Fair Division of an Archipelago

Erel Segal-Halevi

January 30, 2019

Abstract

An archipelago of \( m \) islands has to be divided fairly among \( n \) agents with different preferences. What fraction of the total archipelago value can be guaranteed to each agent? Classic algorithms for fair cake-cutting can give each agent a share worth at least \( 1/n \) of the total value, but this share might be disconnected (spread over multiple islands). When each agent insists on getting a single connected piece (contained in a single island), it is shown that \( 1/(n + m - 1) \) of the total value can be guaranteed, and this fraction is tight. When each agent insists on getting at most \( k \) connected pieces, where \( 1 \leq k \leq m \), it is possible to guarantee at least \( k/(nk + m - k) \) and impossible to guarantee more than \( k/(n + m - 1) \). The paper presents several cases in which the upper bound can be attained. Whether it can always be attained remains a mystery.

Archipelago division has an application to a geometric problem — fair division of a two-dimensional land estate shaped as a rectilinear polygon, where each agent must receive a rectangular piece.

1 Introduction

Consider \( n \) people who inherit \( m \) distant land-estates ("islands") and want to divide the property fairly among them. Classic algorithms for fair cake-cutting (Steinhaus 1948, Even and Paz 1984) can be used to divide each island into \( n \) pieces such the value of piece \( i \), according to the personal value-measure of person \( i \), is at least \( 1/n \) of the total island value. However, this scheme requires each person to manage properties on \( n \) distant locations, which may be quite inconvenient. An alternative scheme is to consider the
union of all \( m \) islands (the “archipelago”) as a single cake, and partition it into \( n \) pieces using the above-mentioned algorithms. However, this too might give some agents a share that overlaps many distinct islands. For example, with \( m = 5 \) islands and \( n = 3 \) agents, a typical partition might look like this:

where Alice’s share contains 4 disconnected pieces and Carl’s share contains 2 disconnected pieces.

Can we find a more convenient division? For example, can we divide the archipelago fairly such that each agent receives at most 3 disconnected pieces? In general, what is the smallest number \( k \) (as a function of \( m \) and \( n \)) such that there always exists a fair division of the archipelago in which each agent receives at most \( k \) disconnected pieces?

This paper studies the following more general question. We are given an integer parameter \( k \geq 1 \), and it is required to give each agent a share with at most \( k \) disconnected pieces (i.e., the share of each agent may overlap at most \( k \) different islands). What is the largest fraction \( r(m, k, n) \) such that each agent can be guaranteed at least a fraction \( r(m, k, n) \) of his total archipelago value?

An obvious upper bound is \( r(m, k, n) \leq 1/n \). It becomes an equality when \( k \geq m \), since then we can just divide each island separately using classic cake-cutting algorithms.

The question becomes interesting when \( k < m \).

**Section 3** proves an upper bound of \( r(m, k, n) \leq k/(n + m - 1) \). In particular, when each agent insists on getting a single connected piece, at most \( 1/(n + m - 1) \) of the total value can be guaranteed.

**Section 4** shows that, when the islands are arranged arbitrarily on a line, classic cake-cutting algorithms can be adapted for dividing the line. With such adaptation it is possible to provide a lower bound (when \( k \leq m \)): \( r(m, k, n) \geq k/(nk + m - k) \).

When \( k = 1 \), this matches the upper bound, so \( r(m, k = 1, n) = 1/(n + m - 1) \). However, for \( k \geq 2 \) there is a substantial gap between the bounds. This hints that more sophisticated division algorithms may be able to attain a better value-guarantee. The following sections show that, indeed, the upper bound can be attained in several cases.
Section 5 shows that the upper bound is attainable when all agents have the same ranking on islands. I.e., they agree on which island is the most valuable, which is the second-most valuable, which is the least valuable, etc.

Section 6 shows that the upper bound is attainable when there are $n = 2$ agents (with arbitrary valuations), so $r(m, k, n = 2) = \min \left( \frac{1}{2}, \frac{k}{m+1} \right)$.

Section 7 shows that the upper bound is attainable when $k = 2$, so $r(m, k = 2, n) = \min \left( \frac{1}{n}, \frac{2}{m+n-1} \right)$.

The proofs in the paper are mostly based on elementary counting arguments and the pigeonhole principle. An exception is the proof in Section 7, which is based on a proof by Luria (2013) regarding the existence of a non-empty bipartite envy-free matching.

Since there are various cases in which the upper bound is attainable, it is reasonable to conjecture that this is always possible:

**Archipelago Conjecture.** For all $m \geq 1, k \geq 1, n \geq 1$:

$$r(m, k, n) = \min \left( \frac{1}{n}, \frac{k}{n+m-1} \right).$$

The smallest case in which this conjecture is open is $n = 3$ agents with three different rankings on islands, $k = 3$ pieces per agent and $m = 4$ islands: the lower bound is $3/(9+4-3) = 3/10$ and the upper bound is $1/3$. Some additional directions for future work are presented in Section 8.

The above results provide a partial answer to the dual question: if each agent should be guaranteed at least $1/n$ of the total value, how many pieces should each agent get (as a function of $m$ and $n$)? The upper bound on $r$ implies a lower bound $k \geq 1+(m-1)/n$. It is tight whenever the Archipelago Conjecture is true.

**Appendix A** presents an application of archipelago division to the problem of dividing a two-dimensional land estate shaped as a rectilinear polygon, when each agent needs to get a rectangular land-plot. If the land-estate has $T$ reflex vertices (vertices with internal angle $270^\circ$), then it is possible to allocate to each agent a rectangle worth at least $1/(n+T)$ of the total polygon value, and impossible to guarantee more.

### 1.1 Related work

Recently there is a lot of interest in various variants of the fair division problem. These are too many to list here. A relatively recent survey can be found
in the Wikipedia pages “Fair division”, “Fair cake-cutting”, “Proportional cake-cutting”, “Envy-free cake-cutting” and “Fair item assignment”.

Most works on cake-cutting either require to give each agent a single connected piece, or ignore connectivity altogether. The natural intermediate case, in which each agent should receive at most a fixed number \(k\) of disconnected pieces, has only recently been studied by Arunachaleswaran and Gopalakrishnan (2018). They consider a connected cake, i.e., a single island. They quantify the gain, as a function of \(k\), in the optimal social welfare, defined as the sum of values of all agents or the minimum value per agent. Based on their results, they conjecture that this gain grows linearly with \(\min(n,k)\). This is in par with our Archipelago Conjecture.

Two subsets of the cake-cutting literature are particularly related to the present paper.

Some works aim to minimize the number of cuts required to attain various fairness and efficiency goals [Webb, 1997; Shishido and Zeng, 1999; Barbanel and Brams, 2004, 2014; Alijani et al., 2017]. These works usually assume that the cake itself is connected. Moreover, their goal is to minimize the global number of cuts, while the goal in the present paper is to ensure that every individual agent is not given too many disconnected pieces.

Some works consider the multi-cake division problem [Cloutier et al., 2010; Lebert et al., 2013; Nyman et al., 2017], where several disjoint cakes are being divided. In these works, it is assumed that each agent must get a part of every cake. This is opposite to the present paper, in which each agent wants to overlap as few cakes (islands) as possible.

