Anonymity-Preserving Space Partitions

Úrsula Hébert-Johnson
Department of Computer Science, University of California, Santa Barbara, CA, USA

Chinmay Sonar
Department of Computer Science, University of California, Santa Barbara, CA, USA

Subhash Suri
Department of Computer Science, University of California, Santa Barbara, CA, USA

Vaishali Surianarayanan
Department of Computer Science, University of California, Santa Barbara, CA, USA

Abstract

We consider a multidimensional space partitioning problem, which we call Anonymity-Preserving Partition. Given a set P of n points in $\mathbb{R}^d$ and a collection H of m axis-parallel hyperplanes, the hyperplanes of H partition the space into an arrangement $\mathcal{A}(H)$ of rectangular cells. Given an integer parameter $t > 0$, we call a cell C in this arrangement *deficient* if $0 < |C \cap P| < t$; that is, the cell contains at least one but fewer than t data points of P. Our problem is to remove the minimum number of hyperplanes from H so that there are no deficient cells. We show that the problem is NP-complete for all dimensions $d \geq 2$. We present a polynomial-time d-approximation algorithm, for any fixed d, and we also show that the problem can be solved exactly in time $(2d - 0.924)^k m^{O(1)} + O(n)$, where k is the solution size. The one-dimensional case of the problem, where all hyperplanes are parallel, can be solved optimally in polynomial time, but we show that a related Interval Anonymity problem is NP-complete even in one dimension.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Anonymity, Hitting Set, LP, Constant Approximation, Fixed-Parameter Tractable, Space Partitions, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.32

Funding *Subhash Suri*: in part supported by NSF grant CCF-1814172.

Acknowledgements We thank Daniel Lokshtanov for discussions relating to our approximation algorithm for Anonymity-Preserving Partition, as well as an anonymous reviewer, whose comments helped to improve the aforementioned result.

1 Introduction

Consider the following geometric problem. We are given a set P of n points and a family H of m axis-parallel hyperplanes in $\mathbb{R}^d$. The hyperplanes of H partition the space into an arrangement $\mathcal{A}(H)$ of rectangular cells. Given an integer parameter $t > 0$, we call a cell C *deficient* if $0 < |C \cap P| < t$; that is, the cell contains at least one but fewer than t data points of P. We then ask: What is the minimum number of hyperplanes we must delete so that there are no deficient cells? See Figure 1 for an example. The problem turns out to be nontrivial even in two dimensions and, in fact, also in one dimension under a dual formulation.

While we are mainly interested in this as a natural geometric problem, it can also be relevant in the study of data anonymity. For instance, given a real-valued *scalar* data set, a common technique for *group anonymization* is to partition the domain into *buckets*, defined by a set of boundary values $\{x_1, x_2, \ldots, x_l\}$. Given an integer target $t > 0$, the buckets are chosen to ensure that any bucket $[x_i, x_{i+1}]$ is either empty or contains at least t different

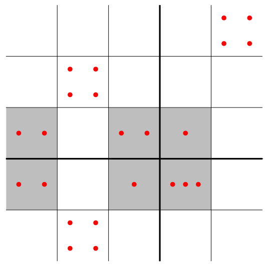


Figure 1 A 2-dimensional Anonymity-Preserving Partition instance with $t = 4$. The deficient cells are highlighted in gray and the two bold lines denote the optimal solution.

data records, thereby ensuring t-anonymity for each individual data value. Generalizing this to multidimensional data, the buckets are defined independently for each of the d axes, which geometrically creates a set of axis-parallel hyperplanes – the hyperplanes with normals parallel to the i-th coordinate axis correspond to the bucketing of the i-th dimension. Given a set of multidimensional data points and a set of candidate hyperplanes, the problem of discarding the fewest number of hyperplanes to achieve t-anonymity is precisely our space partitioning problem. For instance, one can imagine points being user locations in a two-dimensional coordinate system, and the problem is to specify those locations to within some "longitude" and "latitude" values so that every user's location is t-anonymized. Inspired by these connections, we have chosen to call our problem Anonymity-Preserving Partition for convenience, but our research focus in this work is purely algorithmic, and not related to anonymity.

Space partitioning problems are fundamental to many domains, including computational geometry, databases, robotics, etc. [12, 4, 6, 9, 5, 2]; however, to the best of our knowledge, this particular partition problem has not been studied. In computational geometry, for instance, space partitioning is frequently used for *range query* data structures such as kD-trees, range trees, etc. [7, 22, 1, 18, 20]. The primary focus in those algorithms is a hierarchical partitioning of the space to represent a set of points so that all points inside a query range can be reported efficiently. In contrast, our goal is to sparsify the (flat) partition induced by a given set of hyperplanes. A different type of multidimensional partitioning is investigated in [15, 21], where the goal is to partition a d-dimensional array, with nonnegative entries, into a fixed number of subarrays with roughly equal weights. Those approaches are motivated by an interest in constructing a compact *histogram* of the multidimensional data. In contrast, in our anonymizing partition, the goal is not to balance the weight but rather to avoid small-weight regions. In addition, while in the histogram problem the array is partitioned into arbitrarily arranged rectangular boxes, in our setting the partition is induced by full hyperplanes. In [17], LeFevre et al. also consider an anonymity-related partitioning problem, but they compute an arbitrary rectangular subdivision, not an arrangement of hyperplanes. They also show that their problem is NP-complete, but their proof requires the dimension of the space to be unbounded – in particular, $d \geq n$ in the constructed instances. In contrast, we show our problem is NP-complete even for dimension $d = 2$.

1.1 Our Contributions

We now discuss the main results of this paper. Given a set P of n points in $\mathbb{R}^d$, a set H of m axis-parallel hyperplanes, and an integer target $0 < t \leq n$, we define a *deletion set* to be a subset of hyperplanes so that no cell in the remaining arrangement is deficient. The goal of the Anonymity-Preserving Partition problem is to find a minimum deletion set.

For notational convenience, suppose $H_i \subseteq H$ is the subset of planes whose normals are parallel to the i-th coordinate axis, for $i = 1, 2, \ldots, d$. Then, if the number of nonempty families H_i is p, then our problem is essentially a p-dimensional problem, for $p \leq d$. If $p = 1$, then it is easy to solve the problem optimally using dynamic programming in time $O(nm)$. Surprisingly, we show that the problem is already NP-hard if $p = 2$, namely, the input is two-dimensional.

We then propose a polynomial-time p-approximation algorithm for the problem for any fixed $p \leq d$. For this, we reduce the problem to a variant of the well-known HITTING SET problem which we show to have an approximation algorithm using LP rounding. The approximate solution for the reduced HITTING SET instance will yield a p-approximate solution for our problem. We also give an FPT algorithm for the problem, with running time $(2d - 0.924)^k m^{O(1)} + O(n)$. From now on, for convenience of the reader, we assume that $p = d$ and state the results in terms of d.

Finally, we also introduce an interval anonymity problem in one dimension which can be viewed as a geometric dual of ANONYMITY-PRESERVING PARTITION when $d = 1$ – the roles of lines and points are interchanged. Specifically, we are given a set P of n points, which we call *markers*, a multiset S of m *segments* (intervals) on the real line $\mathbb{R}$, and an (integer) anonymity parameter $0 < t \leq n$. The set of markers P partitions S into equivalence classes, where two segments s, s' are in the same class if they contain the same set of marker points, namely, $s \cap P = s' \cap P$. We say a segment is *nonempty* if it contains at least one marker. We call an equivalence class consisting of nonempty segments *deficient* if it contains less than t segments. In the INTERVAL ANONYMITY problem, the aim is to remove a minimum number of points from P so that every nonempty segment of S belongs to a non-deficient equivalence class. For motivation, one can imagine segments as movement trajectories of m users, and markers as location sensors, and the goal is to report user locations in such a way that each user has t-anonymity. Somewhat surprisingly, this one-dimensional problem turns out to be NP-hard.

