Submitted to *Econometrica*

INCENTIVE-COMPATIBLE AND STRONGLY FAIR CAKE CUTTING MECHANISMS FOR GENERAL VALUATIONS

JENS WITKOWSKI
Frankfurt School of Finance & Management

RUPERT FREEMAN
Darden School of Business, University of Virginia

The classical cake cutting setting is concerned with dividing a resource, modeled by the $[0, 1]$ interval, and allocating subintervals to different agents. A recent result shows that there does not exist a deterministic cake cutting mechanism that is both incentive compatible and even only one of proportional or envy-free (the latter is restricted to non-wasteful mechanisms). In principle, randomization can circumvent this impossibility, but known solutions either require restrictive assumptions on the valuation functions or only provide non-constructive existence arguments. In this work, leveraging proper scoring rules, we design a class of randomized mechanisms that are ex ante incentive compatible, ex ante proportional, and ex ante envy-free. Moreover, within this class, we identify the Seeded Probabilistic Allocation mechanism, which is ex ante *strictly* incentive compatible, *ex post strongly* proportional and *ex post strongly* envy-free. This result is tight in the sense that additionally achieving ex post incentive compatibility is impossible as it would violate the aforementioned impossibility result.

KEYWORDS: fair resource allocation, mechanism design, proper scoring rules.

Jens Witkowski: j.witkowski@fs.de

Rupert Freeman: FreemanR@darden.virginia.edu

We thank Aurélien Baillon, Fabian Krüger, Nicolas Lambert, Hervé Moulin, David C. Parkes, Ariel Procaccia, Clemens Puppe, Omer Tamuz, as well as the attendees of the Duke Fuqua Decision Sciences seminar, the USC Economic Theory Group Research seminar, the emLyon Quantitative Methods in Business seminar, the KIT Workshop on Economics, Finance and Statistics, and the 2024 INFORMS Advances in Decision Analysis conference for helpful feedback. Both authors contributed equally to this manuscript.

1. INTRODUCTION

The cake-cutting problem models a situation in which a divisible resource must be split among several agents (Steinhaus (1948)). The resource (the "cake") is heterogeneous and agents may disagree on the relative value of different parts of the cake. The cake-cutting problem captures a range of real-world settings such as negotiating land borders (Chambers (2005)), divorce settlements (Brams and Taylor (1996)), and sharing some valuable resource such as a supercomputer across time (Procaccia (2013)).

Slightly more formally, the cake is modeled as the $[0,1]$ interval, with each of n agents having a valuation function that induces an additive utility for every (positive-measure) subset of $[0,1]$. Given these valuation functions, the goal is to divide the cake and allocate the resulting pieces in a way that is fair to all agents. Two particularly prominent notions of fairness are *proportionality* (Steinhaus (1948)), which dictates that every agent should receive her "fair share," i.e., a piece of cake that she values at least as much as receiving a random piece of length $1/n$, and *envy-freeness* (Foley (1967)), which says that no agent should prefer someone else's piece to her own. In addition to these fairness properties, we would like our procedure ("mechanism") to be *incentive compatible*, ensuring that no agent can benefit from misreporting her true valuation.

Dating back to the original work of Steinhaus, the cake-cutting literature has largely been focused on achieving fairness. (For a general overview, we point the reader to the survey by Procaccia (2016).) A recent line of work has focused on additionally achieving incentive compatibility, but results have been mixed. Building on several earlier results (Aziz and Ye (2014), Bei et al. (2017), Menon and Larson (2017)), an explanation for these mixed results was recently provided by Bu et al. (2023), who showed that no deterministic cake-cutting mechanism can simultaneously satisfy incentive compatibility and proportionality. Moreover, the impossibility continues to hold if proportionality is replaced by envy-freeness and non-wastefulness (which requires that the entire cake is allocated), and even when valuations are restricted to being piecewise constant (i.e., "step functions").

At a higher level, there are two approaches to circumvent this strong impossibility while retaining the desired incentive and fairness guarantees: further restricting the space of admissible valuation functions and allowing randomization. (For a summary of the properties achieved by known incentive-compatible mechanisms, including those introduced in this work, see Table I.) With respect to further restricting the space of admissible valuation functions, the only positive results for deterministic, incentive-compatible, and proportional cake cutting mechanisms requires restricting to piecewise uniform valuations, a highly constrained class in which every agent's valuation function takes at most two values (0 and an appropriately normalized constant). For this restrictive setting, Chen et al. (2013) design a deterministic mechanism that is incentive compatible, proportional, and envy-free, but wasteful. For two agents, Bei et al. (2020) achieve the same properties while also guaranteeing non-wastefulness.

Randomization has proven successful in generalizing positive results beyond piecewise uniform valuation functions. For general valuations, it is well known that a division of the cake into n pieces with every agent having utility exactly $1/n$ for every piece (a "perfect partition") always exists (Neyman (1946)). All known (randomized) mechanisms that achieve both incentive compatibility and proportionality heavily rely on, and typically return, perfect partitions. The main idea underlying these mechanisms is to assign pieces of a perfect partition to agents uniformly at random, thus guaranteeing ex ante incentive compatibility along with ex post proportionality and ex post envy-freeness (Mossel and Tamuz (2010), Chen et al. (2013)). Note however that perfect partitions do not exploit differences in agents' valuations, leaving agents no better off in expectation than the naive mechanism that ignores agent reports and simply allocates the entire cake to a uniformly picked agent (i.e., possible "gains from trade" are not realized). Motivated by that fact, Mossel and Tamuz (2010) extend the perfect partition mechanism to one that is ex ante *strongly* proportional, at the cost of sacrificing ex post envy-freeness (ex post weak proportionality is retained). Strong proportionality demands that every agent obtains utility *strictly higher* than $1/n$ unless every agent reports the same valuation (where this is impossible). Similarly, strong envy-freeness demands that all agents *strictly* prefer their own piece to that of any other agent reporting a different valuation. Mossel and Tamuz's extended mechanism proceeds by choosing a random allocation of the cake and implementing that allocation if it gives every agent strictly more than $1/n$ utility; otherwise, it reverts to a perfect partition.

However, none of these randomized "mechanisms" are algorithms, in the sense that they do not provide a constructive method for returning an allocation, since they rely on computing a perfect partition. Neyman only proved that perfect partitions always exist but his proof is non-constructive, and to this day no algorithm is known for general valuation functions.

In search of *constructive* cake-cutting mechanisms ("algorithms"), other work has explored the space of randomized mechanisms for valuation functions that are not fully general but more permissive than piecewise uniform. When valuations are piecewise linear, Chen et al. (2013) show that perfect partitions can be computed, allowing for the perfect-partition-based mechanisms in the previous paragraph to be implemented. For piecewise constant valuations, Aziz and Ye (2014) define the Constrained Mixed Serial Dictatorship mechanism, which is ex ante incentive compatible, ex post (weakly) proportional, and, unlike the perfect partition mechanism, sometimes generates gains from trade, but is not envy-free unless there are only two agents.

In this work, we define *Competitive Cake Scoring Rules (CCSRs)*, a class of ex ante strictly incentive compatible, ex ante strongly proportional, and ex ante strongly envy-free mechanisms. (Strict incentive compatibility demands that reporting truthfully yields *strictly higher* utility than any other report.) We instantiate two representatives of this class: the first is the subclass of *OneCut* mechanisms, which accommodate fully general valuation functions, though without providing ex post fairness guarantees. The second is the *Seeded Probabilistic Allocation (SPA)* mechanism, which preserves ex ante incentive compatibility but implements both fairness criteria ex post for a very general class of valuation functions. This class, which we term *relaxed Lipschitz*, includes all Lipschitz-continuous valuation functions while also allowing for a finite number of discontinuities, an extremely mild condition that accommodates all but highly pathological functions. In particular, this class is significantly more general than piecewise-linear valuation functions, which in turn are more general than piecewise-constant and piecewise-uniform valuation functions. Note that further strengthening ex ante incentive compatibility to hold ex post is impossible due to the aforementioned result of Bu et al. (which even holds for piecewise constant valuations). As others have argued (e.g., Budish et al. (2013)), satisfying fairness properties ex post is crucial because agents evaluate the fairness of their *realized* allocation, whereas the objective of incentive compatibility is to ensure proper incentives at time of reporting (i.e., before the randomness of the mechanism materializes).