2 Preliminaries

There is a set \(C\) (“the archipelago”) that contains \(m \geq 1\) disjoint subsets (“the islands”). In most of the paper, it is assumed that \(C\) is a subset of the real line and the islands are pairwise-disjoint intervals; in particular, each island \(j \in \{1, \ldots, m\}\) corresponds to the open interval \((j-1, j)\), and the archipelago \(C\) corresponds to the interval \((0, m)\). This assumption is made for convenience only — it does not imply that there is any geographic proximity between e.g. the island \((0, 1)\) and the island \((1, 2)\).

There are \(n \geq 1\) agents. Each agent has a value-measure \(V_i\) on subsets of \(C\), which is an integral of an integrable value-density function \(v_i\): \(V_i(X) = \int_{x \in X} v_i(x)dx\). The notation \(V_i(x_1, x_2)\), where \(x_1, x_2\) are real numbers, is used
as a shorthand for $V_i((x_1, x_2)) =$ the value of the open interval $(x_1, x_2)$. Note that $V_i$ is absolutely continuous with respect to length, so the value of the closed interval $[x_1, x_2]$ is the same.

There is a fixed integer constant $k \geq 1$, which denotes the maximum number of disconnected pieces that an agent can use.

For each agent $i$, define the function $W_i$ on subsets of $C$, as the value of the $k$ most-valuable connected pieces contained in that subset. In particular, $W_i(C)$ is the value of the $k$ most valuable islands in the entire archipelago. If a subset $X \subseteq C$ is made of at most $k$ connected pieces, then $W_i(X) = V_i(X)$; otherwise, in general $W_i(X) \leq V_i(X)$. If agent $i$ is allocated a subset $X_i$ made of more than $k$ disconnected pieces, he can use only at most $k$ pieces, so the actual utility he gets from $X_i$ is $W_i(X_i)$.

The function $r(m, k, n)$ is defined as the maximum, over all partitions $(X_1, \ldots, X_n)$ of $C$, of $\min_i W_i(X_i)/V_i(C)$. I.e., it is the highest fraction that can be guaranteed to all agents under the constraint of giving each agent at most $k$ disconnected pieces. The goal of this paper is to calculate upper and lower bounds on $r(m, k, n)$.

An allocation $(X_1, \ldots, X_n)$ is called envy-free if for every two agents $i, j$, $W_i(X_i) \geq W_i(X_j)$. I.e., each agent feels that the best $k$ intervals in his own share are at least as good as the best $k$ intervals in any other share.

## 3 Upper Bound

Before presenting the archipelago division algorithms, it is useful to have an upper bound on what such algorithms can hope to achieve.

**Proposition 3.1.** When dividing an archipelago of $m$ islands among $n$ agents and each agent must get at most $k$ connected pieces, it is impossible to guarantee all agents more than:

$$\min \left( \frac{1}{n}, \frac{k}{n+m-1} \right) = \begin{cases} \frac{k}{n+m-1} & m+n-1 \geq nk \\ \frac{1}{n} & m+n-1 \leq nk \end{cases}$$

of their total archipelago value.

*Proof.* It is obviously impossible to guarantee more than $1/n$. It remains to prove that it is impossible to guarantee more than $k/(n+m-1)$. Suppose that all agents have the same value-measure: they value islands $1, \ldots, m-1$
as 1 and island \( m \) as \( n \). So the total archipelago value for each agent \( i \) is \( V_i(C) = n + m - 1 \). Now there are two cases:

1. At least one agent gets all his \( k \) pieces in \( k \) of the islands \( 1, \ldots, m - 1 \). This agent receives a value of at most \( k \).

2. All \( n \) agents get at least one of their pieces in island \( m \). At least one such agent receives a value of at most 1 from that island. This agent receives a value of at most \( k - 1 \) from his/her other \( k - 1 \) pieces; therefore his/her total value is at most \( k \).

So, at least one agent receives a value of at most \( \frac{k}{n + m - 1} V_i(C) \).

\[ \square \]

4 General Lower Bound

This section presents a lower bound that is valid for any \( m, k \) and \( n \).

**Theorem 4.1.** For all \( m \geq 1, k \geq 1, n \geq 1 \), it is possible to divide an archipelago of \( m \) islands among \( n \) agents, giving each agent at most \( k \) connected pieces with a total value of at least:

\[ \frac{k}{nk + \max[k, m] - k} = \begin{cases} 
\frac{k}{nk + m - k} & m \geq k \\
1/n & m \leq k 
\end{cases} \]

of his total archipelago value.

Two different proofs are given, each of which has an added benefit. The first proof (subsection 4.1) also provides a polynomial-time algorithm for finding the allocation; the second one (subsection 4.2) also guarantees envy-freeness — each agent believes that his share is at least as good as the shares of all others.

It is an open question whether both benefits can be attained simultaneously. Stromquist (2008) proved that, in the classic case in which \( m = 1 \) and \( k = 1 \), an envy-free allocation among \( n \geq 3 \) agents cannot be found using a finite number of queries. Aziz and Mackenzie (2016) proved that, without any restriction on the number of pieces per agent (i.e, \( k = \infty \)), an envy-free allocation can be found using a finite number of queries. The intermediate cases in which \( k \in [2, \infty) \) are still open.
In both proofs, the archipelago is first mapped into the real interval $(0, m)$, such that each island is mapped into an interval between two consecutive integers. So one (arbitrary) island is mapped into $(0, 1)$, another island is mapped into $(1, 2)$, etc., and the last one is mapped into $(m - 1, m)$.

The valuations of all agents are normalized such that for each agent $i$: $V_i(C) = nk + \max[k, m] - k$. The goal now is to give each agent a subset of $(0, m)$ that overlaps at most $k$ different islands (i.e., contains at most $k - 1$ integer points), and whose total value for the agent is at least $k$.

The proofs use the following technical lemma.

**Lemma 4.2.** Suppose a piece $X_i$ overlaps $m_i$ islands and has $V_i(X_i) \geq \max[k, m_i]$. Then agent $i$ can take from $X_i$ at most $k$ intervals whose total value is at least $k$, i.e.: $W_i(X_i) \geq k$.

**Proof.** If $m_i \leq k$, then the value of all $X_i$ is at least $k$, and the agent can take it all.

If $m_i > k$, then the value of the entire $X_i$ is at least $m_i$, so the average value per overlapped island is at least 1. By the pigeonhole principle, the $k$ most valuable intervals in $X_i$ are worth together at least $k$. \qed

### 4.1 Proof of theorem 4.1 with a polynomial-time algorithm

The division algorithm presented below is recursive on $n$. The pre-condition for each recursion step is that, for each agent $i$, the archipelago contains $m_i \leq m$ adjacent islands whose total value for $i$ is at least $nk + \max[k, m_i] - k$. I.e., for each $i$ there exists integers $d_i \geq 0$ and $m_i \geq 1$ such that:

$$V_i(d_i, d_i + m_i) \geq nk + \max[k, m_i] - k.$$  

When $n = 1$, the archipelago contains some $m_i$ islands whose total value for $i$ is at least $\max[k, m_i]$, so by Lemma 4.2 the agent’s utility from the entire archipelago is at least $k$.