2 NP-Hardness of Anonymity-Preserving Partition

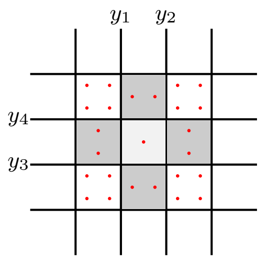
In this section, we prove that ANONYMITY-PRESERVING PARTITION is NP-hard even in two dimensions. This problem is easy to solve in one dimension, which we discuss in Section 3.

Let (P, H, t) be an instance of ANONYMITY-PRESERVING PARTITION in two dimensions. Without loss of generality, we assume that $H_1, H_2 \subseteq H$ are the sets of hyperplanes having normals parallel to the x- and y-axes, respectively. Furthermore, we denote the hyperplanes $h_1 \in H_1$ and $h_2 \in H_2$ by equations of the form $h_1 = x'$ and $h_2 = y'$, respectively, where $x', y' \in \mathbb{R}$ are constants. To show NP-hardness, we reduce from a structured variant of SAT called LINEAR NEAR EXACT SATISFIABILITY (LNES), which is known to be NP-complete [11]. The main idea here is to associate literals with hyperplanes and clauses with deficient cells, and to make satisfying assignments correspond to deletion sets.

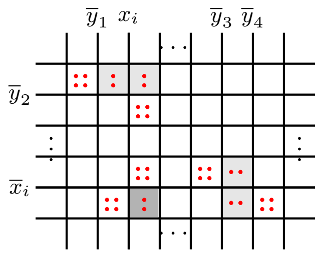
▶ **Theorem 1.** ANONYMITY-PRESERVING PARTITION *is NP-complete for all dimensions* $d \geq 2$.

Proof. Clearly, the decision version of our problem belongs to NP. We now show NP-hardness for just $d = 2$ as these instances can be easily embedded into any higher dimension. An instance J of LNES consists of $5s$ clauses, for $s \in \mathbb{N}$, and is denoted by

$$\mathcal{C} = \{U_1, V_1, U_1', V_1', \cdots, U_s, V_s, U_s', V_s'\} \cup \{C_1, \cdots, C_s\}.$$



(a) This figure shows nine nonempty cells corresponding to an auxiliary clause $C := (y_1 \vee y_2 \vee y_3 \vee y_4)$. The middle cell with one point is an *auxiliary cell*, and the four gray cells on its boundary are *shadow auxiliary cells*. The nonempty white cells denote the *helpers*.



(b) This figure shows core cells and variable cells. We consider the following four core clauses: $U_i := (\overline{y}_1 \vee x_i), V_i := (\overline{y}_2 \vee x_i), U_i' := (\overline{y}_3, \overline{x}_i), V_i' := (\overline{y}_4, \overline{x}_i)$. Moreover, we assume the literals y_1, y_3, y_4 are associated with the hyperplanes in H_2 forming the auxiliary cells, and y_2 is associated with the hyperplane in H_1. The *core cells* are colored light gray, and the *variable cell* is colored dark gray.

Figure 2 Example construction of auxiliary, core, and variable cells.

We refer to the first $4s$ clauses as the *core* clauses, and the remaining s clauses as the *auxiliary* clauses. The set of variables consists of s *main variables* $x_1, \ldots, x_s$ and $4s$ *shadow variables* $y_1, \ldots, y_{4s}$. Each core clause consists of two literals (one corresponding to a main variable, and the other to a shadow variable) and it has the following structure: $\forall\, i \in [s], U_i \cap V_i = \{x_i\}$ and $U_i' \cap V_i' = \{\overline{x}_i\}$.

Each main variable x_i occurs exactly twice as a positive literal and twice as a negative literal. The main variables only occur in the core clauses. Each shadow variable makes two appearances: as a positive literal in an auxiliary clause and as a negative literal in a core clause. Each auxiliary clause consists of four literals, each corresponding to a positive occurrence of a shadow variable.

The LNES problem asks whether, given a set of clauses with the aforementioned structure, there exists an assignment τ of truth values to the variables such that *exactly one* literal in every core clause and *exactly two* literals in every auxiliary clause evaluate to TRUE under τ.

Construction. We construct the set of hyperplanes $H = H_1 \cup H_2$ by adding hyperplanes placed at integer coordinates starting at one, i.e., $H = \{h_1 = x' \mid x' \in \{1, 2, \ldots, 3qs\}\} \cup \{h_2 = y' \mid y' \in \{1, 2, \ldots, 3qs\}\}$. These hyperplanes are numbered from left to right and top to bottom. For $i, j \in \mathbb{N}$, let $\square_{(i,j)}$ denote a 1×1 cell $[i, i+1] \times [j, j+1]$ on $\mathcal{A}(H)$. We set $q = 5s + 4$ (recall s is a parameter from the LNES instance) which is sufficiently larger than the desired size of the deletion set ($5s$). During the construction, we use q hyperplanes between a cluster of non-empty cells introduced so the sets remain independent, i.e., deleting lines from one cluster does not affect the other. We set the target t to 4. We associate a hyperplane from H with each of the $10s$ literals (H may contain additional hyperplanes which are not associated with any literal). Of these $10s$ hyperplanes, $8s$ are associated with the shadow literals and $2s$ with the main literals. By default, each cell in $\mathcal{A}(L)$ is empty. We introduce the nonempty cells and organize them into the following three groups (also, we describe the locations of the $4s$ hyperplanes associated with the positive shadow literals in the auxiliary cells group, and the locations of the remaining hyperplanes in the core cells group):

- **Auxiliary cells**: We introduce a set of nine nonempty cells for each auxiliary clause. For $i \in [s]$, we call $\square_{(qi,qi)}$ the *auxiliary cell* for clause C_i. The first two literals in C_i are associated with the two adjacent hyperplanes $x = qi$ and $x = qi + 1$ from H_1, and the remaining two literals are associated with the hyperplanes $y = qi$ and $y = qi + 1$ from H_2.[1] We add one point to $\square_{(qi,qi)}$ (note that $1 < t/2$). Moreover, we add $t/2$ points to each of $\square_{(qi-1,qi)}$, $\square_{(qi+1,qi)}$, $\square_{(qi,qi+1)}$, $\square_{(qi,qi-1)}$, and refer to them as *shadow cells*, while we add t points to each of $\square_{(qi-1,qi-1)}$, $\square_{(qi-1,qi+1)}$, $\square_{(qi+1,qi-1)}$, $\square_{(qi+1,qi+1)}$, and refer to them as *helpers* (see Fig. 2a). Observe that for each C_i, one needs to remove at least two of the four hyperplanes associated with the shadow literals appearing in C_i forming the corresponding auxiliary cell $\square_{(qi,qi)}$. This is to ensure that we have at least t points in all the remaining cells among the nine initial cells without exceeding the $5s$ deletion limit.

