Pizza Sharing is PPA-hard

Argyrios Deligkas* John Fearnley† Themistoklis Melissourgos‡

Abstract

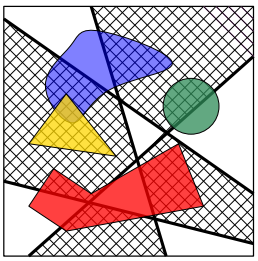
We study the computational complexity of finding a solution for the straight-cut and square-cut pizza sharing problems. We show that computing an ε-approximate solution is PPA-complete for both problems, while finding an exact solution for the square-cut problem is FIXP-hard. Our PPA-hardness results apply for any $\varepsilon < 1/5$, even when all mass distributions consist of non-overlapping axis-aligned rectangles or when they are point sets, and our FIXP-hardness result applies even when all mass distributions are unions of squares and right-angled triangles. We also prove that the decision variants of both approximate problems are NP-complete, while the decision variant for the exact version of square-cut pizza sharing is $\exists\mathbb{R}$-complete.

*Royal Holloway University of London, UK. email: argyrios.deligkas@rhul.ac.uk
†University of Liverpool, UK. email: john.fearnley@liverpool.ac.uk
‡University of Essex, UK. email: themistoklis.melissourgos@essex.ac.uk

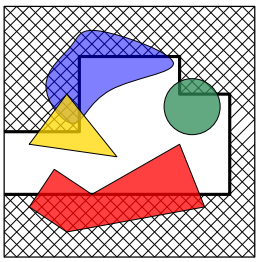
1 Introduction

Mass partition problems ask to fairly divide measurable objects that are embedded into Euclidean space [RS20]. Perhaps the most popular mass partition problem is the *ham sandwich* problem, in which three masses are given in three-dimensional Euclidean space, and the goal is to find a single plane that cuts all three masses in half. Recently, there has been interest in *pizza sharing* problems, which are mass partition problems in the two-dimensional plane, and in this work we study the computational complexity of such problems.

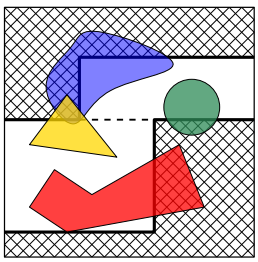
In the *straight-cut* pizza sharing problem, we are given $2n$ two-dimensional masses in the plane, and we are asked to find straight lines (see Figure 1a for a depiction) that simultaneously bisect all of the masses. It has been shown that this problem always has a solution for when we have n straight lines available: the first result on the topic showed that solutions always exist when $n = 2$ [BPS19], and this was subsequently extended to show existence for all n [HK20].



(a) A set of straight-cuts with four lines.



(b) A square-cut-path with six turns (not y-monotone).



(c) A y-monotone square-cut-path with four turns.

Figure 1: An example with 4 masses and various partitions of the plane into two regions, namely the shaded and non-shaded one. In a solution, each region contains half the area of each mass.

Another related problem is the *square-cut* pizza sharing. In this problem, there are n masses in the plane, and the task is to simultaneously bisect all masses using cuts, but the method of generating the cuts is different. Specifically, we seek a *square-cut*, which consists of a single path that is the union of horizontal and vertical line segments. See Figure 1b and Figure 1c for two examples of square-cuts. Intuitively, we can imagine that a pizza cutter is placed on the plane, and is then moved horizontally and vertically without being lifted in order to produce the cut. Note that the path is allowed to wrap around in the horizontal axis: if it exits the left or right boundary, then it re-appears on the opposite boundary. So the cut in Figure 1c is still considered to be a single square-cut.

It has been shown by [KRPS16] that, given n masses, there always exists a *square-cut-path* (termed SQUARE-*path*) which makes at most $n - 1$ turns and simultaneously bisects all of the masses. This holds even if the SQUARE-path is required to be *y-monotone*, meaning that someone moving on the path would either never head South or never head North (e.g., Figure 1c).

Two-dimensional fair division is usually called *land division* in the literature. Land division is a prominent topic of interest in the Economics and AI communities that studies ways of fairly allocating two-dimensional objects among n agents [Cha05, SHNHA17, SNHA20, ESS21, AD15, IH09, Hüs11]. Popular mathematical descriptions of fair division problems first appeared in [Ste48], and since then, the existence of allocations under various fairness criteria have been extensively studied, together with algorithms that achieve them. These problems find applications from division of resources on land itself, to the Law of the Sea [SS03], to redistricting [LRY09, LS14].

Consensus halving is a problem that asks us to split a one-dimensional resource into two parts such that n agents have equal value in both parts. Here, we study the same fairness criterion for

n agents, but for a two-dimensional resource. One can see that when we have the same fairness criterion at hand for any k-dimensional resource, $k \geq 2$, we can always translate the problem into its one-dimensional version, by integrating each agent's measure to a single dimension. Then a solution can be given by applying consensus halving. However, the solutions we get by doing so, are not taking into account the dimensionality of the problem, and as a result they might produce very unnatural solutions to a high-dimensional problem. For example, in land division, applying consensus halving would produce two parts, each of which can possibly be a union of $\lceil n/2 \rceil$ disjoint land strips. *Can we get better solutions by exploiting all the dimensions of the problem?*

In this work we investigate different cutting methods of the two-dimensional objects, and in particular, two pizza sharing methods for which a solution is guaranteed. While based on intuition one might assume that exploiting the two dimensions would allow the complexity of finding a solution to be lower, our results show that this is not the case. We present polynomial time reductions from the one-dimensional problem to the two-dimensional problems showing that the latter ones are at least as hard as the former, i.e., PPA-hard.

Computational complexity of fair division problems. There has been much interest recently in the computational complexity of fair division problems. In particular, the complexity class PPA has risen to prominence, because it appears to naturally capture the complexity of solving these problems. For example, it has recently been shown by [FRG18, FRG19] that the consensus halving problem, the ham sandwich problem, and the well-known necklace splitting problem are all PPA-complete.

More generally, PPA captures all problems whose solution is verifiable in polynomial time and is guaranteed by the Borsuk-Ulam theorem. Finding an approximate solution to a Borsuk-Ulam function, or finding an exact solution to a linear Borsuk-Ulam function are both known to be PPA-complete problems [Pap94, DFMS21]. The existence of solutions to the ham sandwich problem, the necklace splitting problem, and indeed the square-cut pizza sharing problem can all be proved via the Borsuk-Ulam theorem[1].

Theorem 1 (Borsuk-Ulam)**.** *Let $f : S^d \to \mathbb{R}^d$ be a continuous function, where S^d is a d-dimensional sphere. Then, there exists an $x \in S^d$ such that $f(x) = f(-x)$.*

The other class of relevance here is the class FIXP, defined by Etessami and Yannakakis [EY10]. This is the class of problems that can be reduced in polynomial time to the problem of finding an exact fixed point of a Brouwer function. It is known by the aforementioned work, that FIXP contains the problem Square Root Sum, which has as input positive integers $a_1, \ldots, a_n$ and k, and asks whether $\sum_{i=1}^{n} \sqrt{a_i} \leq k$. The question of whether Square Root Sum is in NP has been open for more than 40 years ([GGJ76, Pap77, Tiw92]). Furthermore, since there exist Brouwer functions that only have irrational fixed points, it is not expected that FIXP will be contained in FNP. In [DFMS21], it was shown that exact consensus halving is FIXP-hard.

Our contribution. We study the computational complexity of the straight-cut and square-cut pizza sharing problems, and we specifically study the cases where (i) all mass distributions are unions of weighted polygons (continuous version), and (ii) we are given unweighted point sets (discrete version). We show that it is PPA-complete to find approximate solutions for the continuous and discrete versions of the two problems, while their decision variants are NP-complete. Also, for the continuous version of the square-cut pizza sharing problem, we show that finding an exact solution is FIXP-hard, while its decision variant is $\exists\mathbb{R}$-complete.

[1] It has also been shown by [CS17] that the Borsuk-Ulam theorem is equivalent to the *ham sandwich theorem* which states that the volumes of any n compact sets in $\mathbb{R}^n$ can always be simultaneously bisected by an $(n-1)$-dimensional hyperplane.

To the best of our knowledge, currently, there are no problems in computational geometry with PPA-hardness results other than discrete ham sandwich [FRG19]. We also note that pizza sharing problems do not need a circuit as part of the input, which makes them in some sense more "natural" than problems that are specified by circuits. Other known PPA-hard problems of this kind are one-dimensional, such as consensus halving [FHSZ20] and necklace splitting [FRG19], or problems with unbounded dimensions, such as discrete ham sandwich. Here we show the first known PPA-hardness result for a "natural" two-dimensional problem. It is worth mentioning that shortly after the appearance of our result, Schnider in [Sch21] proved the following: (a) the discrete version of straight-cut pizza sharing where each mass is represented by unweighted points is PPA-complete, and (b) for a more general input representation than ours, to find an exact solution in its continuous version is FIXP-hard, and the decision variant is $\exists\mathbb{R}$-complete.

For both the straight-cut and the square-cut pizza sharing problems, namely ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing, we show that it is PPA-complete to find an ε-approximate solution for any constant $\varepsilon \in (0, 1/5)$. This holds even when $n + n^{1-\delta}$ lines are permitted in a straight-cut pizza sharing instance with $2n$ mass distributions, and when $n - 1 + n^{1-\delta}$ turns of the square-cut path are permitted in a square-cut pizza sharing instance with n mass distributions, for any constant $\delta \in (0, 1]$. Furthermore, the PPA-hardness holds even when each mass distribution is uniform over polynomially many axis-aligned rectangles, and there is no overlap between any two mass distributions. The inapproximability for such high values of ε is possible due to a recent advancement in the inapproximability of consensus halving [DFHM25].

The PPA membership of straight-cut and square-cut pizza sharing holds even for inverse polynomial and inverse exponential ε, respectively, and for weighted polygons with holes (arguably, a very general type of allowed input). To prove the PPA membership of the square-cut pizza sharing problem, we first turn the original topological proof by [KRPS16] into an algorithmic one. Furthermore, we show that there is a constant $\varepsilon > 0$ such that it is NP-complete to decide whether an ε-approximate solution of straight-cut pizza sharing with at most $n - 1$ lines (resp. an ε-approximate solution of square-cut pizza sharing with at most $n - 2$ turns) exists. All of these results hold also for the discrete version of the problems.

We then turn our attention to the computational complexity of finding an *exact* solution to the square-cut problem. We show that the problem of finding a Square-path with at most $n - 1$ turns that exactly bisects n masses is FIXP-hard. This hardness result applies even if all mass distributions are unions of weighted axis-aligned squares and right-angled triangles. In order to prove this, we reduce from the problem of finding an exact Consensus-Halving solution [DFMS21]. Regarding the decision variant of the square-cut problem, we show that deciding whether there exists an exact solution with at most $n - 2$ turns is $\exists\mathbb{R}$-complete, where $\exists\mathbb{R}$ consists of every decision problem that can be formulated in the existential theory of the reals (see Section 2 for its definition). All of our hardness results are summarized in Table 1 and Table 2.

From a technical viewpoint, our PPA membership result for straight-cut pizza sharing is based on a reduction that transforms mass distributions to point sets in general position and then employs a recent result by [Sch21]. For the membership results of square-cut pizza sharing, our proof strategy is different, since we are able to directly reduce it to the ε-Borsuk-Ulam problem (see Definition 26). Our hardness results are obtained by reducing from the consensus halving problem, historically the first fair-division problem shown to be PPA-complete [FRG18].

Further related work. Since mass partitions lie in the intersection of topology, discrete geometry, and computer science there are several surveys on the topic; [BFHZ18, DLGMM19, Mat08, Živ17] focus on the topological point of view, while [AE+99, Ede12, KK03, Mat02]

Hardness	ε	Lines	Pieces	Overlap	Theorem
		Point sets			
PPA	$1/5$	$n + n^{1-\delta}$	-	-	14
NP	c	$n - 1$	-	-	16
		Mass distributions			
PPA	$1/5$	$n + n^{1-\delta}$	poly(n)	1	6
NP	c	$n - 1$	poly(n)	1	7

Table 1: A summary of our hardness results for ε-Straight-Pizza-Sharing. Here, c and δ are absolute, positive constants. "Lines" refers to the number of cut-lines allowed in a solution. "Pieces" refers to the maximum number of distinct polygons that define every mass distribution. "Overlap" refers to the maximum number of different mass distributions allowed to contain any point of $[0, 1]^2$.

Hardness	ε	Turns	Pieces	Overlap	Theorem
		Point sets			
PPA	$1/5$	$n - 1 + n^{1-\delta}$	-	-	15
NP	c	$n - 2$	-	-	17
		Mass distributions			
PPA	$1/5$	$n - 1 + n^{1-\delta}$	poly(n)	1	9
NP	c	$n - 2$	poly(n)	1	10
FIXP	0	$n - 1$	6	3	19
$\exists\mathbb{R}$	0	$n - 2$	6	3	21

Table 2: A summary of our hardness results for ε-Square-Pizza-Sharing. Here, "turns" refers to the number of turns a solution (Square-path) is allowed to have. The definitions of c, δ and the semantics of "pieces", and "overlap" are the same as those of Table 1.

focus on computational aspects. Consensus halving [SS03] is the mass partition problem that received the majority of attention in Economics and Computation so far [DFH21, FRFGZ18, FRG19, FHSZ20, FRHSZ21]. Recently, Haviv [Hav22] showed PPA-completeness of finding fair independent sets on cycle graphs, having as a starting point the latter problem.

2 Preliminaries

Mass distributions and point sets. A *mass distribution* μ on $[0,1]^2$ is a measure on the plane such that all open subsets of $[0,1]^2$ are measurable, $0 < \mu\left([0,1]^2\right) < \infty$, and $\mu(S) = 0$ for every subset of $[0,1]^2$ with dimension lower than 2. For any given $d \in \mathbb{N}^*$ we denote $[d] := \{1, 2, \ldots, d\}$, and we denote by $\bigsqcup$ the union of disjoint sets. For every $S \in [0,1]^2$ we denote by area(S) the Lebesgue measure of S on $\mathbb{R}^2$, i.e., the area of S. Let a mass distribution μ be described by a finite set of non-overlapping regions $a_1, a_2, \ldots, a_d$, i.e., $\bigsqcup_{j=1}^{d} a_j = [0,1]^2$, such that $\sum_{j=1}^{d} \mu(a_j) = \mu\left([0,1]^2\right)$. Then, μ is *piece-wise uniform* if for every j and every $S \subseteq a_j$ it holds that $\mu(S) = w_j \cdot \text{area}(S)$ for some *weight* $w_j > 0$ independent of S. When additionally $w_j = w_k$ for all $j, k \in [d]$ then the mass distribution is called *uniform*. The *support* of mass distribution $i \in [n]$, denoted by $supp(i)$, is the area $A_i \subseteq [0,1]^2$ which has the property that for every $S \subseteq A_i$ with area$(S) > 0$ we have $\mu_i(S) > 0$. Let $N := \{I \subseteq [n] : \bigcap_{i \in I} supp(i) \neq \emptyset\}$. A set of mass distributions $\mu_1, \ldots, \mu_n$, or *colours*, has *overlap* k if $\max_{I \in N} |I| = k$. Finally, a mass distribution is *normalised* if $\mu([0,1]^2) = 1$. For ease of presentation, all our additive approximation results on the continuous versions of the problems assume that all mass distributions are normalised, which is without loss of generality.

A point set $P = (p_1, p_2, \ldots, p_d)$ on $[0,1]^2$ consists of $d \in \mathbb{N}^*$ many non-overlapping point masses. Throughout this work, the points that will be considered in the discrete versions of our problems have the same finite weight, so when we partition them (by partitioning $[0,1]^2$), it suffices to measure the cardinality of the points in each part.

Set of straight-cuts. A set of *straight-cuts*, or *cut-lines*, or simply *lines* defines subdivisions of the plane R. Figure 1a shows an example of a set of straight-cuts. Each line creates two half-spaces, and arbitrarily assigns number "0" to one and "1" to the other. A subdivision of R is labeled "+" (and belongs to R^+) if its parity is odd (according to the labels given to the half-spaces) and "−" (and belongs to R^-) otherwise. Observe that by flipping the numbers of two half-spaces defined by a line, we flip all the subdivisions' labels. Thus, there are only two possible labelings of the subdivisions.

Square-cut-path. A *square-cut-path*, denoted for brevity Square-path, is a non-crossing directed path that is formed only by horizontal and vertical line segments and in addition it is allowed to "wrap around" in the horizontal dimension. Figure 1b and Figure 1c show two examples of Square-paths. A *turn* of the path is where a horizontal segment meets with a vertical segment. A Square-path is *y-monotone* if all of its horizontal segments are monotone with respect to the y axis. Any Square-path partitions the plane R into two regions, namely, R^+ and R^-, so that the following holds: for any two points of the plane, if the straight line that connects them intersects once the path, then the two points have opposite labels.[2]

Pizza sharing. A set of lines (resp. a Square-path) *ε-bisects* a mass distribution μ, if $|\mu(R^+) - \mu(R^-)| \leq \varepsilon$. It *simultaneously ε-bisects* a set of mass distributions $\mu_1, \ldots, \mu_n$ if $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$ for every $i \in [n]$.

[2]Notice that a path can pass multiple times from the same point.

Definition 2. *For any $n \geq 1$, the problem ε-*Straight-Pizza-Sharing *is defined as follows:*

- ***Input:*** *$\varepsilon \geq 0$, and mass distributions $\mu_1, \mu_2, \ldots, \mu_{2n}$ on $[0,1]^2$.*
- ***Output:*** *A partition of $[0,1]^2$ to R^+ and R^- using at most n lines such that for each $i \in [2n]$ it holds that $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$.*

Definition 3. *For any $n \geq 1$, the problem ε-*Square-Pizza-Sharing *is defined as follows:*

- ***Input:*** *$\varepsilon \geq 0$, and mass distributions $\mu_1, \mu_2, \ldots, \mu_n$ on $[0,1]^2$.*
- ***Output:*** *A partition of $[0,1]^2$ to R^+ and R^- using a* Square*-path with at most $n-1$ turns such that for each $i \in [n]$ it holds that $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$.*

In [HK20] and [KRPS16] it was proved that ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing, respectively, always admit a solution for arbitrary absolutely continuous masses with respect to the Lebesgue measure (i.e., area), and for any $\varepsilon \geq 0$ (see Theorem 1 of the former, and Theorem 2.4 of the latter work). While the aforementioned results hold for such general measures, for the computational problems ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing we need a standardized way to describe the input, and therefore restrict to particular classes of measures. We consider the class of mass distributions that are defined by weighted simple polygons with holes. This class consists of mass distributions that can be succinctly represented in the input of a Turing machine, while at the same time provide great expressive power.

In particular, we will use the standard representation of 2-dimensional simple polygons in computational geometry problems, that is, a directed chain of points[3]. Consider a polygon that is defined by k points $p_i = (x_i, y_i)$, where $x_i, y_i \in [0,1] \cap \mathbb{Q}$, for $i \in [k]$, which form a directed chain $C = (p_1, \ldots, p_k)$. This chain represents a closed boundary defined by the line segments (p_i, p_{i+1}) for $i \in [k-1]$ and a final one (p_k, p_1). Since we consider polygons with holes, we need a way to distinguish between the polygons that define a boundary whose interior has strictly positive weight and polygons that define the boundary of the holes (whose interior has zero weight). We will call the former *solid* and the latter *hollow* polygon. To distinguish between the two, we define a solid polygon to be represented by directed line segments with counterclockwise orientation, while a hollow polygon to be represented similarly but with clockwise orientation. Furthermore, each solid polygon C_s, its weight w and its $r \geq 0$ holes $C_{h_1}, C_{h_2}, \ldots, C_{h_r}$ in the interior, are grouped together in the input to indicate that all these directed chains of points represent a single polygon $(w, C_s, C_{h_1}, \ldots, C_{h_r})$.

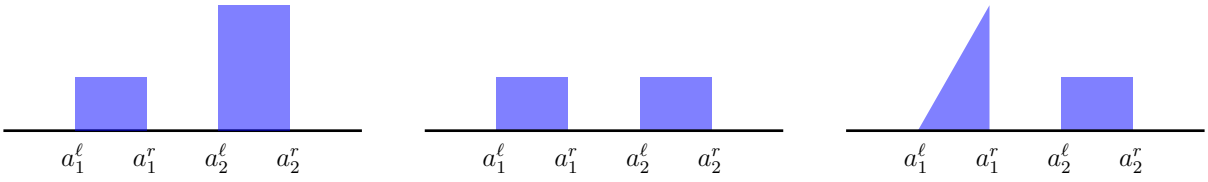
Although it is not hard to construct instances of ε-Straight-Pizza-Sharing (resp. ε-Square-Pizza-Sharing) where n lines (resp. $n-1$ turns for any Square-path) are necessary in order to constitute a solution, there might be cases where a solution can be achieved with fewer lines (resp. a Square-path with fewer turns). Hence, we also study the decision variant of these problems, in which we ask whether we can find a solution with at most k lines (resp. k turns), where $k < n$ (resp. $k < n-1$). Note also that, due to the normalization assumption of the considered measures, $\varepsilon \in [0,1]$.

Consensus halving. The main hardness results of this work are proved by reductions from the consensus halving problem.

[3]From this point on, whenever we refer to polygons we will implicitly assume that they are simple polygons.

In the ε-Consensus-Halving problem, there is a set of n agents with *valuation density functions* $v_i : [0,1] \to \mathbb{R}_{\geq 0}$, $i \in [n]$. For any given interval $[a,b]$, let us denote $v_i([a,b]) := \int_a^b v_i(x)\,dx$. The goal is to find a partition of the $[0,1]$ interval into subintervals labelled either "+" or "−", using at most n cuts. This partition should satisfy that for every agent i, the total value for the union of subintervals $\mathcal{I}^+$ labelled "+" and the total value for the union of subintervals $\mathcal{I}^-$ labelled "−" is the same up to ε, i.e., $|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| \leq \varepsilon$. Furthermore, in order for ε to be meaningful, we consider normalized valuation functions, that is, $v_i([0,1]) = 1$ for all $i \in [n]$, which implies that $\varepsilon \in [0,1]$. In our results, we will use the following types of valuation functions (see Figure 2 for a depiction).