In summary, SPA improves upon the state of the art along several dimensions. Relative to deterministic mechanisms, most notably the one by Chen et al. (2013) for piecewise uniform valuations, SPA allows for much more general valuation functions. Relative to the (randomized) perfect-partition-based mechanisms, it achieves both fairness properties

	Determ.	Vals.	Constr.	IC	Prop.	EF	Non-Wastef.
Chen et al. (2013)	**Yes**	PWU	**Yes**	**Weak**	Weak	Weak	No
Naive Random	No	**General**	**Yes**	**Weak**	$\mathbb{E}$ Weak	$\mathbb{E}$ Weak	**Yes**
Aziz and Ye (2014)	No	PWC	**Yes**	$\mathbb{E}$ Weak	Weak	No	**Yes**
Chen et al. (2013),[1]	No	PWL	**Yes**	$\mathbb{E}$ Weak	Weak	Weak	**Yes**
Mossel and Tamuz (2010)	No	**General**	No	$\mathbb{E}$ Weak	Weak	Weak	**Yes**
Mossel and Tamuz (2010)	No	**General**	No	$\mathbb{E}$ Weak	$\mathbb{E}$ Strong[2]	No	**Yes**
OneCut	No	**General**	**Yes**	$\mathbb{E}$ **Strict**	$\mathbb{E}$ Strong	$\mathbb{E}$ Strong	**Yes**
SPA	No	**General**[3]	**Yes**	$\mathbb{E}$ **Strict**	**Strong**	**Strong**	**Yes**
Bu et al. (2023)[4]	~~**Yes**~~	~~PWC~~		~~**Weak**~~	~~Weak~~		

TABLE I

Comparison of properties satisfied by, to our knowledge, all known incentive-compatible cake-cutting mechanisms. Naive Random refers to the mechanism that allocates the entire cake uniformly at random, OneCut and SPA refer to the two new mechanisms developed in this paper. We highlight in bold undominated versions of each property (weak incentive compatibility and ex ante strict incentive compatibility are incomparable). [1]The constructive version of the perfect partition mechanism for piecewise linear valuations is (only) due to Chen et al. (2013). [2]The mechanism is both ex ante strongly proportional and weakly proportional. [3]We assume valuation functions to be "relaxed Lipschitz" (Definition 11), a mild, technical condition that is met by all reasonable valuation functions, including all those with only finitely many Lipschitz violations. [4]This row depicts the impossibility result due to Bu et al. (2023); the crossed-out properties cannot be achieved simultaneously. Note that the impossibility continues to hold when weak proportionality is replaced by weak envy-freeness and non-wastefulness.

strongly, and, crucially, is a *constructive* mechanism for general valuation functions. Even restricted to the special case of piecewise constant valuation functions, we are not aware of any mechanism that achieves this combination of properties.

On a technical level, the closest work to ours is that of Freeman et al. (2023). They consider the setting in which a number of homogeneous divisible items are to be allocated among a set of agents. Drawing inspiration from earlier work in information elicitation (Kilgour and Gerchak (2004), Lambert et al. (2008), Lambert et al. (2015)), they define the class of *Competitive Scoring Rules (CSRs)*, which are deterministic, strictly incentive compatible, envy-free, and proportional. The item allocation setting is a special case of piecewise constant valuation functions in cake cutting, namely the case where all agents and the mechanism agree a priori where the "steps begin and end." Importantly, this additional restriction beyond piecewise constant valuations simplifies the cake-cutting problem enough to circumvent Bu et al.'s impossibility. Our Competitive Cake Scoring Rules (CCSRs) generalize CSRs to the full cake-cutting setting. Crucially, in contrast to the divisible-item setting of Freeman et al. (2023), the general cake-cutting setting requires an allocation decision for each of the infinitely many "crumbs" (real-valued numbers in $[0, 1]$). Moreover, crumbs cannot be divided and hence each crumb needs to be allocated to exactly one agent.

CCSRs are parameterized by the choice of a (continuous-outcome) proper scoring rule, which are accuracy measures traditionally used to incentivize and evaluate probabilistic forecasts (Brier (1950), Good (1952), Gneiting and Raftery (2007)). Given a particular scoring rule, CCSRs specify marginal allocation probabilities for every crumb. We show that any mechanism that implements these marginal probabilities satisfies ex ante strict incentive compatibility, ex ante strong proportionality, and ex ante strong envy-freeness. However, implementing these marginal probabilities is nontrivial. Naively, one might hope to allocate each crumb independently but this is impossible as the mechanism cannot iterate through infinitely many crumbs. Our first approach to implementing CCSR marginals is to define the OneCut family of mechanisms. These mechanisms proceed by choosing a random cut point, using the two resulting parts of the cake to define outcomes for a binary random variable, and then applying a binary proper scoring rule to these outcomes. This approach is grounded in a known construction of continuous-outcome scoring rules from binary ones (Matheson and Winkler (1976)). The random cut point effectively samples a threshold; scoring the utilities for the piece below this threshold ensures that, in expectation over all cuts, the allocation aligns with the desired CCSR marginal probabilities.

Satisfying the fairness properties ex post is non-trivial. For example, note that even if the cake were to be finely discretized and the resulting pieces allocated independently (e.g., using the average crumb allocation probability of each discretized piece), then, even ignoring the incentive issues due to the fact that the marginal probabilities would not be implemented exactly, it would still be possible for one agent to receive the entire cake. The Seeded Probabilistic Allocation (SPA) mechanism, which we develop to implement ex post fairness, instead reduces the number of random draws to a single "seed," conditioned on which the mechanism is deterministic while still implementing the CCSR marginals. It makes a case distinction based on whether all agent reports are distinct or if two or more are identical, as the conditions to satisfy strong proportionality and strong envy-freeness differ significantly between these two cases. Distinct reports ensure that ex ante strong proportionality and ex ante strong envy-freeness imply some "slack" in the sense that every agent expects *strictly* more than $1/n$ utility and *strictly* more utility for her own piece than for that of any other agent. SPA leverages this by using an increasing parameter m such that, as m increases, the cake is divided into finer and finer parts with adjacent parts being allocated to different agents, leading to each agent's ex post utility converging to her ex ante utility. This is crucial because it means that, since the ex ante utilities are strictly higher than the thresholds required for strong proportionality and strong envy-freeness, there is a finite number of cuts after which the ex post utilities exceed these thresholds, guaranteeing that the mechanism satisfies the *ex post* fairness properties.

When two or more agents submit identical reports, it is impossible for these agents to strictly prefer their own piece to that of the others. Instead, ex post strong envy-freeness requires that utilities between these agents are exactly equal, and, in contrast to the all-distinct case, merely approximating expected utility is no longer sufficient. To take care of this case, SPA treats all agents with identical reports as a single "super agent" and utilizes a subroutine that divides the corresponding "super piece" amongst them in such a way that they value all of the individual pieces exactly equally. Moreover, and crucially, the subroutine also ensures that all agents outside of the "super agent" value the individual pieces of the "super piece" approximately equally through increasing the number of cuts, similar to the all-distinct case described above. Ex ante strong envy-freeness guarantees that all agents outside the "super agent" strictly prefer their own pieces to the *average* of the "super agent" pieces. Hence, having each of the "super agent" pieces be valued approximately equally by all outside agents is sufficient for ex post strong envy-freeness. Finally, note that while we have only argued about strong envy-freeness, strong envy-freeness implies strong proportionality for non-wasteful mechanisms (Barbanel (1996)).