When $n > 1$, we ask each agent $i$ to mark a point $x_i \in C$ in the following way. Define the following real function $u_i : (d_i, d_i + m_i) \rightarrow \mathbb{R}$:

$$u_i(x) := V_i(d_i, x) - \max[k, \text{ceil}(x - d_i)]$$

Intuitively, $u_i(x)$ is the value of the interval $(d_i, x)$, minus the number of islands it overlaps ($\text{ceil}(x - d_i)$), rounded up to $k$. This function has the following properties:
For every non-integer $x$, $u_i(x)$ is continuous and weakly-monotonically-increasing.

For every integer $x$, $u_i(x)$ is either continuous and weakly-monotonically-increasing (when $x - d_i < k$), or it has a discontinuous jump of size $-1$ (when $x - d_i \geq k$).

For $x \to (d_i)^+$, $u_i(x) \to 0 - k = -k$.

For $x \to (d_i + m_i)^-$, $u_i(x) \to V_i(d_i, d_i + m_i) - \max[k, m_i] \geq nk - k$.

Since $u_i(x)$ never jumps upwards, its image must contain all the values in $[-k, nk - k]$. An example is illustrated below for $d_i = 0, m_i = 11, k = 2, n = 4$, and a uniform value-measure:

Each agent $i$ marks a point $x_i$ such that $u_i(x_i) = 0$. If there is more than one such point, the agent selects one arbitrarily. The algorithm cuts at the leftmost mark $x_* := \min_i x_i$, gives $(0, x_*)$ to the leftmost cutter, and divides the remaining archipelago among the remaining $n - 1$ agents.

The leftmost cutter (say, agent $i$) receives $(0, x_i)$, which contains the interval $(d_i, x_i)$. By definition of $u_i$, this piece has $V_i(d_i, x_i) = \max[k, \text{ceil}(x_i - d_i)]$. It overlaps $\text{ceil}(x_i - d_i)$ islands. Hence, by Lemma 4.2, the agent can get from this piece at most $k$ sub-intervals with a total value of at least $k$.

It remains to prove that, for each remaining agent $j \neq i$, the remaining archipelago satisfies the precondition for recursion with $n - 1$ agents. It is sufficient to prove that:

$$V_j(x_j, d_j + m_j) \geq (n - 1) \cdot k + \max[k, \text{ceil}(d_j + m_j - x_j)] - k \quad (1)$$
For each remaining agent \( j \neq i \), \( x_j \geq x_i \), so the remaining archipelago contains the interval \((x_j, d_j + m_j)\). This interval overlaps \( \text{ceild}(d_j + m_j - x_j) \) islands, and its value is:

\[
V_j(x_j, d_j + m_j) = V_j(d_j, d_j + m_j) - V_j(d_j, x_j)
\]

\[
\geq (nk + \max[k, m_j] - k) - \max[k, \text{ceild}(x_j - d_j)] \quad \text{(by definition of } x_j) \\
\geq (n - 1) \cdot k + \max[k, m_j] - \max[k, \text{ceild}(x_j - d_j)]
\]

(2)

Define \( m_L := \text{ceild}(x_j - d_j) \) and \( m_R := \text{ceild}(d_j + m_j - x_j) \). By the properties of the ceiling, \( m_L + m_R \leq m_j + 1 \) (the sum equals \( m_j \) when \( x_j \) is integer and \( m_j + 1 \) otherwise). By comparing (1) and (2) and substituting the expressions for \( m_L, m_R, m_j \), it can be seen that it is sufficient to prove the following inequality:

\[(n - 1) \cdot k + \max[k, m_L + m_R - 1] - \max[k, m_L] \geq (n - 1) \cdot k + \max[k, m_R] - k \]

\[\iff \max[k, m_L + m_R - 1] \geq \max[k, m_L] + \max[k, m_R] - k \]

There are several cases depending on the relation between \( k, m_L, m_R \).

(*) If \( k \geq m_L + m_R - 1 \), then also \( k \geq m_L, k \geq m_R \) (since both \( m_L \) and \( m_R \) are at least 1), so both sides equal \( k \). So from now on assume \( m_L + m_R - 1 > k \) and consider the inequality:

\[m_L + m_R - 1 \geq \max[k, m_L] + \max[k, m_R] - k \]

(*) If \( k \geq m_L \), then the right-hand side is \( \max[k, m_R] \) and the left-hand side is larger than \( k \) and at least as large as \( m_R \).

(*) If \( k \geq m_R \), then the right-hand side is \( \max[k, m_L] \) and the left-hand side is larger than \( k \) and at least as large as \( m_L \).

(*) Otherwise, the right-hand side is \( m_L + m_R - k \) which is at most the left-hand side since \( k \geq 1 \).

Remark 4.3. The above algorithm requires \( n \) recursion steps. Using a halving technique similar to the algorithm of [Even and Paz (1984)](https://example.com), it is possible to reduce this to \( O(\log(n)) \).

When \( n \) is odd, the algorithm proceeds as above. When \( n \) is even, each agent \( i \) marks a “half point” \( x_i \) for which \( u_i(x_i) = nk/2 - k \) (recall that the range of \( u_i \) contains \([-k, nk-k]\) so such a point exists). If there is more than one such point, the agent selects one arbitrarily.
The algorithm then cuts the interval \((0, m)\) at the median of the agents’ half-points, and divides each part recursively among the \(n/2\) agents whose mark is in that part. So each agent \(i\) in the leftmost half shares an archipelago that contains the interval \((d_i, x_i)\), and each agent \(j\) in the rightmost half shares an archipelago that contains the interval \((x_j, d_j + m_j)\). Similarly to the above proof, it is possible to prove that \((d_i, x_i)\) contains some \(m_L := \lceil x_i - d_i \rceil\) adjacent islands whose total value for \(i\) is at least \(nk/2 + \max[k, m_L] - k\), and \((x_j, d_j + m_j)\) contains some \(m_R := \lceil m_j + d_j - x_j \rceil\) adjacent islands whose total value for \(j\) is at least \(nk/2 + \max[k, m_R] - k\), so the precondition for recursion is satisfied for all agents.

The number of agents in each subset is halved every at most two recursion steps, so at most \(2 \log_2(n)\) recursion steps are needed.

**Remark 4.4.** A simpler algorithm for the case \(k = 1\) was published as Example 4.2 in Segal-Halevi et al. (2017) and Lemma 11 in Segal-Halevi (2018). The total archipelago value is \(n + m - 1\) and it is required to give each agent a single connected piece with a value of at least 1. The algorithm is recursive on \(m\). When \(m = 1\), \(V_1(C) = n\), so classic algorithms can be used to give each agent a connected piece worth at least 1. Suppose \(m > 1\). Pick an arbitrary island and call it \(C'\). Order the agents in a descending order of \(V_1(C') \geq \cdots \geq V_n(C')\). Let \(n'\) be the largest index such that \(V_{n'}(C') \geq n'\) (analogously to the famous \(h\)-index used to evaluate researchers), or 0 if already \(V_1(C') < 1\).