- *k-block*: consists of at most k non-overlapping (but possibly adjacent) intervals $[a_1^\ell, a_1^r], \ldots, [a_k^\ell, a_k^r]$ where interval $[a_j^\ell, a_j^r]$ has density $c_j > 0$, and 0 otherwise. So, $v([a_j^\ell, x]) = (x - a_j^\ell) \cdot c_j$ for every $x \in [a_j^\ell, a_j^r]$.
- *k-block uniform*: k-block, where the density of every interval is $c > 0$ (same for all blocks).
- *k-block-triangle*: union of a k-block valuation function and an extra interval $[a_1^\ell, a_1^r]$, where interval $[a_1^\ell, a_1^r]$ has density $2 \cdot (x - a_1^\ell) \cdot c_1$ for some $c_1 > 0$, therefore $v([a_1^\ell, x]) = (x - a_1^\ell)^2 \cdot c_1$ for every $x \in [a_1^\ell, a_1^r]$. Also, $(a_1^\ell, a_1^r) \cap [a_j^\ell, a_j^r] = \emptyset$ for every $j \in [k]$.



(a) 2-block valuation (b) 2-block uniform valuation (c) 1-block-triangle valuation

Figure 2

Complexity classes. ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing are examples of *total* problems, which are problems that always have a solution. The complexity class TFNP (Total Function NP) defined in [MP91], contains all total problems whose solutions can be verified in polynomial time.

In this work, we will focus on a well-known subclass of TFNP, namely PPA, defined by Papadimitriou [Pap94]. This class captures problems whose totality is guaranteed by the *parity argument* on undirected graphs: if there is an odd-degree vertex then there is another one. In the typical PPA problem, EndOfUndirectedLine, we are given a Boolean circuit N with input of size n and output of size $2n$, and the circuit has a poly(n) size description. The input represents the identity of a vertex and the output represents the identities of (at most) two other vertices. If for two vertices i, j we have $j \in N(i)$ and $i \in N(j)$, then we consider an undirected edge between them. This implies an undirected graph structure where the maximum degree of any vertex is 2. The problem is, given a vertex of degree 1, to find any other vertex of degree 1. Now notice that the graph size is 2^n, whereas the input is poly(n) large, therefore, common algorithms that would solve the problem in case the graph was described explicitly are no longer useful. As discussed earlier, since the definition of PPA, many problems have been shown to be complete for the class, yet most of them require a circuit description in their input. The more interesting cases of PPA-completeness are for problems with more "natural" inputs, in the sense that they require no such circuit description. The pizza sharing problems we study here are among those ones.

The complexity class $\exists\mathbb{R}$ consists of all decision problems that can be formulated in the *existential theory of the reals (ETR)* [Mat14, Sch09]. In other words, problems that can be written in ETR form: $\exists\vec{P} \in \mathbb{R}^m \cdot \Phi$, where Φ is a Boolean formula using connectives $\{\wedge, \vee, \neg\}$ over polynomials with domain $\mathbb{R}^m$ for some $m \in \mathbb{N}$ compared with the operators $\{<, \leq, =, \geq, >\}$. It is known that $\mathsf{NP} \subseteq \exists\mathbb{R} \subseteq \mathsf{PSPACE}$ [Can88], and it is generally believed that $\exists\mathbb{R}$ is distinct from the other two classes. The class FETR ($\mathsf{Function}\ \exists\mathbb{R}$) consists of all search problems whose decision variant is in $\exists\mathbb{R}$. The class TFETR is the subclass of FETR which contains only problems that admit a solution (i.e. all the instances of their decision variant are "yes" instances). Both FETR and TFETR were introduced in [DFMS21] as the natural analogues of FNP and TFNP in the real RAM model of computation. For a definition of the real RAM model we refer the reader to the detailed work of Erickson, van der Hoog, and Miltzow [EvM20].

In this work, our focus regarding complexity classes of TFETR will be on the class FIXP. FIXP was defined in [EY10] and captures problems whose totality is guaranteed by Brouwer's fixed point theorem [Bro11]. An instance of a typical problem in FIXP consists of the description of a continuous function $g : D \to D$, where D is a nonempty, compact, and convex set. g is represented by an *algebraic circuit*, and a solution of the instance is any $x \in D$ such that $g(x) = x$. An algebraic circuit is a circuit that operates on real numbers, and uses gates from the set $\{c, +, -, \times c, \times, \max, \min\}$; a c-gate outputs the constant c, a $\times c$-gate multiplies the input by a constant c, and all other gates behave according to their standard definitions, where $c \in \mathbb{Q}$. It is worth noting that, since each of these gates' output is a continuous function of its input, any function g constructed using those gates is continuous on D.

3 Hardness results

Here we show all hardness results regarding the exact and approximate versions of our pizza sharing problems for mass distributions, as well as for point sets. For the PPA- and NP-hardness results on mass distributions, the instances we construct are such that there is no overlap between any two mass distributions. Notice that the case of non-overlapping mass distributions is the most simple type of an instance, since we can easily reduce it to one where an arbitrarily large number of masses overlap.[4]

Our PPA-hardness proofs of the continuous versions of ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing are via reductions from the ε-Consensus-Halving problem (Theorem 6 and Theorem 9, respectively). Consequently, by a general construction (Lemma 13), we reduce those to their discrete versions to get PPA-hardness. We also show that the decision variants of the approximation problems are NP-hard by using our PPA-hardness constructions to reduce from the respective decision variant of Consensus-Halving from [FRFGZ18] (Theorem 7 and Theorem 10, respectively), and the NP-hardness is attained in the discrete problems too. Finally, for the exact version of Square-Pizza-Sharing, via reductions from exact Consensus-Halving, we show that the problem is FIXP-hard (Theorem 19), while its decision variant is $\exists\mathbb{R}$-hard (Theorem 21).

[4]Given an ε-Straight-Pizza-Sharing (resp. ε-Square-Pizza-Sharing) instance with $2n$ (resp. n) mass distributions, we can pick an arbitrary distribution i and create an extra $(2n+1)$-st (resp. $(n+1)$-st) distribution by copying i. Then, a solution to the resulting instance is a solution to the initial instance, and vice versa. Notice that in the latter instance at least 2 mass distributions overlap, and we can repeat this "copying" procedure as many times as needed to achieve any number of overlapping distributions.

3.1 Hardness of approximate Straight-Pizza-Sharing

We start by proving that ε-Straight-Pizza-Sharing is PPA-hard for any $\varepsilon < 1/5$, even for very simple mass distributions. We prove our result via a reduction from ε-Consensus-Halving with k-block valuations, which for the special case of 3-block uniform valuations has been shown to be PPA-complete [DFHM25]. In addition, we explain how to combine the machinery of our reduction with that of [FRFGZ18] in order to get NP-hardness for ε-Straight-Pizza-Sharing, where $\varepsilon > 0$ is a small constant.

The reduction. We reduce from Consensus-Halving with $2n$ agents, and for each agent we create a corresponding mass in Straight-Pizza-Sharing. Firstly, we finely discretize the $[0, 1]$ interval into blocks and we place the blocks on $y = x^2$, where $x \geq 0$. So, the $[0, 1]$ interval corresponds to a part of the quadratic equation. This guarantees that every line can cut this "bent" interval at most twice and in addition the part of each mass that is in R^+ is almost the same as value of the corresponding agent for the piece of $[0, 1]$ labelled with "+".

Next we show how to construct an instance I_P of $(\varepsilon - \varepsilon')$-Straight-Pizza-Sharing with $2n$ mass distributions, for any constant $r \geq 1$, and $1/n^r \leq \varepsilon' < \varepsilon \leq 1$, given an instance I_{CH} of ε-Consensus-Halving with $2n$ agents with k-block valuations.

Let $c_{\max} := \max_{i\in[2n],m\in[k]} c_{im}$, where c_{im} is the value density of agent i's m-th block in I_{CH}, and observe that $c_{\max} \geq 1$ since the total valuation of any agent over $[0, 1]$ is 1. In what follows, it will help us to think of the interval $[0, 1]$ in I_{CH} as being discretized in increments of $d := \frac{1}{\lceil 8\cdot n\cdot c_{\max}/\varepsilon' \rceil}$. We refer to the subinterval $[(j-1)\cdot d, j\cdot d]$ as the *j-th d-block* of interval $[0, 1]$ in I_{CH}, for $j \in [1/d]$.

We now describe the instance I_P. We consider two kinds of square tiles; $1/d$ large square tiles of size $\frac{d^2}{24} \times \frac{d^2}{24}$, each of which contains $2n$ smaller square tiles of size $\frac{1}{2n}\frac{d^2}{24} \times \frac{1}{2n}\frac{d^2}{24}$ on its diagonal. We will call the former type *big-tile* and denote it by t_j and the latter one *small-tile* and denote it by t_{ij} for some $i \in [2n]$, $j \in [1/d]$.

For every agent $i \in [2n]$ of I_{CH} we will create a uniform mass distribution μ_i that consists of at most $1/d$ many axis-aligned small-tiles. Each big-tile t_j is centered at $(\frac{jd}{2}, \frac{j^2d^2}{4})$, $j \in [1/d]$, and in it, each small-tile t_{ij}, $i \in [2n]$, belonging to mass distribution μ_i has its bottom left corner at $\left(\frac{jd}{2} - \frac{d^2}{48} + \frac{(i-1)d^2}{48n}, \frac{j^2d^2}{4} - \frac{d^2}{48} + \frac{(i-1)d^2}{48n}\right)$. Each small-tile t_{ij} contains total mass (belonging to μ_i) of $v_{ij} \cdot \frac{2}{n}\left(\frac{d^2}{24}\right)^2$, where v_{ij} is the total value that agent i has for the j-th d-block in I_{CH}. Observe that, by definition, we have $v_{ij} \leq d \cdot c_{\max} \leq \frac{\varepsilon'}{8n} < \frac{1}{8n}$, where the last inequality comes from the fact that $\varepsilon' < 1$, and therefore $v_{ij} \cdot \frac{2}{n}\left(\frac{d^2}{24}\right)^2$ fits inside the small tile of size $\frac{1}{2n}\frac{d^2}{24} \times \frac{1}{2n}\frac{d^2}{24}$. In particular, the mass inside t_{ij} has width $\frac{1}{2n}\frac{d^2}{24}$ and height $v_{ij} \cdot \frac{4d^2}{24}$. Finally, it is easy to check that all big-tiles are in $[0, 1]^2$: by construction, the big-tiles are placed such that the 1-st big-tile's bottom-left corner has the smallest x- and y-coordinates, while the $1/d$-th big-tile's top-right corner has the largest x- and y-coordinates among points that belong to mass distributions of I_P. The aforementioned points' coordinates are $\left(\frac{d}{2} - \frac{d^2}{48}, \frac{d^2}{4} - \frac{d^2}{48}\right)$, and $\left(\frac{1}{2} - \frac{d^2}{48} + \frac{d^2}{24}, \frac{1}{4} - \frac{d^2}{48} + \frac{d^2}{24}\right) = \left(\frac{1}{2} + \frac{d^2}{48}, \frac{1}{4} + \frac{d^2}{48}\right)$, respectively, and both are in $[0, 1]^2$ since $d \leq 1$. Figure 3 and Figure 4 depict our construction.

Next, we prove the following auxiliary claim.

Claim 4. *Any straight line in $[0, 1]^2$ cannot have distance at most $\frac{d^2}{24}$ with more than two centers of big-tiles.*

Proof. For the sake of contradiction, suppose there are three tiles, t_a, t_b, t_c, with centers $p_j =$

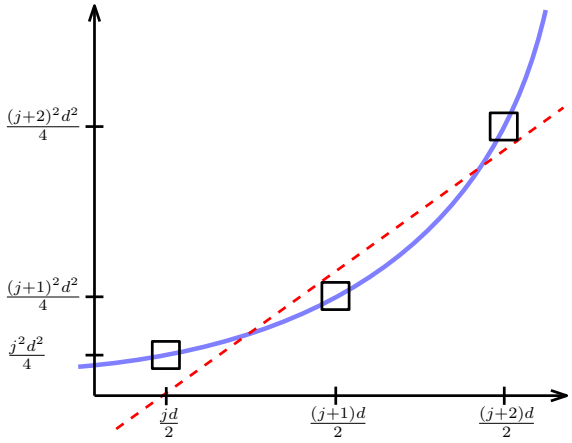
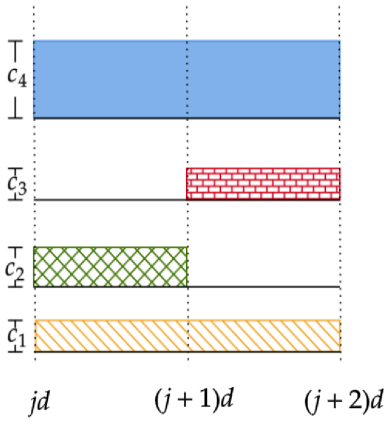
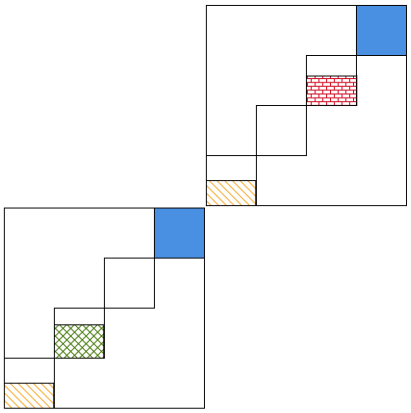


Figure 3: Placing the big-tiles on the $y = x^2$ curve. The j-th, $(j+1)$-st and $(j+2)$-nd big-tiles are centered on the curve. Their size is small enough to prevent any straight line (red/dashed) from intersecting more than two big-tiles.



(a) The $(j+1)$-st (left) and $(j+2)$-nd d-block (right) of the Consensus-Halving instance.



(b) The $(j+1)$-st (left) and the $(j+2)$-nd (right) big-tile of the Straight-Pizza-Sharing and the Square-Pizza-Sharing instance with their small-tiles. The small-tiles contain the mass distributions of the $(j+1)$-st and $(j+2)$-nd d-block, respectively.

Figure 4: The construction for a part of an instance with four mass distributions.

(x_j, y_j), $j \in \{a, b, c\}$, such that every p_j has distance at most $\frac{d^2}{24}$ from a line ℓ. Then, let us move ℓ in parallel until it passes through p_a. Next, rotate the line around p_a until it passes through p_b as well. These two movements of ℓ resulted in a new line ℓ' whose distance from p_c is at most $3 \cdot \frac{d^2}{24} = \frac{d^2}{8}$, since each movement costed an extra distance of at most $\frac{d^2}{24}$.

The distance between ℓ' and p_c is

$$\frac{|(y_b - y_a)(x_a - x_c) - (y_a - y_c)(x_b - x_a)|}{\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}} = \frac{\left|(d^2/4)(a - c)(b - c)\right|}{\sqrt{1 + (d^2/4)(a + b)}},$$

where the equality comes after substituting the coordinates of the centers of t_a, t_b and t_c as defined by our construction and simplifying the expression. Now recall that $a, b, c \in [1/d]$, which means that they are integers, and additionally, pairwise different, since otherwise they would be less than three distinct points. This means that we can bound from below the minimum distance between ℓ' and p_c for $|a - c| = |b - c| = 1$ and $a = b = 1/d$. In other words, the minimum distance between ℓ' and p_c is at least

$$\frac{d^2/4}{\sqrt{1 + (d^2/4)(2/d)}} = \frac{d^2/4}{\sqrt{1 + d/2}} > \frac{d^2}{8},$$

a contradiction. □

Now we are ready to prove the following.

Lemma 5. *Fix constants $\delta \in (0, 1]$, $r \geq 1$, and let $1/n^r \leq \varepsilon' < \varepsilon \leq 1$. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_m\}$ be a set of lines, where $m \leq n + n^{1-\delta}$. If $\mathcal{L}$ is a solution to $(\varepsilon - \varepsilon')$-*Straight-Pizza-Sharing *instance I_P, then we can find in polynomial time a solution to ε-*Consensus-Halving *instance I_{CH} with at most $2(n + n^{1-\delta})$ cuts.*

Proof. We will first prove that there is no line that intersects more than two big-tiles of I_P. This comes almost directly from Claim 4. In particular, recall that each big-tile has size $\frac{d^2}{24} \times \frac{d^2}{24}$, therefore it fits inside a circle with radius $\frac{d^2}{24}$, whose center is the barycenter of the big-tile. If any line could intersect more than two big-tiles, then it would intersect also their corresponding circles, which means that it would have distance at most $\frac{d^2}{24}$, contradicting Claim 4.

Now, given a solution of I_P, we define the cuts and labels for the Consensus-Halving instance, I_{CH}. We consider the big-tiles of I_P in sequential order and we add one cut at $j \cdot d$ whenever we find two big-tiles t_j and t_{j+2} that belong to different regions, i.e. "+", "−", or vice versa. This change of region can happen at most $2(n + n^{1-\delta})$ times. Hence, we have at most $2(n + n^{1-\delta})$ cuts in the instance I_{CH}. Each d-block of I_{CH} follows the label of its corresponding big-tile of the solution of I_P, except for those that correspond to intersected big-tiles. The latter d-blocks are arbitrarily given one of the two labels.

The aforementioned arbitrary labeling of the intersected big-tiles will cause some extra discrepancy in the solution of I_{CH}. In particular, each such big-tile will be adding to each valuation v_i, $i \in [2n]$ of I_{CH}, discrepancy $2 \cdot v_{ij} \leq 2 \cdot d \cdot c_{\max} \leq \frac{\varepsilon'}{4n}$, by construction of I_{CH}. Since $|\mathcal{L}| \leq n + n^{1-\delta}$, and each line of $\mathcal{L}$ can intersect two big-tiles, all lines of $\mathcal{L}$ collectively add discrepancy v_i of value at most $2(n + n^{1-\delta}) \cdot \frac{\varepsilon'}{4n} \leq \varepsilon'$.

Let us denote by $\mathcal{I}^+$ and $\mathcal{I}^-$ the regions in I_{CH} as translated from the solution of I_P according to the aforementioned process. Then, $|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| \leq |\mu_i(R^+) - \mu_i(R^-)| + \varepsilon' \leq (\varepsilon - \varepsilon') + \varepsilon' = \varepsilon$. Therefore, this is a solution to ε-Consensus-Halving.

Finally, notice that the construction of I_P from I_{CH} is a polynomial time reduction. In particular, since $\varepsilon' \geq 1/n^r$ for some constant $r \geq 1$, we have that $1/d = \lceil 8 \cdot n \cdot c_{\max}/\varepsilon' \rceil \leq \lceil 8 \cdot n \cdot c_{\max} \cdot n^r \rceil$ is a value polynomial in the input size, and the creation of each of the $1/d$ big-tiles can be done in polynomial time. □

This, together with the fact that ε-Consensus-Halving is PPA-hard for any $\varepsilon < 1/5$, due to [DFHM25], implies the main theorem of this section.

Theorem 6. *ε-Straight-Pizza-Sharing with $2n$ mass distributions is PPA-hard for any constant $\varepsilon < 1/5$, even when $n + n^{1-\delta}$ lines are allowed for any given constant $\delta \in (0, 1]$, every mass distribution is uniform over polynomially many rectangles, and there is no overlap between any two mass distributions.*

We will now shift our attention to studying the decision variant of the problem, where we are asking to find a solution that uses at most $n - 1$ straight lines, and notice that there is no guarantee for such a solution. We employ the NP-hard instances of ε-Consensus-Halving for their constant ε from [FRFGZ18], and reduce them according to the above reduction procedure to $(\varepsilon - \varepsilon')$-Straight-Pizza-Sharing instances for some ε' that is inverse polynomial in the input size, e.g., $\varepsilon' = 1/n^2$. This gives the following.

Theorem 7. *There exists a constant $\varepsilon > 0$ for which it is NP-hard to decide if an ε-Straight-Pizza-Sharing instance with $2n$ mass distributions admits a solution with at most $n - 1$ lines, even when every mass distribution is uniform over polynomially many rectangles, and there is no overlap between any two mass distributions.*

3.2 Hardness of approximate Square-Pizza-Sharing

In this section, we prove hardness results for ε-Square-Pizza-Sharing. We provide a reduction from ε-Consensus-Halving with k-block valuations, which was shown to be PPA-complete in [DFHM25] for any constant $\varepsilon < 1/5$ even for 3-block uniform valuations. Also, the machinery that we present, combined with the reduction by [FRFGZ18], implies NP-hardness for the decision variant of ε-Square-Pizza-Sharing, where $\varepsilon > 0$ is a small constant.

The reduction. We reduce from a general ε-Consensus-Halving instance to an ε-Square-Pizza-Sharing instance, and the idea is to create a mass for each agent. For any constant $r \geq 1$, and $1/n^r \leq \varepsilon' < \varepsilon \leq 1$, given an instance I_{CH} of ε-Consensus-Halving with n agents and k-block valuations we will show a polynomial time construction to an $(\varepsilon - \varepsilon')$-Square-Pizza-Sharing instance I_{SC}.

For our construction, we will use the same components as those in the proof of Lemma 5. In particular, let $c_{\max} := \max_{i\in[n],m\in[k]} c_{im}$, where c_{im} is the value density of agent i's m-th block in I_{CH} (and again note that $c_{\max} \geq 1$ since the total valuation of the agent over $[0, 1]$ is 1). Similarly to the aforementioned proof, we will discretize the $[0, 1]$ interval of I_{CH} in increments of $d := \frac{1}{\lceil 4\cdot n\cdot c_{\max}/\varepsilon' \rceil}$. Also, let us restate that for any given $j \in [1/d]$, the subinterval $[(j - 1) \cdot d, j \cdot d]$ is called j-th d-block of interval $[0, 1]$ in I_{CH}.

We will be using the same gadgets that were constructed for the proof of Lemma 5, namely the big-tiles, which contain small-tiles. In particular, we have $1/d$ square big-tiles of size $d \times d$, each of which contains n square small-tiles of size $\frac{d}{n} \times \frac{d}{n}$ on its diagonal. For any $i \in [n]$, $j \in [1/d]$, we denote them by t_j and t_{ij}, respectively.

