ is shorthand for x consecutive copies of φ_z .

3. We compute the following constants:

- For each column placement $\varphi_i = (s_1, s_2, \dots, s_r) \in \mathcal{T}'$ we compute the total cost of its “vertical borders” $\text{bd}_i^{\text{vert}}$ as follows:

$$\text{bd}_i^{\text{vert}} = \sum_{\forall z \in [r-1]} \text{border}_D(s_z, s_{z+1}).$$

- We also compute the total “horizontal cost”, which depends only on D and Q_f (since identical column placements do not have horizontal borders), as follows:

$$\text{cost}_h = \sum_{\forall z \in [r], w \in [t-1]} \text{border}_D(\varphi_{f(w)}^{-1}(z), \varphi_{f(w+1)}^{-1}(z)).$$

- For each distinct $s \in \mathcal{S}$ let $\#_s$ contain the number of occurrences of s in \mathcal{S} .
 - For each distinct $s \in \mathcal{S}$ and φ_i let $\#_s^i$ contain the number of occurrences of s in φ_i .
4. We construct and solve an p -OPT-ILP instance \mathcal{I} to compute the multiplicity function h which contains the “vertical cost” variable cost_v , the variables $h(1), \dots, h(t)$ and the following constraints:

- a) For each distinct $s \in \mathcal{S}$: $\#_s = \sum_{\forall z \in [t]} h(z) \cdot \#_s^z$.

- b) $\forall z \in [t] : h(z) > 0$.

- c) $\text{cost}_v = \sum_{\forall z \in [t]} h(z) \cdot \text{bd}_z^{\text{vert}}$.

- d) Minimize cost_v .

The intuition of the constraints is as follows. Constraints of type a) ensure that the choice of multiplicities does not introduce too many/too few occurrences of some probe s in the array. By the constraints of type b) it is ensured that the multiplicities are strictly positive. With help of constraint c) the vertical border cost for a certain choice of multiplicities is computed, which is in turn minimized by constraint d).

5. Finally, for each choice of D , \mathcal{T}' and f we store $\text{cost}_v + \text{cost}_h$ and the table of values $h = (h(1), \dots, h(t'))$ from the optimal solution of \mathcal{I} . After the branching is complete, we choose an arbitrary branch with minimum $\text{cost}_v + \text{cost}_h$ and read the values D, f, h associated with this branch. The algorithm then outputs (φ, D) where φ is computed from the template Q_f given by f and the multiplicity function given by h .

Running time. The number of branches processed after Step 1 and Step 2 is bounded by $c^{c^{O(\ell)}} 2^{c^{\ell \cdot r}} c^{c^{O(\ell \cdot r)}} = c^{c^{O(\ell \cdot r)}}$ and this branching can be initialized in $O(|\mathcal{S}|)$ time. Step 3 and the construction of \mathcal{I} can both also be completed in linear time, assuming multisets are implemented via a multiplicity function. \mathcal{I} contains $t \leq c^{\ell \cdot r}$ variables and has size linear in \mathcal{S} , and can thus be solved in time at most $c^{c^{O(\ell \cdot r)}} \cdot |\mathcal{S}|$ by Theorem 2. The time required to process Step 5 is easily seen to be dominated by Step 1 and 4.

Correctness. Assume for a contradiction that the algorithm outputs (φ, D) but there exists an optimal solution (φ', D') such that $\text{BL}(\varphi', D') < \text{BL}(\varphi, D)$. Consider the template Q'_f and multiplicity function h' associated with φ' . During the computation of our algorithm, the branch of Q'_f and D' had correctly computed the cost'_h component of $\text{BL}(\varphi', D')$. Furthermore, since (φ', D') is optimal, we obtain that h' must be an optimal solution for the p -OPT-ILP instance \mathcal{I}' constructed for this branch; let cost'_v be the output of \mathcal{I}' . Then $\text{BL}(\varphi', D') = \text{cost}'_v + \text{cost}'_h$ implies that $\text{cost}'_v + \text{cost}'_h < \text{cost}_v + \text{cost}_h$, which contradicts the assumed choice of branch D and Q_f in Step 5. \square

4.3 Fpt-Algorithm for P-BMP $_{c,o}^e$

Given an $r \times m$ array, a mask \mathcal{M} is called *trivial* if $\mathcal{M}(i, j) \neq \text{“-”}$ for all $i \in [r], j \in [m]$. Given a deposition sequence D , we say that a subsequence D' of D is *primal* if it is obtained from D by deleting all characters which are associated with a trivial mask. Notice that the border length of each mask associated with each character in a primal sequence is at least one, and the border length of all trivial masks is 0. For the purpose of providing concise running times, we use n to denote the size of the input.