If \(n' = 0\) then just discard \(C'\). Otherwise use a classic algorithm to divide \(C'\) among the *winners* — the agents 1, \ldots, \(n'\). By definition, each winner values \(C'\) as at least \(n'\). Hence classic algorithms guarantee each winner a connected subset of \(C'\) with value at least 1.

The \(n - n'\) losers value \(C'\) as less than \(n' + 1\), so they value the remaining archipelago \(C \setminus C'\) as more than \((n + m - 1) - (n' + 1) = (n - n') + (m - 1) - 1\). This is an archipelago of \(m - 1\) islands, so it can be divided recursively among the remaining \(n - n'\) agents, giving each agent a connected piece with value at least 1. Note that this is true even when \(n' = 0\).

A straightforward way to extend this simple algorithm to \(k > 1\) is to replace each agent with \(k\) “virtual agents” with the same value function, run the above algorithm, and then assign to each agent the union of the

\[1\text{The algorithm generalized an idea of Chris Culter in http://math.stackexchange.com/q/461675.}\]
intervals allocated to his virtual agents. This yields a value-guarantee of \( k/(nk + m - 1) \), which is worse than the one of Theorem 4.1.

4.2 Proof of theorem 4.1 with envy-freeness

Like in subsection 4.1, it is assumed that the islands are mapped in an arbitrary order into the \( m \) intervals \((0, 1), \ldots, (m - 1, m)\).

Consider any partition of the interval \((0, m)\) into \( n \) connected parts (intervals) \( X_1, \ldots, X_n \). In each such partition, it is possible to ask each agent \( i \) which of the \( n \) parts he prefers. The agent would answer such a question using the function \( W_i \), i.e., he would calculate for each part \( X_j \), the value of the best \( k \) sub-intervals in \( X_j \), and select the piece \( j \) with the highest \( W_i(X_j) \). This selection satisfies two properties: (1) An agent always weakly prefers a non-empty piece over an empty piece; (2) The set of partitions in which an agent prefers the piece with index \( j \), is a closed subset of the space of all partitions. This is because \( W_i \) is a continuous function. Stromquist (1980) and Su (1999) prove that, whenever the preferences of \( n \) agents satisfy these properties, there exists a connected envy-free partition of \((0, m)\). Let this partition be \((X_1, \ldots, X_n)\).

It remains to prove that, in this envy-free partition, each agent can find in his share at most \( k \) connected pieces with a value of at least \( k \), i.e., \( \forall i: W_i(X_i) \geq k \). Since the allocation is envy-free, it is sufficient to prove that for each \( i \) there exists a part \( X_j \) with \( W_i(X_j) \geq k \).

For the sake of proof, first discard all the parts \( X_j \) for which \( V_i(X_j) < k \). Suppose \( n - d \) such parts are discarded (for some \( d \in \{0, \ldots, n\} \)). Recall that the total archipelago value is normalized to \( nk + \max[k, m] - k \); therefore the total value of the remaining \( d \) parts is larger than \( (d - 1)k + \max[k, m] \). This expression is at least 0, so the remaining value is larger than 0, so at least one part remains, so in fact \( d \geq 1 \).

Suppose w.l.o.g. that the remaining parts are \( X_1, \ldots, X_d \). Suppose that each part \( X_j \) contains \( m_j \) sub-intervals. The total number of sub-intervals in \( X_1, \ldots, X_n \) is at most \( n + m - 1 \) (since there are \( m - 1 \) integer points and \( n - 1 \) cuts made by the algorithm). After removing \( n - d \) parts, each of which contains at least one interval, the total number of sub-intervals is at most
\(d + m - 1\), so:

\[
\sum_{j=1}^{d} m_j \leq d + m - 1.
\] (3)

On the other hand, the total value of the remaining parts is:

\[
\sum_{j=1}^{d} V_i(X_j) > (d - 1) \cdot k + \max[k, m] \geq (d - 1) \cdot 1 + m = d + m - 1
\] (4)

Comparing (3) and (4) implies that \(\sum_{j=1}^{d} V_i(X_j) \geq \sum_{j=1}^{d} m_j\), so for at least one \(j \in \{1, \ldots, d\}\):

\[V_i(X_j) \geq m_j.\]

Moreover, since we started by discarding the parts whose value is less than \(k\), in fact: \(V_i(X_j) \geq \max[k, m_j]\). Hence, by Lemma 4.2, \(W_i(X_j) \geq k\).  

**Remark 4.5.** When \(k = 1\), Theorem 4.1 is optimal — its value-guarantee is \(1/(n + m - 1)\) which exactly equals the upper bound of Proposition 3.1. Similarly, when \(k \geq m\), Theorem 4.1 guarantees \(1/n\) which is clearly optimal. However, when \(1 < k < m\), Theorem 4.1 does not match the upper bound. Intuitively, the reason is that it does not use all the freedom allowed by the problem — it begins by ordering the islands arbitrarily on a line, and then gives each agent a contiguous subset of that line; it does not try to re-arrange the islands based on their values. The following example shows that, with such an approach, it is indeed impossible to guarantee a larger value than Theorem 4.1.

Suppose all agents have the same value-measure: they value the \(m - 1\) leftmost islands \((0, 1), \ldots, (m - 2, m - 1)\) as 1, and the rightmost island \((m - 1, m)\) as \(nk - k + 1\), so the total archipelago value is \(nk + m - k\). Now there are three cases:

1. At least one agent gets all his \(k\) pieces in \(k\) of the islands \(1, \ldots, m - 1\). This agent receives a value of at most \(k\).

2. All \(n\) agents get a piece in island \(m\), and the leftmost agent (who gets the leftmost interval of \((0, m)\)) gets a value of at most 1 from island \(m\). This leftmost agent can get at most \(k - 1\) other islands, so his total value is at most \(k\).
3. All $n$ agents get a piece is island $m$, and the leftmost agent gets a value of more than 1 from island $m$. Then, the remaining value in island $m$ is less than $nk - k = (n - 1)k$. Therefore, at least one of the $n - 1$ remaining agents gets a value of less than $k$.

To overcome this impossibility, it is required to re-order the islands based on their value. The following sections present three different ways to do this, in three different situations.

## 5 Identical Island Ranking

This section shows that it is possible to get a value-guarantee that matches the upper bound of Proposition 3.1 whenever all agents agree on the island ranking. I.e., it is possible to map the $m$ islands to $(0, 1), (1, 2), \ldots, (m - 1, m)$, such that for every agent $i \in \{1, \ldots, n\}$, $V_i(0, 1) \leq V_i(1, 2) \leq \cdots \leq V_i(m - 1, m)$.

**Theorem 5.1.** It is possible to divide an archipelago of $m$ islands among $n$ agents that rank the islands in the same order, giving each agent at most $k$ connected pieces with a total value of at least:

$$
\min \left( \frac{1}{n}, \frac{k}{m + n - 1} \right) = \begin{cases} 
\frac{1}{n} & \text{when } m < nk - n + 1 \\
\frac{k}{(m + n - 1)} & \text{when } m \geq nk - n + 1
\end{cases}
$$

of his total archipelago value.

**Proof.** It is sufficient to prove the theorem for $m \geq nk - n + 1$, since otherwise we can add dummy zero-value islands so that the total island count becomes $nk - n + 1$; the value guarantee remains $k/kn = 1/n$.