In this construction, however, the positioning of big-tiles will be different than that of the aforementioned proof. In particular, we will be placing them on the diagonal of $[0, 1]^2$ as shown in Figure 4b. For every agent i we will create a uniform mass distribution μ_i that consists of at most $1/d$ many axis-aligned small-tiles. For each $j \in [1/d]$, the bottom-left corner of big-tile t_j is at $((j - 1)d, (j - 1)d)$. In it, each small-tile t_{ij}, $i \in [n]$, belonging to mass distribution μ_i has its bottom left corner at $\left((j - 1)d + (i - 1)\frac{d}{n}, (j - 1)d + (i - 1)\frac{d}{n}\right)$.

Inside a given big-tile t_j, each small-tile t_{ij} contains total mass (belonging to μ_i) of $v_{ij} \cdot \frac{4d^2}{n}$, where v_{ij} is agent i's total value for the j-th d-block in I_{CH}. The mass inside the small-tile is

rectangular, with width $\frac{d}{n}$ and height $v_{ij} \cdot 4d$. Observe that $v_{ij} \leq d \cdot c_{\max} \leq \frac{\varepsilon'}{4n} < \frac{1}{4n}$, where the first inequality is by definition of a d-block in I_{CH}, the second one is by definition of d, and the last one is due to $\varepsilon' < 1$. Therefore the total mass of $v_{ij} \cdot \frac{4d^2}{n}$ fits inside the small-tile. By definition of the big-tiles positioning and size, it is straightforward that they are inside $[0,1]^2$. Figure 4 depicts our construction.

Now we are ready to prove the following lemma.

Lemma 8. *Fix constants $\delta \in (0,1]$, $r \geq 1$, and let $2/n^r \leq \varepsilon' < \varepsilon \leq 1$. Let a* Square*-path with at most $n-1+n^{1-\delta}$ turns be a solution to $(\varepsilon-\varepsilon')$-*Square-Pizza-Sharing *instance I_{SC}. Then we can find in polynomial time a solution to ε-*Consensus-Halving *instance I_{CH} with at most $n+n^{1-\delta}$ cuts.*

Proof. We have to specify how a solution of I_{SC}, i.e., a Square-path, is translated back to a solution of I_{CH}, i.e., a set of cuts. This is identical to the one used in the proof of Lemma 5. In particular, we consider again the big-tiles in sequential order and we add one cut at $j \cdot d$ whenever we find two big-tiles t_j and t_{j+2} that belong to different regions. Suppose that, following the aforementioned procedure, the next I_{CH} cut falls at $j \cdot d'$ for some $d' > d$. If t_j belongs to region "+" (resp. "−") and t_{j+2} belongs to "−" (resp. "+"), then the interval $[j \cdot d, j \cdot d']$ gets label "+" (resp. "−"), and vice versa. It is easy to see that this translation takes polynomial time.

What remains is to prove that the translation of the aforementioned solution of a $(\varepsilon-\varepsilon')$-Square-Pizza-Sharing instance I_{SC} into a solution of the ε-Consensus-Halving instance I_{CH} is indeed correct. Notice that, if the solution of I_{SC} has $r \in \mathbb{N}$ many turns on the Square-path, there can be at most $r+1$ small-tiles that are intersected by it (since there are $r+1$ line segments). For our reduction, let $r = n-1+n^{1-\delta}$ for any constant $\delta \in (0,1]$. Consider sequentially the big-tiles $t_1, \ldots, t_{1/d}$, and without loss of generality, let $t'_1, \ldots, t'_{r+1}$ be its subset, where t'_j is the j-th big-tile that has an intersected small-tile. In the big-tile sequence of $t_1, \ldots, t_{1/d}$, the change of region can happen at most $n+n^{1-\delta}$ times. Therefore, we have at most $n+n^{1-\delta}$ cuts in I_{CH} with the corresponding labels as defined previously. Each d-block of I_{CH} follows the label of the corresponding big-tile in I_{SC}, except for those corresponding to intersected big-tiles. These d-blocks are given an arbitrary label.

The above translation of the Square-path to I_{CH} cuts indicates that, each line segment of the Square-path that intersects a big-tile, introduces a discrepancy between the "+" and "−" regions of I_{CH}, of value at most $2 \cdot c_{\max} \cdot d \leq \frac{\varepsilon'}{2n}$ for each valuation v_i, $i \in [n]$. Taking into account the entire Square-path, this results in total discrepancy of at most $(n+n^{1-\delta}) \cdot \frac{\varepsilon'}{2n} \leq \varepsilon'$ for each agent i.

We now denote by $\mathcal{I}^+$ and $\mathcal{I}^-$ the regions in I_{CH} according to the above translation from a solution of I_{SC} to a solution of I_{CH}. We have $|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| \leq |\mu_i(R^+) - \mu_i(R^-)| + \varepsilon' \leq (\varepsilon-\varepsilon') + \varepsilon' = \varepsilon$, therefore, this is indeed a solution to ε-Consensus-Halving.

Again, notice that the construction we described can be done in polynomial time, since the creation of each big-tile can be done in polynomial time, and we have $1/d$ many big-tiles, where $1/d = \lceil 4 \cdot n \cdot c_{\max}/\varepsilon' \rceil \leq \lceil 4 \cdot n \cdot c_{\max} \cdot n^r \rceil$ for some constant $r \geq 1$. □

The above, together with the fact that ε-Consensus-Halving is PPA-hard for any $\varepsilon < 1/5$ ([DFHM25]) implies the main theorem of this section.

Theorem 9. ε-Square-Pizza-Sharing *with n mass distributions is* PPA*-hard for any constant $\varepsilon < 1/5$, even when $n-1+n^{1-\delta}$ turns are allowed in the* Square*-path for any given constant $\delta \in (0,1]$, every mass distribution is uniform over polynomially many rectangles, and there is no overlap between any two mass distributions.*

Similarly to the case of the decision variant of ε-Straight-Pizza-Sharing (Theorem 7), we can get the following result by reducing from the NP-hard instances of ε-Consensus-Halving for constant ε.

Theorem 10. *There exists a constant $\varepsilon > 0$ for which it is NP-hard to decide if an ε-Square-Pizza-Sharing instance with n mass distributions admits a solution consisting of a Square-path with at most $n-2$ turns, even when every mass distribution is uniform over polynomially many rectangles, and there is no overlap between any two mass distributions.*

3.3 Hardness of discrete Straight-Pizza-Sharing and Square-Pizza-Sharing

In this section, we study the discrete versions of Straight-Pizza-Sharing and Square-Pizza-Sharing.

Definition 11. *For any $n \geq 1$, the problem ε-Discrete-Straight-Pizza-Sharing is defined as follows:*

- ***Input:*** *$\varepsilon \geq 0$, and $2n$ point sets $P_1, P_2, \ldots, P_{2n}$ on $[0,1]^2$.*
- ***Output:*** *One of the following.*
  - *(a) Three points that can be intersected by the same line.*
  - *(b) A partition of $[0,1]^2$ to R^+ and R^- using at most n lines such that for each $i \in [2n]$ it holds that $||P_i \cap R^+| - |P_i \cap R^-|| \leq \varepsilon \cdot |P_i|$.*

A point that is intersected by a line does not belong to any of R^+, R^-.

Definition 12. *For any $n \geq 1$, the problem ε-Discrete-Square-Pizza-Sharing is defined as follows:*

- ***Input:*** *$\varepsilon \geq 0$, and n point sets $P_1, P_2, \ldots, P_n$ on $[0,1]^2$.*
- ***Output:*** *One of the following.*
  - *(a) Two points with the same x- or y-coordinate.*
  - *(b) A partition of $[0,1]^2$ to R^+ and R^- using a y-monotone Square-path with at most $n-1$ turns such that for each $i \in [n]$ it holds that $||P_i \cap R^+| - |P_i \cap R^-|| \leq \varepsilon \cdot |P_i|$.*

A point that is intersected by a line does not belong to any of R^+, R^-.

Notice that the first kind of allowed output for both problems (Definition 11(a), Definition 12(a)) is a witness that the input points are not in general position or that their x- or y-coordinates are not unique, respectively, which can be checked in polynomial time. The second kind of output (Definition 11(b), Definition 12(b)) is the one that is interesting and can encode the hard instances studied here. In case the first kind of output does not exist, the other one is guaranteed to exist due to [Sch21] for ε-Discrete-Straight-Pizza-Sharing, while for ε-Discrete-Square-Pizza-Sharing its existence is guaranteed for every $\varepsilon \in [0,1]$ due to a reduction we present in Section 4.3 which shows PPA membership.

The definition of ε-Discrete-Straight-Pizza-Sharing is a slightly modified form of the one that appears in [Sch21], where it is referred to as DiscretePizzaCutting. In our definition, we avoid having to "promise" an input of points that are in general position, by allowing as output a witness of an inappropriate input. Furthermore, the definition we present is more general, since it accommodates an approximation factor ε; in particular, for $\varepsilon < \min_i\{1/|P_i|\}$ we get the definition of the aforementioned paper. Similarly, we define ε-Discrete-Square-Pizza-Sharing, which to the best of our knowledge, has not been stated in previous work. Note that, if the input of any of the discrete versions of the problems consists of points that are in general position and furthermore have pairwise different x- and y-coordinates (a property that the instances in our reductions have), in any ε-Discrete-Straight-Pizza-Sharing solution, a line can only intersect up to two points, while in any ε-Discrete-Square-Pizza-Sharing solution, a line segment can intersect up to one point.

Recall that in ε-Square-Pizza-Sharing we are given n mass distributions, while in ε-Straight-Pizza-Sharing we are given $2n$ mass distributions as input. In Appendix A, we describe a general construction that takes as an input $q \in \{n, 2n\}$ mass distributions $\mu_1, \ldots, \mu_q$ normalized on $[0,1]^2$, represented by weighted polygons with holes (see Section 2 for the detailed description), and turns it into q sets of points $P_1, \ldots, P_q$ on $[0,1]^2$. The points that constitute those sets' union are in general position, and additionally, they have unique x- and y-coordinates. We prove that, if a set of at most n lines or a Square-path of at most $n-1$ turns partitions $[0,1]^2$ into R^+ and R^- such that $||P_i \cap R^+| - |P_i \cap R^-|| \leq (\varepsilon - \varepsilon') \cdot |P_i|$, then the same set of lines or Square-path, respectively, separates the mass distributions such that $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$.

Let $N \geq 2n$ be the input size of any of our two pizza sharing problems, and let the smallest area triangle in the mass distributions' triangulation be α. For any $\varepsilon' < \varepsilon$, where ε and α are at least inverse polynomial in N, the construction results in a polynomial time reduction from ε-Straight-Pizza-Sharing to $(\varepsilon - \varepsilon')$-Discrete-Straight-Pizza-Sharing and from ε-Square-Pizza-Sharing to $(\varepsilon - \varepsilon')$-Discrete-Square-Pizza-Sharing. The reduction can be performed in time polynomial in the input size and in $1/\alpha$. It consists of two parts: (i) First we "pixelate" the mass distributions finely enough so that they are represented by a sufficiently large number of pixels. This will ensure a high enough "resolution" of the pixelated distributions. (ii) The pixels will then be turned into points, which we have to perturb in order to guarantee they are in general position and with unique coordinates, as required. In particular, in Appendix A, we prove the following.

Lemma 13. *Let N be the input size of an approximate pizza sharing problem (either ε-Straight-Pizza-Sharing or ε-Square-Pizza-Sharing) whose triangulation has no triangle with area less than $\alpha > 0$. Also, let $\varepsilon' \in \left[\frac{6}{N^c}, \varepsilon\right)$, where $c > 0$ is a fixed constant, and $\frac{6}{N^c} < \varepsilon < 1$. Then, the instance can be reduced in time $\mathrm{poly}(N, 1/\alpha)$ to its approximate discrete version, that is, $(\varepsilon - \varepsilon')$-Discrete-Straight-Pizza-Sharing or $(\varepsilon - \varepsilon')$-Discrete-Square-Pizza-Sharing, respectively.*

Given Theorem 6 and Theorem 9, and since their instances are constructed such that α is an at least inverse polynomial function of the input size, Lemma 13 implies the following hardness results.

Theorem 14. *ε-Discrete-Straight-Pizza-Sharing with $2n$ point sets is PPA-hard for any constant $\varepsilon < 1/5$, even when $n + n^{1-\delta}$ lines are allowed for any given constant $\delta \in (0, 1]$.*

Theorem 15. *ε-Discrete-Square-Pizza-Sharing with n point sets is PPA-hard for any constant $\varepsilon < 1/5$, even when $n - 1 + n^{1-\delta}$ turns are allowed in the Square-path for any given constant $\delta \in (0, 1]$.*

We note that PPA-hardness for ε-Discrete-Straight-Pizza-Sharing was so far known only for any $\varepsilon \in [0, \min_i\{1/|P_i|\})$ (which is equivalent to $\varepsilon = 0$), due to [Sch21]. Since, in the aforementioned paper's constructions, $\min_i\{1/|P_i|\} \in O(1/\text{poly}(N))$, our result strengthens the hardness of the problem significantly.

Lemma 13 is general enough to allow us to derive NP-hardness results for the decision variants of the two discrete versions of the pizza sharing problems. If we ask for a solution with at most $n-1$ straight lines or $n-2$ turns in $(\varepsilon-\varepsilon')$-Discrete-Straight-Pizza-Sharing and $(\varepsilon-\varepsilon')$-Discrete-Square-Pizza-Sharing, respectively, then we can easily reduce to them from the instances of Theorem 7 and Theorem 10, picking ε' to be some inverse polynomial function of N, e.g., $\varepsilon' = 1/N$. In particular, we get the following.

Theorem 16. *There exists a constant $\varepsilon > 0$ for which it is NP-hard to decide whether a solution of ε-Discrete-Straight-Pizza-Sharing with $2n$ point sets and at most $n-1$ lines exists.*

Theorem 17. *There exists a constant $\varepsilon > 0$ for which it is NP-hard to decide whether a solution of ε-Discrete-Square-Pizza-Sharing with n point sets and a Square-path with at most $n-2$ turns exists.*

3.4 Hardness of exact Square-Pizza-Sharing

In this section, we show hardness results for *exact* Square-Pizza-Sharing, that is, Definition 3 for $\varepsilon = 0$. We prove that solving Square-Pizza-Sharing is FIXP-hard and that deciding whether there exists a solution for Square-Pizza-Sharing with fewer than $n-1$ turns is $\exists\mathbb{R}$-hard.

As mentioned earlier, computing an exact solution of a FIXP-hard problem may require computing an irrational number. To showcase this for Square-Pizza-Sharing, consider the following simple instance. Let us have a single mass distribution in the shape of a right-angled triangle, whose corners are on $(0,1)$, $(1,1)$, and $(1,0)$. It is normalised, i.e., its total mass is 1, therefore its weight is 2. An exact solution of Square-Pizza-Sharing is either a horizontal or a vertical straight line (with 0 turns), which cuts the triangle such that each half-space has half of the mass, that is, $1/2$. One can easily check that the solution is either the horizontal line $y = \sqrt{2}/2$, or the vertical line $x = \sqrt{2}/2$.

We provide a main reduction from (exact) Consensus-Halving to (exact) Square-Pizza-Sharing, whose gadgets are then employed to show the $\exists\mathbb{R}$-hardness of the decision variant. This time, as a starting point, we will use the instances of Consensus-Halving constructed as end-points of the reductions in [DFMS21]. In particular, when clear from context, $I_{\text{CH}}^{\text{DFMS}}$ will denote an arbitrary instance satisfying properties 1, 2, and 3 (see next paragraph). When requesting a solution of $I_{\text{CH}}^{\text{DFMS}}$ with at most n cuts, we get FIXP-hardness, while when requesting to decide if it is solvable with $n-1$ cuts we get $\exists\mathbb{R}$-hardness. Both of these results are due to [DFMS21], and hold even for 6-block-triangle valuations. We note that here the input consists of sets of points with rational coordinates, i.e., we describe polygons by their vertices. For a detailed description of the input representation, see Section 2. In the aforementioned work, the problems' input is n algebraic circuits capturing the cumulative valuation of n agents on $[0,1]$, which are piece-wise polynomials of maximum degree 2. In particular, since their (density) valuation functions are piece-wise linear, the input of Square-Pizza-Sharing suffices to consist of very simple shapes, namely, only rectangles and triangles.

The reduction. Here we show the main reduction, which will conclude with the proof that finding an exact solution to Square-Pizza-Sharing is FIXP-hard. Then, this reduction's construction will be used to show that the problem's decision variant is $\exists\mathbb{R}$-hard. We reduce

from a Consensus-Halving instance $I_{\text{CH}}^{\text{DFMS}}$ with n agents and 6-block-triangle valuations to a Square-Pizza-Sharing instance I_{SC} with n mass distributions.

The key difference between this reduction and the previous reductions on the approximate versions is that the starting point of the reduction, i.e., instance $I_{\text{CH}}^{\text{DFMS}}$, besides rectangular-shaped (constant) pieces, also contains triangular-shaped (linear) pieces in the valuation density functions for some agents. More specifically, all of the following properties hold for $I_{\text{CH}}^{\text{DFMS}}$:

1. the valuation function of every agent is 4-block-triangle, or 6-block;
2. for any given agent $i \in [n]$, every triangle (linear piece) of her valuation function has height 2 and belongs to exactly one interval of interest of the form $[x_j, x_{j+1}]$ for $j \in [m]$, where $m \leq 12n + 1$ (see below for the definition of those intervals);
3. for every agent $i \in [n]$ there exists an interval $[a_i, b_i]$ that contains more than half of their total valuation (i.e., more than 1/2 cumulative valuation), and in addition, for every $i' \neq i$ we have $(a_i, b_i) \cap (a_{i'}, b_{i'}) = \emptyset$.

Also, in this reduction, the resulting Square-Pizza-Sharing instances will contain *weighted* mass distributions (see definition in Section 2).

The first step is to partition $[0, 1]$ of $I_{\text{CH}}^{\text{DFMS}}$ into subintervals that are defined by *points of interest*. We say that a point $x \in [0, 1]$ is a point of interest if it coincides with the beginning or the end of a valuation block or triangle of an agent; formally, x is a point of interest if $x \in \{a_{ij}^{\ell}, a_{ij}^{r}\}$ for some agent $i \in [n]$ and some valuation block j or triangle of hers (for this notation see Section 2). These points conceptually split $[0, 1]$ into *intervals of interest*, since in between any pair of consecutive points of interest, all agents have a non-changing valuation density. Let $0 =: x_1 < x_2 < \ldots < x_m < x_{m+1} := 1$ denote the points of interest, and each interval of interest $[x_j, x_{j+1}]$, $j \in [m]$ is called the *j-th subinterval*. Observe that $m \leq 12n + 1$: as a base case consider a single block or triangle produces at most two points of interest and therefore at most three intervals, so $m \leq 3$; for any block or triangle we add, we increase by at most 2 the points of interest; each valuation function has either 4 blocks and 1 triangle, or 6 blocks, therefore the total number of blocks and triangles is at most $n \cdot 6$ (by Property 1 above).

Here it is important to mention that in our reduction we are allowed to only use FIXP gates (see Section 2 for details). As representation of the $I_{\text{CH}}^{\text{DFMS}}$ instance,[5] for each agent $i \in [n]$ we consider ℓ_i ordered pairs of points in $[0, 1]$ interpreted as consecutive intervals' endpoints, together with their corresponding valuation density function in the form of a circuit: $\left((r_k^{(i)}, r_{k+1}^{(i)}), f_k^{(i)}(x)\right)_{k \in [\ell_i]}$, where $r_1^{(i)} := 0$ and $r_{\ell_i+1}^{(i)} := 1$ for all $i \in [n]$. In particular, according to $I_{\text{CH}}^{\text{DFMS}}$, for any $i \in [n]$, $k \in [\ell_i]$, either $f_k^{(i)}(x) = c_{ik}$ or $f_k^{(i)}(x) = 2(x - r_k^{(i)})c_{ik}$, where $c_{ik} \geq 0$ is a constant.

The next step is to do some preprocessing of the input in order to incorporate the intervals of interest. The points of interest can be identified using an algebraic circuit by a sorting network (e.g., [Knu98]) which takes as input all points $\left(r_k^{(i)}\right)_{i \in [n], k \in [\ell_i]}$ and outputs them in non-decreasing order $(x_j)_{j \in [m+1]}$. Then, we turn the $I_{\text{CH}}^{\text{DFMS}}$ instance representation into the following form for each agent $i \in [n]$: $\left((x_{j,k}^{(i)}, x_{j+1,k}^{(i)}), f_k^{(i)}(x)\right)_{k \in [\ell_i], j \in [m]}$, where

$$x_{j,k}^{(i)} = \max\{r_k^{(i)}, \min\{x_j, r_{k+1}^{(i)}\}\}.$$

[5] This is not the same input representation as the one defined in [DFMS21] for exact Consensus-Halving. However, it is easy to check that the FIXP-hardness reduction of the aforementioned work goes through if we require this new input representation of Consensus-Halving.

Observe that, by definition, for any $i \in [n]$, $k \in [\ell_i]$, and $j \in [m]$, either $[x_j, x_{j+1}] \cap [r_k^{(i)}, r_{k+1}^{(i)}] = [x_j, x_{j+1}]$, or $[x_j, x_{j+1}] \cap [r_k^{(i)}, r_{k+1}^{(i)}]$ is singleton or empty. So, we have the following cases:

(a) $x_j \leq r_k^{(i)}$ (and $x_{j+1} \leq r_k^{(i)}$): The first inequality implies $x_j \leq r_{k+1}^{(i)}$ and so, $x_{j,k}^{(i)} = r_k^{(i)}$. The second inequality implies $x_{j+1} \leq r_{k+1}^{(i)}$), and so, $x_{j+1,k}^{(i)} = r_k^{(i)}$. Therefore, $x_{j,k}^{(i)} = x_{j+1,k}^{(i)} = r_k^{(i)}$.

(b) $x_j \geq r_{k+1}^{(i)}$ (and $x_{j+1} \geq r_{k+1}^{(i)}$): Similarly to above case, $x_{j,k}^{(i)} = x_{j+1,k}^{(i)} = r_{k+1}^{(i)}$.

(c) $x_j \geq r_k^{(i)}$ and $x_{j+1} \leq r_{k+1}^{(i)}$: By definition, $x_j \leq x_{j+1}$, so we get $x_{j,k}^{(i)} = x_j$ and $x_{j+1,k}^{(i)} = x_{j+1}$.

