Fair and Square: Cake-Cutting in Two Dimensions

Erel Segal-Halevi Shmuel Nitzan Avinatan Hassidim Yonatan Aumann*

Abstract

A polygonal land-estate ("cake") has to be divided among n agents. The division should satisfy the following two requirements: (a) Each piece should have a pre-specified geometric *shape*, such as a square. (b) Each agent should receive a piece with a *value* above a given threshold. The value of a piece is defined as the integral of a given value-density function over the piece. Each agent has a possibly different value-density, yet each agent should agree that the value of his piece is above the fairness threshold. Each of the two requirements has been studied before on its own: the geometric requirement is common in polygon decomposition problems, and the value requirement is common in the classic economics problem known as "fair cake-cutting". Our research combines these requirements. We present algorithms for dividing a square cake in a way both *fair* (in value) and *square* (in shape). The value guarantee per agent is $\Theta(1/n)$, where the constants depend on the cake shape.

1 Introduction

Several people inherited a land-estate. How can they divide it fairly among them?

Geometric division. Division problems are abundant in computational geometry. A survey from 2000 [5] lists over 100 papers about different variants of such problems. A typical problem involves a given polygon C and a given family S of polygons (triangles, squares, rectangles, star-shapes, spirals, etc). C should be partitioned to several components which are elements of S (henceforth S-*elements*). The partition should satisfy such requirements as: minimizing the number of pieces, minimizing the total perimeter of the pieces, etc. Sometimes it is also required that the pieces have the same area, e.g. [2, 6]. But the value of land is much more than its shape and area. For example, a land-plot near the sea may have a very different value than a land-plot with exactly the same shape and area in the middle of the desert. Geometric partition problems do not handle such considerations.

Economic division. Division problems are also abundant in economics and social choice. The land division problem is often called *cake-cutting* [3, 8]. There, value considerations are of key importance. Moreover, economists acknowledge that different people have different valuations. One person may prefer the sea-shore while another person may prefer the mountains. Hence, value is defined on an agent-by-agent basis: each agent i has a bounded and integrable *value-density* function on the cake, $v_i : C \to \mathbb{R}$. The value of a piece X to agent i is defined as the integral of the value-density: $V_i(X) := \int_{x\in X} v_i(x)dx$. The V_i thus defined are *nonatomic measures*, so there are no atoms which cannot be fairly divided.

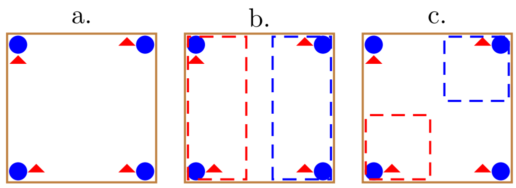

Figure 1: Dividing a square fairly to two agents.

An *allocation* of C is an n-tuple $X_1, \dots, X_n$ of pairwise-disjoint subsets of C: $X_1 \cup \cdots \cup X_n \subseteq C$. An allocation is called *fair* or *proportional* if every agent is allotted a piece he values as at least $1/n$ the total cake value: $\forall i : V_i(X_i)/V_i(C) \geq 1/n$. Algorithms for finding fair allocations have been used since Biblical times. A famous algorithm for two agents is "cut and choose": the cake C is partitioned to two parts C', C'' which have the same value for Alice ($V_A(C') = V_A(C'') = V_A(C)/2$); the part more valuable to Bob is given to Bob and the other part is given to Alice. Thus both Alice and Bob are guaranteed a piece worth at least half their total cake value. This algorithm has been generalized to n agents in the 1940s [10] and many new algorithms have been published over the years [7]. But in contrast to the geometric partition problems, most of these algorithms do not pay much attention to the geometric shape of the pieces. Typically, C is assumed to be a 1-dimensional interval and the pieces are either intervals or a countable collection thereof (see the full paper [9] for some exceptions). While a 1-dimensional division can be projected on a two-dimensional cake, the resulting pieces might be long and narrow slivers that are unusable in practice.