Normalize the value-measures such that for every agent $i$, $V_i(C) = m + n - 1$. The goal is now to give each agent $i$ a subset $X_i \subset C$ that overlaps at most $k$ islands and has $V_i(X_i) \geq k$.

The algorithm is recursive on $n$. When $n = 1$, the archipelago contains $m \geq k$ islands and its total value is $m$, so the value of the $k$ most-valuable islands is at least $k$.

For $n > 1$ the algorithm proceeds as follows.

- Map the islands into $(0, m)$ from worst to best, such that the least-valuable island is mapped to $(0, 1)$ and the most-valuable island is mapped to $(m - 1, m)$.
• Ask each agent $i$ to specify an integer $l_i \in \{0, \ldots, m-k\}$ and a number 
$x_i \in [k-1, k)$ such that $V_i(l_i, l_i + x_i) = k$. In words, agent $i$ claims
$k-1$ whole islands, and a part (possibly empty) of a $k$-th island, whose
total value for him is exactly $k$. It will be shown below how each agent 
can find these numbers.

• Pick a “winner” — an agent $i$ with a smallest $l_i$. If two or more agents 
have the same smallest $l_i$, pick an agent with a smallest $x_i$. If two or 
more agents have the same smallest $l_i$ and $x_i$, pick one arbitrarily.

• Give the winner his claimed piece. The winner now has at most $k$
connected intervals with a total value of exactly $k$, so he can go home 
happily.

• The remaining archipelago has $m - k + 1$ islands; divide it recursively 
among the $n-1$ remaining agents. It will be shown below that, for each 
agent who follows the algorithm below for choosing $l_i$ and $x_i$, the value 
of the piece given to the winner is at most $k$. Hence the value of the 
remaining archipelago is at least $(m+n-1) - k = (m-k+1)+(n-1)-1$, 
so the precondition for recursion is satisfied.

It remains to specify how each agent $i$ selects $l_i$ and $x_i$. The selection of $l_i$
should satisfy the following properties:

1. $V_i(l_i, l_i + k - 1) \leq k$;
2. $V_i(l_i, l_i + k) > k$.

Once $l_i$ is found, it is easy to find an $x_i \in [k-1, k)$ such that $V(l_i, l_i + x_i) = k$.

Define a function $U_i$, which assigns to each $X \subseteq C$ the value of $X$ minus 
the number of islands overlapping $X$. So $U_i(C) = V_i(C) - m = (m + n - 1) - m = n - 1$.

Using this notation, the goal of agent $i$ is to find $l_i$ with the following 
properties:

1. $U_i(l_i, l_i + k - 1) \leq 1$;
2. $U_i(l_i, l_i + k) > 0$.

It is easy to satisfy each property on its own:
1. Property 1 is satisfied by $l_i = 0$. **Proof:** the interval $(0, k - 1)$ contains the least-valuable $k - 1$ islands. By assumption, $C$ contains at least $nk - n + 1$ islands. So by the pigeonhole principle, $U_i(0, k - 1) \leq \frac{k-1}{nk-n+1} U_i(C) < \frac{k-1}{n(k-1)}(n-1) = \frac{1}{n}(n-1) < 1$.

2. Property 2 is satisfied by $l_i = m - k$. **Proof:** the interval $(m - k, m)$ contains the most-valuable $k$ islands. By the pigeonhole principle, $U_i(m - k, m) \geq \frac{k}{m} U_i(C) = \frac{k}{m}(n-1) > 0$.

To find an $l_i$ that satisfies both properties, initialize $l_i := 0$, so it satisfies property 1. If $l_i$ satisfies property 2 too, then we are done. Otherwise, we have $U_i(l_i, l_i+k) \leq 0$. Set $l'_i := l_i+1$. Now, $U_i(l'_i, l'_i+k-1) = U_i(l_i+1, l_i+k) \leq U_i(l_i, l_i+k) + 1 \leq 0 + 1 = 1$, so $l'_i$ satisfies property 1 too. Set $l_i := l'_i$ and continue.

Since $m - k$ satisfies property 2, there exists a smallest integer $l_i \leq m - k$ that satisfies property 2. By the above argument, this smallest $l_i$ satisfies property 1 too. So agent $i$ can report $l_i$ and $x_i$ as required.

It remains to show that the piece given to agent $i$ is worth at most $k$ for the $n - 1$ agents $j \neq i$. Since $l_i$ is minimal, there are two cases:

- If $l_j = l_i$, then the piece $(l_i, l_i+x_i)$ is contained in the piece $(l_j, l_j+x_j)$ since the winner is selected such that $x_i \leq x_j$. Therefore $V_j(l_i, l_i+x_i) \leq V_j(l_j, l_j+x_j) = k$.

- If $l_j > l_i$, then, assuming agent $j$ followed the above strategy for picking $l_j$, he has $U_j(l_i, l_i+k) \leq 0$. Therefore, $V_j(l_i, l_i+k) \leq k$, so $V_j(l_i, l_i+x_i) \leq k$ too. \[\Box\]

Theorem 5.1 can be slightly strengthened: it works even when only $n - 1$ agents have the same islands ranking.

**Theorem 5.2.** Given an archipelago of $m$ islands, and $n$ agents of whom at least $n - 1$ rank the islands in the same order, it is possible to give each agent at most $k$ connected pieces with a total value of at least:

$$\min\left(\frac{1}{n}, \frac{k}{m+n-1}\right) = \begin{cases} 1/n & \text{when } m < nk - n + 1 \\ k/(m+n-1) & \text{when } m \geq nk - n + 1 \end{cases}$$

of his total archipelago value.
Proof. Suppose that agents 1, . . . , n − 1 have the same ranking. Apply the algorithm of Theorem 5.1 with these n − 1 agents. Let \( i \in \{1, \ldots, n - 1\} \) be the winning agent and \((l_i, x_i)\) the winning bid. Ask agent \( n \) to evaluate the piece \((l_i, l_i + x_i)\):

- If \( V_n(l_i, l_i + x_i) \leq k \), then give this piece to agent \( i \);
- Otherwise, give this piece to agent \( n \).

In both cases, by the same arguments of Theorem 5.1, the remaining archipelago contains \( m - k + 1 \) islands and its value for all remaining \( n - 1 \) agents is at least \((m - k + 1) - (n - 1) - 1\), so the precondition for recursion holds. \( \square \)

6 Two Agents

This section proves that it is possible to attain the upper bound of Proposition 3.1 when there are \( n = 2 \) agents. While this result is implied by Theorem 5.2, the proof below has an added benefit — it also guarantees envy-freeness.

Theorem 6.1. It is possible to divide an archipelago of \( m \) islands between 2 agents, giving each agent at most \( k \) connected pieces with a total value of at least:

\[
\min \left( \frac{1}{2}, \frac{k}{m+1} \right) = \begin{cases} 
\frac{1}{2} & \text{when } m < 2k - 1 \\
\frac{k}{m+1} & \text{when } m \geq 2k - 1
\end{cases}
\]

of his total archipelago value. Moreover, the allocation is envy-free and can be found by an efficient algorithm.