This way, using FIXP gates we managed to copy the valuation density function $f_k^{(i)}(x)$ to all the intervals $[x_j, x_{j+1}]$ that are inside $[r_k^{(i)}, r_{k+1}^{(i)}]$, while for those that are outside of it, we created artificial singleton intervals which do not contribute to the cumulative valuation function (since $f_k^{(i)}$ is a density function).

Now we are ready to construct the Square-Pizza-Sharing instance I_{SC}. For each subinterval $j \in [m]$ of $I_{\mathrm{CH}}^{\mathrm{DFMS}}$, we will construct a tile t_j of size $d_j \times d_j$ in which we will place our measures, where $d_j := x_{j+1} - x_j$, and notice that always $d_j > 0$. These tiles will be placed diagonally in $[0,1]^2$, starting from the bottom-left corner (see Figure 5b for a depiction). The points describing t_j are $LL := (x_j, x_j)$, $HL := (x_{j+1}, x_j)$, $HH := (x_{j+1}, x_{j+1})$, and $LH := (x_j, x_{j+1})$. Inside each tile t_j, we place triangles $z_{j,k}^{(i)}$ for all $i \in [n], k \in [\ell_i]$, each with vertices HL, HH, and LH. Triangle $z_{j,k}^{(i)}$ has weight $w_{j,k}^{(i)} = d_j^{-2} \cdot (x_{j+1,k}^{(i)} - x_{j,k}^{(i)}) \cdot f_k^{(i)}(x_{j+1,k}^{(i)})$. Notice that here we have also used a division gate.[6] Also, we place triangles $z_{j,k}^{*(i)}$ for all $i \in [n], k \in [\ell_i]$, each with vertices LL, HL, and HH. Triangle $z_{j,k}^{*(i)}$ has weight $w_{j,k}^{*(i)} = d_j^{-2} \cdot (x_{j+1,k}^{(i)} - x_{j,k}^{(i)}) \cdot f_k^{(i)}(x_{j,k}^{(i)})$ (notice the change in the argument of $f_k^{(i)}$).

By the above construction, if $f_k^{(i)}(x) = c_{ik}$, then $w_{j,k}^{(i)} = w_{j,k}^{*(i)} = d_j^{-2} \cdot (x_{j+1,k}^{(i)} - x_{j,k}^{(i)}) \cdot c_{ik}$, and by multiplying all parts with d_j^2 we get that for every colour $i \in [n]$ the area of the j-th subinterval equals its measure inside tile t_j, and has square shape. Similarly, if $f_k^{(i)}(x) = 2(x - r_k^{(i)})c_{ik}$, then $w_{j,k}^{(i)} = d_j^{-2} \cdot (x_{j+1,k}^{(i)} - x_{j,k}^{(i)}) \cdot 2(r_{k+1}^{(i)} - r_k^{(i)})c_{ik}$, and $w_{j,k}^{*(i)} = 0$. By multiplying both sides of the former equation with $d_j^2/2$ we get that for every colour $i \in [n]$ the area of the j-th subinterval equals its measure inside tile t_j, and has triangular shape (see Figure 5).

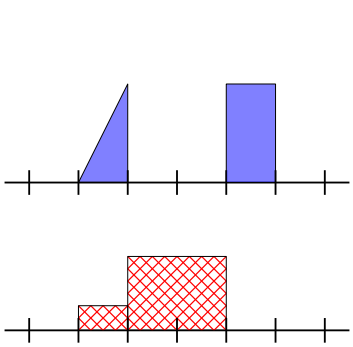
Now we need to show how an exact solution of I_{SC}, that is, a Square-path with $n-1$ many turns is mapped back to a solution of $I_{\mathrm{CH}}^{\mathrm{DFMS}}$ with n cuts. This is straightforwardly done in the following way. Let the solution be represented by an ordered tuple of $(n-1)$ points $(p_1, \ldots, p_{n-1})$ interpreted as the turns of the Square-path. Then, p_1's coordinates correspond to the first two cuts in $[0,1]$ of $I_{\mathrm{CH}}^{\mathrm{DFMS}}$, and for the remaining $n-2$ points, those with even index encode a cut at their y-coordinate, while those with odd index encode it at their x-coordinate.

It remains to prove that this is a solution of $I_{\mathrm{CH}}^{\mathrm{DFMS}}$. To do this, we will use the following crucial observation.

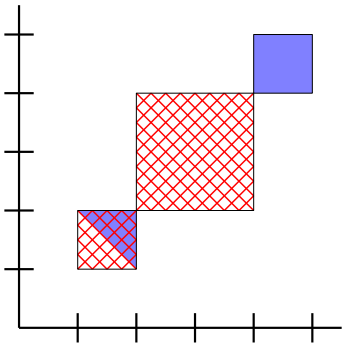
Claim 18. *In any solution of I_{SC} created by $I_{\mathrm{CH}}^{\mathrm{DFMS}}$, there can be no turn inside a tile.*

Proof. The truth of the statement can become apparent if one considers Property 3 of the above facts on $I_{\mathrm{CH}}^{\mathrm{DFMS}}$, together with the fact that the endpoints of each $[a_i, b_i]$ are points of interest.

[6] The division gate is among those available in FIXP and can be used as long as there is no division with 0. However, it is known that the definition of the class does not need it since there are FIXP-hard problems that do not use this gate. For a proof of this, see [EY10].



(a) Part of the Consensus-Halving instance with two agents and the corresponding regions of interest.



(b) The corresponding part of the Square-Pizza-Sharing instance.

Figure 5

It is implied then, that in any of the $I_{\text{CH}}^{\text{DFMS}}$ solutions, there needs to be at least one cut in each interval $[a_i, b_i]$ for each $i \in [n]$. And since we are allowed to draw at most n cuts, there will be a single cut in each of those intervals. Also, due to the fact that a_i, b_i are points of interest, each cut in $[a_i, b_i]$ belongs to a different subinterval, and therefore, there will be exactly n cuts in n distinct subintervals. Focusing now on our I_{SC} construction, the Square-path with $n-1$ turns consists of a total of n horizontal and vertical line segments. If any of those does not intersect any tile, then this Square-path will correspond to a set of at most $n-1$ cuts in $I_{\text{CH}}^{\text{DFMS}}$, which cannot be a solution. Therefore, every line segment intersects some tile.

Notice that, due to the diagonal placement of tiles, any Square-path that is a solution has to be positively x-monotone and positively y-monotone, i.e., to have a "staircase" form. Also, this diagonal placement of tiles dictates that, if there was a turn of the Square inside a tile, then two line segments are used to intersect it instead of one. This means that at most $n-1$ tiles will be intersected, and therefore this translates to a set of at most $n-1$ cuts in $I_{\text{CH}}^{\text{DFMS}}$, which cannot be a solution. □

Put differently, having a turn inside a tile would be a "waste", and in our instances, all turns are needed in order for a solution to exist. Now, if we focus on any tile t_j for some $j \in [m]$, as shown in the construction, it will contain exactly the same measure that the corresponding colour has inside the j-th subinterval of $I_{\text{CH}}^{\text{DFMS}}$. Furthermore, if the density is rectangular then the measure in the tile is square. Having a square on the entire region of the tile allows us to immediately translate a line segment of Square-path that intersects the square into a $I_{\text{CH}}^{\text{DFMS}}$ cut, since we translate in the same manner both horizontal and vertical such segments. Similarly, if the density is triangular, then so is the measure in the tile half of whose area it occupies, and the right angle of the triangle is at the top-right of the tile. This again allows us to translate any line segment, horizontal or vertical, straightforwardly into a cut of $I_{\text{CH}}^{\text{DFMS}}$, since the measure at the bottom or left, respectively, part of the triangle has exactly the same area as that of the corresponding triangle's part at the left of the $I_{\text{CH}}^{\text{DFMS}}$ cut.

The above analysis shows that, for any $i \in [n]$, in each individual tile the total "+" mass is equal to the total "+" value of the corresponding subinterval. Therefore, given a solution to I_{SC} where the total mass of R^+ will equal that of R^- for every $i \in [n]$, the induced cuts on I_{CH} constitute a solution. Finally, by construction of $I_{\text{CH}}^{\text{DFMS}}$, at any given point in $[0, 1]$, no more than three agents have positive valuation density, therefore, I_{SC} has overlap at most 3. Due to the FIXP-hardness of exact Consensus-Halving shown in [DFMS21], we get the following.

Theorem 19. Square-Pizza-Sharing *is* FIXP*-hard even when every mass distribution consists of at most six pieces that can be unit-squares or right-angled triangles, and have overlap at most 3.*

We can also show that deciding whether there exists an exact Square-Pizza-Sharing solution with $n-2$ turns is $\exists\mathbb{R}$-hard. For this, we will use a result of [DFMS21], where it was shown that deciding whether there exists an exact Consensus-Halving solution with n agents and $n-1$ cuts is $\exists\mathbb{R}$-hard. We give a reduction from this version of Consensus-Halving to the decision problem for Square-Pizza-Sharing. The reduction uses the same ideas that we presented for the FIXP-hardness reduction.

Before we prove the theorem, let us give a brief sketch of the $\exists\mathbb{R}$-hardness proof of the aforementioned Consensus-Halving decision variant of [DFMS21]. We are given an instance of the following problem which was shown to be $\exists\mathbb{R}$-complete (Lemma 15 of the aforementioned paper).

Definition 20 (Feasible$_{[0,1]}$)**.** *Let $p(x_1,\ldots,x_m)$ be a polynomial. We ask whether there exists a point $(x_1,\ldots,x_m)\in[0,1]^m$ that satisfies $p(x_1,\ldots,x_m)=0$.*

Given the polynomial p, we first normalize it so that the sum of the absolute values of its terms is in $[0,1]$ (thus not inserting more roots), resulting in a polynomial q. Then, we separate the terms that have positive coefficients from those that have negative coefficients, thus creating two positive polynomials q_1,q_2 such that $q=q_1-q_2$. Therefore, $p(\vec{x})=0$ for some $\vec{x}\in[0,1]^m$ if and only if $q_1(\vec{x})=q_2(\vec{x})$. We then represent q_1,q_2 in a circuit form with gates that implement the operations $\{c,+,\times c,\times\}$ (where $c\in[0,1]\cap\mathbb{Q}$ is a constant input, and $\times c$ is multiplication by constant). By the scaling we know that $q_1,q_2\in[0,1]$, and in addition, the computation of the circuit using the aforementioned operations can be simulated by a Consensus-Halving instance with $n-1$ agents, where $n-1\in\text{poly}(\#gates)$; the argument of p becomes a set of "input" cuts and according to the circuit implementation by Consensus-Halving, two output cuts encode q_1 and q_2. Finally, checking whether $q_1=q_2$ is true is done by an additional n-th agent that can only be satisfied (have her total valuation split in half) if and only if $q_1(\vec{x})=q_2(\vec{x})$ for some $\vec{x}\in[0,1]^m$. In other words, the Consensus-Halving instance has a solution if and only if there are "input" cuts $\vec{x}=(x_1,\ldots,x_m)\in[0,1]^m$ that force the rest of the cuts (according to the circuit implementation) that encode the values $v_{m+1},\ldots,v_{n-1}$ at the output of each of the circuit's gates such that they also cut the n-th agent's valuation in half without the need for an additional cut.

We are now ready to prove the following theorem.

Theorem 21. *It is $\exists\mathbb{R}$-hard to decide if an exact* Square-Pizza-Sharing *instance admits a solution with a* Square*-path with at most $n-2$ turns, even when every mass distribution consists of at most six pieces that can be unit-squares or right-angled triangles, and have overlap at most 3.*

Proof. We will use exactly the aforementioned technique from [DFMS21] up to the point where we have a Consensus-Halving instance that checks whether $q_1=q_2$. Then, we use the gadgets described in the FIXP-hardness reduction (proof of Theorem 19) that reduce the valuation functions of n agents in Consensus-Halving into mass distributions of a Square-Pizza-Sharing instance with n colours. According to Claim 18, in any solution of the resulting Square-Pizza-Sharing instance, the Square-path does not have turns inside any unit-square. This means that each horizontal/vertical segment of the Square-path that cuts a unit square in Square-Pizza-Sharing has a 1-1 correspondence to a cut of a Consensus-Halving solution, thus a Consensus-Halving solution that uses $n-1$ cuts would correspond to a Square-path with

$n-1$ line segments, i.e., $n-2$ turns. Therefore, if and only if there is a SQUARE-path that solves SQUARE-PIZZA-SHARING with n colours and $n-2$ turns, there is a $(n-1)$-cut that solves CONSENSUS-HALVING with n agents. Equivalently, there is a $\vec{x} \in [0,1]^m$ such that $p(\vec{x}) = 0$, making the FEASIBLE$_{[0,1]}$ instance satisfiable. □

4 Membership results

Up to this point, we have showed that STRAIGHT-PIZZA-SHARING and SQUARE-PIZZA-SHARING are PPA-hard for their approximate versions for any discrepancy $\varepsilon < 1/5$, even when the input consists of point sets. We have also showed that the decision variants of the problems are NP-hard. Furthermore, we have studied the exact version of SQUARE-PIZZA-SHARING and proved that it is FIXP-hard, while its decision variant is $\exists\mathbb{R}$-hard.

In this section, we present membership results for the exact and approximate versions of the aforementioned pizza sharing problems. Our PPA membership result for STRAIGHT-PIZZA-SHARING is achieved by reducing the problem to its discrete version, which was recently shown by [Sch21] to be in PPA (Theorem 22). For SQUARE-PIZZA-SHARING, our results revolve around the original existence proof of [KRPS16] which, additionally to the Borsuk-Ulam theorem, uses other involved topological techniques. We present a new algorithmic proof based on the original one, where now the only topology tool used is the Borsuk-Ulam theorem (Theorem 24). Then, by showing how to algorithmically compute the measure contained in the positive part of an arbitrary SQUARE-path (Appendix B), we turn this into a PPA membership proof (Theorem 27). Finally, we study the corresponding decision variants of the problems and acquire NP membership for the approximate continuous and discrete versions of the problems and $\exists\mathbb{R}$ membership of exact SQUARE-PIZZA-SHARING.

4.1 Membership results for approximate STRAIGHT-PIZZA-SHARING

Here we prove that ε-STRAIGHT-PIZZA-SHARING with $2n$ mass distributions is in PPA for any $\varepsilon \in \Omega(1/\text{poly}(N))$ and $\alpha \in \Omega(1/\text{poly}(N))$, where N is the input size and α is the smallest area among the triangles of the triangulated mass distributions of ε-STRAIGHT-PIZZA-SHARING. This answers a big open question left from [DFM22], where PPA membership was elusive. We also show that deciding whether a solution with at most $n-1$ straight lines exists is in NP. Both those results are derived by reducing the problem to its discrete version, namely DISCRETE-STRAIGHT-PIZZA-SHARING, where instead of mass distributions, the input consists of points, and the goal is to bisect (up to one point) each of the $2n$ point sets using at most n straight lines. DISCRETE-STRAIGHT-PIZZA-SHARING, was recently shown by [Sch21] to be in PPA.

PPA membership. We will use Lemma 13 in a straightforward way. In particular, given an ε-STRAIGHT-PIZZA-SHARING instance for some $\varepsilon \in \Omega(1/\text{poly}(N))$, we will pick an $\varepsilon' < \varepsilon$ as prescribed in the aforementioned lemma, and reduce our problem to DISCRETE-STRAIGHT-PIZZA-SHARING in time $\text{poly}(N, 1/\alpha)$.

Theorem 22. *ε-STRAIGHT-PIZZA-SHARING with $2n$ weighted mass distributions with holes is in PPA for any $\varepsilon \in \Omega(1/\text{poly}(N))$ and $\alpha \in \Omega(1/\text{poly}(N))$, where N is the input size and α is the smallest area among the triangles of the triangulated mass distributions.*

NP membership. Observe that Lemma 13 shows how to turn a set of $2n$ mass distributions (weighted polygons with holes) into a set of (unweighted) points, which, if cut by at most $2n$ straight lines, will result to an approximate cut of the mass distributions relaxed by an extra additive $\varepsilon' \in \Omega(1/N^c)$ for any $c > 0$. When the number of straight lines is at most $n-1$, this

gives straightforwardly a reduction to 0-Discrete-Straight-Pizza-Sharing, since we can check how many of the polynomially many points of the latter instance are in R^+ and R^-.

Theorem 23. *Deciding whether ε-Straight-Pizza-Sharing with $2n$ weighted mass distributions with holes has a solution with at most $n-1$ straight lines is in* NP *for any $\varepsilon \in \Omega(1/\text{poly}(N))$ and $\alpha \in \Omega(1/\text{poly}(N))$, where N is the input size and α is the smallest area among the triangles of the triangulated mass distributions.*

4.2 Membership results for approximate Square-Pizza-Sharing

Here we show that the problem of finding a solution to ε-Square-Pizza-Sharing is in PPA even for exponentially small ε, while deciding whether there exists a solution (a Square-path) with at most $k \in \mathbb{N}$ turns is in NP. The latter is almost immediate by the fact that any candidate solution is verifiable in polynomial time. The former is shown by reducing ε-Square-Pizza-Sharing to ε-Borsuk-Ulam which is in PPA (e.g., see [DFMS21]).

Existence of a Square-Pizza-Sharing solution. We begin by proving that a solution to exact Square-Pizza-Sharing always exists (and therefore, a solution to the approximate version exists too). This proof holds for arbitrary mass distributions, but for our algorithmic results, we will only consider the case where the mass distributions are unions of polygons with holes. Our proof is based on that of Karasev, Roldán-Pensado, and Soberón [KRPS16]. However, they use more involved techniques from topology, which we would like to avoid since our goal is to give an algorithmic proof.

Theorem 24 (originally by [KRPS16])**.** *Let n be a positive integer. For any n mass distributions in $\mathbb{R}^2$, there is a path formed by only horizontal and vertical segments with at most $n-1$ turns that splits $\mathbb{R}^2$ into two sets of equal size in each measure. Moreover, the path is y-monotone.*

Proof. Let S^n denote the L_1 sphere in $n+1$ dimensions. The Borsuk-Ulam theorem states that if $f : S^n \to \mathbb{R}^n$ is a continuous function, then there exists a point $\vec{z} \in S^n$ such that $f(\vec{z}) = f(-\vec{z})$. Consider n mass distributions in $\mathbb{R}^2$. For some given $d \in \mathbb{N}^*$, we will show how to decode Square-paths from points in S^d, and then we will construct a function $f : S^d \to \mathbb{R}^n$ for which $f(\vec{z}) = f(-\vec{z})$ implies that the Square-path corresponding to $\vec{z}$ is a solution to the Square-Pizza-Sharing problem. In particular, we will show a general way to decode Square-paths (with $d-1$ turns) from points in S^d for a suitable dimension $d \in \mathbb{N}^*$ to be determined later. It will turn out that, to guarantee a Square-Pizza-Sharing solution given n measures, the Borsuk-Ulam theorem will require $d = n$. However, we will have d undetermined for as long as we can, making the proof transparent enough to help in the understanding of the cases where $d \neq n$ (see Theorem 28, Theorem 32, and their proofs).

For the sake of simplicity, we normalize them so that they fit inside $[0,1]^2$ by scaling them down if needed, preserving their relative positions. Figure 6 gives an overview of our decoding to Square-paths, which actually results in y-monotone Square-paths. First, we will consider the case of even d, and then explain how the odd d case is proved. We will decode $\vec{z} := (z_0, z_1, \ldots, z_d) \in S^d$ into horizontal slices $y_1, y_3, \ldots, y_{d-1}$ and vertical cuts $x_1, x_3, \ldots, x_{d-1}$. For ease of presentation, let us also define $y_0 := 0$ and $y_{d+1} := 1$. We set $y_1 := |z_0|$, and $y_j := y_{j-2} + |z_{j-2}| + |z_{j-1}|$, for $j \in \{3, 5, \ldots, d+1\}$. It is immediate that $y_{d+1} = \sum_{j=0}^{d} |z_j|$, which equals 1 as it should (since $\vec{z} \in S^d$). So, the bottom *strip* $[y_0, y_1]$ has width $|z_0|$, and the strips $[y_j, y_{j+2}]$ for $j \in \{1, 3, \ldots, d-1\}$ have width $|z_j| + |z_{j+1}|$.

Let us focus on one such strip $[y_j, y_{j+2}]$ to define its respective x_j. We have the following cases. (i) If $|z_j| + |z_{j+1}| > 0$ and $z_j \cdot z_{j+1} \geq 0$, then $x_j \in \{0, 1\}$; in particular, if $z_j + z_{j+1} > 0$ then $x_j = 1$, and if $z_j + z_{j+1} < 0$ then $x_j = 0$. (ii) Otherwise, x_j has to satisfy the equation

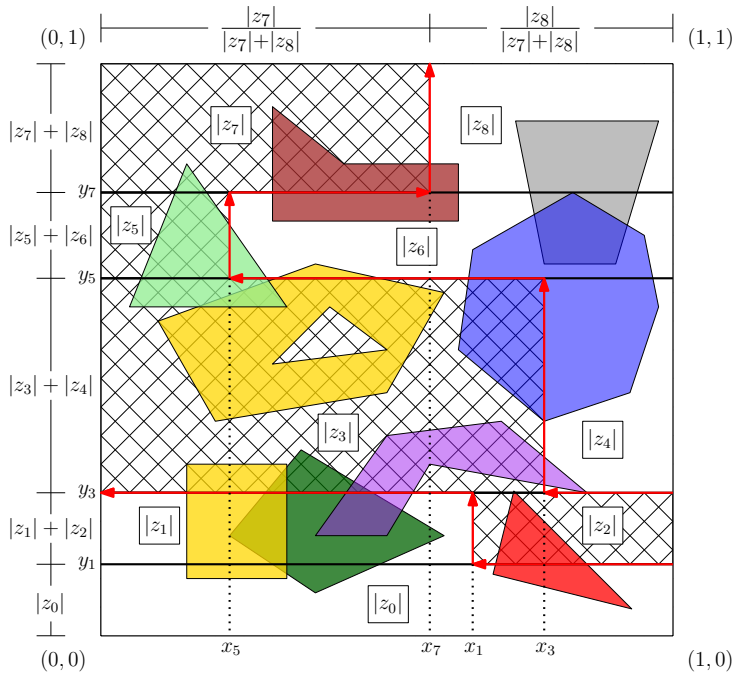


Figure 6: An instance with $n = 8$ polygon-shaped mass distributions and $d = 8$. A vector $(z_0, \ldots, z_8) \in S^8$ corresponds to horizontal slices and vertical cuts, which define a y-monotone SQUARE-path with 7 turns. Here, we have $z_0, z_1, z_4, z_6, z_8 > 0$ and $z_2, z_3, z_5, z_7 < 0$. For an example of the figure's notation, notice that in the top strip $[y_7, 1]$, the area to the left of x_7 is $|z_7|$ and to the right of x_7 is $|z_8|$.