For an illustration of the SPA mechanism, consider $n = 2$ agents with valuations given by beta distributions $y_1 = \text{Beta}(2, 5)$ and $y_2 = \text{Beta}(3, 4)$, as shown in Figure 1 (left side). The resulting marginal probability that crumb $x \in [0, 1]$ is allocated to agent 1 is $s_1(\boldsymbol{y}, x) = -\frac{15}{7}x^7 + 10x^6 - 18x^5 + 15x^4 - 5x^3 + \frac{81}{143}$; agent 2's marginal allocation probabilities are given by $s_2(\boldsymbol{y}, x) = 1 - s_1(\boldsymbol{y}, x)$. (See the supplemental appendix for detailed calculations.) SPA implements these probabilities by first drawing a "seed" $b \in [0, 1]$ uniformly at random (here we assume it comes up as $b = 0.12$) corresponding to the intercept of a line with slope m (initialized at $m = 2$) and "wrapped" into $[0, 1]$ as shown in Figure 1 (red line on right side). Agent 1 receives those crumbs x for which this line is below $s_1(\boldsymbol{y}, x)$, and agent 2 receives the remaining piece. Observe that the marginal selection probabilities are implemented by this process because, for any $x \in [0, 1]$, the line value corresponds to a uniform random draw. In the resulting allocation, agent 1 receives piece $[0,\ 0.2112] \cup [0.44,\ 0.6555] \cup [0.94,\ 1]$, and agent 2 receives piece $[0.2112,\ 0.44] \cup [0.6555,\ 0.94]$. Because the agents' reports differ and SPA is both strongly proportional and strongly envy-free, each agent must have utility strictly greater than 0.5 for their own piece (and thus utility less than 0.5 for the other agent's piece), which is the case in this example, with utilities 0.528 and 0.536 for agents 1 and 2, respectively.

2. MODEL

Let the interval $[0, 1]$ represent a heterogenous cake and let a "piece of cake" $z = z_1 \cup \ldots \cup z_r$ be a finite union of r disjoint subintervals. Denote by $\text{len}(z) = \text{len}(z_1) + \ldots + \text{len}(z_r)$ the length of a piece. We will refer to a "crumb" as an infinitesimal part of the cake, i.e., a real-valued number in $[0, 1]$. The set of $n \geq 2$ agents is denoted by $[n] = \{1, \ldots, n\}$. (For any natural number k, we define shorthand $[k] = \{1, \ldots, k\}$.) Let $\mathcal{P}$ denote the set of all integrable functions from $[0, 1]$ to $[0, \infty)$. Each agent i has a valuation function (sometimes simply "valuation") $v_i \in \mathcal{P}$ over the cake, indicating the agent's relative valuation for each crumb. As is standard in the cake-cutting literature, we assume that valuations are normalized, i.e., $\int_0^1 v_i(x)\,\mathrm{d}x = 1$ for all agents i. The space of admissible valuation functions is sometimes further restricted, e.g., to piecewise-constant valuations (Procaccia (2013)).

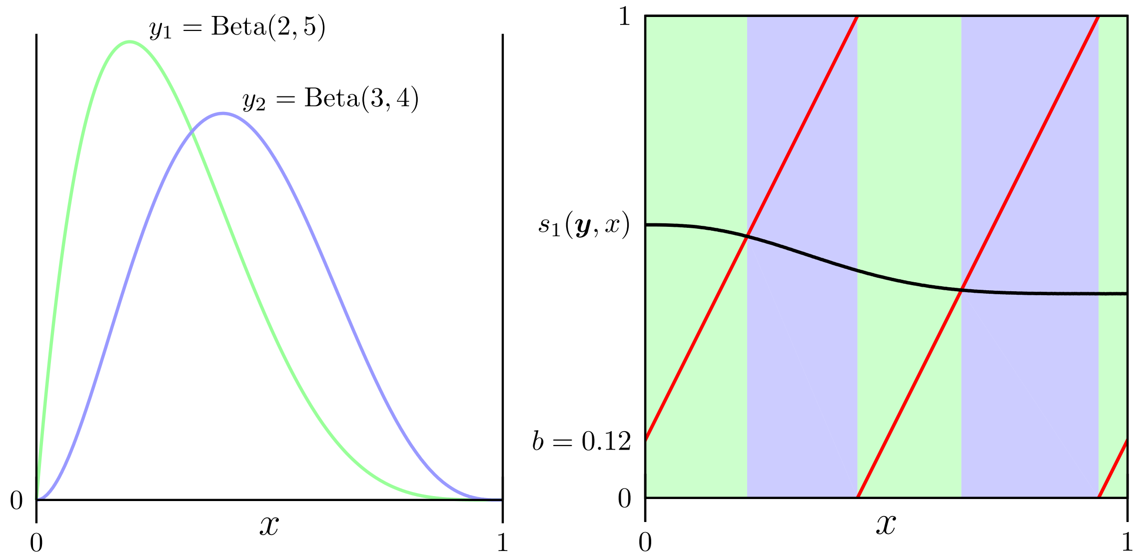

FIGURE 1.—Valuations (left side) and marginal crumb allocation probabilities with realized allocation highlighted by color (right side).

An agent's utility for a piece of cake z is given by $u_i(z, v_i) = \int_{x \in z} v_i(x)\, \mathrm{d}x$.[1] Note that this implies that utilities are additive and non-atomic. The latter property allows us to ignore the boundaries of intervals, and in particular allows us to treat two intervals as disjoint if their intersection is a singleton. An *allocation* $\boldsymbol{a} = (a_1, \dots, a_n)$ of the cake is a vector of pairwise disjoint pieces a_i such that piece a_i is allocated to agent i.

DEFINITION 1: A *deterministic cake-cutting mechanism* $\mathcal{D}$ takes as input a profile of all agents' reports $\boldsymbol{y} = (y_1, \dots, y_n)$, where $y_i : [0,1] \to [0, \infty)$ is a reported (normalized) valuation function[2] for each agent i, and outputs an allocation $\mathcal{D}(\boldsymbol{y})$. We use $\mathcal{D}_i(\boldsymbol{y})$ for the piece of cake allocated to agent i by $\mathcal{D}$. A *(randomized) cake-cutting mechanism* $\mathcal{M}$ is a distribution over deterministic cake-cutting mechanisms, and we refer to the set of deterministic mechanisms that are assigned non-zero probability by $\mathcal{M}$ as $\mathrm{supp}(\mathcal{M})$.

Note that the space of randomized cake-cutting mechanisms is strictly larger than the space of deterministic cake-cutting mechanisms because any deterministic mechanism can

[1]Technically, the subscript denoting the agent is not required for u because it is implied by the second argument, but we include it when doing so improves clarity.

[2]If a restriction is placed on the valuation functions, then the reported valuations follow the same restriction.

be represented by a point distribution. Going forward, unless otherwise specified, we refer to randomized cake-cutting mechanisms simply as "mechanisms."

A key focus of this work is the design of incentive-compatible mechanisms, i.e., mechanisms that do not incentivize agents to misreport their (private) valuations. The definition we use is that of dominant-strategy incentive compatibility, which requires that it is in each agent's best interest to report truthfully, regardless of the reports of others. This is in contrast to the weaker concept of Bayes-Nash incentive compatibility, which only requires truthful reporting to be an equilibrium.