Proof. It is sufficient to prove the theorem for \( m \geq 2k - 1 \), since if \( m < 2k - 1 \), we can add \( 2k - 1 - m \) dummy islands whose value for both agents is 0, and get the same value guarantee of \( k/(2k - 1 + 1) = 1/2 \).

One of the agents, say Alice, evaluates all \( m \) islands, orders them in increasing order of their value, and arranges them on the interval \((0, m)\) in the following way. The first (least-valuable) island is mapped to the leftmost subinterval \((0, 1)\). The second island is mapped to the rightmost subinterval \((m - 1, m)\). The third is mapped to \((1, 2)\), the fourth is mapped to \((m - 2, m - 1)\), and so on. So, the less valuable islands are mapped to the left and right ends of \((0, m)\) alternately, while the more valuable islands are mapped
to the center of $(0, m)$. Note that this ordering is different than the one used in Theorem 5.1.

Alice cuts the interval $(0, m)$ into two pieces that are equivalent in her eyes, i.e., the best $k$ intervals in the leftmost half have the same value for her as the best $k$ intervals in the rightmost half. Formally, Alice picks some $x_A \in (0, m)$ such that $W_A(0, x_A) = W_A(x_A, m)$. There exists such $x_A$ by the intermediate value theorem, since $W_A(0, x)$ is a continuous function of $x$ that increases from 0 towards $W_A(C)$, and $W_A(x, m)$ is a continuous function of $x$ that decreases from $W_A(C)$ towards 0.

Bob now chooses the half that he prefers (the half with the larger $W_B$) and takes his best $k$ intervals from it, and Alice takes her best $k$ intervals from the remaining half. This obviously yields an envy-free allocation. It remains to prove the value guarantee.

We first prove an auxiliary claim: we prove that Alice can always make her cut in $[k - 1, m - k + 1]$, i.e., she never has to cut inside one of the $k - 1$ leftmost islands or inside one of the $k - 1$ rightmost islands. Proof of auxiliary claim. Consider first the leftmost islands — those mapped to $(0, k - 1)$. For each such island, the next island in Alice’s ordering is mapped to $(m - k + 1, m)$. Therefore, $V_A(0, k - 1) \leq V_A(m - k + 1, m)$. Since both these intervals overlap less than $k$ islands, $W_A(0, k - 1) \leq W_A(m - k + 1, m)$ too. The assumption $m \geq 2k - 1$ implies $k - 1 < k \leq m - k + 1$, so $W_A(0, k - 1) \leq W_A(k - 1, m)$. Hence Alice has a halving-point to the right of $k - 1$: $x_A \geq k - 1$. A similar consideration applies to the rightmost islands — those mapped to $(m - k + 1, m)$. For each such island, the next island in Alice’s ordering is mapped to $(1, k)$. Therefore, $V_A(m - k + 1, m) \leq V_A(1, k) \leq V_A(0, k)$. Since both these intervals overlap at most $k$ islands, $W_A(m - k + 1, m) \leq W_A(0, k)$ too. The assumption $m \geq 2k - 1$ implies that $m - k + 1 \geq k$, so $W_A(m - k + 1, m) \leq W_A(0, m - k + 1)$. Hence Alice has a halving-point to the left of $m - k + 1$: $x_A \leq m - k + 1$. This completes the proof of the auxiliary claim.

The auxiliary claim shows that we can actually require Alice to make her cut in $[k - 1, m - k + 1]$, so that each half contains at least $k - 1$ whole islands. Moreover, if Alice’s cut is in $(k - 1, m - k + 1)$, then each half contains at least $k$ sub-intervals. In the edge cases in which $x_A = k - 1$ or $x_A = m - k + 1$, we add to one of the halves, a dummy interval whose value is 0, so that in

\begin{itemize}
  \item[2] The proof is based on ideas by lulu and Gregory Nisbet in https://math.stackexchange.com/q/3045731/29780.
\end{itemize}
all cases, each half contains at least \( k \) sub-intervals.

Normalize the valuations such that the total archipelago value for each agent is \( m + 1 \). Denote the two halves by \( X_L, X_R \). Suppose that \( X_L \) contains some \( m_L \geq k \) intervals and \( X_R \) contains some \( m_R \geq k \) intervals. The total number of intervals in both halves is \( m_L + m_R \leq m + 1 \) (this is true whether or not we have added a dummy interval). Since \( m_L + m_R \leq V_i(X_L) + V_i(X_R) \), for each agent \( i \), there exists some index \( j \in \{L, R\} \) such that \( m_j \leq V_i(X_j) \). Since \( m_j \leq k \), we have \( V_i(X_j) \geq \max[k, m_j] \). Hence, by Lemma 4.2 \( W_i(X_j) \geq k \). Since the division is envy-free, each agent’s final utility is at least \( k \).

**Remark 6.2.** One could think of other ways to obtain an envy-free division of an archipelago. For example, the islands can be treated as *indivisible goods*. Then, the *envy-cycles* algorithm of Lipton et al. (2004) can be used to find a so-called *EF1 allocation* of the islands — an allocation that is envy-free up to at most a single island. Note that this algorithm works even for non-additive valuations so it can be used with the \( W_i \) functions. Once the envy is limited to a single island, this island can be divided such that envy is completely eliminated.

The problem with this scheme is that it does not guarantee each agent at least \( k/(m + 1) \) of the total archipelago value. In general, when valuations are not additive, envy-freeness does not necessarily guarantee a high value to all agents. The challenge is to simultaneously guarantee both envy-freeness and a high value per agent.

### 7 Two Pieces Per Agent

This section shows that the upper bound can be attained whenever \( k = 2 \).

The proof will use the concept of *bipartite envy-free matching* (Luria, 2013).

**Definition 7.1.** Let \( G = (X + Y, E) \) be a bipartite graph. A *bipartite envy-free matching* (BEFM) in \( G \) is a matching between a subset \( X_M \subseteq X \) and a subset \( Y_M \subseteq Y \) such that no unmatched vertex in \( X \) is adjacent to a matched vertex in \( Y \), i.e.:

\[
\forall x \in X \setminus X_M : \forall y \in Y_M : (x, y) \not\in E.
\]

An unmatched \( x \) does not “envy” any matched \( x' \), because it does not “like” any matched \( y' \) anyway. Any perfect matching is envy-free, and the
empty matching is envy-free too. The following theorem provides a sufficient condition for the existence of a non-empty BEFM (here \( N_E(X) \) is the set of neighbors of \( X \)):

**BEFM theorem.** Let \( G = (X + Y, E) \) be a bipartite graph. If \( |N_E(X)| \geq |X| \), then \( G \) has a non-empty bipartite-envy-free matching.

Note that this condition is strictly weaker than Hall’s condition for the existence of a perfect matching, which requires \( |N_E(X')| \geq |X'| \) to hold for all subsets \( X' \subseteq X \). The BEFM theorem was proved by Luria (2013). The proof can be converted to a polynomial-time algorithm that finds a BEFM when the sufficient condition holds; see Segal-Halevi (2019).