$(|z_j|+|z_{j+1}|)\cdot x_j = |z_j|$; in other words, if $|z_j|+|z_{j+1}| > 0$ and $z_j \cdot z_{j+1} < 0$, then $x_j = \frac{|z_j|}{|z_j|+|z_{j+1}|}$, whereas if $|z_j|+|z_{j+1}| = 0$, then x_j can take any value in $[0, 1]$. Notice that in the latter subcase, $z_j = z_{j+1} = 0$, and the width of the strip is 0, so no matter what the value of x_j is, it will not contribute to the SQUARE-path's structure.

Finally, we need to label each of the (at most two) parts of each strip. In case (i) above, if $z_j + z_{j+1} > 0$ (resp. $z_j + z_{j+1} < 0$), then the whole strip belongs to R^+ (resp. R^-), depicted with a non-shaded (resp. shaded) area. In case (ii), if $z_j \geq 0$ and $z_{j+1} \leq 0$ (resp. $z_j \leq 0$ and $z_{j+1} \geq 0$), then the part of the strip to the left of x_j belongs to R^+ (resp. R^-) and that to the right of x_j belongs to R^- (resp. R^+). Similarly, for the strip $[y_0, y_1]$, if $z_0 > 0$ (resp. $z_0 < 0$) then the whole strip belongs to R^+ (resp. R^-), while if $z_0 = 0$ its width is zero and there is no need to specify where it belongs.

Now, having the slices y_j, the cuts x_j, and the labels (R^+ or R^-) of the slices they define, we can use Algorithm 1 to recover the underlying SQUARE-path.

What remains is the definition of the required Borsuk-Ulam function. For any given point $\vec{z} = (z_0, z_1, \ldots, z_d) \in S^d$, the Borsuk-Ulam function is defined to be the total "+" (positive) measure on $[0, 1]^2$ induced by $\vec{z}$, and we denote $f_i(\vec{z}) = \mu_i(R^+; \vec{z})$ for $i \in [n]$. The total positive measure $\mu_i(R^+; \vec{z})$ is a continuous function of its variables: by our definition of slices, cuts, and labelling, the area of R^+ is the sum of individual areas $|z_j|$ for those $j \in \{0, 1, 2, \ldots, d\}$ for which $z_j > 0$ (see non-shaded area of Figure 6); the slices y_j and the cuts x_j are continuous functions of the z_j's;[7] in each such individual area, the contained measure $j \in \{0, 1, \ldots, d\}$

[7]It is true that, e.g., for some fixed $z_{j+1} > 0$ and some moving $z_j \to 0^-$ we have $x_j \to 0^+$, while when $z_j = 0$, immediately $x_j = 1$. However, this is considered a continuous behaviour of x_j since it is allowed to wrap around in the horizontal dimension. Also, notice that this has no effect on the sign of any other individual area, creating

Algorithm 1 Mapping $(z_0, z_1, \ldots, z_d)$ to a SQUARE-path with $d-1$ turns

Input: A vector $(z_0, z_1, \ldots, z_d) \in S^d$.
Output: A SQUARE-path with $d-1$ turns.

Let $s_1 := |z_0|$, and $s_j := |z_{j-2}| + |z_{j-1}|$ for $j \in \{3, 5, \ldots, d+1\}$ (resp. $j \in \{3, 5, \ldots, d+2\}$) when d is even (resp. odd).
Find the set $T = \{t_1 \ldots, t_r\} \subseteq \{1, 3, \ldots, d+1\}$ (resp. $\{1, 3, \ldots, d+2\}$) when d is even (resp. odd), where $t_1 < \cdots < t_r$, and for each $\ell \in [r]$ it holds that $s_{t_\ell} > 0$.
if $z_0 > 0$ **then**
    create an artificial cut $x_0 = 1$ in strip $[y_0, y_1]$, and set $t_0 := 0$
if $z_0 < 0$ **then**
    create an artificial cut $x_0 = 0$ in strip $[y_0, y_1]$, and set $t_0 := 0$
if d odd **and** $z_d > 0$ **then**
    create an artificial cut $x_d = 1$ in strip $[y_d, y_{d+2}]$
if d odd **and** $z_d < 0$ **then**
    create an artificial cut $x_d = 0$ in strip $[y_d, y_{d+2}]$
Give an upward direction to all cuts within strips.
For any given $\ell \in [r]$, let $x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}$ denote the directed horizontal line segment belonging to slice y_{t_ℓ} that connects the head of cut $x_{t_{\ell-1}}$ and the tail of cut x_{t_ℓ}. Also, let $\overline{x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}}$ denote its complementary directed line segment with the same start and end points that wraps around the x-axis.
$\ell \leftarrow 1$
while $\ell \leq r$ **do**
    if $x_{t_{\ell-1}}, x_{t_\ell} \in (0,1)$ **and** $z_{t_{\ell-1}} \cdot z_{t_\ell} > 0$ **then**
        create $x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}$
    if $x_{t_{\ell-1}}, x_{t_\ell} \in (0,1)$ **and** $z_{t_{\ell-1}} \cdot z_{t_\ell} < 0$ **then**
        create $\overline{x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}}$
    if $x_{t_{\ell-1}}, x_{t_\ell} \in \{0,1\}$ **and** $x_{t_{\ell-1}} \neq x_{t_\ell}$ **then**
        create $x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}$
    if $x_{t_{\ell-1}} \in \{0,1\}$ **and** $x_{t_\ell} \in (0,1)$ **then**
        if $z_{t_\ell} > 0$ **then**
            create $x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}$
        if $z_{t_\ell} < 0$ **then**
            create $\overline{x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}}$
    if $x_{t_{\ell-1}} \in (0,1)$ **and** $x_{t_\ell} \in \{0,1\}$ **then**
        if $z_{t_{\ell-1}} > 0$ **then**
            create $x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}$
        if $z_{t_{\ell-1}} < 0$ **then**
            create $\overline{x_{t_{\ell-1}} y_{t_\ell} x_{t_\ell}}$
    $\ell \leftarrow \ell + 1$
Remove artificial cuts from strips $[y_0, y_1]$ and $[y_d, y_{d+2}]$ (odd d case), if any.

changes continuously with the boundaries of the area; also, one can easily check that in order for an individual area corresponding to z_j to change sign, z_j will have to pass from 0, and the magnitude $|z_j|$ of the area is a continuous function of z_j.

no discontinuities. Moreover, the only case where the cut of a strip can arbitrarily take values in $(0,1)$ is when for its components z_j, z_{j+1} we have $z_j = z_{j+1} = 0$, in which case the strip's width is also 0, and so it does not contribute to $\mu(R^+; \vec{z})$.

When d is odd, the decoding of $\vec{z} = (z_0, z_1, \ldots, z_d)$ into a SQUARE-path is similar. The horizontal slices are $y_1, y_3, \ldots, y_d$, and we set $y_0 := 0$ and $y_{d+2} := 1$. These define strips similarly to the even d case. The vertical cuts are $x_1, x_3, \ldots, x_{d-2}$, meaning that the bottom strip $[y_0, y_1]$ *and* the top strip $[y_d, y_{d+2}]$ are not vertically cut.

Also, it is easy to see that one could consider the path to be again y-monotone but in the opposite direction, meaning that there is no line segment pointing upwards.

Given n measures, if $d = n$ the Borsuk-Ulam theorem applies on f, ensuring that there exist two antipodal points $\vec{z}^*, -\vec{z}^* \in S^n$ such that $f(\vec{z}^*) = f(-\vec{z}^*)$. Notice that for any $i \in [n]$, $f_i(-\vec{z}) = \mu_i(R^-; \vec{z})$, since by flipping the signs of the variables of $\vec{z}$, we consider the "$-$" (negative) measure of $[0,1]^2$ induced by (the SQUARE-path of) $\vec{z}$. Therefore, when $f(\vec{z}^*) = f(-\vec{z}^*)$ we will have $\mu_i(R^+; \vec{z}^*) = \mu_i(R^-; \vec{z}^*)$ for every $i \in [n]$, that is, in each of the n measures, the positive total measure equals the negative one. Therefore, the SQUARE-path corresponding to $\vec{z}^*$ (see Algorithm 1) is a solution to SQUARE-PIZZA-SHARING. The total number of turns of the directed path is $d - 1 = n - 1$. □

Remark 25. *Note that a symmetric proof exists, where the slices are vertical instead of horizontal, and the cuts within the strips are horizontal instead of vertical. The analysis is similar to the one we give here, and it guarantees the existence of a* SQUARE*-path which is allowed to wrap around in the vertical dimension, it bisects all n measures and is x-monotone with either no line segment heading left or no line segment heading right.*

PPA membership. The following theorem shows PPA membership of ε-SQUARE-PIZZA-SHARING via a reduction to the ε-BORSUK-ULAM problem which is in PPA. The latter problem was introduced and shown to be in PPA by [Pap94] with its definition involving essentially a polynomial-time algorithm for the computation of the Borsuk-Ulam function. In [DFMS21], the definition of the problem uses an algebraic circuit as the representation of that function. The PPA membership is shown via a reduction to TUCKER (see [Pap94]), and for both versions of the ε-BORSUK-ULAM problem the reduction goes through. Here we state the most inclusive version of the problem.

Definition 26 (ε-BORSUK-ULAM)**.**

- ***Input:*** *$\varepsilon > 0$, and a continuous function $f : S^d \to \mathbb{R}^d$ whose Lipschitz constant is claimed to be λ. The function can be presented as either an algebraic circuit or a polynomial-time algorithm.*
- ***Task:*** *Find one of the following.*
  - *(a) Two points $x, y \in S^d$ such that $\|f(x) - f(y)\|_\infty > \lambda \cdot \|x - y\|_\infty$.*
  - *(b) A point $x \in S^d$ such that $\|f(x) - f(-x)\|_\infty \leq \varepsilon$.*

If the first task is accomplished, then we have found witnesses $x, y \in S^d$ that function f is not λ-Lipschitz continuous in the L_∞-norm as required.[8] But if the second task is accomplished then we have an approximate solution to the Borsuk-Ulam problem.

Theorem 27. *ε-SQUARE-PIZZA-SHARING for weighted polygons with holes is in* PPA*.*

[8] The λ-Lipschitzness of f is necessary for the correctness of the reduction from ε-BORSUK-ULAM to TUCKER in [DFMS21], where the latter problem is known to be in PPA. In particular, the reduction triangulates the S^d sphere such that the triangulation's vertices have distance at most $O(\varepsilon/\lambda)$, therefore, if f is not Lipschitz continuous (i.e., λ is unbounded), the solutions of TUCKER are not guaranteed to correspond to ε-BORSUK-ULAM solutions.

Proof. We will turn our existence proof of Theorem 24 into a polynomial time reduction from ε-Square-Pizza-Sharing to ε-Borsuk-Ulam in which the Borsuk-Ulam function is computable via a polynomial-time algorithm. Given the ε-Square-Pizza-Sharing instance with n mass distributions, we just have to construct the Borsuk-Ulam function $f : S^n \to \mathbb{R}^n$ using the procedure described in Appendix B, for $d = n$. The entire procedure involving the preprocessing part and the construction of f is a polynomial-time algorithm.

Furthermore, the function captures the R^+ part of each involved colour $i \in [n]$ by creating a Square-path as described in the proof of Theorem 24, where we showed that f is continuous. Also, it is easy to verify from the final step of the construction in Appendix B.2 that f is piece-wise polynomial with respect to $\vec{z}$, and therefore it is λ-Lipschitz continuous for $\lambda = \max_{j=0}^{n} \left\{ \sup_{\vec{z}} \left\| \frac{\partial f(\vec{z})}{\partial z_j} \right\|_\infty \right\}$ (and note that points where $f_i(\vec{z})$, $i \in [n]$ is non-differentiable do not matter for Lipschitzness). By the construction of f (Appendix B.2), one can see that λ is constant: the polynomial pieces of f are of degree at most 2, and the partial derivative of each f_i is determined by the rational points (given in the input) that define the polygons.

So far we have showed how to formulate any given instance of ε-Square-Pizza-Sharing as an ε-Borsuk-Ulam instance in polynomial time. What remains is to show how to turn a solution $\vec{z}^*$ of the latter to a solution of the former again in polynomial time. As we showed in the proof of Theorem 24, any $\vec{z} \in S^n$ can be efficiently translated into a Square-path using Algorithm 1. The Square-path corresponding to $\vec{z}^*$ (for which we have $\|f(\vec{z}^*) - f(-\vec{z}^*)\|_\infty \leq \varepsilon$) is the solution to ε-Square-Pizza-Sharing. To see this, notice that from the aforementioned proof we have $f_i(\vec{z}^*) = \mu_i(R^+; \vec{z}^*)$ and $f_i(-\vec{z}^*) = \mu_i(R^-; \vec{z}^*)$, therefore, $\|\mu_i(R^+; \vec{z}^*) - \mu_i(R^-; \vec{z}^*)\|_\infty \leq \varepsilon$. Finally, since $\vec{z}^* \in S^n$, the Square-path has at most $n - 1$ turns as required by an ε-Square-Pizza-Sharing solution. □

NP membership. Here we show that for any $k, n \in \mathbb{N}$, deciding whether there exists a solution for ε-Square-Pizza-Sharing with n measures and at most k turns is in NP. Notice that, for any such instance, we can verify a candidate solution in polynomial time. In particular, suppose we are given a Square-path with at most k turns that splits $[0, 1]^2$ into R^+ and R^- regions. Let the path be represented by the starting and ending points of its line segments, and the regions be defined by labels to the left and right of each vertical segment. Each measure $i \in [n]$ consists of a set of polygons with holes which can be preprocessed in polynomial time as described in Appendix B.1 to result in only axis-aligned right-angled triangles. Then, using the Square-path, the measures $\mu_i(R^+)$ and $\mu_i(R^-)$ (in a similar way) can be computed in polynomial time as described in Appendix B.2, where now $d = k + 1$. Finally, for the given ε, we can check whether $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$ is true for all $i \in [n]$. Therefore, we get the following.

Theorem 28. *Deciding whether there exists a* Square*-path with at most $k \in \mathbb{N}$ turns that is a solution of ε-*Square-Pizza-Sharing *with $n \in \mathbb{N}$ mass distributions is in* NP.

4.3 Membership results for discrete Square-Pizza-Sharing

It has already been shown in [Sch21] that ε-Discrete-Straight-Pizza-Sharing is in PPA even for $\varepsilon = 0$. We complete the picture regarding inclusion of discrete pizza sharing problems, by showing that ε-Discrete-Square-Pizza-Sharing is also in PPA for $\varepsilon = 0$, and therefore, for every $\varepsilon \in [0, 1]$. We will reduce ε-Discrete-Square-Pizza-Sharing to ε'-Square-Pizza-Sharing for $\varepsilon = 0$ and $\varepsilon' = 1/2N$, where N is the input size. Finally, as one would expect from the discrete version, its decision variant is in NP since its candidate solutions are verifiable in polynomial time.

PPA membership. Consider an instance of ε-Discrete-Square-Pizza-Sharing with point sets $P_1, \ldots, P_n$, and denote $P := P_1 \cup \cdots \cup P_n$. We intend to turn each point into a mass of non-zero area. To do so, we need to first scale down the landscape of the points, to create some excess free space as a "frame" around them. It suffices to scale down by an order of 3 and center it in the middle of $[0,1]^2$, that is, to map each point (x,y) to $\left(\frac{1}{3}+\frac{x}{3}, \frac{1}{3}+\frac{y}{3}\right)$. Now all our points are in $[1/3, 2/3]^2$. For convenience, for each $i \in [n]$, we will be still denoting by P_i the new set of points after scaling and centering.

Now, we check whether for any pair $i \neq j$ we have $P_{i,j} := P_i \cap P_j \neq \emptyset$, which means that two points of two point sets have identical positions. Consider all $P_{i,j} \neq \emptyset$ and let their union be P'. Now consider all points that do not belong in P', that is $R := P \setminus P'$. We want to find the minimum positive difference in the x- and y-coordinates between any pair of points in R. Let a point of R be denoted $p_t = (x_t, y_t)$, and let us denote

$$x_{\min} := \min_{\substack{p_a, p_b \in R \\ x_a \neq x_b}} |x_a - x_b|, \qquad \text{and by} \qquad y_{\min} := \min_{\substack{p_a, p_b \in R \\ y_a \neq y_b}} |y_a - y_b|,$$

and finally, $d := \min\{x_{\min}, y_{\min}\}$.

We now turn each point of P_i into an axis-aligned square of size $\frac{d}{3} \times \frac{d}{3}$, with its bottom-left corner having the point's coordinates. Notice that the total area of the squares is $|P_i| \cdot \frac{d^2}{9}$, therefore, by setting the weight of each square to $\frac{9}{|P_i| d^2}$ we have the full description of a mass distribution μ_i. Notice that, since $d \leq 1/3$, all of the mass distributions are in $[1/3 - 1/3 \cdot 3, 2/3 + 1/3 \cdot 3] = [2/9, 7/9]^2$.

Lemma 29. *Any* Square*-path that is a solution to the resulting* ε-Square-Pizza-Sharing *instance for* $\varepsilon = 1/2N$*, can be turned into a solution of* ε'-Discrete-Square-Pizza-Sharing *for* $\varepsilon' = 0$ *in polynomial time.*

Proof. If a horizontal (resp. vertical) line segment of the Square-path solution intersects two squares (of any two mass distributions), this means that their corresponding points in Discrete-Square-Pizza-Sharing had the same y- (resp. x-) coordinate. To see this, without loss of generality, suppose that the two squares are intersected by the same horizontal line segment, and that their corresponding points do not have the same y-coordinate. Then, the distance between their bottom-left corners is positive but no greater than $d/3$, which implies that $d \leq d/3$ (by definition of d), a contradiction. A symmetric argument holds for two squares that are intersected by the same vertical line segment. In both the above cases, we return as a solution to 0-Discrete-Square-Pizza-Sharing the two corresponding points of the squares, which are of the kind of "Output (a)" in Definition 12.

Let $P_{\max} := \max_{i \in [n]} |P_i|$. Suppose $|P_i|$ is odd for every $i \in [n]$. Then, since we are asking for an $1/2N$-Square-Pizza-Sharing solution, its Square-path cannot be non-intersecting with any of the squares, otherwise $|\mu_i(R^+) - \mu_i(R^+)| \geq 1/|P_i| \geq 1/P_{\max} > 1/2P_{\max} \geq 1/2N$, a contradiction. Therefore, at least one square of P_i is intersected, and this holds for every $i \in [n]$. If no line segment of the Square-path intersects two squares, we conclude that the Square-path with $n-1$ turns and n line segments will intersect at most n squares. But since, as discussed above, at least one square of each mass distribution has to be intersected by Square, we get that Square intersects exactly one square of each mass distribution. Each side, R^+ and R^- of the Square-path includes at least $\lfloor |P_i|/2 \rfloor$ entire squares for every $i \in [n]$, and therefore, their bottom-left corners, i.e., the corresponding points of Discrete-Square-Pizza-Sharing. This is a solution to the 0-Discrete-Square-Pizza-Sharing instance.

Now suppose $|P_i|$ is even for some $i \in [n]$. We can remove an arbitrary square from all mass distributions that come from point sets with even cardinality, and perform the aforementioned

reduction to $1/2N$-Square-Pizza-Sharing. Then, let us call "i-th segment" the one that intersects one square of P_i, called i-th square, and let it be a vertical segment, without loss of generality. By placing back the removed square, its bottom-left corner will be: (i) either on opposite sides with that of the i-th square, (ii) or on the same side (notice that due to the allowed discrepancy $1/2N < 1/2P_{\max}$, the i-th segment cannot fall on the bottom-left corner of the i-th square). Then, in case (i) each side contains exactly $|P_i|/2$ bottom-left corners of squares (i.e., points). By scaling up the positions of Square-path's segments (recall that we have scaled down), this is a solution to the 0-Discrete-Square-Pizza-Sharing instance. In case (ii), suppose without loss of generality that both the inserted square and the bottom-left corner of the i-th square are on the left side of the i-th segment. We modify the Square-path by shifting the i-th segment to the left such that it is now located $d/3$ to the left of i-th square's bottom-left corner. Notice that this position is to the right of the inserted square since there is at least $2d/3$ distance between the two squares. We do the same for every $i \in [n]$ has even number of points/squares. Then, each side of the Square-path for every $i \in [n]$ has exactly $|P_i|/2$ bottom-left corners of squares which represent the initial points. After scaling up the positions of the Square-path's segments, this is a solution to 0-Discrete-Square-Pizza-Sharing.

Finally, it is clear that the aforementioned operations can be performed in poly(N) time. □

By the PPA membership of $1/2N$-Square-Pizza-Sharing (see Theorem 27), we get the following.

Theorem 30. ε-Discrete-Square-Pizza-Sharing *is in* PPA *for any* $\varepsilon \in [0, 1]$.

NP membership. It is also easy to see that, by checking whether each of the points of each P_i is in R^+ or R^- as defined by a candidate Square-path solution, we can decide in polynomial time if indeed it is a solution or not to 0-Discrete-Square-Pizza-Sharing (since the points are polynomially many in N, by definition).

Theorem 31. *Deciding whether there exists a* Square*-path solution to* ε-Discrete-Square-Pizza-Sharing *is in* NP *for any* $\varepsilon \in [0, 1]$.

4.4 $\exists\mathbb{R}$ membership for the decision variant of exact Square-Pizza-Sharing

Here we show that deciding whether there exists an exact Square-Pizza-Sharing solution (Square-path) with at most k turns and n measures is in $\exists\mathbb{R}$, for any $k, n \in \mathbb{N}$. To do so, we turn our decision problem into an ETR formula in polynomial time. As discussed in Section 2, an ETR expression has the form: $\exists\vec{P} \in \mathbb{R}^m \cdot \Phi$, where Φ is a Boolean formula using connectives $\{\wedge, \vee, \neg\}$ over polynomials with domain $\mathbb{R}^m$ for some $m \in \mathbb{N}$ compared with the operators $\{<, \leq, =, \geq, >\}$. The ETR problem is to decide whether there is a truth assignment $\vec{P} \in \mathbb{R}^m$ that satisfies Φ.