Observation 4. For any instance of P-BMP e and BMP e , the number of primal sequences is bounded by $\sum_{i=1}^o c^i \leq o \cdot c^o$.

Additionally, since the number of “borders” between distinct probes is bounded from below by the number of distinct probes, we obtain:

Observation 5. Given a multiset \mathcal{S} of probes. For any YES-instance of P-BMP e and BMP e over \mathcal{S} , the number of distinct probes in \mathcal{S} is upper-bounded by $o + 1$.

Lemma 5. For any instance of P-BMP e and BMP e , any primal sequence D' corresponds to at most one good deposition sequence D . Furthermore, there exists an algorithm which runs in time $\mathcal{O}(o \cdot n)$ and which either computes this D from D' or correctly outputs that no such D exists.

Proof. We provide the polynomial time algorithm to compute D from D' ; uniqueness follows by the fact that the algorithm is deterministic.

ALGORITHM(D')

- 1 ($i := 1$)
- 2 Check whether a trivial mask for any character $x \in \Sigma$ can be applied.
- 3 If not, go to 5.
- 4 If yes, apply it, set $D := D + x$, and go to 2.
- 5 Apply the mask for $D'[i]$. Set $D := D + D'[i]$.
- 6 $i := i + 1$.
- 7 If ($i \leq |D'|$) then go to 2.
- 8 Check whether a trivial mask for any character $x \in \Sigma$ can be applied.
- 9 If not, go to 11.
- 10 If yes, apply it, set $D := D + x$, and go to 8.
- 11 If there remains a nonempty probe s , then **reject**.
- 12 **Output** D .

The algorithm runs in time $O(|D'| \cdot (c + |\mathcal{S}| \cdot \max_{s \in \mathcal{S}} |s|)) = O(o \cdot n)$. Correctness follows from the definition of primal sequences. \square

Theorem 4. P-BMP $_{c,o}^e$ is fixed-parameter tractable, and there exists an algorithm for P-BMP $_{c,o}^e$ which runs in time $\mathcal{O}(oc^o \cdot (n + o^2))$.

Proof. This algorithm builds upon Observation 4. We can branch on all primal sequences. For each candidate sequence D' we check whether the primal sequence corresponds to a deposition sequence D via Lemma 5. For each such D , we compute and store $\text{BL}(\varphi, D)$. Finally, a solution with a minimum $\text{BL}(\varphi, D)$ is selected. Observe that an applicable trivial mask can be found in linear time. Along with Observation 5, this yields a total runtime of $\mathcal{O}(oc^o \cdot (n + o^2))$ by Proposition 1 and Lemma 5. \square

4.4 Fpt-Algorithm for BMP $_{c,o}^e$

For a multiset \mathcal{S} and $s \in \mathcal{S}$, we denote by \mathcal{S}^{-s} the set of sequences in \mathcal{S} which are distinct from s . An instance (\mathcal{S}, r, m, o) of BMP $_{c,o}^e$ is then called *s-enveloped* if $|\mathcal{S}^{-s}| \leq o^2$.

Lemma 6. Any instance (\mathcal{S}, r, m, o) of BMP $_{c,o}^e$ such that $r > o$ and $m > o$ which is not *s-enveloped* for any $s \in \mathcal{S}$ is a no-instance.

Proof. Consider any placement φ . For $s \in \mathcal{S}$, we say that a column (or row) is *s-uniform* (w.r.t. φ) if all cells in the column (or row) are only assigned sequences which are not distinct from s . Furthermore, we say that a column (or row) is *uniform* if all cells in the column (or row) are not distinct from some sequence in \mathcal{S} .