**Theorem 7.2.** It is possible to divide an archipelago of \( m \) islands among \( n \) agents, giving each agent at most \( k = 2 \) connected pieces with a total value of at least:

\[
\min \left( \frac{1}{n}, \frac{2}{m + n - 1} \right) = \begin{cases} 
\frac{1}{n} & \text{when } m < n + 1 \\
\frac{2}{m + n - 1} & \text{when } m \geq n + 1
\end{cases}
\]

of his total archipelago value.

**Proof.** If \( m < n + 1 \), add dummy islands to make \( m = n + 1 \); the value-guarantee remains the same \((1/n)\). Normalize the archipelago value to \( m + n - 1 \). The goal now is to give each agent a piece overlapping at most \( 2 \) islands with a value of at least \( 2 \). The algorithm is recursive on \( n \).

If \( n = 1 \), then the single agent values the archipelago as \( m \) and \( m \geq 2 \), so the agent values the two most valuable islands as at least \( 2 \).

Suppose \( n > 1 \). Define an island as barren if all \( n \) agents value it as at most \( 2 \). The algorithm proceeds according to the number of barren islands.

**Case #1.** All \( m \) islands are barren. Map the islands into \((0, m)\) as in Theorem 5.1 according to the ranking of one of the agents arbitrarily (say, Alice). Ask Alice to find an integer \( l_A \in \{0, \ldots, m - 1\} \) for which \( V_A(l_A, l_A + 1) \leq 2 \) while \( V_A(l_A, l_A + 2) > 2 \); she can find such \( l_A \) in exactly the same way as in the proof of Theorem 5.1.

Since all islands are barren, for every agent \( i \), \( V_i(l_A, l_A + 1) \leq 2 \). So for every agent \( i \) for whom \( V_i(l_A, l_A + 2) > 2 \), there exists some \( x_i \in [1, 2) \) such that \( V_i(l_A, l_A + x_i) = 2 \). There exists at least one such agent (Alice). For every agent \( j \) for whom \( V_j(l_A, l_A + 2) \leq 2 \), define \( x_j = \infty \). Choose an
agent with a smallest $x_i$ (breaking ties arbitrarily) and give him/her the piece $(l_A, l_A + x_i)$. This piece overlaps two islands and its value for $i$ is 2, so $i$ can go home happily. The remaining archipelago contains $m - 1$ islands. The remaining $n - 1$ agents value the piece given to $i$ as at most 2, so they value the remaining islands as at least $(m + n - 1) - 2 = (m - 1) + (n - 1) - 2$, so they can divide it recursively among them.

**Case #2.** There are between 1 and $m - 1$ barren islands. Pick one barren island and map it to $(0, 1)$. Pick one non-barren island and map it to $(1, 2)$. By definition, for every agent $i$, $V_i(0, 1) \leq 2$. So for every agent $i$ for whom $V_i(0, 2) > 2$, there exists some $x_i \in [1, 2)$ such that $V_i(0, x_i) = 2$. Since $(1, 2)$ is not barren, there exists at least one such $i$. For every agent $j$ for whom $V_j(0, 2) \leq 2$, define $x_j = \infty$. Similarly to Case #1, choose an agent $i$ with a smallest $x_i$ and give him/her the piece $(0, x_i)$. Agent $i$ goes home happily, and the remaining $n - 1$ agents can divide the remaining $m - 1$ islands recursively among them.

**Case #3.** No island is barren, i.e, each island is valued as more than 2 by at least one agent. Consider the bipartite graph in which the agents are on one side, the islands are on the other side, and there is an edge from agent $i$ to island $j$ iff agent $i$ values island $j$ as more than 2. Since no island is barren, the number of neighbors of all $n$ agents is $m$. Since $m \geq n$, the BEFM theorem implies that the graph has a bipartite envy-free matching of islands to agents. Give each matched agent his matched island, and recurse with the remaining agents. Let $l \geq 1$ be the size of the matching. The remaining archipelago contains $m - l$ islands. There are $n - l$ remaining agents. By the envy-freeness of the matching, each remaining agent is not connected to any matched island. So each remaining agent values the allocated islands at most $2l$, and the remaining archipelago at least $(m + n - 1) - 2l = (m - l) + (n - l) - 1$. So the precondition for recursion is met. □

## 8 Future Work

In light of the various cases in which the upper bound of Proposition 3.1 can be attained, it is reasonable to conjecture that it can always be attained:

**Archipelago Conjecture.** It is possible to divide an archipelago of $m$ islands among $n$ agents, giving each agent at most $k$ pieces with a total value
of at least:

$$\min\left(\frac{1}{n}, \frac{k}{m+n-1}\right) = \begin{cases} 1/n & \text{when } m < nk - n + 1 \\ k/(m+n-1) & \text{when } m \geq nk - n + 1 \end{cases}$$

of the total archipelago value.

The conjecture is proved whenever $k \geq m$ or $k \leq 2$ or $n \leq 2$ or when at least $n - 1$ agents rank the islands in the same order. Thus, the smallest case in which the conjecture is open is $n = 3$ agents with three different rankings on islands, $k = 3$ pieces per agent and $m = 4$ islands: the lower bound is $3/(9 + 4 - 3) = 3/10$ and the upper bound is $1/3$.

Another question for the future is what happens if different agents have different preferences for the number of pieces: some agents insist on getting a small number of pieces even if that entails a smaller value-guarantee, while others insist on getting a large value even if it requires many pieces. Is it possible to give a personalized guarantee to each agent? For example, suppose $n = 2$ and each agent $i \in \{1, 2\}$ wants at most $k_i$ pieces, is it possible to guarantee to agent $i$ a value of at least $k_i/(m + 1)$?

Finally, it may be interesting to replace the hard constraint of at most $k$ pieces per agent, with a soft model where the number of pieces decreases the value by some known amount. For example, suppose that managing pieces on $k$ different islands incurs an expense of $E(k)$. Then the value of agent $i$ from an allotment $X_i$ overlapping $k_i$ islands is $V_i(X_i) - E(k_i)$. Given some reasonable assumptions on the function $E$, what value-guarantees are possible?

Acknowledgments

I am grateful to Chris Culter, Zur Luria, Yuval Filmus, Gregory Nisbet and lulu for their helpful ideas.

The research was partly funded by the Doctoral Fellowships of Excellence Program at Bar-Ilan University, the Mordecai and Monique Katz Graduate Fellowship Program, and the Israel Science Fund grant 1083/13.
A Application: Dividing a Rectilinear Polygon

The archipelago division algorithm can be used to fairly divide a two-dimensional land-estate shaped like a rectilinear polygon — a polygon whose angles are 90° or 270°.

Rectilinearity is a common assumption in polygon partition problems [Keil, 2000]. The “complexity” of a rectilinear polygon is characterized by the number of its reflex vertices — vertices with a 270° angle. We denote this number by $T$. A rectangle — the simplest rectilinear polygon — has $T = 0$. The polygon below has $T = 4$ reflex vertices (circled):

A natural requirement in land division settings is to give each agent a rectangular piece. This requirement can be satisfied using an archipelago division algorithm.

**Theorem A.1.** It is possible to divide a rectilinear polygon with $T$ reflex vertices among $n$ agents giving each agent a rectangle with value at least $1/(n + T)$ of his total polygon value.