DEFINITION 2: A mechanism $\mathcal{M}$ is *ex ante (weakly) incentive compatible* if, for all agents i, all valuations v_i, and all profiles of reports $\boldsymbol{y}$, it holds that

$$\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i((y_1,\ldots,v_i,\ldots,y_n)),v_i)\big] \geq \mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i((y_1,\ldots,y_i,\ldots,y_n)),v_i)\big].$$

A mechanism $\mathcal{M}$ is *ex post (weakly) incentive compatible* if, for all $\mathcal{D} \in \text{supp}(\mathcal{M})$, all agents i, all valuations v_i, and all profiles of reports $\boldsymbol{y}$, it holds that

$$u_i(\mathcal{D}_i((y_1,\ldots,v_i,\ldots,y_n)),v_i) \geq u_i(\mathcal{D}_i((y_1,\ldots,y_i,\ldots,y_n)),v_i).$$

$\mathcal{M}$ is (ex ante / ex post) *strictly* incentive compatible if the corresponding inequality is strict whenever $y_i \neq v_i$.

Note that weak incentive compatibility does not preclude the existence of other, non-truthful equilibria, which might be worse.[3] In contrast, strict incentive compatibility implies full implementation, i.e., in addition to being a dominant strategy for each agent, truthful reporting by all agents is also the *unique* equilibrium.

[3]As a concrete example, consider the random serial dictatorship mechanism, which first chooses an ordering of the agents uniformly at random, and then, to the first agent in the ordering, allocates those parts of the cake that she values positively. Subsequent agents get allocated those parts that they value positively of those parts that have not already been allocated to an earlier agent. This mechanism is only weakly incentive compatible, because, for example, agents are not penalized for reporting positive value for the entire cake even if there exist parts they have no value for. As a consequence, there can exist non-truthful dominant-strategy equilibria, in which, for example, proportional divisions are not implemented even for complementary valuations, such as when two agents each value opposite sides of the cake.

Proportionality (Steinhaus (1948)) requires that each agent receives a utility of at least $1/n$, i.e., their "fair share" of the cake. We will sometimes refer to proportionality as *weak* proportionality to contrast it with strong proportionality (Definition 4).

DEFINITION 3: A cake-cutting mechanism $\mathcal{M}$ is *ex ante (weakly) proportional* if, for all profiles of reports $\boldsymbol{y}$ and all agents i, it holds that $\underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] \geq 1/n$. $\mathcal{M}$ is *ex post (weakly) proportional* if, for all $\mathcal{D} \in \operatorname{supp}(\mathcal{M})$, all profiles of reports $\boldsymbol{y}$, and all agents i, it holds that $u_i(\mathcal{D}_i(\boldsymbol{y}), y_i) \geq 1/n$.

Note that, in general, the inequalities in Definition 3 are tight in that it is impossible to guarantee strictly more than 1/n to even one agent while still guaranteeing at least 1/n to everyone else. In particular, if all agents report the same valuation, then the sum of all agents' utilities is upper-bounded by 1. Hence, the following definition strengthens the proportionality requirement only for those cases where not all reported valuations are the same, i.e., whenever at least two agents report distinct valuations. While it would already strengthen proportionality to require that just *one* agent receives strictly more than 1/n utility under this condition, it turns out that these cases allow for *all* agents to receive strictly more than 1/n utility.[4] This was already mentioned by Steinhaus (1948) and studied formally by Dubins and Spanier (1961).

DEFINITION 4: A mechanism $\mathcal{M}$ is *ex ante strongly proportional* if it is ex ante proportional and, additionally, if, for all profiles of reports $\boldsymbol{y}$ with $y_j \neq y_k$ for some $j, k \in [n]$, it holds that $\underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] > 1/n$ for all agents i. A mechanism $\mathcal{M}$ is *ex post strongly proportional* if it is ex post proportional and, additionally, if, for all $\mathcal{D} \in \operatorname{supp}(\mathcal{M})$ and all profiles of reports $\boldsymbol{y}$ with $y_j \neq y_k$ for some $j, k \in [n]$, it holds that $u_i(\mathcal{D}_i(\boldsymbol{y}), y_i) > 1/n$ for all agents i.

[4]Prior work (Barbanel (1996)) defines a *strongly fair allocation* to be one in which every agent receives strictly more than $1/n$ utility. Our definition of *strongly proportional mechanisms* requires that the mechanism outputs a strongly fair allocation whenever one exists. In more recent work, Mossel and Tamuz (2010) consider strong proportionality under the name *super fairness*. We adopt the term "strongly proportional" in order to reflect that Definition 4 is a strengthening of proportionality specifically, and not of other fairness notions such as envy-freeness.

Envy-freeness (Foley (1967)) requires that no agent ever prefer another agent's piece to her own. Analogous to proportionality, we sometimes refer to this as *weak* envy-freeness.

DEFINITION 5: A mechanism $\mathcal{M}$ is *ex ante (weakly) envy-free* if, for all profiles of reports $\boldsymbol{y}$ and all agents i, j, it holds that $\underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\left[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\right] \geq \underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\left[u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)\right]$. A mechanism $\mathcal{M}$ is *ex post (weakly) envy-free* if, for all $\mathcal{D} \in \text{supp}(\mathcal{M})$, all profiles of reports $\boldsymbol{y}$, and all agents i, j, it holds that $u_i(\mathcal{D}_i(\boldsymbol{y}), y_i) \geq u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)$.

Note that, in general, and analogous to Definition 3, the inequalities in Definition 5 are tight in that it is impossible to guarantee that every agent strictly prefers her own piece of cake to that of every other agent for all admissible reports. In particular, if two agents i and j report the same valuation, i.e., $y_i = y_j$, and agent i strictly prefers her own piece to that of agent j, i.e., $u_i(\mathcal{M}_i(\boldsymbol{y}), y_i) > u_i(\mathcal{M}_j(\boldsymbol{y}), y_i)$, then agent j must also strictly prefer agent i's piece to her own, i.e., $u_j(\mathcal{M}_i(\boldsymbol{y}), y_i) > u_j(\mathcal{M}_j(\boldsymbol{y}), y_i)$, hence violating envy-freeness. The following definition requires that every agent i strictly prefers her own piece of cake to that of any agent j who reports a different valuation function $y_j \neq y_i$.[5]

DEFINITION 6: A mechanism $\mathcal{M}$ is *ex ante strongly envy-free* if it is ex ante envy-free and, additionally, if, for all profiles of reports $\boldsymbol{y}$ and all agents i, j with $y_j \neq y_i$, it holds that $\underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\left[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\right] > \underset{\mathcal{D}\sim\mathcal{M}}{\mathbf{E}}\left[u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)\right]$. A mechanism $\mathcal{M}$ is *ex post strongly envy-free* if it is ex post envy-free and, additionally, if, for all $\mathcal{D} \in \text{supp}(\mathcal{M})$, all profiles of reports $\boldsymbol{y}$, and all agents i, j with $y_i \neq y_j$, it holds that $u_i(\mathcal{D}_i(\boldsymbol{y}), y_i) > u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)$.

Finally, a mechanism $\mathcal{M}$ is *non-wasteful* if it allocates the entire cake, i.e., if $\cup_{i\in[n]}\mathcal{M}_i(\boldsymbol{y}) = [0, 1]$ for all profiles $\boldsymbol{y}$. As is the case for the weak versions of the properties, strong envy-freeness implies strong proportionality for non-wasteful mechanisms (Barbanel (1996)).

3. A CLASS OF EX ANTE INCENTIVE COMPATIBLE AND STRONGLY FAIR MECHANISMS

In this section, we introduce a class of mechanisms that satisfy all of the strong fairness and incentive properties from Section 2 in expectation. The class is defined by a set of

[5] Barbanel (1996) defines strong envy-freeness as a property of allocations. Our definition of strong envy-freeness aligns with that of Barbanel in the sense that a mechanism is strongly envy-free if it only outputs strongly envy-free allocations.

sufficient conditions ensuring these properties. While they do not directly define a mechanism, we nevertheless provide an explicit example later in the section. As a key tool in our design, we exploit proper scoring rules, which are typically used as reward functions in probabilistic forecasting and as loss functions in machine learning.