Each non-uniform column and each non-uniform row contains at least one tuple of neighboring distinct sequences, which (regardless of D) contributes to an increase of $\text{BL}(\varphi, D)$ by at least 1. Hence any solution (φ, D) of (\mathcal{S}, r, m, o) must contain at most

o rows and at most o columns which are not uniform. Furthermore, if there exists an s -uniform column (or row) for some $s \in \mathcal{S}$, then all other uniform columns (rows) must also be s -uniform—otherwise φ would contain more than o non-uniform rows (columns), which we have already argued cannot happen.

To complete the proof, consider the possible cells where a sequence which is distinct from s may appear. Clearly such sequences may only appear in the at most o non-uniform columns and in the at most o non-uniform rows, and these intersect in at most o^2 cells. \square

We now consider two specific subcases of the problem before giving the theorem.

Lemma 7. *There is an algorithm which solves any instance (\mathcal{S}, r, m, o) of $\text{BMP}_{c,o}^e$ such that $m > 2o$ and $r > 2o$ in time $\mathcal{O}(o^3 \cdot c^o \cdot (n + o^2))$.*

Proof. By Lemma 6, there is either a sequence $s \in \mathcal{S}$ which represents the majority of sequences in \mathcal{S} , or (\mathcal{S}, r, m, o) is a no-instance; since only at most one quarter of sequences in \mathcal{S} are distinct from s , the sequence s is unique and can be computed in time $|\mathcal{S}|$.

Next, by Corollary 1 (and the symmetric statement for rows), we can assume without loss of generality that all s -uniform columns and all s -uniform rows are placed consecutively in φ . Notice that in this case only the first and last o columns and rows can be non- s -uniform. Since any sequence q distinct from s can only be placed in columns and rows that are not s -uniform, the number of possibilities for $\varphi(q)$ is bounded by $4o^2$.

We now summarize the algorithm. First, we find s in time $|\mathcal{S}|$. Second, for each of the at most o^2 sequences q distinct from s we branch on the at most $4o^2$ possible values of $\varphi(q)$, resulting in a placement φ . Third, for each such choice of φ we use the algorithm for $\text{P-BMP}_{c,o}^e$ from Theorem 4 to find an optimal deposition sequence D and store the obtained $\text{BL}(\varphi, D)$. Finally, we choose a tuple (φ, D) with a minimum $\text{BL}(\varphi, D)$. The bound on the running time follows from Theorem 4. \square

Lemma 8. *There is an algorithm which solves any instance (\mathcal{S}, r, m, o) of $\text{BMP}_{c,o}^e$ such that $m > 2o$ and $r \leq 2o$ in time $n \cdot c^{o^{O(o)}}$.*

Proof. By Observation 5, we obtain that the number of distinct column placements is bounded by $o^r \leq o^{2o}$.

Now we reuse the algorithm given in the proof of Theorem 3 with the only difference that in Step 1 we branch on primal sequences and compute the corresponding (good) deposition sequence in polynomial time. The number of primal sequences is bounded by $o \cdot c^o$ (Observation 4), the time required to compute the corresponding deposition sequence is bounded $O(o \cdot n)$ by Lemma 5. For each fixed deposition sequence, the running time of steps 2–4 of the algorithm in Theorem 3 is bounded by $c^{o^{O(o)}}$, and hence the runtime bound of $o^{2o} \cdot (o \cdot n + n \cdot c^{o^{O(o)}}) = n \cdot c^{o^{O(o)}}$. \square

Theorem 5. *$\text{BMP}_{c,o}^e$ is fixed parameter tractable, and there exists an algorithm for $\text{BMP}_{c,o}^e$ which runs in time $n \cdot c^{o^{O(o)}}$.*

Proof. In case $m > 2o$ and $r > 2o$ we use the algorithm described in the proof of Lemma 7. In case $m > 2o$ and $r \leq 2o$ (or, by symmetry, if $m \leq 2o$ and $r > 2o$) we use the algorithm described in the proof of Lemma 8. In case $m \leq 2o$ and $r \leq 2o$ we branch over all of the at most $(4o^2)!$ placements φ , resulting in at most $(4o^2)!$ instances of P-BMP $_{c,o}^e$ which can be solved individually in time $\mathcal{O}(oc^o \cdot (n + o^2))$ by Theorem 4. \square

5 Conclusion

In this work we considered the parameterized complexity of BMP e and P-BMP e , two fundamental problems related to the optimal design of microarrays, with respect to combinations of parameters centered around the number of distinct characters c . We presented fpt-algorithms for both BMP e and P-BMP e if the maximum probe length and the number of rows are viewed as additional parameters (c, ℓ, r) ; and if the border length is the additional parameter (c, o) . In addition, we showed that P-BMP e parameterized by c and ℓ is in FPT. For c, r (and also c alone) we showed paraNP-hardness for both BMP e and P-BMP e . Hence, under the usual complexity theoretic assumptions, one cannot hope to find an fpt-algorithm for these settings.