Our division. We claim that both geometric shape and fair value are important. The input in our problem is a polygonal cake C, a family S of polygons

*Bar-Ilan University, Ramat-Gan, Israel. {erelsgl,shmuelnitzan,avinatanh,yaumann}@gmail.com.
Research is partially funded by ISF grant 1083/13, Doctoral Fellowships of Excellence Program and Mordecai and Monique Katz Graduate Fellowship Program at Bar-Ilan University.

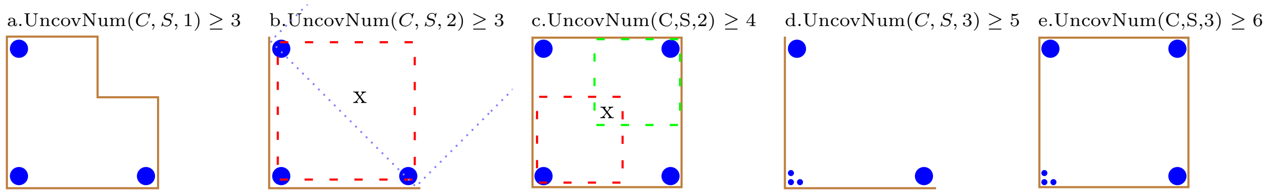

Figure 2: Uncover numbers of various cakes. S is the family of squares.

and n nonatomic value-measures $V_1, \ldots, V_n$. We are looking for S-*allocations* - allocations of C in which $\forall i : X_i \in S$. A simple example, illustrated in Figure 1, shows that the ideal of fairness, giving each agent at least $1/n$ of the total cake value, cannot always be satisfied. Here C is square and there are $n = 2$ agents, Disc and Triangle. The value-density of the Disc agent is 1 inside the discs and 0 outside; the value-density of the Triangle agent is 1 inside the triangles and 0 outside (a). When S is the family of rectangles, it is easy to give each agent 1/2 of the total value (b). When S is the family of squares, it is easy to give each agent 1/4 of the total value (c), but impossible to give both agents more than 1/4. This motivates the following:

Definition 1 ***Prop***(C, S, n) *is the largest proportion* $f \in [0, 1]$ *such that, for every set of* n *nonatomic value-measures* $(V_1, \ldots, V_n)$*, there exists an* S*-allocation* $(X_1, ..., X_n)$ *of* C *for which* $\forall i : V_i(X_i)/V_i(C) \geq f$.

One-dimensional cake-cutting algorithms, e.g. [4], imply that Prop$(Interval, Intervals, n) = 1/n$. By a projection argument, this also implies that Prop$(Rectangle, Rectangles, n) = 1/n$. As we have just seen (Figure 1), Prop$(Square, Squares, 2) \leq 1/4$.

Prop(C, S, n) is a purely geometric function: its value depends only on n and the geometric shapes of C and S. Intuitively, it describes how well the family S can be used to fairly divide C. This function is the focus of our research. We study many different combinations of C and S. For every such combination, we look for impossibility results like Figure 1 proving upper bounds on Prop, and division algorithms proving lower bounds on Prop. In the present abstract we illustrate some of our methods focusing on several simple cases: the cake C is a square, a quarter-plane or an unbounded plane (where the value-density functions always have a bounded support), and S is the family of squares. Section 2 presents impossibility results and Section 3 presents division algorithms. Many other combinations are described in [9], including arbitrarily-shaped cakes and arbitrary fat pieces.

2 Impossibility results

The key geometric tools used in our impossibility results are *uncovers* and *uncover-numbers*. They generalize the *anti-squares* studied e.g. by [1], who also show the duality between them and square-covers.

Definition 2 *(a) Let* I *be a set of discs contained in* C*.* I *is called an* n*-*S***-uncover*** *in* C *if in any set of* n *pairwise-disjoint* S*-elements contained in* C*, at least one* S*-element overlaps only at most one disc of* I*.*

(b) The n*-*S***-uncover number*** *of* C*, UncovNum*(C, S, n)*, is the maximum cardinality of an* n*-*S*-uncover in* C*.*

Some examples are illustrated in Figure 2. In (a), C is a rectilinear hexagon. The three discs are a 1-square-uncover, because any 1 square contained in C overlaps at most one disc. In (b), C is a quarter-plane and the three discs are a 2-square-uncover: any square that overlaps two or more discs must contain the "x" in its interior. Hence, in any set of 2 disjoint squares contained in C, at least one square overlaps at most one disc. Similarly in (c), C is a square and the four discs are a 2-square-uncover.