- **Core cells**: For each *core clause*, we introduce two nonempty cells. For each *main variable* x_i, we construct eight cells for the four core clauses U_i, V_i, U_i', V_i' together. Without loss of generality, let $U_i := (\overline{y}_1 \vee x_i)$, and $V_i := (\overline{y}_2 \vee x_i)$. Define $z_i = q(s+2i)$ for convenience.[2] We call $\square_{(z_i,z_i)}$ and $\square_{(z_i+1,z_i)}$ the *core cells* corresponding to the clauses U_i, V_i, respectively. We add two points to each of these cells and associate the common hyperplane $x = z_i + 1$ from H_1 to the literal x_i. Next, two cases arise according to the orientation of the hyperplanes associated with the literals y_1, y_2, say $p(y_1), p(y_2)$ (recall that orientation of these hyperplanes is decided while constructing the *auxiliary cells*):
  1. *$p(y_1) \in H_1$:* We associate the hyperplane $y = z_i$ from H_2 which forms the upper boundary of $\square_{(z_i,z_i)}$ with $\overline{y}_1$, and add four points to $\square_{(z_i,z_i-1)}$. Similarly, if $p(y_2) \in H_1$, we associate the hyperplane $y = z_i + 1$ from H_2 which forms the lower boundary of $\square_{(z_i+1,z_i)}$ with $\overline{y}_2$, and add four points to $\square_{(z_i+1,z_i+1)}$.
  2. *$p(y_1) \in H_2$:* We associate the hyperplane $x = z_i$ from H_1 which is the left boundary of $\square_{(z_i,z_i)}$ with $\overline{y}_1$, and add four points to $\square_{(z_i-1,z_i)}$. Similarly, if $p(y_2) \in H_2$, we associate the hyperplane $x = z_i + 1$ from H_1 which is the right boundary of $\square_{(z_i+1,z_i)}$ with $\overline{y}_2$, and add four points to $\square_{(z_i+2,z_i)}$.

  The construction above ensures that hyperplanes associated with y_i and $\overline{y}_i$ have *orthogonal* normals. We call the two nonempty cells introduced in either of the cases above as *shadow core cells.*

  We associate the literal $\overline{x}_i$ to the hyperplane $y = z_i + q + 1$ from H_2, and use a procedure symmetric to the one above to construct four nonempty cells. Here, $\square_{(z_i+1,z_i+q)}$ and $\square_{(z_i+1,z_i+q+1)}$ are *core cells* for the clauses U_i', V_i', respectively (note that, here, the two core cells are one below the other as opposed to side-by-side as we did for x_i). We complete the rest of the construction as described above. For an example, refer to Fig. 2b. Observe that removal of the hyperplane associated with the positive literal x_i makes both core cells (corresponding to U_i, V_i) non-deficient as these are merged together. Alternatively, removing the hyperplane corresponding to each $\overline{y}_1, \overline{y}_2$ makes the core cells non-deficient. The case of the literal $\overline{x}_i$ and the core clauses U_i', V_i' is symmetric.
- **Variable cells**: Recall that our construction of core cells ensures that for each main and shadow variable, the two hyperplanes associated with its two literals have orthogonal normals. Next, we introduce three nonempty cells for each of these variables. For each main variable x_i, the two hyperplanes associated with x_i and $\overline{x}_i$ form the top and left boundaries of the cell $\square_{(z_i+1,z_i+q+1)}$. We refer to $\square_{(z_i+1,z_i+q+1)}$ as a *variable cell*, and add two points to it. Furthermore, we add four points each to $\square_{(z_i,z_i+q+1)}$, $\square_{(z_i+1,z_i+q)}$, and call them *literal cells.* These cells are adjacent to the left and the upper boundaries of the variable cell. Refer to Fig. 2b.

[1] If for a *main variable* x_i, the two shadow variables appearing in the core clauses U_i, V_i are also the first two or the last two literals for some auxiliary clause, then we associate those literals with a pair of orthogonal hyperplanes $y = qi$ and $x = qi$ rather than with the default of a pair of parallel hyperplanes described earlier.

[2] Observe that we add an offset of qs so that the core and auxiliary cells are independent.

Next, we repeat the same procedure of introducing three nonempty cells for each shadow variable at the intersection of the hyperplanes associated with its literals. Notice that it is imperative to remove at least one of the two hyperplanes associated with the two literals for **every** variable so as to merge and make the variable cell non-deficient while staying within the deletion budget of $5s$ hyperplanes.

For the constructed Anonymity-Preserving Partition instance I, we ask if there exists a deletion set with size at most $5s$. We now turn to the argument of equivalence.

Forward direction. Recall that we start with an instance J of LNES. Let τ be a satisfying assignment for J; then we claim that the set S consisting of $5s$ hyperplanes associated with $5s$ literals set to TRUE under τ gives a valid deletion set for I. We now show that $\mathcal{A}(H \setminus S)$ does not contain any deficient cell. First, we observe that τ sets exactly one of the two literals associated with each of the $5s$ variables to TRUE (since τ is a valid assignment). Hence, the deficient variable cell introduced for each variable (see the dark gray cell from Fig. 2b) is merged with one of the literal cells and becomes non-deficient. Next, for each auxiliary clause C_i for $1 \leq i \leq s$, exactly two literals are set to TRUE. From the construction of the auxiliary cells group, one can verify that removing exactly two of the four hyperplanes associated with the four literals in C_i makes the auxiliary cell and the four shadow cells non-deficient (see Fig. 2a). Similarly, τ sets exactly one literal from each core clause to TRUE. Hence, we remove exactly one hyperplane on the boundary of each deficient core cell. Due to this, the core cell merges with either a shadow core cell or another core cell, making it non-deficient (see Fig. 2b). This accounts for all the deficient cells in I; hence, we conclude our argument for the forward direction.

Reverse direction. Let S be a valid deletion set of size at most $5s$; we construct an assignment τ for J by setting the literals associated with hyperplanes in S to TRUE. From the construction of the variable cells, we first observe that S contains exactly one of the two hyperplanes associated with the two literals for each of the $5s$ variables in J (since $|S| \leq 5s$). Hence, S is a valid SAT assignment, i.e., each variable is either set to TRUE or FALSE. Next, using a counting argument, we show that τ is a satisfying assignment for J. Recall that each main variable x_i occurs twice as a positive literal and twice as a negative literal in the core clauses. Hence, the s literals associated with the s main variables set to TRUE under τ satisfy exactly $2s$ core clauses. Next, for the remaining $2s$ core clauses, τ sets exactly one negative shadow literal appearing in each of those clauses to TRUE. This is because from the construction of a core cell corresponding to each core clause, at least one of the two hyperplanes associated with the literals in the clause must be in S (and literals corresponding to main variables cannot be set to TRUE for this set of core clauses). Similarly, τ sets at least two positive shadow literals appearing in each auxiliary clause to TRUE. At this stage, we use a counting argument: Among the $4s$ shadow literals set to TRUE under τ, exactly $2s$ negative shadow literals and exactly $2s$ positive shadow literals are TRUE (due to the argument above). Hence, with s main literals and $2s$ negative shadow literals set to TRUE, each core clause is satisfied exactly once. With $2s$ positive shadow literals set to TRUE, each auxiliary clause is satisfied exactly twice. This completes the proof for the reverse direction. ◀

3 Approximation and FPT Algorithms

In this section, we present a d-approximation algorithm for Anonymity-Preserving Partition. We first note that an $O(d)$-approximation can be easily achieved using a Hitting Set approximation, since we have a set system of VC dimension $O(d)$ [13, 8].