We will use the proof of Theorem 24 and the explicit construction of the Borsuk-Ulam function from Appendix B. Recall that the aforementioned sections provide a polynomial time algorithm that takes the problem's input, i.e., n sets of polygons with holes, and computes a Borsuk-Ulam function, $f : S^d \to \mathbb{R}^n$, for some given $d \in \mathbb{N}^*$. This is done by first mapping any point $\vec{z} \in S^d$ to a Square-path with at most $d-1$ turns, and then computing f_i, $i \in [n]$, which captures the i-th measure's intersection with the R^+ region (where the R^+, R^- regions have been determined by the Square-path). In Appendix B.2 we showed how to explicitly construct f in polynomial time and showed that it is piece-wise polynomial.

Now notice that for every $i \in [n]$, $f_i(\vec{z}) = \mu_i(R^+; \vec{z})$, therefore, $f_i(\vec{z}) = f_i(-\vec{z})$ is equivalent to $\mu_i(R^+; \vec{z}) = \mu_i(R^+; -\vec{z}) = \mu_i(R^-; \vec{z})$, where the last equality comes by the definition of the

Square-path decoded from $\vec{z}$. In other words, $\vec{z} \in S^d$ is a solution to $f(\vec{z}) = f(-\vec{z})$ if and only if its corresponding Square-path is a solution to $\mu(R^+; \vec{z}) = \mu(R^-; \vec{z})$. Let us set $d = k + 1$. What remains is to turn the decision problem of whether such a $\vec{z}$ exists into an ETR formula.

We start by replacing the domain $\vec{z} \in S^{k+1}$ with $\vec{Z} \in \mathbb{R}^{k+2}$ and adding in the ETR formula the constraint $\sum_{j=0}^{k+1} |Z_j| = 1$. Then, we turn all the aforementioned polynomial time computations (that result to f) into ETR form. We can use a standardized method to do so in a generic manner by efficiently expressing in ETR form any computation belonging to NP. In particular, it is clear that any such computation can be turned in polynomial time into a Boolean satisfiability (SAT) formula, and sequentially transform it into a CNF formula. To turn this formula into an ETR expression is easily done in the following way (e.g., see [BPR06]). Consider the CNF formula B over $m \in \mathbb{N}$ Boolean variables $\{x_1, \ldots, x_m\}$. This can be turned in polynomial time into an equisatisfiable ETR formula: $\exists \vec{X} \in \mathbb{R}^m \cdot \bigwedge_{i=1}^{m}((X_i = 0) \vee (X_i = 1)) \wedge B'$, where B' is constructed in the following way. For each $i \in [m]$, let $y_i \in \{x_i, \neg x_i\}$ be a literal in B. We turn all disjunctions $y_j \vee y_k \vee \cdots \vee y_\ell$ of B into the inequality $Y_j + Y_k + \cdots + Y_\ell > 0$ in B', where for each $i \in [m]$, $Y_i = X_i$ if $y_i = x_i$ and $Y_i = 1 - X_i$ if $y_i = \neg x_i$. Finally, the auxiliary variables F_i, F'_i, $i \in [n]$, contain the values of $f_i(\vec{z})$'s and $f_i(-\vec{z})$'s computed from B'.

The above induce the following ETR expression, which is true if and only if there is an exact solution (Square-path) with at most k turns for the Square-Pizza-Sharing problem with n measures.

$$\exists(\vec{Z}; \vec{X}; \vec{F}) \in \mathbb{R}^{k+2+m+2n} \cdot \left(\sum_{j=0}^{k+1} |Z_j| = 1 \right) \wedge \bigwedge_{i=1}^{m} ((X_i = 0) \vee (X_i = 1)) \wedge B' \wedge \left(\bigwedge_{i=1}^{n} F_i = F'_i \right).$$

This gives us the following theorem.

Theorem 32. *Deciding whether there exists a* Square*-path with at most $k \in \mathbb{N}$ turns that is an exact solution for* Square-Pizza-Sharing *with $n \in \mathbb{N}$ mass distributions is in $\exists\mathbb{R}$.*

5 Conclusions

For ε-Straight-Pizza-Sharing we have shown that finding a solution is PPA-complete for any $\varepsilon \in [1/N^c, 1/5)$, where N is the input size and $c > 0$ is a constant. This result holds for both its continuous (even when the input contains only axis-aligned squares) and its discrete version. We have also shown that the same result holds for ε-Square-Pizza-Sharing, where the PPA membership holds even for inverse exponential ε. One open question that remains is "Can we prove PPA membership of ε-Straight-Pizza-Sharing for inverse exponential ε?". For the decision variant of both these problems, we show that there exists a small constant ε such that they are NP-complete. For both these problems and their search/decision variants, a big open question is "What is the largest constant ε for which the problem remains PPA-hard and NP-hard, respectively?". The most interesting open question is "Are there any algorithms that guarantee a solution in polynomial time for some constant $\varepsilon \in [1/5, 1)$, even when slightly more lines (resp. turns in a Square-path) are allowed?".

We have also shown that exact Square-Pizza-Sharing is FIXP-hard. The interesting question that needs to be settled are "Is the problem in the class BU (defined in [DFMS21]) similarly to exact Consensus-Halving?", and "For what complexity class is the problem complete?". Schnider in [Sch21] showed that exact Straight-Pizza-Sharing is FIXP-hard for a more general type of input than ours. So, some natural questions are "When the input consists of weighted polygons with holes, is the problem FIXP-hard and/or inside BU?", and "Is it complete for any of the two classes?". For a strong approximation version of Consensus-Halving, [BHH21]

showed that the problem is BU_a-complete. We conjecture that the same holds for the two pizza sharing problems studied here.

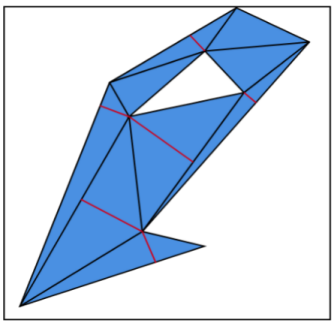
In the Square-Pizza-Sharing problem, a path is allowed to wrap around on either the x-axis or the y-axis, suggesting a cylindrical shape of the underlying space. It would be intriguing to study the case of a torus or even that of a plane. Another couple of questions that remain open are: "What is the complexity of ε-Straight-Pizza-Sharing and ε-Square-Pizza-Sharing when every mass distribution consists of a constant number of non-overlapping rectangles?", and finally, "What is the complexity of the pizza sharing problems when we ask to fairly split the plane into more than two equal parts?".

A Proof of Lemma 13

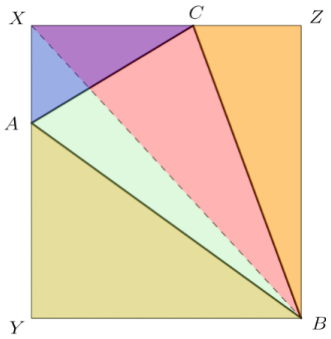
Pixelation. We will start with the task of pixelating the mass distributions. Consider the input of an ε-Square-Pizza-Sharing or an ε-Straight-Pizza-Sharing instance, that is, $q \in \{n, 2n\}$ mass distributions, respectively, on $[0,1]^2$ consisting of weighted polygons with holes (see Section 2 for details on the input representation). Let the instance's input size be $N \geq 2n$ (by definition). As a first step, we perform a *pixelation* procedure: each polygon will be turned into a union of smaller squares that will have approximately the same total area as the polygon.

As shown in Appendix B.1, it is possible to decompose a polygon into a union of disjoint non-obtuse triangles (Proposition 38). Each of those triangles' area is rational since it can be computed by adding and subtracting the areas of five right-angled triangles with rational coordinates, which additionally, are axis-aligned (Proposition 39). Furthermore, the cardinality of the non-obtuse triangles is a polynomial function in the input size of the polygon's description, that is, the coordinates of the points that define its corners and the value that defines its weight. Therefore, the exact area of any mass distribution can be computed in polynomial time.

As we have discussed earlier, in order for our approximation parameter $\varepsilon \in [0,1]$ to make sense, we consider normalized mass distributions, meaning that $\mu_i\left([0,1]^2\right) = 1$ for all $i \in [q]$. Notice that it can be the case that some mass distributions have total area constant (in which case their weight is constant), while some others might have total area exponentially small (and therefore exponentially large weight). Therefore, in our analysis, we make sure that the "resolution" we provide to any polygon F is *relative* to its actual area $\text{area}(F)$ rather than its measure $\mu_i(F)$.



(a) The triangulation with only non-obtuse triangles. After the standard triangulation, extra line segments (in red colour) are added to ensure non-obtuseness.



(b) A non-obtuse triangle $\widehat{ABC}$ and its decomposition into axis-aligned right-angled triangles: $\text{area}(\widehat{ABC}) = \text{area}(\widehat{XYB}) + \text{area}(\widehat{XBZ}) - \text{area}(\widehat{AYB}) - \text{area}(\widehat{XAC}) - \text{area}(\widehat{CBZ})$.

Figure 7

Consider some polygon F of mass distribution μ_i for $i \in [q]$ and let it be triangulated into non-obtuse triangles, all with strictly positive area. We will focus on one of F's non-obtuse triangles, $\widehat{ABC}$ (see Figure 7b) with area $S := \text{area}\left(\widehat{ABC}\right) > 0$ and perimeter $T > 0$. Notice that since $\widehat{ABC}$ is in $[0,1]^2$, we have $S \leq 1$ and $T \leq 3 \cdot \sqrt{2} < 5$. Suppose that among all triangles of all μ_i's, the minimum area triangle has area α. The first step is to pixelate $\widehat{ABC}$. Let our *pixels* be thought of as squares of size $t \times t$ for $t := \frac{1}{\lceil 15N^{1+c}/\alpha \rceil}$, where $c > 0$ is any fixed constant. In other words, consider an axis-aligned square grid on $[0,1]^2$ with edge length t, where the closed region defined by four edges is called a pixel (see Figure 8a for a depiction). We create a pixel for $\widehat{ABC}$ if and only if the pixel's intersection with $\widehat{ABC}$ has strictly positive area. Then, the pixelated version of the triangle, denoted $\widehat{ABC}_p$, has area $S + S'$, where S' is the excess area induced by the pixels intersected by the three sides of the triangle. By definition, S' is at least 0, and at most the area of pixels that intersect the three sides of $\widehat{ABC}$. Therefore, by denoting the number of such pixels for each side by n_{AB}, n_{BC}, n_{CA} and referring to Figure 7b, we have $n_{AB} \leq \left\lceil \frac{AY}{t} \right\rceil + 1 + \left\lceil \frac{YB}{t} \right\rceil + 1 \leq \frac{AY}{t} + \frac{YB}{t} + 4$, and similarly for n_{BC}, n_{CA}. Now also notice that $t \leq \frac{\alpha}{15N^{1+c}} \leq \frac{S}{15} < \frac{S}{3T}$, where the second inequality comes from the fact that $S \geq a$ by definition, and the third one is due to $T < 5$ as argued above. Then, we also have to use the known formula that connects S and T, namely,

$$S = \sqrt{\frac{T}{2}\left(\frac{T}{2} - AB\right)\left(\frac{T}{2} - BC\right)\left(\frac{T}{2} - CA\right)} < \sqrt{\left(\frac{T}{2}\right)^4} = \frac{T^2}{4},$$

which implies $\frac{S}{T} < \frac{T}{4}$, and therefore $t < \frac{T}{12}$.

Putting everything together, we get

$$\begin{aligned}
S' &\leq t^2 \cdot (n_{AB} + n_{BC} + n_{CA}) \\
&\leq t^2 \cdot \left(\frac{AY}{t} + \frac{YB}{t} + \frac{BZ}{t} + \frac{ZC}{t} + \frac{CX}{t} + \frac{XA}{t} + 12\right) \\
&\leq t \cdot ((AY + YB) + (BZ + ZC) + (CX + XA) + 12t) \\
&\leq t \cdot (2 \cdot AB + 2 \cdot BC + 2 \cdot CA + 12t) \qquad \text{(since } AB, BC, CA \text{ are hypotenuses)} \\
&= t \cdot (2 \cdot T + 12t) \qquad \text{(since } t < \frac{T}{12}\text{)}, \\
&< 3 \cdot T \cdot t \\
&< \frac{\alpha}{N^{1+c}} \qquad \text{(since } T < 5\text{)}. \qquad (1)
\end{aligned}$$

We want to bound the proportion of excess area due to the pixelation compared to the triangle's actual area, that is, S'/S. We have

$$\frac{S'}{S} < \frac{\alpha}{S \cdot N^{1+c}},$$

which, together with he fact that $S \geq \alpha$ (by definition of α), gives the following.

Claim 33. *The pixelation of $\widehat{ABC}$ results in $\widehat{ABC}_p$, where area$\left(\widehat{ABC}_p\right) < \left(1 + \frac{\alpha}{S \cdot N^{1+c}}\right) \cdot$ area$\left(\widehat{ABC}\right)$. In particular, area$\left(\widehat{ABC}_p\right) < \left(1 + \frac{1}{N^{1+c}}\right) \cdot S$*

Consider a straight line ℓ that cuts $\widehat{ABC}_p$, splitting it into two shapes L_p and R_p with areas area(L_p) and area(R_p), respectively. Let the corresponding two shapes that ℓ creates when

intersecting $\widehat{ABC}$ be L and R, with areas area(L) and area(R), respectively. Also, let M be the set of pixels of $\widehat{ABC}_p$ that are intersected by ℓ, and M_L, M_R be a partition of M. When clear from the context, we will slightly abuse the notation by denoting D the set of points in the union of squares on $[0,1]^2$ corresponding to pixel set D.

Claim 34. *The total area of M is at most $\alpha/5N^{1+c}$.*

Proof. We start from the easy observation that the length of $\ell \cap \widehat{ABC}_p$ is at most $\sqrt{2} < 2$ since we are in $[0,1]^2$. Therefore, the number of pixels that ℓ intersects is at most $\frac{2}{t}$ resulting to their area being at most $t^2 \cdot \frac{2}{t} = 2t < \frac{\alpha}{5N^{1+c}}$. □

Claim 35. *For any disjoint M_L, M_R with $M_L \cup M_R = M$, we have $|(area(L) - area(R)) - (area(L_p \cup M_L) - area(R_p \cup M_R))| \leq 2\alpha/N^{1+c}$.*

Proof. It suffices to show that $0 \leq \text{area}(L_p \cup M_L) - \text{area}(L) \leq 2\alpha/N^{1+c}$. The first inequality is easy to see, since $L \subseteq L_p \cup M_L$, which implies $\text{area}(L) \leq \text{area}(L_p \cup M_L)$. For the second inequality, we have

$$\begin{aligned}
\text{area}(L) &> \frac{\text{area}(L_p)}{1 + \alpha/SN^{1+c}} && \text{(by Claim 33)} \\
&= \text{area}(L_p) \cdot \left(1 - \frac{\alpha}{SN^{1+c} + \alpha}\right) \\
&\geq (\text{area}(L_p \cup M_L) - \text{area}(M)) \cdot \left(1 - \frac{\alpha}{SN^{1+c}}\right) \\
&\geq \text{area}(L_p \cup M_L) - \frac{\alpha}{5N^{1+c}} - \frac{\alpha}{SN^{1+c}} \cdot \left(\text{area}(L_p \cup M_L) - \frac{\alpha}{5N^{1+c}}\right) && \text{(by Claim 34)} \\
&\geq \text{area}(L_p \cup M_L) - \frac{\alpha}{5N^{1+c}} - \frac{\alpha}{SN^{1+c}} \cdot \text{area}\left(\widehat{ABC}_p\right) \\
&> \text{area}(L_p \cup M_L) - \frac{\alpha}{5N^{1+c}} - \frac{\alpha}{SN^{1+c}} \cdot S\left(1 + \frac{1}{N^{1+c}}\right) && \text{(by Claim 33)} \\
&\geq \text{area}(L_p \cup M_L) - \frac{\alpha}{5N^{1+c}} - \frac{3}{2}\frac{\alpha}{N^{1+c}} && \text{(since } N \geq 2\text{)} \\
&\geq \text{area}(L_p \cup M_L) - \frac{2\alpha}{N^{1+c}}.
\end{aligned}$$

Similarly, $0 \leq \text{area}(R_p \cup M_R) - \text{area}(R) \leq 2\alpha/N^{1+c}$, or equivalently, $-2\alpha/N^{1+c} \leq -(\text{area}(R_p \cup M_R) - \text{area}(R)) \leq 0$. Therefore,

$$-2\alpha/N^{1+c} \leq (\text{area}(L_p \cup M_L) - \text{area}(L)) - (\text{area}(R_p \cup M_R) - \text{area}(R)) \leq 2\alpha/N^{1+c}$$

□

Recall that, in an ε-Straight-Pizza-Sharing solution, at most $2n$ lines can intersect $\widehat{ABC}$ and $\widehat{ABC}_p$ (even though the standardized version of the problem requires at most n straight lines, as we showed in Theorem 6, PPA-hardness holds even for at most $n + n^{1-\delta}$ lines for any constant $\delta \in (0,1]$). Similarly for an ε-Square-Pizza-Sharing solution, since its Square-path comprises of at most $n-1$ turns, i.e., n straight line segments (and again, by Theorem 9 PPA-hardness holds even for at most $n + n^{1-\delta}$ line segments for any constant $\delta \in (0,1]$) By inductively applying Claim 34 and Claim 35 N times (recall that $N \geq 2n$), we get the following.

Lemma 36. *Let at most $2n$ straight lines intersect $\widehat{ABC}_p$, and the side of each pixel be $\alpha/15N^{1+c}$ for any $c > 0$. Also, let M be the set of its pixels that are intersected by the lines, and M_L, M_R be an arbitrary partition of M. Then, area(M) $\leq \alpha/5N^c$, and furthermore, $|(area(L) - area(R)) - (area(L_p \cup M_L) - area(R_p \cup M_R))| \leq 2\alpha/N^c$.*

Turning pixels into points in general position with unique x- and y-coordinates. So far, for ε-Square-Pizza-Sharing and ε-Straight-Pizza-Sharing, with $q \in \{n, 2n\}$ mass distributions, respectively, we have described how to turn each distribution μ_i, $i \in [q]$ into a pixelated version of it. We will now turn each of those pixels into a set of points. Let μ_i consist of b many polygons, and recall that each polygon has its own weight $w_{i,j} > 0$, $j \in [b]$. Let $T^{i,j} \in \mathcal{F}_i$ be a non-obtuse triangle belonging to the j-th polygon of μ_i, and $\mathcal{F}_i$ be the set of these triangles composing μ_i. Suppose a pixel contains triangles (whose area is strictly positive) $\{T^{i,k}\}_{k \in D}$ for some $D \subseteq [b]$, and let us denote $w^i_{\max} := \max_{k \in D} w_{i,k}$. Observe that, due to our assumption that the mass distributions are normalised, we have $1 = \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \text{area}(T^{i,j}) \geq \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \alpha$, therefore,

$$w^i_{\max} \leq \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \leq 1/\alpha. \tag{2}$$

We will place $\lceil w^i_{\max} \cdot N^c \rceil$ points at the pixel's bottom-left corner, that is, all having the same position. Each of the pixels of mass distribution i has no weight, as desired, and they form a set P_i. Notice that each of the P_i's we created contains at most $\frac{2N^c}{t^2\alpha} \leq \frac{512N^{2+3c}}{\alpha^3}$ points, i.e., polynomially many in the instance's description size and $1/\alpha$. That is because its pixels can be at most $\frac{1}{t} \cdot \frac{1}{t} \leq \left(\frac{15N^{1+c}+1}{\alpha}\right)^2 \leq \left(\frac{16N^{1+c}}{\alpha}\right)^2 \leq \frac{256N^{2+2c}}{\alpha^2}$ many, with each pixel containing at most $\lceil w^i_{\max} \cdot N^c \rceil \leq \frac{N^c}{\alpha} + 1 \leq \frac{2N^c}{\alpha}$ points.

Observe, however, that in the discrete version of the pizza sharing instance we created, the points of the $q \in \{n, 2n\}$ point sets lie on vertices of a square grid with edge length $t = \frac{1}{\lceil 15N^{1+c}/\alpha \rceil}$. These points are not guaranteed to be in general position or with unique x- and y-coordinates, and therefore, not all solutions of that instance can be translated back to a solution of the original corresponding continuous version of the instance. For the rest of this section, we will show how to turn this instance into one where the points in $P_1 \cup \cdots \cup P_q$ are in general position and with unique x- and y-coordinates. Additionally, we will ensure that each point remains in its original pixel.

For each pixel, let us create a $k \times k$ square grid with edge length $\frac{t}{k+1}$, where $k := \frac{48n^2N^{2c}}{t^6\alpha^2}$, which is placed at the center of each pixel. The purpose is to place each point of $P_1 \cup \cdots \cup P_q$ belonging to a pixel on the pixel's corresponding grid, such that all the aforementioned points are in general position and have unique x- and y-coordinates.

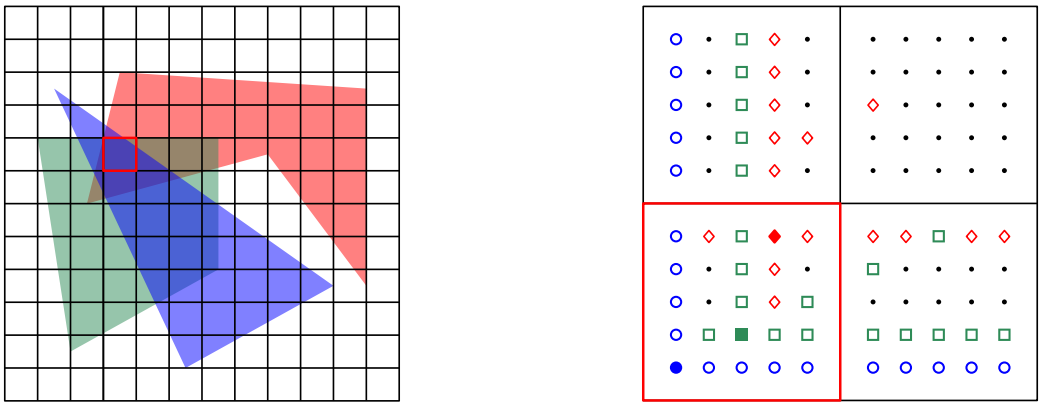
As shown above, for every $i \in [q]$, $|P_i| \leq \frac{2N^c}{t^2\alpha}$. Therefore, the points that a pixel can contain are at most $q \cdot \frac{2N^c}{t^2\alpha} \leq \frac{4nN^c}{t^2\alpha} =: G$ many. Suppose now we place the points of the pixel in distinct vertices of the corresponding grid.