New uncovers can be constructed from existing ones using *deflation*. Let I_K and I_M be two copies of the 2-square-uncover of (b). Remove the bottom-left disc of I_M and deflate I_K such that all its three discs are contained in the previous location of that bottom-left disc. The result is the set of 5 discs in (d). We claim that it is a 3-square-uncover: there is at most one square overlapping two discs of the deflated I_K and at most one square overlapping two discs of I_M, so all in all there are at most two disjoint squares overlapping two discs of the arrangement in (d).

In general, we can prove the following *Deflation Lemma*: if I_K is a k-S-uncover containing K discs and I_M is an m-S-uncover containing M discs, then (under certain conditions that we omit here) it is possible to deflate I_K into I_M to get a $(k + m - 1)$-S-uncover containing $K + M - 1$ discs. This lemma allows us to construct (d) from (b) (with $m = k = 2$ and $M = K = 3$) and to construct (e) from (b)+(c) ($m = k = 2$ and $M = 4$ and $K = 3$).

Let I_M be an m-S-uncover with M discs. By recursively applying the Deflation Lemma, we can (under certain conditions) deflate I_M into one of its own discs to get uncovers with as many discs as we want. For every $n \geq 1$, it is possible to get an $(n-1)(m-1)+1$-S-uncover having $(n - 1)(M - 1) + 1$ discs. Tak-

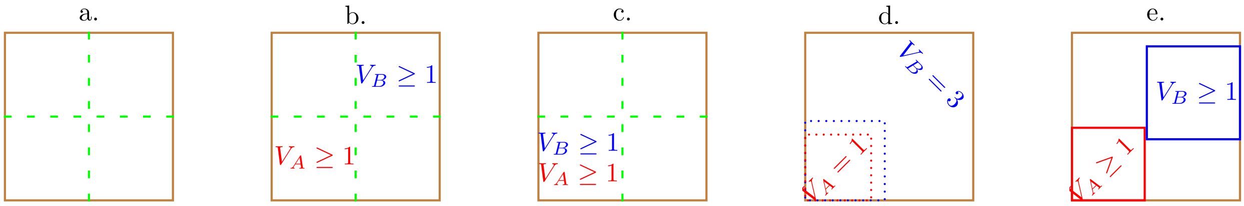

Figure 3: Division algorithm for dividing a square cake to 2 agents who want square pieces.

ing I_M to be the 2-square-uncover in (b), we build, for every $n \geq 2$, an n-square-uncover I_{2n-1} having $2n-1$ discs in a quarter-plane, proving that $\forall n \geq 1$: UncovNum$(QuarterPlane, Squares, n) \geq 2n-1$.

Deflating the set I_{2n-3} into the bottom-left disc of (c) gives, for every $n \geq 2$, an n-square-uncover having $2n$ discs in a square. This proves that $\forall n \geq 2$: UncovNum$(Square, Squares, n) \geq 2n$.

The following lemma links the uncover numbers to fair cake-cutting:

Lemma 1 *For every cake C, family S and $n \geq 1$:*

$$Prop(C, S, n) \leq 1/UncovNum(C, S, n)$$

Proof. Let $m =$ UncovNum(C, n, S) and let I_m be an n-S-uncover of cardinality m in C. Assume that the cake C is a desert and the elements of I_m are water-pools. Assume that all n agents have the same value-density function, which assigns a value of 1 to each pool and is 0 outside the pools. By Definition 2, in every allocation of n S-elements, at least one S-element overlaps at most one pool. The agent receiving this piece has a value of at most $1 = V(C)/m$. ☐

With the uncover numbers from above, we get:

Corollary 2 *a. Prop$(Square, Squares, n) \leq 1/(2n)$; b. Prop$(QuarterPlane, Squares, n) \leq 1/(2n-1)$.*

3 Division algorithms

The key geometric tools used in our positive results are *covers* and *cover numbers*.