Unfortunately, the constant factors in these HITTING SET approximations tend to be large, and in fact a much simpler greedy algorithm can directly give us a $2d$-approximation as follows: while there exists a deficient cell C, we remove all of its (at most) $2d$ bounding hyperplanes, and iterate until no deficient cell remains. The approximation guarantee follows because for each deficient cell, the optimal solution must remove at least one hyperplane and the greedy algorithm removes $2d$ hyperplanes. Thus, the main challenge is to improve on this naive bound, which is the main result of this section.

Our algorithm first reduces the ANONYMITY-PRESERVING PARTITION problem to a special case of HITTING SET in which all sets have a small size, and then we design an LP-rounding-based algorithm to obtain a d-approximation for this problem. We also present a fixed-parameter tractable algorithm running in time $(2d-0.924)^k m^{O(1)}+O(n)$ parameterized by the solution size k.[3]

The one-dimensional case of ANONYMITY-PRESERVING PARTITION can be easily solved in linear time; please see Appendix A for a proof of the following result:

▶ **Theorem 2.** *The* ANONYMITY-PRESERVING PARTITION *problem in one dimension can be solved in time $O(mn)$, where m is the number of hyperplanes and n is the number of points. Further, if every cell in the arrangement is nonempty, then it can be solved in time $O(m+n)$.*[4]

3.1 A d-Approximation Algorithm

We start by defining a HITTING SET variant. Given a universe of elements U and a family $\mathcal{F}$ of subsets of U, the HITTING SET problem asks us to find a minimum-sized set $S \subseteq U$ such that S intersects with every set in $\mathcal{F}$. When every set in $\mathcal{F}$ has size at most l, we call it the l-HITTING SET problem.

▶ **Lemma 3.** *Given an instance (P, H, t) of the d-dimensional* ANONYMITY-PRESERVING PARTITION *problem, we can construct an instance $(U, \mathcal{F})$ of $2d$-*HITTING SET *such that $U = H$, $|\mathcal{F}| \leq |H|^{2d}$, and $(U, \mathcal{F})$ has a hitting set of size k if and only if (P, H, t) has a deletion set of size k, for any $k \in \mathbb{N}$.*

Proof. Given an instance (P, H, t) of ANONYMITY-PRESERVING PARTITION, we construct a $2d$-HITTING SET instance with universe $U = H$ and the family $\mathcal{F}$ being the set of all nonempty subsets X of H such that $\mathcal{A}(X)$ has a deficient cell and such that X contains at most two hyperplanes from each H_i with $1 \leq i \leq d$.

▷ Claim 4. If (P, H, t) has a deletion set of size k, then $(U, \mathcal{F})$ has a hitting set of size k.

Proof. Let $H' \subseteq H$ be a deletion set of size k for (P, H, t). Then, there is no deficient cell in $\mathcal{A}(H \setminus H')$. Since $U = H$, we now show that H' is also a hitting set of $(U, \mathcal{F})$. Suppose not; then there is a set X in $\mathcal{F}$ that has no hyperplanes from H' in it. We know by the construction of $\mathcal{F}$ that X has a cell that is deficient in $\mathcal{A}(X)$. Observe that even if we add any new hyperplanes to the arrangement $\mathcal{A}(X)$, there will still be a deficient cell. Thus, $\mathcal{A}(H \setminus H')$ will have a deficient cell, which contradicts our assumption that H' was a deletion set. ◁

[3] Fixed-parameter tractability (FPT) is studied in the realm of parameterized complexity. FPT algorithms admit running time of the form $f(k)n^{O(1)}$, where k is the parameter under consideration and n is the size of the instance [10].

[4] We assume the points and hyperplanes in the input are sorted.

$$\begin{aligned} &\text{minimize} \sum_{h \in H} x_h \\ &\text{s.t.} \quad \sum_{h \in F} x_h \geq 1 \quad \forall\, F \in \mathcal{F} \\ &\qquad x_h \in [0,1] \quad \forall\, h \in H \end{aligned}$$

Figure 3 LP for $2d$-Hitting Set.

▷ Claim 5. If $(U, \mathcal{F})$ has a hitting set of size k, then (P, H, t) has a deletion set of size k.

Proof. Let H' be a hitting set of $(U, \mathcal{F})$ of size k. Since $U = H$, we now show that H' is also a deletion set of (P, H). Suppose not; then there is a cell C that is deficient in $\mathcal{A}(H \setminus H')$. Let X be the set of hyperplanes adjacent to C in this arrangement. Since all the hyperplanes in H are axis parallel and we are in the d-dimensional version of the problem, it follows that X contains at most two hyperplanes from each H_i with $1 \leq i \leq p$. Also, observe that $\mathcal{A}(X)$ has the cell C in it. Since C is deficient, by construction of the family $\mathcal{F}$, we know X must be in $\mathcal{F}$. But since $H' \cap X = \emptyset$, this contradicts the fact that H' is a hitting set. ◁

This completes the proof of Lemma 3. Observe that the VC-dimension of the constructed set system is $2d$, hence, rounding algorithm from [13] would give an $O(d)$-approximation. ◀

We now observe the following simple fact:

▶ **Lemma 6.** *For each set $X \in \mathcal{F}$ of the $2d$-Hitting Set instance $(U, \mathcal{F})$ obtained by applying the reduction in Lemma 3 to (P, H, t), it holds that $|H_i \cap X| \leq 2$, for $1 \leq i \leq d$.*

Our approximation algorithm uses LP rounding; see Figure 3. While the integrality gap of this LP is known to be at most d, the proof is *non-constructive* [3, Theorem 1][5] and therefore it is not known how to efficiently compute a rounded solution with approximation factor less than $2d$. (The size of each set in the LP is $2d$ and so in any fractional LP solution each set is only guaranteed to have some variable with value at least $\frac{1}{2d}$. Thus a straightforward rounding of the LP solution only leads to a $2d$-approximation.) Our main contribution, therefore, is to design a polynomial-time rounding algorithm that achieves a d-approximation for $2d$-Hitting Set, and thus also for d-dimensional Anonymity-Preserving Partition.

▶ **Theorem 7.** *For every fixed dimension $d \geq 2$, there exists a polynomial-time algorithm that given a d-dimensional* Anonymity-Preserving Partition *instance, computes a deletion set with size at most d times the optimal size.*

Proof. We describe our rounding algorithm for $d = 2$ and defer the general case to Appendix B. We first use Lemma 3 to reduce the 2-dimensional Anonymity-Preserving Partition instance to a Hitting Set instance $(U = H_1 \cup H_2, \mathcal{F})$. Observe that by Lemma 6, for each set $X \in \mathcal{F}$, we have $|H_1 \cap X| \leq 2$ and $|H_2 \cap X| \leq 2$. We now give a 2-approximation algorithm for $(U, \mathcal{F})$ by extending the integrality gap result for the LP in [3] (see Figure 3).

[5] Note that in [3], Theorem 1 shows the integrality gap for a variant of hypergraph Vertex Cover. It is fairly straightforward to see that the Hitting Set instances obtained by applying the reduction in Lemma 3 can be equivalently expressed as instances of that same hypergraph Vertex Cover variant; hence, Lemma 3 also gives a reduction to hypergraph Vertex Cover.

For completeness, we first include the proof that the integrality gap is at most 2, and then describe our algorithm.