Proper Scoring Rules.

Proper scoring rules (Brier (1950), Good (1952), Gneiting and Raftery (2007)) are scoring functions that are used in two ways: to evaluate the accuracy of probabilistic predictions and to incentivize the truthful reporting of privately-held, probabilistic beliefs. Given the focus of this work, we introduce them taking the incentive perspective. Consider a future event, such as the fraction of votes that a particular party obtains in an election, that will take an outcome $x \in [0,1]$. Furthermore, consider an agent i with a private belief $p_i \in \mathcal{P}$ corresponding to the subjective probability density function over possible outcomes x. A principal seeking to truthfully elicit p_i from the agent can employ a proper scoring rule, which ensures the agent maximizes her expected score by reporting her private belief truthfully. The temporal order is as follows: first, the agent reports a forecast given by probability density function $y_i \in \mathcal{P}$, which may or may not coincide with her private belief p_i. Second, an outcome x materializes, and, third, the proper scoring rule assigns the agent a score that depends on the agent's reported forecast and the materialized outcome. In general, the outcomes could be any real value but we restrict to outcomes in the $[0,1]$ interval.

DEFINITION 7: A *continuous-outcome scoring rule* R is a function that maps a report $y_i \in \mathcal{P}$ and an outcome $x \in [0,1]$ to a score $R(y_i, x) \in \mathbb{R} \cup \{-\infty\}$. R is *(weakly) proper* if, for all $y_i, p_i \in \mathcal{P}$,

$$\mathop{\mathbf{E}}_{X \sim p_i} R(p_i, X) \geq \mathop{\mathbf{E}}_{X \sim p_i} R(y_i, X).$$

R is *strictly proper* if the inequality is strict for all $y_i \neq p_i$. R is *bounded* if there exist $\underline{R}, \overline{R} \in \mathbb{R}$ such that $R(y_i, x) \in [\underline{R}, \overline{R}]$ for all $y_i \in \mathcal{P}, x \in [0,1]$.

Positive-affine transformations of proper scoring rules preserve (strict) properness, and there exist infinitely many proper scoring rules since any (strictly) convex function on $[0,1]$ yields a (strictly) proper scoring rule (Gneiting and Raftery (2007, Theorem 1)). A commonly used bounded scoring rule is the *continuous ranked probability score* (CRPS, Matheson and Winkler (1976)), which we will regularly refer to throughout the paper and give here in its quadratic and normalized form to yield scores between 0 and 1.

PROPOSITION 1: *(Matheson and Winkler (1976)) The* continuous ranked probability score $R_{CRPS}(y_i, x) = 1 - \int_0^1 \big(Y_i(w) - \mathbb{1}\{w \geq x\}\big)^2 \, dw$, *where* $Y_i(w) = \int_0^w y_i(w')\, dw'$ *is the cumulative distribution implied by* y_i*, is strictly proper.*

Class of Mechanisms.

In this section, we describe a class of mechanisms for the cake-cutting setting, which specify marginal probabilities with which each crumb must be allocated to the agents.

DEFINITION 8: Mechanism $\mathcal{M}$ is a *Competitive Cake Scoring Rule (CCSR)* if and only if, for all agents i and all crumbs $x \in [0,1]$, agent i receives crumb x with probability

$$\Pr_{\mathcal{D}\sim\mathcal{M}}(x \in \mathcal{D}_i(\boldsymbol{y})) = s_i(\boldsymbol{y}, x) = \frac{1}{n} + \frac{1}{n}\left(R(y_i, x) - \frac{1}{n-1}\sum_{j \neq i} R(y_j, x)\right), \tag{1}$$

where R is a (continuous-outcome) proper scoring rule bounded by $[0,1]$. Mechanism $\mathcal{M}$ is a *strict Competitive Cake Scoring Rule* if R is strictly proper.

Observe that even fixing a scoring rule R does not immediately define a single mechanism. This is because any mechanism induces a *joint* distribution over crumbs, and CCSRs only constrain the marginals of that joint. That is, even fixing R leaves open how different crumbs are correlated with one another. At first sight, one might consider developing this into a concrete mechanism by randomly allocating each crumb independently, but this does not work because the number of crumbs is infinite. Note that the naive mechanism that allocates the entire cake to an agent that is selected uniformly at random is a CCSR, namely one that uses the constant proper scoring rule, which assigns a score of $R(y,x) = c$ for any $c \in [0,1]$, independent of the report and outcome. Interestingly, the perfect partition mechanism (Mossel and Tamuz (2010), Chen et al. (2013)) is a CCSR for the same constant scoring rule but it relies on a more complex and non-constructive characterization of the joint distribution over crumbs. We will introduce several CCSRs with non-trivial proper scoring rules in the remainder of this paper.

THEOREM 2: *Competitive Cake Scoring Rules are ex ante incentive compatible, ex ante proportional, ex ante envy-free, and non-wasteful. Strict Competitive Cake Scoring Rules are ex ante strictly incentive compatible, ex ante strongly proportional, ex ante strongly envy-free, and non-wasteful.*

To build intuition for CCSRs and how their properties are enabled by the application of strictly proper scoring rules, consider the simple case of $n = 2$ agents, and take the perspective of agent 1. Her CCSR crumb allocation probabilities are given by $s_1((y_1, y_2), x) = \frac{1}{2} + \frac{1}{2}\Big(R(y_1, x) - R(y_2, x)\Big)$. First note that ex ante (strict) incentive compatibility is "inherited" from R's (strict) properness because $s_1((y_1, \cdot), x)$ is a positive-affine transformation of $R(y_1, x)$. Now, for the fairness properties, and still taking the perspective of agent 1, there are two cases to distinguish for any given y_2: if $v_1 = y_2$, then, by incentive compatibility, matching agent 2's report, i.e., $y_1 = v_1 = y_2$, is optimal, resulting in crumb allocation probabilities of 0.5 for all $x \in [0, 1]$ and hence expected utility of 0.5. The second case arises when $v_1 \neq y_2$. If agent 1 misreported her true valuation and again matched agent 2's report, i.e., $y_1 = y_2$, she would again obtain an expected utility of 0.5. However, because of strict incentive compatibility, we know that agent 1 is better off reporting $y_1 = v_1 \neq y_2$, so it must be the case that her expected utility is higher than 0.5. This shows ex ante strong proportionality because she receives 0.5 expected utility according to her reported valuation if her report matches that of agent 2, and strictly more if they disagree. For strong envy-freeness, observe that if agent 1 has expected utility more than 0.5 for her own piece, then she must assign less than 0.5 to agent 2's piece.

As a simple example of a CCSR, we first introduce the OneCut family of mechanisms, which are parameterized by *binary* scoring rules, i.e., scoring rules for random variables that take only one of two outcomes.

DEFINITION 9—Binary Scoring Rule: A *binary scoring rule* R is a function that maps a report $y_i \in [0, 1]$ and a binary outcome $x \in \{0, 1\}$ to a score $R(y_i, x) \in \mathbb{R} \cup \{-\infty\}$. R is *(weakly) proper* if, for all $y_i, p_i \in [0, 1]$, it holds that $\mathbf{E}_{X \sim p_i}\big[R(p_i, X)\big] \geq \mathbf{E}_{X \sim p_i}\big[R(y_i, X)\big]$. R is *strictly proper* if the inequality is strict for all $y_i \neq p_i$. R is *bounded* if there exist $\underline{R}, \overline{R} \in \mathbb{R}$ such that $R(y_i, x) \in [\underline{R}, \overline{R}]$ for all $y_i \in [0, 1]$, $x \in \{0, 1\}$.