Definition 3 *(a) An S-**cover of** C is a set of S-elements whose union equals C.*
*(b) The S-**cover number** of C, CoverNum(C, S), is the minimum cardinality of an S-cover of C.*

For example, it is easy to see that for the hexagon in Figure 2/a, CoverNum$(C, Squares) = 3$, since it can be covered by a set of 3 squares (and no set of 1 or 2 squares can cover it).

Lemma 3 *For every cake C and family S:*

$$Prop(C, S, 1) \geq 1/CoverNum(C, S)$$

Proof. By the pigeonhole principle, if there is an S-cover of C with m elements, then one of these elements must have a value of at least $V(C)/m$. ☐

We now present an introductory division algorithm. C is a square, S the family of squares and there are $n = 2$ agents, Alice and Bob. Without loss of generality, we scale the value-densities of the agents such that the value of the entire cake is exactly 4 for both of them. By the example of Figure 1 and by Corollary 2, we already know that the largest value that can be guaranteed to both agents is 1. Our algorithm indeed guarantees each agent a value of at least 1.

The algorithm is illustrated in Figure 3 and it proceeds as follows. (a) Partition C to 4 quarters in a 2×2 grid and calculate the value of each quarter according to each agent. For each agent, select a quarter with a value of at least 1 (such a quarter must exist by the pigeonhole principle). (b) If the selected quarters are different, then give each agent his/her selected quarter and finish. Here it is easy to satisfy both agents since each agent prefers a different geographic region - Alice likes the south-west and Bob likes the north-east. (c) If the selected quarter is the same, then (d) for each agent, mark inside the selected quarter, a corner-square with a value of exactly 1 for that agent. (e) Cut a corner-square between the two marks. Give it to the agent associated with the smaller mark (Alice, in this case); that agent obviously receives a value of at least 1. For the other agent (Bob), the value of the remaining L-shape is at least 3. Cut a square from the remaining L-shape with a value of at least 1 (such a square must exist by Lemma 3) and give it to Bob. Note that Bob's square is larger than Alice's square; this makes sense, since Alice won a square in the south-west, which, according to both agents, is a valuable region. So Bob is compensated by winning a larger plot.

This algorithm proves Prop$(Square, Squares, 2) \geq 1/4$, which matches the upper bound of Corollary 2.

There are several ways to generalize this algorithm to n agents. We present here high-level sketches of the algorithms and refer the reader to [9] for more details.

Algorithm 1. For dividing a square cake, we use a "recursive quartering" technique. The cake is partitioned to four quarters as in Figure 3/a. The value of each quarter according to each agent is calculated.

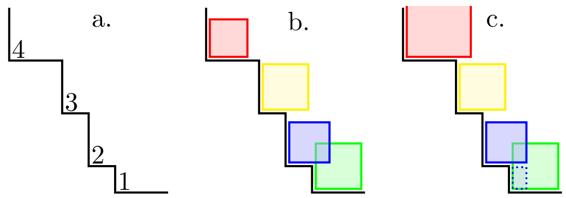

Figure 4: Dividing a 4-staircase.

The agents are assigned to quarters based on their values: each agent is assigned to a quarter which is more valuable to that agent (considering the number of other agents assigned to that quarter). Then, each quarter is divided recursively to the agents assigned to it (handling some special cases which we omit here). If done correctly, this algorithm guarantees to each agent a value of at least 1, when the value of the original cake is normalized to $4n-4$. This proves that: $\text{Prop}(Square, Squares, n) \geq 1/(4n-4)$. This has a multiplicative gap of 2 from the upper bound of Corollary 2. We know how to close this gap in the special case of n agents with identical value measures.