On our agenda for future work is to settle the question whether there is an fpt-algorithm for BMP e , parameterized by c, ℓ . Another direction for future research is to study further (structural) parameters for these two problems. Furthermore, in our complexity analysis we plan to consider more sophisticated target functions that take other criteria in addition to the border length into account.

References

- [1] K. Andreev and H. Räcke. Balanced graph partitioning. *Theory Comput. Syst.*, 39(6):929–939, 2006.
- [2] M. Chatterjee, S. Mohapatra, A. Ionan, G. Bawa, R. Ali-Fehmi, X. Wang, J. Nowak, B. Ye, F. A. Nahhas, K. Lu, S. S. Witkin, D. Fishman, A. Munkarah, R. Morris, N. K. Levin, N. N. Shirley, G. Tromp, J. Abrams, S. Draghici, and M. A. Tain-sky. Diagnostic markers of ovarian cancer by high-throughput antigen cloning and detection on arrays. *Cancer Research*, 66(2):1181–1190, 2006.
- [3] M. Cretich and M. Chiari. *Peptide Microarrays Methods and Protocols*, volume 570 of *Methods in Molecular Biology*. Human Press, 2009.
- [4] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.
- [5] A. E. Feldmann. *Balanced Partitions of Grids and Related Graphs*. PhD thesis, ETH Zürich, 2012.
- [6] A. E. Feldmann. Fast balanced partitioning is hard even on grids and trees. *Theoretical Computer Science*, 485:61–68, 2013.

- [7] M. R. Fellows, D. Lokshтанov, N. Misra, F. A. Rosamond, and S. Saurabh. Graph layout problems parameterized by vertex cover. In *ISAAC*, Lecture Notes in Computer Science, pages 294–305. Springer, 2008.
- [8] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.
- [9] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, Berlin, 2006.
- [10] S. Fodor, J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu, and D. Solas. Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251(4995):767–773, 1991.
- [11] A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
- [12] D. Gerhold, T. Rushmore, and C. T. Caskey. DNA chips: promising toys have become powerful tools. *Trends in Biochemical Sciences*, 24(5):168–173, 1999.
- [13] R. Kannan. Minkowski’s convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.
- [14] H. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8:538–548, 1983.
- [15] C. Li, P. Wong, Q. Xin, and F. Yung. Approximating border length for DNA microarray synthesis. In *Proc. 5th TAMC*, pages 410–422, 2008.
- [16] C. Melle, G. Ernst, B. Schimmel, A. Bleul, S. Koscielny, A. Wiesner, R. Bogumil, U. Möller, D. Osterloh, K.-J. Halbhuber, and F. von Eggeling. A technical triade for proteomic identification and characterization of cancer biomarkers. *Cancer Research*, 64(12):4099–4104, 2004.
- [17] A. Popa, P. W. H. Wong, and F. C. C. Yung. Hardness and approximation of the asynchronous border minimization problem - (extended abstract). In *TAMC*, Lecture Notes in Computer Science, pages 164–176. Springer, 2012.
- [18] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander. Class prediction and discovery using gene expression data. In *Proc. 4th RECOMB*, pages 263–272, 2000.
- [19] J. B. Welsh, L. M. Sapinoso, S. G. Kern, D. A. Brown, T. Liu, A. R. Bauskin, R. L. Ward, N. J. Hawkins, D. I. Quinn, P. J. Russell, R. L. Sutherland, S. N. Breit, C. A. Moskaluk, H. F. Frierson, Jr., and G. M. Hampton. Large-scale delineation of secreted protein biomarkers overexpressed in cancer tissue and serum. *PNAS*, 100(6):3410–3415, 2003.