1. Any pair of those defines a line, and to satisfy the "general position" condition, that line should not intersect any other point that is placed on any other pixel's grid. There are at most $\binom{G}{2}$ many such lines. For each line, we will forbid the placement of other points on it by removing the grid vertices it intersects throughout all pixels' grids. This means that, for each line, at most k grid vertices have to be removed due to a pair of points of a single pixel. Therefore, overall, at most $\binom{G}{2} \cdot k \cdot \frac{1}{t^2}$ grid vertices have to be removed to satisfy the "general position" condition.

2. Each point defines one horizontal and one vertical line that intersects it. To satisfy the "uniqueness of x- and y-coordinates" condition, we have to forbid any other point from being placed on these two lines. To do so, it suffices that among all pixels' grids we remove the vertices that are intersected by these two lines. Each line removes at most k grid vertices in a single pixel, so overall, at most $G \cdot 2 \cdot k \cdot \frac{1}{t^2}$ grid vertices have to be removed to satisfy the "uniqueness of x- and y-coordinates" condition.

In total, at most $\binom{G}{2} \cdot k \cdot \frac{1}{t^2} + G \cdot 2 \cdot k \cdot \frac{1}{t^2} = \frac{k}{t^2}\frac{G^2+3G}{2} \leq \frac{2kG^2}{t^2}$ grid vertices have to be removed from each pixel's grid in order to respect the above conditions. For each pixel's grid, since it initially contained k^2 vertices, its remaining vertices that can be used for points of the pixel to be placed on are at least $k^2 - \frac{2kG^2}{t^2} = \frac{9G^4}{t^4} - \frac{6G^4}{t^4} = \frac{3G^4}{t^4} \geq G$, where the first equality comes by our choice of $k := \frac{48n^2N^{2c}}{t^6\alpha^2} = \frac{3G^2}{t^2}$. Recall that each pixel contains at most G points, therefore there are enough vertices for them to be placed on.

So far we have shown that we can slightly perturb each point from the bottom left corner of a pixel so that it remains in the pixel, and furthermore, all points are in general position with unique x- and y-coordinates. It remains to show that we can do this perturbation in polynomial time. Indeed, it is easy to check that the following procedure achieves this task and requires only polynomially many steps: choose an arbitrary pixel and an arbitrary grid vertex in it to place one of the pixel's points on; from all pixels, remove all other grid vertices which have the same x- or y-coordinate with any of the pair's points (this can be done in polynomial time by exhaustively checking each of the k^2/t^2 grid vertices); next, while there is still a point of that pixel which has not been placed on its grid, place it on one of the non-removed vertices of its grid, then remove every other vertex from all pixels if (i) they have the same x- or y-coordinate, or (ii) if they are colinear with the pair formed by the new point and any of the older points (this again can be done by checking colinearity in polynomial time for all grid vertices); repeat the previous step for all pixels' points until all points from all pixels (polynomially many) have been placed on the respective grids.



(a) An example of the pixelated masses. All three masses overlap on the red pixel.

(b) The bottom-left pixel corresponds to the red pixel of Figure 8a.

Figure 8: Turning pixels into points in general position with unique x- and y-coordinates. In Figure 8b, three of the red pixel's points are placed one by one in its grid. First, the blue point (shaded disk) is placed on an arbitrary vertex; this "forbids" all points that follow to be placed on the blue hollow disks. Then the green point (shaded square) is placed on an arbitrary "non-forbidden" vertex; this, in turn, "forbids" the green hollow squares. Finally, the red point (shaded rhombus) is placed on one of the remaining vertices; this "forbids" the red hollow rhombuses. The black dots denote "non-forbidden" vertices where the next available point from the corresponding pixel can be placed on.

Lemma 37. *The perturbed points are in general position, they have unique x- and y-coordinates, and each has remained in its initial pixel.*

We are now ready to prove the lemma.

Proof of Lemma 13. Consider the input of an ε-Square-Pizza-Sharing or an ε-Straight-

Pizza-Sharing instance, meaning, the description of $q \in \{n, 2n\}$ sets, respectively, of weighted polygons with holes on $[0,1]^2$. By definition, its size is $N \in \Omega(n)$. In time $\text{poly}(N)$, we perform a triangulation of each polygon into non-obtuse triangles, therefore, in total we require $\text{poly}(N)$ time for this task. Then, we perform the "pixelation" procedure, which requires, for each of q mass distributions, checking whether each of $1/t^2 = \lceil 225N^{2+2c}/\alpha^2 \rceil$ pixels has a non-empty intersection with a triangle. This task can be performed in $\text{poly}(N, 1/\alpha)$ time, since $c > 0$ is a fixed constant. Next, the points of each point set P_i created from the respective pixels are perturbed so that they are in general position, and they have unique x- and y-coordinates (Lemma 37). This procedure has as a result that Output (a) of Definition 11 and Definition 12 cannot be produced. Finally, notice that the number of points to be fairly divided is $\text{poly}(N, 1/\alpha)$.

We claim that the lines $\ell_1, \ldots, \ell_m$ for some $m \leq 2n$ that are a solution to ε-Discrete-Straight-Pizza-Sharing or the line segments of the Square-path solution of ε-Discrete-Square-Pizza-Sharing (recall that in Theorem 6 and Theorem 9 we allowed almost $2n$ lines and line segments, respectively) are also a solution to $(\varepsilon - \varepsilon')$-Straight-Pizza-Sharing and $(\varepsilon - \varepsilon')$-Square-Pizza-Sharing, respectively, for any $\varepsilon, \varepsilon'$ with $6/N^c \leq \varepsilon' < \varepsilon \leq 1$, where $c > 0$ is a constant. What remains is to show the correctness of this statement. Notice that, after "pixelation", we placed a set of at most $2N^c/\alpha$ points at the bottom-left corner of the corresponding pixel. Then, we perturbed each point such that it remained in its initial pixel while ensuring that all points are in general position and unique x- and y-coordinates.

Consider now a line (resp. a line segment) ℓ that is part of a solution of the $(\varepsilon - \varepsilon')$-Discrete-Straight-Pizza-Sharing (resp. $(\varepsilon - \varepsilon')$-Discrete-Square-Pizza-Sharing) instance, and intersects the corresponding non-obtuse triangle from mass distribution $i \in [q]$, where $q = 2n$ (resp. $q = n$). The triangle has been pixelated, and its corresponding points belonging to P_i have been created. As we discussed above, each point is inside its corresponding pixel. The line that cuts through the triangle can be thought of as intersecting a set M of the pixels of the triangle's pixelated version. The points that correspond to the pixels of L_p are clearly in the L-part of the cut; the points that correspond to the pixels of R_p are clearly in the R-part of the cut. No matter what part of the cut the points corresponding to M join, Claim 35 and Lemma 36 apply. In the aforementioned results, notice that $L_p \cup M_L$ and $R_p \cup M_R$ correspond to regions that are defined by whole pixels, meaning that they are the union of pixel regions. Therefore, their respective areas are of the form $k_L \cdot t^2$ and $k_R \cdot t^2$, where $k_L, k_R \in \mathbb{N}$ represent the number of pixels on each part of the cut. By construction, if our triangle at hand belongs to μ_i and has weight $w_{i,j}$, then each of the k_L pixels contains at least $\lceil w_{i,j} \cdot N^c \rceil$ points, and similarly for k_R.

Suppose we have turned the ε-Straight-Pizza-Sharing (resp. ε-Square-Pizza-Sharing) to $(\varepsilon - \varepsilon')$-Discrete-Straight-Pizza-Sharing (resp. $(\varepsilon - \varepsilon')$-Discrete-Square-Pizza-Sharing) as described above, so that all points are in general position (resp. have unique x- and y-coordinates). A solution for the latter problems always exists due to [Sch21] and Lemma 29. Let us have a solution of any of the latter two problems, that is, a set of lines (resp. line segments) $\ell_1, \ldots, \ell_m$ for $m \leq 2n$, that partition $[0,1]^2$ to R^+ and R^- and for every $i \in [q]$, where $q \in \{2n, n\}$, we have $||P_i \cap R^+| - |P_i \cap R^-|| \leq (\varepsilon - \varepsilon') \cdot |P_i|$.

Recall that $T^{i,j} \in \mathcal{F}_i$ is a non-obtuse triangle which belongs to the j-th polygon of μ_i, and $\mathcal{F}_i$ is the set of such triangles that compose μ_i. By $T_p^{i,j}$ we denote the pixelated version of $T^{i,j}$, while $M_+^{i,j}, M_-^{i,j}$ are $T_p^{i,j}$'s respective parts of the pixels intersected by lines, that join the R^+ and the R^- sides, respectively. From our earlier analysis, we have an upper bound of $\frac{2N^c}{t^2\alpha}$ for $|P_i|$, however, here we need a better one, namely where a factor of α is removed from the denominator. To that end, we will use the fact that $\sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \text{area}(T^{i,j}) = 1$, or equivalently,

$\sum_{T^{i,j}\in\mathcal{F}_i} w_{i,j}\cdot N^c\cdot \mathrm{area}(T^{i,j}) = N^c$. We have

$$\sum_{T^{i,j}\in\mathcal{F}_i} w_{i,j}\cdot N^c\cdot \mathrm{area}(T^{i,j}) + \sum_{T^{i,j}\in\mathcal{F}_i} w^i_{\max}\cdot N^c\cdot\frac{\alpha}{N^{1+c}}\cdot \mathrm{area}(T^{i,j}) \le N^c + w^i_{\max}\cdot N^c\cdot\frac{\alpha}{N^{1+c}},$$

and since $\sum_{T^{i,j}\in\mathcal{F}_i}\mathrm{area}(T^{i,j})\le 1$, we get

$$\sum_{T^{i,j}\in\mathcal{F}_i} \lceil w_{i,j}\cdot N^c\rceil\cdot \mathrm{area}(T^{i,j}) + \sum_{T^{i,j}\in\mathcal{F}_i} \left\lceil w^i_{\max}\cdot N^c\right\rceil\cdot\frac{\alpha}{N^{1+c}}\cdot \mathrm{area}(T^{i,j}) \le N^c + 1 + \left(w^i_{\max}\cdot N^c + 1\right)\cdot\frac{\alpha}{N^{1+c}}.$$

Now observe that, after pixelation, only the pixels at the boundary of each triangle $T^{i,j}$ can correspond to $\left\lceil w^i_{\max}\cdot N^c\right\rceil$ points instead of $\lceil w_{i,j}\cdot N^c\rceil$. Therefore, using the notation of Equation (1), where $S := \mathrm{area}(T^{i,j})$, only at most a fraction S'/S can correspond to $\left\lceil w^i_{\max}\cdot N^c\right\rceil$ points. So,

$$\begin{aligned}
|P_i| &\le \sum_{T^{i,j}\in\mathcal{F}_i} \lceil w_{i,j}\cdot N^c\rceil\cdot\frac{\mathrm{area}(T^{i,j})}{t^2} + \sum_{T^{i,j}\in\mathcal{F}_i} \left\lceil w^i_{\max}\cdot N^c\right\rceil\cdot\frac{\alpha}{N^{1+c}}\cdot\frac{\mathrm{area}(T^{i,j})}{t^2}\\
&\le \frac{1}{t^2}\cdot\left(N^c + 1 + \left(w^i_{\max}\cdot N^c + 1\right)\cdot\frac{\alpha}{N^{1+c}}\right)\\
&\le \frac{1}{t^2}\cdot\left(N^c + 1 + \left(\frac{N^c}{\alpha} + 1\right)\cdot\frac{\alpha}{N^{1+c}}\right)\\
&\le \frac{1}{t^2}\cdot\left(N^c + 1 + \frac{2}{N}\right)\\
&\le \frac{1}{t^2}\cdot\left(N^c + 3\right), \qquad (3)
\end{aligned}$$

where the second to last inequality comes from the fact that $1\le N^c/\alpha$.

Putting everything together, we have

$$
\begin{aligned}
\left|\mu_i(R^+) - \mu_i(R^-)\right| &= \left| \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \text{area}(T^{i,j} \cap R^+) - \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \text{area}(T^{i,j} \cap R^-) \right| \\
&= \left| \sum_{T^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \big(\text{area}(T^{i,j} \cap R^+) - \text{area}(T^{i,j} \cap R^-)\big) \right| \\
&\leq \left| \sum_{T_p^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot (\text{area}(T_p^{i,j} \cup M_+^{i,j}) - \text{area}(T_p^{i,j} \cup M_-^{i,j})) \right| + \frac{2\alpha}{N^c} \cdot \sum_{T_p^{i,j}} w_{i,j} \\
&\leq \left| \sum_{T_p^{i,j} \in \mathcal{F}_i} w_{i,j} \cdot \left(\frac{|P_{i,j} \cap R^+|}{\lceil w_{i,j} \cdot N^c \rceil} \cdot t^2 - \frac{|P_{i,j} \cap R^-|}{\lceil w_{i,j} \cdot N^c \rceil} \cdot t^2 + 1 \cdot t^2 \right) \right| + \frac{2}{N^c} \\
&\leq t^2 \cdot \left| \frac{1}{N^c} \cdot \sum_{T_p^{i,j} \in \mathcal{F}_i} |P_{i,j} \cap R^+| - \frac{1}{N^c} \cdot \sum_{T_p^{i,j} \in \mathcal{F}_i} |P_{i,j} \cap R^-| \right| + \frac{t^2}{\alpha} + \frac{2}{N^c} \\
&\leq t^2 \cdot \frac{1}{N^c} \cdot \big| |P_i \cap R^+| - |P_i \cap R^-| \big| + \frac{t^2}{\alpha} + \frac{2}{N^c} \\
&\leq t^2 \cdot \frac{1}{N^c} \cdot (\varepsilon - \varepsilon')|P_i| + \frac{t^2}{\alpha} + \frac{2}{N^c} \\
&\leq (\varepsilon - \varepsilon') \cdot \frac{1}{N^c} \cdot (N^c + 3) + \frac{\alpha}{225N^{2+2c}} + \frac{2}{N^c} \qquad \text{(by Equation (3))} \\
&\leq \varepsilon - \varepsilon' + \frac{3}{N^c} + \frac{\alpha}{225N^{2+2c}} + \frac{2}{N^c} \\
&\leq \varepsilon - \varepsilon' + \frac{6}{N^c} \\
&\leq \varepsilon,
\end{aligned}
$$

where the first inequality is acquired by the reverse triangle inequality in combination with Lemma 36, the second inequality is due to the fact that the number of points in a pixel of $T_p^{i,j}$ is $\lceil w_{\max}^i \cdot N^c \rceil$, where $w_{\max}^i \geq w_{i,j}$, by definition, and the last inequality is by definition of ε'. □

B Exact computation of polygons' positive measure given a SQUARE-path

Consider an input of exact SQUARE-PIZZA-SHARING, i.e., the one of Definition 3, where we additionally restrict the mass distributions to be weighted polygons with holes in the representation form described in Section 2. We have so far ensured that our polygons are in $[0,1]^2$. For the computation of function f described in the proof of Theorem 24 we need a way of computing the "positive" measure of a polygon, as dictated by a given $\vec{z} \in S^d$ (and its induced SQUARE-path). To simplify this computation, we will first use some well-known algorithm (e.g., [GJPT78, AAP86, Meh84, GM91]) that triangulates polygons with holes in $O(N \log N)$ time (which is optimal), where N is the input size, without inserting additional vertices. Then, we will triangulate the polygon further to end up with only non-obtuse triangles, which can be decomposed into axis-aligned right-angled triangles. Algorithm 2 describes the two preprocessing

steps. To prove the algorithm's correctness we need to obtain the preliminary results of the following section.

B.1 Computing areas of polygons via axis-aligned right-angled triangle decomposition

Here we first show how an arbitrary triangle can be decomposed into right-angled triangles whose right angle is additionally axis-aligned.

Recall that the Borsuk-Ulam function $f : S^d \to \mathbb{R}^n$ we defined in the proof of Theorem 24, namely, $f(\vec{z}) = \mu(R^+; \vec{z})$ captures the R^+ part of each of n measures when cut by a SQUARE-path (induced by $\vec{z}$) with $d-1$ turns. From this, it is apparent that we need to be able to compute parts of the area of a polygon, depending on where the SQUARE-path cuts it. We first need to preprocess the input by: (i) decomposing each polygon into non-obtuse triangles (Proposition 38), and (ii) decomposing each such triangle into 5 axis-aligned right-angled triangles (Proposition 39). Then, we provide a Borsuk-Ulam function which, given a SQUARE-path, captures the measure found on the R^+ region. The aforementioned decomposition, allows our function to be relatively simple, in the sense that it only needs to consider a single shape, that of axis-aligned right-angled triangles.

We start by showing a simple polynomial time routine that achieves the first decomposition step.

Proposition 38. *Any polygon with holes can be decomposed into non-obtuse triangles in polynomial time.*

Proof. We first use a standard polynomial-time algorithm to triangulate the given polygon, for example, the technique of [AAP86]. Next, we check the obtuseness of each triangle $\widehat{ABC}$ of the triangulation by computing the squared lengths of its sides AB^2, BC^2, AC^2 (each is rational; a sum of squares of rationals), taking the largest one, w.l.o.g. AC^2 and then checking whether $AB^2 + BC^2 < AC^2$. If the inequality is not true then $\widehat{ABC}$ is non-obtuse and we proceed with the next available triangle. Otherwise, we add the line segment BD that starts from B and ends at D on side AC, where $\widehat{BDC} = \widehat{ADB} = 90°$.[9] The coordinates (x_D, y_D) of D are rationals since they are the solution of the following two equations: (a) one that dictates that D is on AC: $\frac{y_D - y_A}{x_D - x_A} = \frac{y_A - y_C}{x_A - x_C}$, and (b) one that captures the fact that BD and AC are perpendicular: $\frac{y_B - y_D}{x_B - x_D} \cdot \frac{y_A - y_C}{x_A - x_C} = -1$. In fact,

$$x_D = \frac{Num}{Den}, \quad \text{and} \quad y_D = \frac{y_A - y_C}{x_A - x_C}(x_D - x_A) + y_A,$$

$$\begin{aligned} \text{where} \quad Num =& (y_A - y_D)[(y_B - y_A)(x_A - x_C) + (y_A - y_C)x_A] + x_B(x_A - x_C)^2, \\ \text{and} \quad Den =& (x_A - x_C)(x_A - x_C + y_A - y_C) \end{aligned}$$

This results in two right-angled triangles $\widehat{ABD}$ and $\widehat{BCD}$. We then proceed with the next available triangle of the triangulation, until there is none left.

It is easy to see that the triangulation results in polynomial many triangles, and the above check and potential split of each triangle requires at most polynomial time. □

[9] We will denote by $\widehat{ABC}$ a triangle with vertices A, B, C and, when clear from context, we will also use the same notation to indicate the *area* of the triangle. Two intersecting line segments AB, BC define two *angles*, denoted $\widehat{ABC}$ and $\widehat{CBA}$. The order of the vertices implies a direction of the segments, i.e., in the former angle we have AB, BC and in the latter we have CB, BA. We consider the direction of the segments and define the angle to be the intersection of the *left* half-spaces of the segments. Therefore $\widehat{ABC} = 360° - \widehat{CBA}$. This order will not matter if clear from context (e.g., in triangles).

At this point, the triangulation of each polygon consists of non-obtuse triangles (see Figure 7a). The next proposition decomposes further each non-obtuse triangle into axis-aligned right-angled triangles.

Proposition 39. *The area of any non-obtuse triangle can be computed using the areas of five axis-aligned right-angled triangles.*

Proof. Consider a non-obtuse triangle $\widehat{ABC}$. A proof by picture is presented in Figure 7b, where we draw a segment from the top-left corner to the bottom-right one, and $\text{area}(\widehat{ABC}) = \text{area}(\widehat{XYB}) + \text{area}(\widehat{XBZ}) - \text{area}(\widehat{AYB}) - \text{area}(\widehat{XAC}) - \text{area}(\widehat{CBZ})$.

The proof is immediate if we show that every non-obtuse triangle $\widehat{ABC}$ can be inscribed inside a rectangle such that all of its vertices touch the rectangle's perimeter and one of them touches a corner of the rectangle while each of the other two vertices touches one of the rectangle's non-incident sides to that corner. To see this, consider a rectangle of minimum perimeter, which contains $\widehat{ABC}$ and its sides are parallel to the axes. Since its perimeter is minimum, each side touches at least one of the triangle's vertices, otherwise the perimeter could be reduced. And since the triangle has only three vertices, at least one of them has to be touching two sides of the rectangle, i.e., a corner of the rectangle. If only one triangle vertex touches a corner of the rectangle, then each of the other two vertices touches one of the non-incident sides of the rectangle's corner, otherwise the rectangle's perimeter can be reduced. If two triangle vertices are on corners of the rectangle, then the third vertex has to be on another corner (i.e., it is a right-angled triangle); otherwise, either the rectangle does not have minimum perimeter (the two corners have a common side), or it is obtuse (the two corners do not have a common side), both being contradictions.

Due to the above, the coordinates to each of the rectangle's corners are $(x_L, y_L), (x_L, y_H), (x_H, y_H)$, and (x_H, y_L), where x_L, x_H, y_L, y_H denote the minimum and maximum x- and y-coordinates of $\widehat{ABC}$'s vertices, respectively. □

Finally, using the above auxiliary results, Algorithm 2 shows how to decompose each given polygon into axis-aligned right-angled triangles in polynomial time.