A widely used bounded binary scoring rule is the *quadratic scoring rule* (Brier (1950)), which we give here in normalized form to yield scores between 0 and 1.

PROPOSITION 3: *(Brier (1950)) The binary* quadratic scoring rule $R_q(y_i, x) = 1 - (y_i - x)^2$ *is strictly proper.*

When used with a binary strictly proper scoring rule, the OneCut family of mechanisms is a strict CCSR, and hence an ex ante strictly incentive-compatible cake-cutting mechanism that is both ex ante strongly proportional and ex ante strongly envy-free. Note that despite their simplicity, OneCut mechanisms are the first to satisfy this combination of properties (see Table I). Moreover, it is interesting that the mechanisms in this family require only a single cut to achieve these properties, no matter the number of agents.

The OneCut family of mechanisms proceeds as follows:

1. Draw a cut point $c \in [0,1]$ uniformly[6] at random, dividing the cake into two pieces $[0,c]$ and $[c,1]$. Let $V_i(c) = \int_0^c v_i(x)\,\mathrm{d}x$ be agent i's utility for the "left" piece $[0,c]$.
2. Allocate piece $[0,c]$ to agent i with probability

$$\frac{1}{n} + \frac{1}{n}\left(R(V_i(c),1) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c),1)\right)$$

and, independently, allocate piece $[c,1]$ to agent i with probability

$$\frac{1}{n} + \frac{1}{n}\left(R(V_i(c),0) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c),0)\right),$$

where R is a binary proper scoring rule bounded by $[0,1]$.

The idea of OneCut mechanisms is to randomly cut the cake into two parts, each of which defines the outcome of a binary random variable.[7] Then, the implied utility that every agent has for each part of the cake is scored using a binary scoring rule, and the resulting scores are transformed in such a way that they yield a probability distribution.

---

[6] While we present the OneCut family with a cut point that is drawn from the uniform distribution, the results generalize to any distribution with full support on $[0,1]$.

[7] For consistency with prior work in forecasting (Matheson and Winkler (1976)), we associate the "left" piece (i.e., part of the cake with lower real values) with the "positive" event outcome (i.e., $x = 1$).

Finally, the transformed scores are used to randomly and independently allocate each part to a single agent.

THEOREM 4: *Every OneCut mechanism is a Competitive Cake Scoring Rule. Every OneCut mechanism instantiated with a binary strictly proper scoring rule is a strict Competitive Cake Scoring Rule.*

Instantiating OneCut with the binary quadratic scoring rule (Proposition 3) yields crumb allocation probabilities corresponding to the CCSR parameterized by CRPS (Proposition 1). This is by design. It is well known that continuous-outcome proper scoring rules can be constructed from binary proper scoring rules by drawing a random threshold, and defining the "yes" outcome (for the binary scoring rule) as those values below the threshold and the "no" outcome as those values above it (Matheson and Winkler (1976)). Instantiating OneCut with a different binary proper scoring rule gives rise to a CCSR whose crumb allocation probabilities correspond to a different continuous-outcome proper scoring rule.

COROLLARY 5: *Every OneCut mechanism is ex ante incentive compatible, ex ante proportional, ex ante envy-free, and non-wasteful. Every OneCut mechanism parameterized with a binary strictly proper scoring rule is ex ante strictly incentive compatible, ex ante strongly proportional, ex ante strongly envy-free, and non-wasteful.*

4. AN EX ANTE INCENTIVE-COMPATIBLE AND EX POST STRONGLY FAIR MECHANISM

A drawback of the simple OneCut family of mechanisms from the previous section is that they obtain their fairness guarantees only in expectation. In an attempt to achieve ex post fairness guarantees, one might consider making many cuts and allocating each of the corresponding parts independently. However, while such an approach may often realize fair allocations, it may nevertheless be the case that, just by chance, the returned allocation is not fair, e.g., because one agent received far more than others. In contrast, the Seeded Probabilistic Allocation (SPA) mechanism, which we introduce in this section, uses a single random "seed" b, conditioned on which all crumb allocations are deterministic. An increasing parameter m divides the cake into finer and finer parts in such a way that, when the parts are allocated to the agents cyclically, CCSR probabilities are preserved. Crucially, as m increases, each agent's ex post utility converges to her ex ante utility, which, by virtue of CCSRs satisfying strong fairness ex ante, is sufficient to guarantee that SPA is ex post strongly proportional and ex post strongly envy-free.

When two or more agents report the same valuation, SPA will utilize the following Common Valuation Subroutine, which is used to divide an interval of cake among k agents with identical valuation function v into k equal-length pieces that all agents value equally. The subroutine proceeds by scanning the provided interval $[\alpha, \beta]$ for the starting point of a subinterval that has the appropriate length $(\beta - \alpha)/k$ and utility $u([\alpha, \beta], v)/k$ (such a subinterval is guaranteed to exist, as we show in Lemma 6). Once the first such subinterval is found, the subroutine is applied recursively on the remaining parts of the original interval, leaving out those parts already allocated. We provide an illustrative, numerical example for the Common Valuation Subroutine in the supplemental appendix.

DEFINITION 10—Common Valuation Subroutine: The following subroutine takes an interval $[\alpha, \beta] \subseteq [0, 1]$, a valuation function v, and an integer $k \leq n$, and it returns k pieces of cake.

1. Let z denote the allocated portion of the cake, initialized as $z = \emptyset$.
2. Let $f(x, z)$ be the position that is $(\beta - \alpha)/k$ "to the right of" x when z is excluded. That is, f maps a position $x \in [\alpha, \beta]$ and a piece z to a position $f(x, z) \in [\alpha, \beta]$ with the property that $\text{len}([x, f(x, z)] \setminus z) = (\beta - \alpha)/k$.
3. For i in $[k]$:
   (a) Initialize x_i at $x_i = \alpha$. Increase x_i until $u([x_i, f(x_i, z)] \setminus z, v) = u([\alpha, \beta], v)/k$, and output the ith piece as $[x_i, f(x_i, z)] \setminus z$.
   (b) Set $z = z \cup [x_i, f(x_i, z)]$.

Lemma 6 shows that the Common Valuation Subroutine partitions the input interval such that the returned pieces have the same length and utility according to the common valuation.

LEMMA 6: *The Common Valuation Subroutine (Definition 10) ensures that of the k pieces, (1) every piece has the same length, (2) every piece has the same utility $u([\alpha, \beta], v)/k$ according to v, and (3) the union of all pieces is $[\alpha, \beta]$ and their intersection is empty.*

Before presenting the full SPA mechanism, we begin with a high-level overview, supported in part by Figure 2. Step 1 of the mechanism is a trivial preprocessing step: If all agents report the same valuation function, then use the Common Valuation Subroutine to allocate the cake. Step 2 computes the CCSR allocation probabilities for every crumb using the CRPS scoring rule and defines functions S_i as the cumulative sums of these probabilities. Steps 3 and 4 initialize $m = 2$, which, roughly speaking, controls the number of cuts being made in the current iteration, and the random "seed" $b \in [0,1]$, respectively. In Step 5, SPA creates a candidate allocation that allocates each crumb x to agent i whenever (the fractional part of) the line $b + m \cdot x$ lies between $S_i(\boldsymbol{y}, x)$ and $S_{i-1}(\boldsymbol{y}, x)$. Step 6 checks whether in this candidate allocation, (1) every agent receives utility strictly higher than $1/n$, and (2) every agent has a strict preference for her own piece over that of any other agent with a differing report. If either condition is not yet met, the algorithm proceeds to Step 4, beginning a new iteration, now with an increased m, resulting in a new, more finely cut allocation. If both conditions are satisfied and all reports are distinct, SPA terminates because, in that case, the two conditions are equivalent to the two (ex post) strong fairness properties. If both conditions are satisfied but not all reports are distinct, then there may be some envy between agents with identical reports. Step 7 addresses this by first combining the pieces of all agents with identical reports to a "super piece," which is then reallocated to the same agents in such a way that they all value their reallocated pieces equally. This is achieved by virtue of the Common Valuation Subroutine and avoids envy within the "super agent." The subtlety here is that, in addition to all agents within the super agent valuing each piece equally, SPA also needs to ensure that every agent *outside of the super agent* values each of the super agent's reallocated pieces approximately equally. This is achieved by repeatedly applying the Common Valuation Subroutine and slicing up the super piece into finer and finer equal-length pieces, so that every piece appears more and more similar "from the outside looking in."