Let $g : U \to [0,1]$ be an optimal fractional hitting set of $(U, \mathcal{F})$ with value $\tau^*(U, \mathcal{F})$. Also, let $\tau(U, \mathcal{F})$ be the size of an optimal integral hitting set of $(U, \mathcal{F})$. Let $B = \{(x_1, x_2) \in [0, \frac{1}{2}] \times [0, \frac{1}{2}] : x_1 + x_2 = \frac{1}{2}\}$, and for each $x = (x_1, x_2) \in B$, let

$$T(x) = \{h \in H_1 : g(h) \geq x_1\} \cup \{h \in H_2 : g(h) \geq x_2\}.$$

In other words, B can be viewed as the set of all points on the line segment $x_1 + x_2 = \frac{1}{2}$ for $x_1, x_2 \in [0, \frac{1}{2}]$, and $T(x)$ can be viewed as the set obtained by rounding g using x_i as the threshold for each H_i.

We now prove that for any $x \in B$, $T(x)$ is a hitting set of $(U, \mathcal{F})$. Suppose not; then there must be a set $X \in \mathcal{F}$ such that $X \cap T(x) = \emptyset$. By the definition of $T(x)$, for each hyperplane $h \in X \cap H_i$, $i \in \{1, 2\}$, it holds that $g(h) < x_i$. Combining this with the fact that $|X \cap H_1| \leq 2$ and $|X \cap H_2| \leq 2$, we get $\sum_{h \in X} g(h) < 2(x_1 + x_2) = 1$. This contradicts the fact that g is a feasible fractional hitting set of $(U, \mathcal{F})$, and thus $T(x)$ is a hitting set.

Observe that for any given $a, b \in [0, 1/2]$ with $a \leq b$, for a uniformly random $x = (x_1, x_2) \in B$, we have $Pr(a \leq x_i \leq b) = \frac{b-a}{1/2}$ for $i \in \{1, 2\}$, i.e., x_1 and x_2 have a uniform distribution over the interval $[0, 1/2]$. We will now use a probabilistic argument to prove that the integrality gap is bounded by 2. If we choose a uniformly random $x = (x_1, x_2)$ from B, and let $E(\cdot)$ denote the expected value, then we have

$$\begin{aligned}\tau(U, \mathcal{F}) \leq E(|T(x)|) = \sum_{h \in H_i, i \in \{1,2\}} Pr(g(h) \geq x_i) &= \sum_{h \in U} \min\left(1, \frac{g(h)}{1/2}\right) \\ &\leq \sum_{h \in U} 2g(h) = 2\tau^*(U, \mathcal{F}).\end{aligned}$$

Let $T \coloneqq \{T(x) : x \in B\}$. By the above argument, there exists $x \in B$ such that $T(x)$ is a hitting set of size at most $2\tau^\star(U, \mathcal{F})$. Thus, to get a 2-approximation we will show that $|T| \leq 2m + 2$ and that T can be constructed in polynomial time (see Appendix B, Algorithm 1 for pseudocode). We now build a set $B' \subset B$ of size at most $2m + 2$ such that $T' \coloneqq \{T(x) : x \in B'\} = T$. We include one point for each hyperplane $h \in H_i$ with $g(h) \leq 1/2$, and we include an arbitrarily chosen point between each consecutive pair of these points on the line $x_1 + x_2 = 1/2$.

Formally, define B_1 and B_2 as follows: For each $h \in H_1$, add $(g(h), 1/2 - g(h))$ to B_1 if $g(h) \leq 1/2$, and for each $h \in H_2$, add $(1/2 - g(h), g(h))$ to B_1 if $g(h) \leq 1/2$. Finally, add the point $(1/2, 0)$ to B_1. Choose a value $\varepsilon > 0$ such that for any distinct $(x_1, x_2), (x_1', x_2') \in B_1$, we have $\varepsilon < |x_1' - x_1|$. For each $x = (x_1, x_2) \in B_1$ such that $x_1 \neq 1/2$, add $(x_1 + \varepsilon, 1/2 - x_1 - \varepsilon)$ to B_2. Finally, add $(0, 1/2)$ to B_2. Now let $B' = B_1 \cup B_2$.

We now prove that $T' = T$. We only need to argue that for all $x \in B \backslash B'$, $T(x) \in T'$. Given $x = (x_1, x_2) \in B \backslash B'$, let $x' = (x_1', x_2')$ be the pair in B_1 having the largest x_1' such that $x_1' < x_1$. If such an x' does not exist, then it is easy to see that $T(y = (0, 1/2)) = T(x)$. If x' exists, then $T(y = (x_1 + \varepsilon, 1/2 - x_1 - \varepsilon)) = T(x)$ since $x \notin B'$. In both cases y is in B' and thus $T(y) = T(x)$ is in T'. This proves that $T' = T$ and that $|T| \leq 2m + 2$. Our approximation algorithm constructs T and outputs the set in T having the smallest size. This completes the proof for $d = 2$. The complete algorithm as well as the details of the general case for dimensions $d > 2$ are presented in Appendix B. ◀

The approximation ratio in Theorem 7 is the best possible that can be obtained using the particular LP formulation from Fig. 3 because it has an integrality gap of d for the constructed hitting set instances [3].

3.2 Fixed-Parameter Tractable Algorithm

Given the equivalence of $2d$-Hitting Set and Anonymity-Preserving Partition (refer to Lemma 3), an FPT algorithm follows easily (when d is a constant). This is because the l-Hitting Set problem is known to admit an exact algorithm running in time[6] $(l - 0.924)^k |U|^{O(1)}$ [14], where k is the size of the hitting set.

▶ **Theorem 8.** *The* Anonymity-Preserving Partition *problem in d dimensions can be solved in time $(2d - 0.92)^k (m)^{O(1)} + O(n)$, where k is the size a minimum deletion set, m is the number of hyperplanes, and n is the number of points.*

4 An NP-hard Anonymity Problem on the Line

In this section, we show that the Interval Anonymity problem is NP-complete and give an exact algorithm running in time $3.08^k n^{O(1)} + O(m)$, where k is the solution size. Recall that here we are given a set P of n points, which we call *markers*, a multiset S of m segments (intervals) on the real line $\mathbb{R}$, and an integral anonymity parameter $t > 0$. For convenience, when we consider any set of points, we consider them to be ordered from left to right according to their relative positions on the line. The set of markers P partitions S into equivalence classes, where two segments s and s' are in the same class if they contain the same set of marker points, namely, $s \cap P = s' \cap P$. We call an equivalence class consisting of nonempty segments *deficient* if it contains less than t segments. The Interval Anonymity problem asks us to remove a minimum number of points from P so that every segment of S belongs to a non-deficient equivalence class. We now show that Interval Anonymity is NP-complete.

▶ **Theorem 9.** Interval Anonymity *is NP-complete, and is NP-hard to approximate within a factor of $(2 - \varepsilon)$, for any $\varepsilon > 0$, assuming the unique games conjecture (UGC).*

Proof. Clearly, the decision version of Interval Anonymity belongs to NP. We give a polynomial-time approximation-preserving reduction from Vertex Cover, which is NP-hard to approximate within a factor less than 2, assuming UGC [16].