Algorithm 2. For dividing a quarter-plane cake, we use a "staircase carving" technique. We want to match the upper bound of Corollary 2 which is $1/(2n-1)$. To do this, we generalize and handle a cake in the shape of a *staircase*. A staircase is a polygonal domain which is bounded in two sides (e.g. left and bottom) but open in the other two sides. A *k-staircase* is a staircase that has k inner corners (see Figure 4/a). A quarter-plane is a 1-staircase. In every k-staircase, it is easy to find an n-square-uncover with $2n-2+k$ discs, so by Lemma 1, $\text{Prop}(k\,staircase, Squares, n) \leq 1/(2n-2+k)$. The following algorithm matches this bound.

Assume that n agents (with different value-densities) value the k-staircase as $2n-2+k$. For each corner j and agent i, mark a corner-square in corner j with a value of exactly 1 for agent i. In each corner, keep only one smallest square. There are two cases. *Easy case*: at least one square is entirely contained in its corner (like the square in corner 4 in Figure 4/b). Cut this square and give it to its agent. The remaining cake is a $(k+1)$-staircase and its value for the remaining $n-1$ agents is at least $2n-3+k = 2(n-1)-2+(k+1)$ so we can divide it to them recursively. *Hard case*: all squares flow over their corners, casting "shadows" on the corners above and/or to their right. We cannot just cut a square because the result will not be a staircase. Fortunately, we can prove the following geometric lemma: there always exists a square whose shadows are entirely contained in the other squares (like the square in corner 2 in Figure 4/c). Cut this square, give it to its agent, and discard its shadow/s. The value of each shadow to the other agents is at most 1; each removed shadow removes one corner and decreases k by 1; hence, the remaining $n-1$ agents still value the remaining cake as at least $2(n-1)-2+k'$ (where k' is the new number of corners) and we can proceed recursively.

This shows that $\text{Prop}(k\,staircase, Squares, n) = 1/(2n-2+k)$. By letting $k=1$ we get a tight result: $\text{Prop}(QuarterPlane, Squares, n) = 1/(2n-1)$.

Algorithm 3. Assume that C is an unbounded plane. Similarly to Algorithm 1, partition it to four quarter-planes and partition the agents to four groups according to their valuations. Then, divide each quarter-plane to its agents using Algorithm 2. A calculation in [9] gives, for all $n \geq 4$: $\text{Prop}(Plane, Squares, n) \geq 1/(2n-4)$. The table below summarizes the results presented in this abstract:

Cake	Lower	Upper
Square	$1/(4n-4)$	$1/(2n)$
1/4-plane	$1/(2n-1)$	$1/(2n-1)$
Plane	$1/n$	$1/(2n-4)$

For an unbounded plane we do not have an upper bound (other than the trivial upper bound of $1/n$). Hence, we conclude with an open question: Is it possible to divide an unbounded plane such that each of n agents receives a square piece worth at least $1/n$? Is it possible to divide the plane "fair and square"?

References

[1] M. O. Albertson and C. J. O'Keefe. Covering Regions with Squares. *SIAM Journal on Algebraic Discrete Methods*, 2(3):240–243, Sept. 1981.

[2] P. V. M. Blagojević and G. M. Ziegler. Convex equipartitions via Equivariant Obstruction Theory. *Israel Journal of Mathematics*, 200(1):49–77, 2014.

[3] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, Feb. 1996.

[4] S. Even and A. Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, Mar. 1984.

[5] J. M. Keil. Polygon Decomposition. In *Handbook of Computational Geometry*, pages 491–518. University of Saskatchewan Saskatoon, Sask., Canada, 2000.

[6] R. Nandakumar and R. Rao. 'Fair' Partitions of Polygons - an Introduction. *Proceedings - Mathematical Sciences*, 122(3):459–467, Nov. 2010.

[7] A. D. Procaccia. Cake Cutting Algorithms. In Brandt, Conitzer, Endriss, Lang and Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13. Cambridge University Press, 2015.

[8] J. M. Robertson and W. A. Webb. *Cake-Cutting Algorithms: Be Fair if You Can*. A K Peters/CRC Press, first edition, July 1998.

[9] E. Segal-Halevi, S. Nitzan, A. Hassidim, and Y. Aumann. Fair and Square: Cake-Cutting in Two Dimensions, Oct. 2015. arXiv preprint 1510.03170.

[10] H. Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, Jan. 1948.