Let $\mathrm{frac}^+(w) = w - \lceil w \rceil + 1$ denote the fractional part of $w \geq 0$ but where integers are mapped to 1 rather than 0. Formally, the mechanism proceeds as follows:

1. If $y_i = y_j$ for all $i, j \in [n]$, allocate the cake using the Common Valuation Subroutine (Definition 10) with interval $[\alpha, \beta] = [0,1]$, the (common) valuation function $v = y_i = y_j$, and $k = n$. Match every agent to exactly one of these pieces; and terminate.
2. Set $R = R_{CRPS}$ in Equation 1 and let $S_i(\boldsymbol{y}, x) = \sum_{j=1}^{i} s_j(\boldsymbol{y}, x)$ be the cumulative sum of the (marginal) CCSR probabilities for every agent $i \in [n]$ and crumb $x \in [0,1]$.
3. Let $m = 2$.

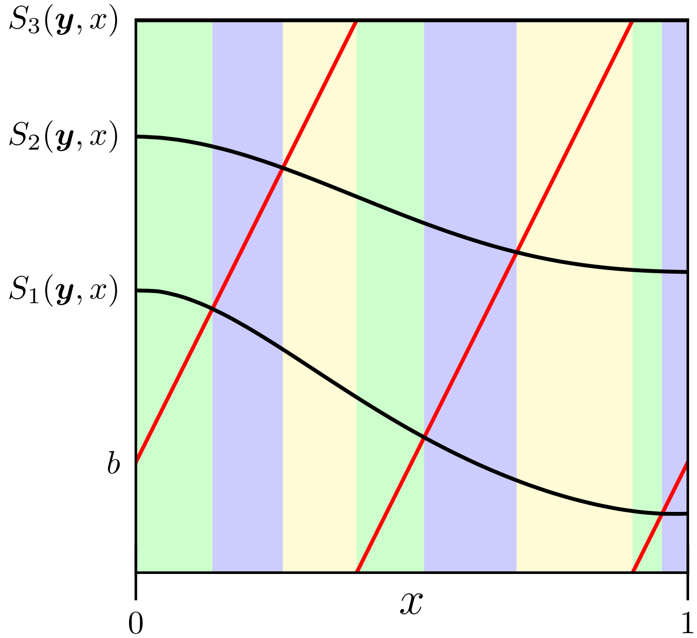

FIGURE 2.— An illustration of the SPA mechanism for $n = 3$ agents. The CRPS scores (Theorem 1) of each valuation function v_i are used in the CCSR formula (Definition 8) to obtain the crumb allocation probabilities s_i. The three functions S_1, S_2, S_3 denoted in black are the cumulative sums of the crumb allocation probabilities. This means that the "height" of the "entitlement zones" between S_i and S_{i-1} corresponds to the crumb allocation probabilities again; for example, the probability that agent 3 is allocated crumb x is $s_3(\boldsymbol{y}, x) = S_3(\boldsymbol{y}, x) - S_2(\boldsymbol{y}, x)$. (Note that, by definition, $S_1(\boldsymbol{y}, x) = s_1(\boldsymbol{y}, x)$ and $S_3(\boldsymbol{y}, x) = 1$.) The red line corresponds to $\text{frac}^+(mx + b)$ with $m = 2$ and random "seed" b. The candidate allocation is shown by the different colors: green for agent 1, blue for agent 2, and yellow for agent 3. Each agent i is allocated those parts of the cake, for which the red line is in her entitlement zone, i.e., between S_i and S_{i-1}. Observe that the uniformly-at-random draw of b ensures that a given crumb is allocated with probability equal to the height of the zone, so that the crumb allocation probabilities are indeed implemented and the strong ex ante properties of CCSRs are guaranteed to hold. If the ex post fairness guarantees do not yet hold, SPA doubles m until they do (after a bounded number of iterations).

4. Draw $b \in [0, 1)$ uniformly at random.
5. Let c denote the union of $\{0, 1\}$ and the set of cut points that are chosen as those x for which $S_i(\boldsymbol{y}, x) = \text{frac}^+(mx + b)$ for some $i \in [n]$, with $0 = c_0 < c_1 < \ldots < c_\ell = 1$ for some a priori unknown ℓ. Create a candidate allocation $\boldsymbol{a}$ that allocates each crumb x to agent $\min\{i : S_i(\boldsymbol{y}, x) > \text{frac}^+(mx + b)\}$, which, for every $k \in [\ell]$, will result in all crumbs $x \in (c_{k-1}, c_k)$ being assigned to the same agent. (The finite number of crumbs

x for which $\text{frac}^+(mx+b) = 1$ have no impact on agents' utilities but are allocated to agent n for completeness.)

6. If the following conditions (C1 and C2) hold, then move to Step 7; else, set $m := 2m$ and return to Step 4.
   (C1) For all $i \in [n]$, $u_i(a_i, y_i) > 1/n$, i.e., all agents receive utility strictly more than $1/n$.
   (C2) For all $i, j \in [n]$ with $y_i \neq y_j$, it holds that $u_i(a_i, y_i) > u_i(a_j, y_i)$, i.e., all agents strictly prefer their own piece to the piece of any agent with a different report.
7. If all reports are distinct, then terminate; else, let $t = 1$.
   (i) Let $G_1, G_2, \ldots, G_q$ denote groups of agents who make identical reports.[8] That is, for all $k, k' \in [q]$ with $k' \neq k$, it holds that $y_i = y_j$ whenever $i, j \in G_k$ and $y_i \neq y_j$ whenever $i \in G_{k'}$ and $j \in G_k$. Let $n_k = |G_k|$ denote the number of agents in G_k.
   (ii) For all $k \in [q]$, divide every subinterval A_k of $\cup_{i \in G_k} a_i$ into t equal-length subintervals, and, using the Common Valuation Subroutine (Definition 10), further divide each of these into n_k subintervals of equal length such that every agent in G_k values each of them equally. Uniformly at random, match every agent to exactly one of these n_k subintervals, the union of which form each agent's piece a_i in the candidate allocation.
   (iii) If $u_i(a_i, y_i) > u_i(a_j, y_i)$ for all i, j with $y_i \neq y_j$, then terminate; else, set $t := 2t$ and return to Step ii.

We want to emphasize that, while we present the mechanism in an iterative fashion where m and t are repeatedly doubled, it is also possible to avoid this by fixing both as a function of the reports right from the start. However, because these values are only upper bounds, the iterative implementation can end with lower values for m and t, leading to fewer cuts.

To guarantee termination, we require a mild condition on the agents' valuation functions. Roughly speaking, the condition says that, with the exception of an arbitrarily small part of the cake, an agent's valuation function cannot be arbitrarily steep. Note that this is extremely mild, and allows for all but highly pathological functions. For example, it allows

[8] Note that, for presentational purposes, we allow for groups consisting of only a single agent. The allocation would be identical if we excluded such single-agent groups.

for beta functions, polynomials, exponentials, and their piecewise generalizations and combinations. In particular, it permits classical valuation functions studied in the cake-cutting literature, such as piecewise linear (and therefore also piecewise constant and piecewise uniform) valuations. A slightly more restrictive but similar assumption has been made in the cake-cutting literature by Cohler et al. (2011).