Construction. Let G be a graph for which we seek a vertex cover of size at most k, and let $n = |V(G)|$. We can assume $k \leq n$. We construct an instance (P, S, t) of Interval Anonymity having $|P| = n + (n-1)k$ and $t = 2$, where we seek the same solution size k. Let $v_1, \ldots, v_n$ be the vertices of G. For each vertex v_i, we create $k+2$ markers labeled as $v_i, v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(k+1)}$, with one exception: the last vertex corresponds to just one marker, v_n. These markers occur in the following order:

$$v_1, v_1^{(1)}, \ldots, v_1^{(k+1)}, \quad \ldots, \quad v_{n-1}, v_{n-1}^{(1)}, \ldots, v_{n-1}^{(k+1)}, v_n.$$

For each $(v_i, v_j) \in E(G)$ with $i < j$, we add the following five (closed) intervals to S: $[v_i, v_j]$, two copies of $[v_i, v_{j-1}^{(k+1)}]$, and two copies of $[v_i^{(1)}, v_j]$. Since $t = 2$, we can see that the deficient intervals are exactly the ones of the form $[v_i, v_j]$.

[6] When $2d \geq 15$, there is an algorithm that runs in time $O(c^k + m)$, $c = d - 1 + \frac{1}{d-1}$ [19].

Proof of equivalence. For any vertex cover $\mathcal{S}$ of G, if we remove the markers (without superscripts) corresponding to the vertices in $\mathcal{S}$, we obtain a solution for the Interval Anonymity instance. For the reverse direction, suppose we have a deletion set $\overline{\mathcal{S}}$ for (P, S, t) of size at most k. Since our segments only have endpoints of the form v_i, $v_i^{(1)}$, or $v_i^{(k+1)}$, we would have had to include in $\overline{\mathcal{S}}$ all of the $(k+1)$-superscripted markers between two consecutive vertices if we wished for these to affect feasibility. Therefore, we can remove from $\overline{\mathcal{S}}$ any superscripted markers and still maintain a feasible solution. Now, $\overline{\mathcal{S}}$ naturally corresponds to a vertex cover for G. ◀

We now turn to a 4-approximation and an exact algorithm for the Interval Anonymity problem. Since this problem only cares about segments s such that $s \cap P \neq \emptyset$, we will from now on assume that for all segments $s \in S$, $s \cap P \neq \emptyset$. Given an instance (P, S, t) of the Interval Anonymity problem, we now associate a set of at most four markers from P to every equivalence class X. We denote this set by M_X. Let s be a segment in X, and let l and r be the leftmost and the rightmost markers in the set $s \cap P$. Also, let l' and r' be the markers in P to the left of l and to the right of r, respectively, if they exist. Then, $M_X = \{l', l, r, r'\}$ is the set containing these markers. Note that l might be equal to r and l' and r' might not exist, and thus M_X is a set of size at most four.

4-Approximation. The idea that each equivalence class can be associated with a set of at most four markers immediately gives us a polynomial-time 4-approximation algorithm and an exact algorithm running in time $4^k(m+n)^{O(1)}$, where k is the size of a minimum deletion set. The key here is to observe that *(i)* All segments in an equivalence class will remain in the same equivalence class in the final solution, and *(ii)* In order to make a deficient equivalence class X non-deficient, we need to remove at least one of the markers from M_X.

Then, the 4-approximation algorithm is as follows: (i) Initialize the deletion set $D = \emptyset$; (ii) Repeatedly pick an arbitrary deficient equivalence class X and add all the markers in M_X to D, as long as there is a deficient equivalence class; (iii) Finally, output D. For the exact algorithm, instead of adding all of the markers from M_X to the deletion set, we guess which one of these markers to add to the deletion set (branching).

We obtain a better exact algorithm for this problem, similarly to the Anonymity-Preserving Partition problem, by reducing to 4-Hitting Set.

▶ **Theorem 10.** *The* Interval Anonymity *problem can be solved in time* $3.08^k n^{O(1)} + O(m)$*, where* k *is the size a minimum deletion set.*

Proof. We first reduce our problem to 4-Hitting Set and then use the known $(3.08)^k|U|^{O(1)}$ time algorithm [14] for 4-Hitting Set to solve our problem. Our focus now is to describe the reduction. Given an instance (P, S, t) of the Interval Anonymity problem, we construct a 4-Hitting Set instance with universe $U = P$ and family $\mathcal{F}$ being the set of all nonempty subsets Q of P of size at most four such that the instance (Q, S, t) contains some deficient equivalence class.

Now we prove the forward direction: If (P, S, t) has a deletion set of size k, then $(U, \mathcal{F})$ has a hitting set of size k. Let $P' \subseteq P$ be a deletion set of size k of (P, S, t). Then, there is no equivalence class in $(P \backslash P', S, t)$ that is deficient. Since $U = P$, we now show that P' is also a hitting set of $(U, \mathcal{F})$. Suppose not; then there is a set $Q \in \mathcal{F}$ that contains no markers from P'. We know by construction of $\mathcal{F}$ that there is some deficient equivalence class X in (Q, S, t). Let s be a segment in X, and let X' be the equivalence class that s belonged to in $(P \backslash P', S, t)$. Since segments in X' always remain together in their resulting equivalence class even after removing additional markers, it is easy to see that if X' is not deficient in $(P \backslash P', S, t)$, then X is not deficient in (Q, S, t). This contradicts the fact that X is deficient and thus completes the forward direction.

Next, we show the reverse direction: If $(U, \mathcal{F})$ has a hitting set of size k, then (P, S, t) has a deletion set of size k. Let P' be a hitting set of $(U, \mathcal{F})$ of size k. Since $U = P$, we now show that P' is also a deletion set of (P, S, t). Suppose not; then there is a deficient equivalence class X in $(P \backslash P', S, t)$. We show that M_X from $(P \backslash P', S, t)$ belongs to $\mathcal{F}$, thus contradicting the fact that P' is a hitting set of $(U, \mathcal{F})$ since M_X does not have any marker from P'. To satisfy an equivalence class E, at least one of the markers in M_E must be deleted. Therefore, deleting all markers from $P \backslash P'$ except those from M_X will make X a deficient equivalence class in (M_X, S, t). Thus, by construction, M_X belongs to $\mathcal{F}$. ◀

5 Conclusion

We considered a natural multidimensional space partitioning problem, showed that it is NP-complete in all dimensions $d \geq 2$, and designed a d-approximation algorithm and an FPT algorithm parameterized by solution size. Although we described our results for the case $p = d$, it is easy to see that the algorithm in fact guarantees a p-approximation for the more general case, where $p \leq d$ is the number of nonempty families of hyperplanes. We also showed that a simple Interval Anonymity problem is NP-complete even in one dimension, and gave approximation and FPT algorithms for that as well. Improving our approximation factors is an interesting open problem.

References

1 P K Agarwal. Geometric Partitioning and Its Applications. *Discrete and Computational Geometry: Papers from the DIMACS Special Year (J. Goodman, R. Pollack, and W. Steiger, eds.)*, pages 1–37, 1991.

2 Pankaj K Agarwal and Micha Sharir. Applications of a new space-partitioning technique. *Discrete & Computational Geometry*, 9(1):11–38, 1993.

3 Ron Aharoni, Ron Holzman, and Michael Krivelevich. On a Theorem of Lovász on Covers in r-partite Hypergraphs. *Combinatorica*, 16(2):149–174, 1996.

4 Walid G Aref and Ihab F Ilyas. Sp-gist: An extensible database index for supporting space partitioning trees. *Journal of Intelligent Information Systems*, 17(2-3):215–240, 2001.

5 J. Baltes and J. Anderson. Flexible binary space partitioning for robotic rescue. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 4, pages 3144–3149 vol.3, 2003.