B.2 Constructing the Borsuk-Ulam function

Here we show how to construct the Borsuk-Ulam function $f : S^d \to \mathbb{R}^n$ given n sets of weighted polygons, where $d, n \in \mathbb{N}$. We will focus on an arbitrary colour $i \in [n]$ and present the coordinate f_i. As discussed earlier, our function will capture the measure μ_i in the R^+ region of any given SQUARE-path. Furthermore, after the preprocessing step achieved by Algorithm 2, the function suffices to be able to capture the measure of simple shapes, namely axis-aligned right-angled triangles.

Consider the $\tau \geq 1$ weighted polygons of the i-th colour, and let us focus on a particular polygon $t \in [\tau]$. We have triangulated the polygon into m_t non-obtuse triangles. Consider one such triangle T_s, corresponding to some $s \in [m_t]$, and the *virtual triangles* $T_s^1, T_s^2, T_s^3, T_s^4, T_s^5$, which are the five axis-aligned right-angled triangles that Algorithm 2 gave as output (also see Figure 7b). W.l.o.g. we consider T_s^1 and T_s^2 to be the *positively contributing triangles* and the rest to be the *negatively contributing triangles*. For each of them, we will be computing the positive measure determined by the SQUARE-path induced from the given point $\vec{z} \in S^d$ (see proof of Theorem 24). By the axis-aligned right-angled triangle decomposition described in the proof of Proposition 39, it suffices to show how to compute parts of areas of such a triangle, for all of its four possible *orientations*: $Q_I, Q_{II}, Q_{III}, Q_{IV}$, where Q_o is the orientation when, by

Algorithm 2 Preprocessing: decomposing polygons into axis-aligned right-angled triangles

Input: A polygon P represented by its ordered vertices (see Section 2).

Output: A set H consisting of 5-tuples; each 5-tuple $s \in [|H|]$ corresponds to a non-obtuse triangle T_s such that $\bigsqcup_{s\in[|H|]} \text{area}(T_s) = \text{area}(P)$; each tuple $(T_s^1, T_s^2, T_s^3, T_s^4, T_s^5)$ consists of 5 axis-aligned right-angled triangles such that $\text{area}(T_s) = \text{area}(T_s^1) + \text{area}(T_s^2) - \text{area}(T_s^3) - \text{area}(T_s^4) - \text{area}(T_s^5)$

Preprocessing step 1:

Run a poly-time algorithm to triangulate the given polygon (e.g., [AAP86]), and let G be the set of the resulting triangles.
while there is an unchecked triangle $T \in G$ **do**
  Check the obtuseness of T
  if T is obtuse **then**
    Split it into two right-angled triangles T_I, T_{II} (Proof of Proposition 38).
    $G \leftarrow G \cup \{T_I, T_{II}\}$

Preprocessing step 2:

$H \leftarrow \emptyset$
while $G \neq \emptyset$ **do**
  Consider a (non-obtuse) triangle $T \in G$.
  Define the tuple $r := ((x_L, y_L), (x_L, y_H), (x_H, y_H), (x_H, y_L))$, where x_L, x_H, y_L, y_H denote the minimum and maximum x- and y-coordinates of T's vertices, respectively.
  Find a point in r which corresponds to a non-right angle of T and name that vertex B (Guaranteed by the proof of Proposition 39).
  Let B be element $r(i)$ for some $i \in [4]$, and name the following points: $Y = r(i \pmod 4 + 1)$, $X = r(i + 1 \pmod 4 + 1)$, and $Z = r(i + 2 \pmod 4 + 1)$.
  Name A the vertex of T located on the segment YX, and C the vertex of T on the segment XZ (Both guaranteed to be in these segments by the proof of Proposition 39).
  Define the tuple $v \leftarrow (\widehat{XYB}, \widehat{XBZ}, \widehat{AYB}, \widehat{XAC}, \widehat{CBZ})$.
  $H \leftarrow H \cup \{v\}$
  $G \leftarrow G \setminus \{T\}$

shifting the triangle so that the vertex of the right angle is on $(0, 0)$, the whole triangle is in the o-th quadrant.

First, we identify the orientation of our triangle. For a fixed colour $i \in [n]$, for each possible category Q_o, $o \in \{I, II, III, IV\}$ we show how to compute the term that an axis-aligned right-angled triangle T_s^v, $s \in [m_t]$, $v \in [5]$, contributes to the Borsuk-Ulam function $f_i(\vec{z})$, where T_s is a non-obtuse triangle $\widehat{ABC}$ as in Figure 7b. We will show this for a Q_I triangle with the help of Figure 9. The constructions for Q_{II}, Q_{III}, Q_{IV} are omitted since they are symmetric to it.

In what follows, for any point W of $[0, 1]^2$ we will denote by x_W, y_W its coordinates. Suppose we are given the Q_I triangle $\widehat{AYB}$ (as in Figure 7b). We are also provided with some SQUARE-path induced by $\vec{z} \in S^d$, and we focus on the strip $[y_j, y_{j+2}]$ for some $j \in \{0, 1, 3, 5, \ldots, d-1\}$ (resp. $j \in \{0, 1, 3, 5, \ldots, d\}$) when d is even (resp. odd), and z_j, z_{j+1} which induce the vertical cut x_j and define an R^+ and an R^- region of the slice (see the proof of Theorem 24 for details). We also add all the artificial cuts needed in the bottom and top strips (see Algorithm 1). We are only interested in the part of $\widehat{AYB}$ in the R^+ region, but since this could be either to the left or to the right of x_j, let us denote by $\text{area}_j^\ell(\widehat{AYB})$ and $\text{area}_j^r(\widehat{AYB})$ the areas of $\widehat{AYB}$ to the right and left part, respectively, of x_j in the j-th slice.

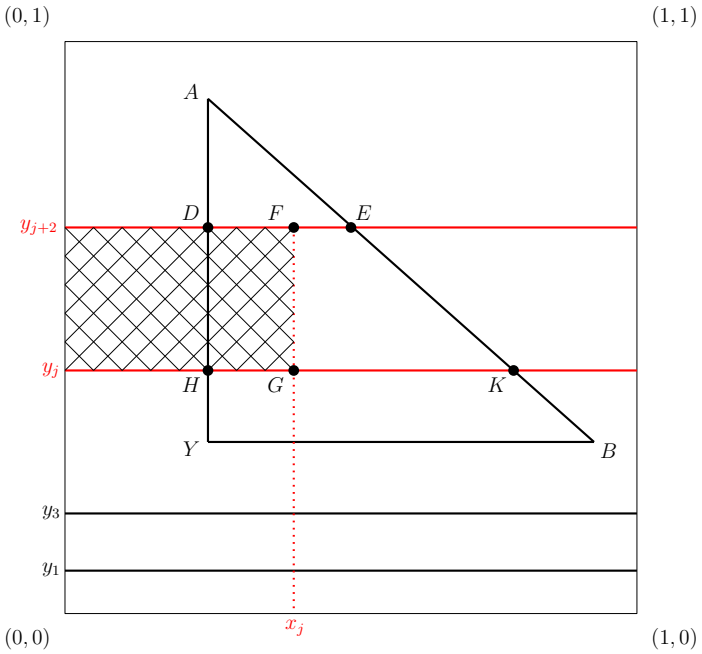


Figure 9: An example of a Q_I triangle $\widehat{AYB}$. The given $\vec{z} \in S^d$ has $z_j < 0$ and $z_{j+1} > 0$, therefore, the R^+ part of $\widehat{AYB}$ is $\text{area}_j^r(\widehat{AYB}) = \text{area}(EFGK)$ and its R^- part is $\text{area}_j^\ell(\widehat{AYB}) = \text{area}(FDHG)$.

Using Figure 9, we have $\text{area}_j^\ell(\widehat{AYB}) = \text{area}(FDHG)$, and $\text{area}_j^r(\widehat{AYB}) = \text{area}(EFGK)$. We have the following cases. (i) $x_j \in (0,1)$: if $z_j \geq 0$ and $z_{j+1} \leq 0$ (resp. $z_j \leq 0$ and $z_{j+1} \geq 0$), then $\text{area}_j^\ell(\widehat{AYB})$ belongs to R^+ (resp. R^-) and $\text{area}_j^r(\widehat{AYB})$ belongs to R^- (resp. R^+). (ii) $x_j \in \{0,1\}$: if $z_j + z_{j+1} \geq 0$ (resp. $z_j + z_{j+1} \leq 0$), then the part of $\widehat{AYB}$ that belongs to the j-th strip, denoted $\text{area}_j(\widehat{AYB})$, belongs entirely to R^+ (resp. R^-).[10] So, w.l.o.g., we can say that $\text{area}_j^\ell(\widehat{AYB})$ and $\text{area}_j^r(\widehat{AYB})$ have opposite signs (and it is possible that one of these areas is 0).

Since the slices y_j, y_{j+2}, and cut x_j, in general, can have values that do or do not intersect $\widehat{AYB}$, we create truncated versions of them as follows:

$$y_j^{tr} := \max\{y_B, \min\{y_A, y_j\}\} = \begin{cases} y_A, & y_j > y_A \\ y_j, & y_j \in [y_B, y_A] \\ y_B, & y_j < y_B, \end{cases}$$

and similarly, $y_{j+2}^{tr} := \max\{y_B, \min\{y_A, y_{j+2}\}\}$, and $x_j^{tr} := \max\{x_A, \min\{x_K, x_j\}\}$. Given these, we need to define properly the y-coordinate of points F, G, and the x-coordinate of points K, E, so that the points stay on the boundary of the trapezoid $EDHK$. Observe that the line passing from points A, B is described by $y = y_B + \frac{x_B - x}{x_B - x_A}(y_A - y_B)$, or equivalently, $x =$

[10] Notice that cases (i) and (ii) include the subcase $z_j = z_{j+1} = 0$. However then, the sign(s) of the (possibly two) parts of the j-th strip do not matter since the strip has width 0 and therefore does not contribute to the Borsuk-Ulam function.

$x_B - \frac{y-y_B}{y_A-y_B}(x_B - x_A)$, so using these we get:

$$\begin{aligned}
y_F &:= \max\left\{y_B, \min\left\{y^{tr}_{j+2}, y_B + \frac{x_B - x^{tr}_j}{x_B - x_A}(y_A - y_B)\right\}\right\},\\
y_G &:= \max\{y_B, \min\{y^{tr}_j, y_F\}\},\\
x_K &:= x_B - \frac{y^{tr}_j - y_B}{y_A - y_B}(x_B - x_A),\\
x_E &:= x_B - \frac{y^{tr}_{j+2} - y_B}{y_A - y_B}(x_B - x_A).
\end{aligned}$$

Now we are ready to compute the length of our line segments. We have, $EF = \max\{0, x_E - x^{tr}_j\}$, $GK = \max\{0, x_K - x^{tr}_j\}$, $FG = y_F - y_G$, $HK = x_K - x_A$, $DE = x_E - x_A$, and $DH = y^{tr}_{j+2} - y^{tr}_j$. Using these, we can compute the quantities of interest:

$$\begin{aligned}
\text{area}^r_j(\widehat{AYB}) &= \text{area}(EFGK) = \frac{(GK + EF)\cdot FG}{2},\\
\text{area}^\ell_j(\widehat{AYB}) &= \text{area}(FDHG) = \text{area}(EDHK) - \text{area}(EFGK)\\
&= \frac{(HK + DE)\cdot DH}{2} - \frac{(GK + EF)\cdot FG}{2}.
\end{aligned}$$

Using the above, we pick the element from $\left\{\text{area}^\ell_j(\widehat{AYB}), \text{area}^r_j(\widehat{AYB})\right\}$ that belongs to R^+, and let us denote this quantity $p^v_s(j)$. This quantity represents the part that *only the j-th strip* contributes to the positive measure of the Borsuk-Ulam function due to triangle T^v_s, for some $s \in [m_t]$ and $v \in [5]$. Consequently, the positive measure that the *entire (unweighted) non-obtuse triangle* T_s contributes to the Borsuk-Ulam function according to the SQUARE-path induced by $\vec{z}$ is

$$q_s := \sum_{j\in J}\left(p^1_s(j) + p^2_s(j) - p^3_s(j) - p^4_s(j) - p^5_s(j)\right),$$

where $J := \{0, 1, 3, 5, \ldots, d-1\}$ (resp. $J := \{0, 1, 3, 5, \ldots, d\}$) when d is even (resp. odd).

Finally, recall that colour $i \in [n]$ has τ many weighted polygons, each of weight w_t, $t \in [\tau]$. Also, each polygon has been decomposed into m_t many non-obtuse triangles. Then, i's positive measure (i.e., the i-th coordinate of the Borsuk-Ulam function) is

$$f_i(\vec{z}) = \sum_{t=1}^{\tau} w_t \sum_{s=1}^{m_t} q_s.$$

Acknowledgements

The second author was supported by the EPSRC grant EP/W014750/1 "New Techniques for Resolving Boundary Problems in Total Search".

References

[AAP86] Takao Asano, Tetsuo Asano, and Ron Y. Pinter. Polygon triangulation: Efficiency and minimality. *Journal of Algorithms*, 7(2):221–231, 1986. doi:10.1016/0196-6774(86)90005-2.

[AD15] Yonatan Aumann and Yair Dombb. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation*, 3(4):23:1–23:16, 2015. doi:10.1287/moor.2019.1016.

[AE+99] Pankaj K Agarwal, Jeff Erickson, et al. Geometric range searching and its relatives. *Contemporary Mathematics*, 223:1–56, 1999.

[BFHZ18] Pavle Blagojević, Florian Frick, Albert Haase, and Günter Ziegler. Topology of the Grünbaum–Hadwiger–Ramos hyperplane mass partition problem. *Transactions of the American Mathematical Society*, 370(10):6795–6824, 2018. doi:10.1090/tran/7528.

[BHH21] Eleni Batziou, Kristoffer Arnsfelt Hansen, and Kasper Høgh. Strong approximate consensus halving and the Borsuk-Ulam theorem. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2021. doi:10.4230/LIPIcs.ICALP.2021.24.

[BPR06] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Existential Theory of the Reals*, pages 505–532. Springer Berlin Heidelberg, 2006. doi:10.1007/3-540-33099-2_14.

[BPS19] Luis Barba, Alexander Pilz, and Patrick Schnider. Sharing a pizza: bisecting masses with two cuts. *arXiv preprint*, abs/1904.02502, 2019. URL: https://arxiv.org/abs/1904.02502.

[Bro11] L. E. J. Brouwer. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1911. doi:10.1007/BF01456931.

[Can88] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the 20th ACM Symposium on Theory of Computing (STOC)*, pages 460–467, 1988. doi:10.1145/62212.62257.

[Cha05] Christopher P. Chambers. Allocation rules for land division. *Journal of Economic Theory*, 121(2):236–258, 2005. doi:10.1016/j.jet.2004.04.008.

[CS17] Karthik C.S. and Arpan Saha. Ham sandwich is equivalent to Borsuk-Ulam. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG)*, pages 24:1–24:15, 2017. doi:10.4230/LIPIcs.SoCG.2017.24.

[DFH21] Argyrios Deligkas, Aris Filos-Ratsikas, and Alexandros Hollender. Two's company, three's a crowd: Consensus-Halving for a constant number of agents. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 347–368, 2021. doi:10.1145/3465456.3467625.

[DFHM25] Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. Constant inapproximability for PPA. *SIAM Journal on Computing*, 54(1):163–192, 2025. doi:10.1137/22M1536613.

[DFM22] Argyrios Deligkas, John Fearnley, and Themistoklis Melissourgos. Pizza sharing is ppa-hard. In *Proceedings 36-th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4957–4965, 2022. doi:10.1609/aaai.v36i5.20426.

[DFMS21] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Computing exact solutions of consensus halving and the Borsuk-Ulam theorem. *Journal of Computer and System Sciences*, 117:75–98, 2021. doi:10.1016/j.jcss.2020.10.006.

[DLGMM19] Jesús De Loera, Xavier Goaoc, Frédéric Meunier, and Nabil Mustafa. The discrete yet ubiquitous theorems of Carathéodory, Helly, Sperner, Tucker, and Tverberg. *Bulletin of the American Mathematical Society*, 56(3):415–511, 2019. doi:10.1090/bull/1653.

[Ede12] Herbert Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10. Springer Science & Business Media, 2012. doi:10.1007/978-3-642-61568-9.

[ESS21] Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. Keep your distance: Land division with separation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 168–174, 2021. doi:10.24963/ijcai.2021/24.

[EvM20] Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. Smoothing the gap between NP and ER. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1022–1033, 2020. doi:10.1109/FOCS46700.2020.00099.

[EY10] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010. doi:10.1137/080720826.

[FHSZ20] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. Consensus-Halving: Does it Ever Get Easier? In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 381–399, 2020. doi:10.1145/3391403.3399527.

[FRFGZ18] Aris Filos-Ratsikas, Søren Kristoffer Still Frederiksen, Paul W. Goldberg, and Jie Zhang. Hardness results for Consensus-Halving. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 24:1–24:16, 2018. doi:10.4230/LIPIcs.MFCS.2018.24.

[FRG18] Aris Filos-Ratsikas and Paul W. Goldberg. Consensus halving is PPA-complete. In *Proceedings of the 50th ACM Symposium on Theory of Computing (STOC)*, pages 51–64, 2018. doi:10.1145/3188745.3188880.

[FRG19] Aris Filos-Ratsikas and Paul W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. In *Proceedings of the 51st ACM Symposium on Theory of Computing (STOC)*, pages 638–649, 2019. doi:10.1145/3313276.3316334.

[FRHSZ21] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. A topological characterization of modulo-p arguments and implications for necklace splitting. In *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2615–2634, 2021. doi:10.1137/1.9781611976465.155.

[GGJ76] M. R. Garey, Ronald L. Graham, and David S. Johnson. Some NP-complete geometric problems. In *Proceedings of the 8th ACM Symposium on Theory of Computing (STOC)*, 1976. doi:10.1145/800113.803626.

[GJPT78] M. R. Garey, David S. Johnson, Franco P. Preparata, and Robert Endre Tarjan. Triangulating a simple polygon. *Information Processing Letters*, 7(4):175–179, 1978. doi:10.1016/0020-0190(78)90062-5.

[GM91] Subir Kumar Ghosh and David M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM Journal on Computing*, 20(5):888–910, 1991. doi:10.1137/0220055.

[Hav22] Ishay Haviv. The complexity of finding fair independent sets in cycles. *Computational Complexity*, 31(2):14, 2022. doi:10.1007/s00037-022-00233-6.

[HK20] Alfredo Hubard and Roman Karasev. Bisecting measures with hyperplane arrangements. *Mathematical Proceedings of the Cambridge Philosophical Society*, 169(3):639–647, 2020. doi:10.1017/S0305004119000380.

[Hüs11] Farhad Hüsseinov. A theory of a heterogeneous divisible commodity exchange economy. *Journal of Mathematical Economics*, 47(1):54–59, 2011. doi:10.1016/j.jmateco.2010.12.001.

[IH09] Karthik Iyer and Michael N Huhns. A procedure for the allocation of two-dimensional resources in a multiagent system. *International Journal of Cooperative Information Systems*, 18(3–4):381–422, 2009. doi:10.1142/S0218843009002051.

[KK03] Atsushi Kaneko and M. Kano. *Discrete Geometry on Red and Blue Points in the Plane-A survey*, volume 25. Springer Berlin, Heidelberg, 2003. doi:10.1007/978-3-642-55566-4_25.

[Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Professional, 1998.

[KRPS16] Roman N. Karasev, Edgardo Roldán-Pensado, and Pablo Soberón. Measure partitions using hyperplanes with fixed directions. *Israel Journal of Mathematics*, 212:705–728, 2016. doi:10.1007/s11856-016-1303-z.

[LRY09] Zeph Landau, Oneil Reid, and Ilona Yershov. A fair division solution to the problem of redistricting. *Social Choice and Welfare*, 32(3):479–492, 2009. doi:10.1007/s00355-008-0336-6.

[LS14] Zeph Landau and Francis Edward Su. *Fair division and redistricting*, volume 624. American Mathematical Society, 2014. doi:10.1090/conm/624/12472.

[Mat02] Jiří Matoušek. *Lectures on discrete geometry*, volume 212. Springer Science, 2002. doi:10.1007/978-1-4613-0039-7.

[Mat08] Jiří Matoušek. *Using the Borsuk-Ulam theorem: lectures on topological methods in combinatorics and geometry*. Springer Science & Business Media, 2008. doi:10.1007/978-3-540-76649-0.

[Mat14] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *arXiv preprint*, abs/1406.2636, 2014. URL: http://arxiv.org/abs/1406.2636.

[Meh84] Kurt Mehlhorn. *Data Structures and Algorithms 3: Multi-Dimensional Searching and Computational Geometry*. Springer-Verlag, 1984. doi:10.1007/978-3-642-69900-9.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991. doi:10.1016/0304-3975(91)90200-L.

[Pap77] Christos H. Papadimitriou. The euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

[RS20] Edgardo Roldán-Pensado and Pablo Soberón. A survey of mass partitions. *arXiv preprint*, abs/2010.00478, 2020. URL: https://arxiv.org/abs/2010.00478.

[Sch09] Marcus Schaefer. Complexity of some geometric and topological problems. In *Proceedings of the 17th International Symposium on Graph Drawing*, pages 334–344, 2009. doi:10.1007/978-3-642-11805-0_32.

[Sch21] Patrick Schnider. The complexity of sharing a pizza. In *Proceedings of the 32nd International Symposium on Algorithms and Computation (ISAAC)*, pages 13:1–13:15, 2021. doi:10.4230/LIPIcs.ISAAC.2021.13.

[SHNHA17] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics*, 70:1–28, 2017. doi:10.1016/j.jmateco.2017.01.007.

[SNHA20] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. Envy-free division of land. *Mathematics of Operations Research*, 45(3):896–922, 2020. doi:10.1287/moor.2019.1016.

[SS03] Forest W. Simmons and Francis E. Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Mathematical social sciences*, 45(1):15–25, 2003. doi:10.1016/S0165-4896(02)00087-2.

[Ste48] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[Tiw92] Prasoon Tiwari. A problem that is easier to solve on the unit-cost algebraic RAM. *Journal of Complexity*, 8(4):393–397, 1992. doi:10.1016/0885-064X(92)90003-T.

[Živ17] Rade Živaljević. *Topological methods in discrete geometry*. Taylor & Francis, 2017. doi:10.1201/9781315119601.