**Proof.** Keil (2000); Eppstein (2010) present efficient algorithms for partitioning a rectilinear polygon into a minimal number of rectangles. A rectilinear polygon with $T$ reflex vertices can be partitioned in time $O(poly(T))$ into at most $T + 1$ rectangles (this number is tight when the vertices of $C$ are in general position). Using such algorithms, partition $C$ into at most $T + 1$ rectangular “islands” and then apply the theorems for archipelago division with $m = T + 1$. By setting $k = 1$ in Theorem 4.1, each agent $i$ gets a single rectangle worth at least $V_i(C)/(n + T)$.

If each agent is willing to receive up to $k$ rectangles, then the value-guarantee increases accordingly. In particular, if the Archipelago Conjecture is true, then the guarantee improves to min $(1/n, k/(n + T))$. 

22
Theorem A.1 is tight in the following sense:

**Proposition A.2.** for every integer \( T \geq 0 \), there exists a rectilinear polygon with \( T \) reflex vertices, in which it is impossible to guarantee every agent a rectangle with a value of more than \( 1/(n + T) \) the total polygon value.

**Proof.** Consider a staircase-shaped polygon with \( T + 1 \) stairs (illustrated for \( T = 4 \)):

All agents have the same value-measure, which is concentrated in the diamond-shapes: the top diamond is worth \( n \) and each of the other diamonds is worth 1. So for all agents, the total polygon value is \( n + T \).

Any rectangle in \( C \) can touch at most a single diamond. There are two cases:

- At least one agent touches one of the \( T \) bottom diamonds. Then, the value of that agent is at most 1.
- All \( n \) agents touch the top diamond. Then, their total value is \( n \) and at least one of them must receive a value of at most 1.

So, either some or all of the agents receive a value of at most \( V_i(C)/(n + T) \).

**References**

Reza Alijani, Majid Farhadi, Mohammad Ghodsi, Masoud Seddighin, and Ahmad S Tajik. 2017. Envy-Free Mechanisms with Minimum Number of Cuts.. In *AAAI*. 312–318.

Eshwar Ram Arunachaleswaran and Ragavendran Gopalakrishnan. 2018. The Price of Indivisibility in Cake Cutting. arXiv preprint 1801.08341.
Haris Aziz and Simon Mackenzie. 2016. A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents. In FOCS 2016. 416–427. arXiv:1604.03655 http://arxiv.org/abs/1604.03655

Julius B. Barbanel and Steven J. Brams. 2004. Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond. Mathematical Social Sciences 48, 3 (Nov. 2004), 251–269. https://doi.org/10.1016/j.mathsocsci.2004.03.006

Julius B. Barbanel and Steven J. Brams. 2014. Two-Person Cake Cutting: The Optimal Number of Cuts. The Mathematical Intelligencer 36, 3 (2014), 23–35. https://doi.org/10.1007/s00283-013-9442-0

John Cloutier, Kathryn L. Nyman, and Francis E. Su. 2010. Two-player envy-free multi-cake division. Mathematical Social Sciences 59, 1 (16 Jan. 2010), 26–37. https://doi.org/10.1016/j.mathsocsci.2009.09.002

David Eppstein. 2010. Graph-Theoretic Solutions to Computational Geometry Problems. In Graph-Theoretic Concepts in Computer Science, Christophe Paul and Michel Habib (Eds.). Lecture Notes in Computer Science, Vol. 5911. Springer Berlin Heidelberg, 1–16. https://doi.org/10.1007/978-3-642-11409-0_1 arXiv:0908.3916v1.pdf

Shimon Even and Azaria Paz. 1984. A Note on Cake Cutting. Discrete Applied Mathematics 7, 3 (March 1984), 285–296. https://doi.org/10.1016/0166-218x(84)90005-2

J. Mark Keil. 2000. Polygon Decomposition. In Handbook of Computational Geometry. University of Saskatchewan Saskatoon, Sask., Canada, 491–518. https://doi.org/10.1016/B978-044482537-7/50012-7

Nicolas Lebert, Frédéric Meunier, and Quentin Carbonneaux. 2013. Envy-free two-player m-cake and three-player two-cake divisions. Operations Research Letters 41, 6 (Nov. 2013), 607–610. https://doi.org/10.1016/j.orl.2013.07.010

R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. 2004. On approximately fair allocations of indivisible goods. In Proceedings of the 5th ACM conference on Electronic commerce (EC ’04). ACM, New York, NY, USA, 125–131. https://doi.org/10.1145/988772.988792
Zur Luria. 2013. Envy-free matching. Mathematics Stack Exchange. [https://math.stackexchange.com/q/621306/29780](https://math.stackexchange.com/q/621306/29780) URL: https://math.stackexchange.com/q/621306/29780 (version: 2013-12-29).

Kathryn Nyman, Francis E. Su, and Shira Zerbib. 2017. Fair division with multiple pieces. arXiv:1710.09477 [http://arxiv.org/abs/1710.09477](http://arxiv.org/abs/1710.09477)

Erel Segal-Halevi. 2018. Redividing the Cake. In Proceedings of IJCAI’18. arXiv:1603.00286 arXiv preprint 1603.00286.

Erel Segal-Halevi. 2019. Bipartite Envy-Free Matching. arXiv preprint 1901.09527.

Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Au- mann. 2017. Fair and square: Cake-cutting in two dimensions. Journal of Mathematical Economics 70 (Feb. 2017), 1–28. [https://doi.org/10.1016/j.jmateco.2017.01.007](https://doi.org/10.1016/j.jmateco.2017.01.007)

Harunor Shishido and Dao-Zhi Zeng. 1999. Mark-Choose-Cut Algorithms For Fair And Strongly Fair Division. Group Decision and Negotiation 8, 2 (1999), 125–137. [https://doi.org/10.1023/a:1008620404353](https://doi.org/10.1023/a:1008620404353)

Hugo Steinhaus. 1948. The problem of fair division. Econometrica 16, 1 (jan 1948), 101–104. [http://www.jstor.org/stable/1914289](http://www.jstor.org/stable/1914289)

Walter Stromquist. 1980. How to Cut a Cake Fairly. The American Mathematical Monthly 87, 8 (Oct. 1980), 640–644. [https://doi.org/10.2307/2320951](https://doi.org/10.2307/2320951)

Walter Stromquist. 2008. Envy-free cake divisions cannot be found by finite protocols. Electronic Journal of Combinatorics 15, 1 (Jan. 2008), #R11. [http://www.emis.ams.org/journals/EJC/Volume_15/PDF/v15i1r11.pdf](http://www.emis.ams.org/journals/EJC/Volume_15/PDF/v15i1r11.pdf) Research paper 11, 10 pp., 91B32.

Francis E. Su. 1999. Rental Harmony: Sperner’s Lemma in Fair Division. The American Mathematical Monthly 106, 10 (Dec. 1999), 930–942. [https://doi.org/10.2307/2589747](https://doi.org/10.2307/2589747)

William A. Webb. 1997. How to cut a cake fairly using a minimal number of cuts. Discrete Applied Mathematics 74, 2 (April 1997), 183–190. [https://doi.org/10.1016/s0166-218x(96)00032-7](https://doi.org/10.1016/s0166-218x(96)00032-7)