6 Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, pages 322–331, 1990.

7 Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

8 Hervé Brönnimann and Michael T Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

9 J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

10 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015.

11 Pratyush Dayal and Neeldhara Misra. Deleting to Structured Trees. In *International Computing and Combinatorics Conference*, pages 128–139. Springer, 2019.

12 Adrian Dumitrescu, Joseph SB Mitchell, and Micha Sharir. Binary space partitions for axis-parallel segments, rectangles, and hyperrectangles. *Discrete & Computational Geometry*, 31(2):207–227, 2004.

13 Guy Even, Dror Rawitz, and Shimon Moni Shahar. Hitting sets when the vc-dimension is small. *Information Processing Letters*, 95(2):358–362, 2005.

14 Fedor V Fomin, Serge Gaspers, Dieter Kratsch, Mathieu Liedloff, and Saket Saurabh. Iterative compression and exact algorithms. *Theoretical Computer Science*, 411(7-9):1045–1053, 2010.

15 Sanjeev Khanna, S. Muthukrishnan, and Mike Paterson. On Approximating Rectangle Tiling and Packing. In *Proceedings of the ninth annual ACM-SIAM Symposium On Discrete Algorithms*, volume 95, page 384. SIAM, 1998.

16 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

17 Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, pages 25–25. IEEE, 2006.

18 Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

19 Rolf Niedermeier and Peter Rossmanith. An efficient fixed-parameter algorithm for 3-hitting set. *Journal of Discrete Algorithms*, 1(1):89–102, 2003.

20 Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

21 Adam Smith and Subhash Suri. Rectangular tiling in multidimensional arrays. *Journal of Algorithms*, 37(2):451–467, 2000.

22 Marc Van Kreveld, Otfried Schwarzkopf, Mark de Berg, and Mark Overmars. *Computational geometry algorithms and applications*. Springer, 2000.

A Proof of Theorem 2

We show that the Anonymity-Preserving Partition problem is easy to solve in the one-dimensional case in time $O(mn)$. Furthermore, this special case can be solved in time $O(m+n)$ if every cell in the arrangement is nonempty. In both cases, we assume the points and hyperplanes in the input are pre-sorted.

Proof (of Theorem 2). We design a dynamic-programming algorithm to solve the problem in the one-dimensional case. Let i be the dimension in which we have a nonempty set of hyperplanes. We have $m = |H_i| = |H|$. We will denote the cells by $f_1, \dots, f_{m+1}$ and the hyperplanes by $h_1, \dots, h_m$, so that they occur in the following order in space:

$$f_1, h_1, f_2, h_2 \dots, h_m, f_{m+1}.$$

Let n_i be the number of points in the cell f_i. We will think of hyperplanes and cells with smaller indices in this ordering as being "to the left."

For each $1 \le i \le m+1$, let L_i be the set of hyperplanes to the left of the cell f_i. We have $L_1 = \emptyset$. For a set of hyperplanes H', let $f_i(H')$ denote the cell containing f_i in the arrangement $\mathcal{A}(H \backslash H')$. For example, if $H' = \{h_1\}$, then $f_2(H')$ is the cell formed by the union of f_1 and f_2. For every $1 \le i \le m+1$ and every $0 \le s \le t$, we define the following value:

$f(i,s) =$ minimum possible size of a set $H' \subseteq L_i$ such that in the arrangement $\mathcal{A}(H \backslash H')$, any nonempty cell to the left of $f_i(H')$ contains at least t points, and the cell $f_i(H')$ contains at least s points.

The value we need to compute is $f(m+1,t)$. We compute $f(m+1,t)$ using the following recursive formula:

$$f(i,s)=\begin{cases}0 & \text{if } i=1 \text{ and } s\le n_1\\ \infty & \text{if } i=1 \text{ and } s>n_1\\ \min\Big(f(i-1,0)+1, f(i-1,t)\Big) & \text{if } i>1 \text{ and } s\le n_i\\ f(i-1,s-n_i)+1 & \text{if } i>1 \text{ and } s>n_i.\end{cases}$$

The return value $f(m+1,t)$ is always finite since we assume $n\ge t$. This concludes the algorithm – we leave the formal proof of correctness to the reader. It is easy to see that the running time is $O(mt+n)$, which is bounded by $O(mn)$.

We now proceed to the case when the instance is not only one-dimensional, but also has the property that every cell in the arrangement is nonempty. In this case, the problem can be solved by a greedy algorithm, which proceeds as follows:

- Initially, set $q=1$ and set $S=\emptyset$.
- Repeat the following steps while $q\le m+1$:
  - Set j to be the smallest j such that $\sum_{i=q}^{j} n_i\ge t$. Set $S'=\{h_q,\ldots,h_{j-1}\}$. (If $j=q$, then S' is empty.) If there is no such j, this means we have reached the last of the cells. In that case, set j to be the largest j such that $\sum_{i=j}^{m+1} n_i\ge t$, set $S'=\{h_j,\ldots,h_m\}$, and break once this iteration is complete.
  - Set $S=S\cup S'$.
  - Set $q=j+1$.
- Return S.

Note that there always exists a j such that $\sum_{i=j}^{m+1} n_i\ge t$ since we assume $\sum_{i=1}^{m+1} n_i=n\ge t$. The formal proof of correctness is straightforward, and we leave it to the reader. ◀

B Proof of Theorem 7 for $d\ge 3$

In this section, we prove Theorem 3 for $d\ge 3$ and provide the pseudocode for the $d=2$ case. Recall that Theorem 7 promises a d-approximation algorithm for the d-dimensional Anonymity-Preserving Partition problem.

Proof (of Theorem 3 – for $d\ge 3$). Given an instance (P,H,t) of the d-dimensional Anonymity-Preserving Partition problem, we use the reduction in Lemma 3 to obtain a $2d$-Hitting Set instance $(U,\mathcal{F})$. Recall that $U=H=\bigcup_{1\le i\le d} H_i$, i.e., U is a union of d disjoint sets of hyperplanes H_i.

Next, we partition U into three sets S_1,S_2,S_3 such that for all $X\in\mathcal{F}$, $|X\cap S_i|\le d$ for $1\le i\le 3$. When d is even, we let

$$S_1=\bigcup_{1\le i\le \frac{d}{2}} H_i,\quad S_2=\bigcup_{\frac{d}{2}+1\le i\le d} H_i,\quad S_3=\emptyset.$$

When d is odd, we let

$$S_1=\bigcup_{1\le i\le \lfloor\frac{d}{2}\rfloor} H_i,\quad S_2=\bigcup_{\lfloor\frac{d}{2}\rfloor+1\le i\le d-1} H_i,\quad S_3=H_d.$$

We define $s_i=\max_{X\in\mathcal{F}}|X\cap S_i|$, for $i\in\{1,2,3\}$. From Lemma 2, we know that for all $X\in\mathcal{F}$, $|X\cap H_i|\le 2$; hence, $s_1+s_2+s_3\le 2d$. We now describe a d-approximation algorithm for $(U,\mathcal{F})$. To this end, we first use a result from [3] which bounds the integrality gap for the

LP from Fig. 3 on the instance $(U, \mathcal{F})$ by d. For completeness, we include the proof from [3], and then build upon it to give an approximation algorithm.

Let $g : U \to [0,1]$ be an optimal fractional hitting set of $(U, \mathcal{F})$ with value $\tau^*(U, \mathcal{F})$. Furthermore, let $\tau(U, \mathcal{F})$ be the size of an optimal integral hitting set. We now construct a set $B \subseteq [0, 1/d]^3$. Fix four points:

$$q_1 = \left(\frac{s_1 + s_2 - s_3}{2ds_1}, 0, \frac{1}{d}\right), \qquad q_2 = \left(\frac{1}{d}, \frac{s_2 + s_3 - s_1}{2ds_2}, 0\right)$$

$$q_3 = \left(\frac{s_1 + s_3 - s_2}{2ds_1}, \frac{1}{d}, 0\right), \qquad q_4 = \left(0, \frac{s_1 + s_2 - s_3}{2ds_2}, \frac{1}{d}\right)$$

and let

$$B^{(1)} = [q_1, q_2], \quad B^{(2)} = [q_3, q_4], \quad B^{(3)} = [q_1, q_3], \quad B^{(4)} = [q_2, q_4],$$

where $[q_i, q_j]$ denotes the line segment between the points q_i and q_j. We define $B = B^{(1)} \cup B^{(2)} \cup B^{(3)} \cup B^{(4)}$.

Notice that the coordinates of q_1, q_2, q_3, q_4 all satisfy the equation $s_1x_1 + s_2x_2 + s_3x_3 = 1$, and hence, this equation is satisfied by all tuples $x = (x_1, x_2, x_3) \in B$. Hence, using an argument similar to that used for $d = 2$, the sets $T(x)$ constructed as follows are indeed hitting sets:

$$T(x) = \{h \in S_1 : g(h) \geq x_1\} \cup \{h \in S_2 : g(h) \geq x_2\} \cup \{h \in S_3 : g(h) \geq x_3\}.$$

Let $T = \{T(x) : x \in B\}$. Next, we define a probability measure μ over B such that for any given $a, b \in [0, 1/d]$ with $a \leq b$, for a randomly chosen tuple $(x_1, x_2, x_3) \in B$, we have $Pr(a \leq x_i \leq b) = \frac{b-a}{1/d}$ for $1 \leq i \leq 3$, i.e., the x_i's have a uniform distribution over the interval $[0, 1/d]$. For $1 \leq i \leq 4$, let μ_i be the uniform measures on the line segments $B^{(i)}$ such that

$$\mu_1(B^{(1)}) = \mu_2(B^{(2)}) = \frac{(s_1 + s_3 - s_2)(s_2 + s_3 - s_1)}{2s_3(s_1 + s_2 - s_3)},$$

$$\mu_3(B^{(3)}) = \frac{(s_2 - s_3)(s_2 + s_3 - s_1)}{s_3(s_1 + s_2 - s_3)},$$

$$\mu_4(B^{(4)}) = \frac{(s_1 - s_3)(s_1 + s_3 - s_2)}{s_3(s_1 + s_2 - s_3)}.$$

We set $\mu = \mu_1 + \mu_2 + \mu_3 + \mu_4$. It can be verified that $\sum_{i=1}^{4} \mu_i(B^{(i)}) = 1$, and hence, $\mu(B) = 1$. At this stage, to argue as in the case $d = 2$ in order to show the bound on the integrality gap, it remains to show that x_i indeed has a uniform distribution on $[0, 1/d]$ for all $1 \leq i \leq 3$.

It is easy to see that for a randomly chosen $x = (x_1, x_2, x_3) \in B$, x_3 has a uniform distribution over $[0, 1/d]$. This is because each μ_i is a uniform measure over $B^{(i)}$, and x_3 takes all values from $[0, 1/d]$ on each $B^{(i)}$ with $1 \leq i \leq 4$. It is easy to see that x_1 is uniform over $B^{(4)}$ using the same argument. Next, we observe that x_1 is uniform on each of the line segments $\left[0, \frac{s_1+s_3-s_2}{2ds_1}\right]$, $\left[\frac{s_1+s_3-s_2}{2ds_1}, \frac{s_1+s_2-s_3}{2ds_1}\right]$, $\left[\frac{s_1+s_2-s_3}{2ds_1}, \frac{1}{d}\right]$. Recall that $\mu_1(B^{(1)}) = \mu_2(B^{(2)})$; hence, the situation for the first and the third line segment is the same. Without loss of generality, assume that $0 \le s_3 \le s_2 \le s_1 \le d$. Hence, we only need to check

$$\frac{\mu_3(B^{(3)})}{\mu_2(B^{(2)})} = \frac{\frac{s_1+s_2-s_3}{2ds_1} - \frac{s_1+s_3-s_2}{2ds_1}}{\frac{s_1+s_3-s_2}{2ds_1}},$$

which indeed holds. Hence, x_1 is uniformly distributed. With a similar argument, it can be shown that x_2 is uniformly distributed. At this stage, similarly to the $d = 2$ case, we can compute the expected size of $T(x)$ to obtain the desired bound d on the integrality gap.

Next, we show that there are only $O(m)$ distinct rounded hitting sets $T(x)$ constructed using $x \in B$. Observe that while traversing on any line segment $B^{(i)}$ for $1 \le i \le 4$, the hitting set $T(x)$ may change at points $x \in B^{(i)}$ for which there exists $1 \le j \le 3$ such that $g(h) = x_j$ for some $h \in S_j$, i.e., when the plane $x_j = g(h)$ intersects B. Note that the hitting set $T(x)$ does not change for the points on the open line segment between two consecutive intersection points on $B^{(i)}$ obtained from the aforementioned planes (here, the open line segment (x_i, x_j) is the set of all points on the line segment $[x_i, x_j]$ except for the endpoints). Since each such plane can have at most four intersection points with B, the number of distinct rounded solutions is $O(m)$, where $m = |U|$.

We iterate through all distinct rounded solutions and return a hitting set with minimum cardinality. This completes the proof of Theorem 7. ◀

Algorithm 1 2-approximation for Anonymity-Preserving Partition in 2 dimensions.

Input: Anonymity-Preserving Partition instance $(P, H = H_1 \cup H_2, t)$
Output: 2-approximate Deletion Set
$U \leftarrow H$
$\mathcal{F} \leftarrow \{X \subseteq U : \mathcal{A}(X) \text{ is deficient}, |X \cap H_i| \le 2, \forall i \in \{1, 2\}\}$
$g \leftarrow$ optimum fractional hitting set of $(U, \mathcal{F})$ ▷ $g : U \to [0, 1]$
$B_1 \leftarrow \{(g(v), 1/2 - g(v)) : v \in H_1, g(v) \le 1/2\} \bigcup$
$\{(1/2 - g(v), g(v)) : v \in H_2, g(v) \le 1/2\} \bigcup \{(1/2, 0)\}$
$\varepsilon \leftarrow$ arbitrary positive value less than $\min\limits_{(x_1,x_2),(x'_1,x'_2) \in B_1} |x_1 - x'_1|$
$B_2 \leftarrow \{(x_1 + \varepsilon, x_2 - \varepsilon) : (x_1, x_2) \in B_1, x_1 \ne 1/2\} \cup \{(0, 1/2)\}$
$B \leftarrow B_1 \cup B_2$
for $x = (x_1, x_2) \in B$ **do**
    $T_x \leftarrow \{v_1 \in H_1 : g(v_1) \ge x_1\} \cup \{v_2 \in H_2 : g(v_2) \ge x_2\}$
$x_{min} \leftarrow \arg\min\limits_{x \in B} |T_x|$
return $T_{x_{min}}$ ▷ $T_{x_{min}}$ is a 2-approximate deletion set