DEFINITION 11: A valuation function v_i is Lipschitz on interval $I \subseteq [0,1]$ if there exists a positive real number L such that for all $x_1, x_2 \in I$, $|v_i(x_1) - v_i(x_2)| \leq L|x_1 - x_2|$. A valuation function v_i is *relaxed Lipschitz* if, for any $\epsilon > 0$, there exists a piece of cake $z = z_1 \cup \ldots \cup z_r$ such that v_i is Lipschitz on each subinterval of z, and it holds that $u_i(z, v_i) \geq 1 - \epsilon$.

With this definition, we are able to prove our main theorem.

THEOREM 7: *When valuations v_i and reports y_i are relaxed Lipschitz for all $i \in [n]$, the Seeded Probabilistic Allocation mechanism is ex ante incentive compatible, non-wasteful, ex post strongly proportional, and ex post strongly envy-free.*

5. DISCUSSION

5.1. *Robertson-Webb Model*

We assume that valuation functions are reported directly to the mechanism. An alternative line of work in cake cutting pertains to the Robertson-Webb query model (Robertson and Webb (1998), Procaccia (2016)), which specifies the type of information that the mechanism can elicit from the agents. More precisely, the model allows for two types of queries: First, a *cut* query provides the agent with the left endpoint of an interval and a concrete utility value, and elicits the right endpoint such that the agent values the interval at exactly the provided utility value. Second, *eval* queries elicit an agent's utility for a provided interval. In that model, Kurokawa et al. (2013) show that no deterministic incentive-compatible and envy-free mechanism terminates in a bounded number of steps. Allowing for randomization, Brânzei and Miltersen (2015) provide a mechanism that is weakly incentive compatible in expectation and, for any ε, outputs an approximate perfect partition, i.e., an allocation such that every agent values every agent's piece between $1/n - \varepsilon$ and $1/n + \varepsilon$ (thus guaranteeing approximate ex post proportionality and approximate ex post envy-freeness).

The OneCut family of mechanisms from Section 3 can also be implemented in the Robertson-Webb model. To see this, observe that the random draw of the cut point can simply be followed by one eval query per agent to determine every agent's utility for the two pieces defined by the cut. The random allocation of the two pieces remains unchanged and hence the distribution over allocations is the same as that of the OneCut mechanisms from Section 3, with the incentive and fairness properties carrying over directly.

5.2. *Beyond Incentive-Compatible Mechanisms*

We have focused on cake-cutting mechanisms that are incentive compatible. Strongly proportional methods for cake cutting have also been studied ignoring incentive compatibility concerns. Dubins and Spanier (1961) show the existence of strongly proportional mechanisms, with Woodall (1986) eventually providing such a (constructive) mechanism. To our knowledge, prior to our work, no algorithms existed that are guaranteed to output strongly envy-free allocations. A related notion is super envy-freeness (Barbanel (1996)), which requires that every agent values her own piece strictly higher than $1/n$ and every other agent's piece strictly lower than $1/n$. Barbanel proved that super envy-free allocations exist if and only if all agents' valuation functions are linearly independent, and Webb (1999) provided a constructive algorithm for finding one whenever it exists. Since a super envy-free allocation (when it exists) is always strongly envy-free, Webb's algorithm also guarantees a strongly envy-free allocation when the valuations are linearly independent. However, when valuations are linearly dependent, Webb's algorithm is not defined. In particular, the linearly dependent case includes valuations that are not all identical and therefore the strict inequalities in the definition of strong envy-freeness apply.

A prominent cake-cutting mechanism is the *Maximum Nash Welfare (MNW)* rule, which returns an allocation maximizing the product of agent utilities. MNW is equivalent to a version of the Competitive Equilibrium from Equal Incomes rule (Segal-Halevi and Sziklai (2019)), from which it follows that MNW satisfies proportionality, envy-freeness, and Pareto optimality[9] (Weller (1985), Segal-Halevi and Sziklai (2019)). However, MNW does

[9]Pareto optimality says that it should not be possible to increase some agent's utility without decreasing some other agent's utility.

not satisfy strong proportionality or strong envy-freeness.[10] Furthermore, no algorithm is known to compute an MNW solution for general valuations (for piecewise constant valuations, Aziz and Ye (2014) provide an algorithm). Exploring the space of (non-incentive-compatible) mechanisms that satisfy strong proportionality and strong envy-freeness along with other desirable properties (such as Pareto optimality) is an interesting question for future work.

5.3. *Chores*

Our results extend easily to the case in which the agents value the cake negatively, a setting that models the assignment of work shifts to employees, chores to household members, or the allocation of undesirable items such as waste or emissions. In this setting, it is also necessary to impose a non-wastefulness requirement, since otherwise the entire cake could simply be discarded. Agents still report normalized valuation functions, but these now define costs rather than utilities. In particular, an agent i's cost for a piece of cake z is given by $c_i(z, v_i) = \int_{x \in z} v_i(x)\, \mathrm{d}x$. The definitions of (strict) incentive compatibility, (strong) proportionality, and (strong) envy-freeness all carry over directly to this setting, with the direction of the inequalities in the respective definitions reversing. In words, incentive compatibility now requires that truthful reporting minimizes the cost of an agent's allocated piece of cake, proportionality requires that every agent receives a cost of at most $1/n$, and envy-freeness requires that an agent's cost for their own piece of cake is at most their cost for any other agent's piece of cake. (The strong versions are analogous to the standard case, requiring that the inequalities in the definitions of proportionality and envy-freeness are strict whenever allowed by the input.)

To apply CCSRs to the chores setting while preserving the properties from Theorem 2, the CCSR definition needs to be modified slightly. To see why, note that CCSRs seek to

[10]Consider an example with $n = 3$ agents, where agent 1 has valuation function uniform over the subinterval $[0, \frac{1}{3}]$ while agents 2 and 3 have valuation function uniform over the entire cake. MNW allocates the piece $[0, \frac{1}{3}]$ to agent 1, and splits the remainder of the cake between agents 2 and 3 (for example, agent 2 receiving $[\frac{1}{3}, \frac{2}{3}]$ and agent 3 receiving $[\frac{2}{3}, 1]$ is one possible MNW solution. This allocation violates strong proportionality because agents 2 and 3 receive utility exactly $\frac{1}{3}$ even though not all reports are identical. Similarly, it violates strong envy-freeness because agents 2 and 3 are exactly indifferent between their own pieces and that of agent 1, even though agent 1 makes a different report than agents 2 and 3.

allocate crumbs to agents with high $v_i(x)$ while, for chores, we want to allocate crumbs to agents with *low* cost $v_i(x)$. To resolve this, it is enough to simply replace the proper scoring rule R with its corresponding *proper loss* $1-R$ in Equation 1. With this adjustment, the crumb allocation probabilities become

$$\Pr_{\mathcal{D}\sim\mathcal{M}}(x \in \mathcal{D}_i(\boldsymbol{y})) = \frac{1}{n} + \frac{1}{n}\left(\frac{1}{n-1}\sum_{j\neq i} R(y_j, x) - R(y_i, x)\right),$$

which is the difference between the average loss incurred by agents other than i for crumb x and the loss incurred by i for crumb x. With this modification, Theorem 2 continues to hold in the chores setting, as do the ex post properties of SPA (Theorem 7).

6. CONCLUSION

This work advances strategyproof cake cutting by introducing Competitive Cake Scoring Rules (CCSRs), a family of randomized mechanisms. A CCSR is specified by (1) a proper scoring rule, typically used to evaluate and compare probabilistic forecasts, to fix the marginal distribution over crumbs and (2) a joint distribution over crumbs consistent with this marginal. Different choices of either component yield distinct mechanisms. When instantiated with a *strictly* proper scoring rule, every CCSR satisfies ex ante strict incentive compatibility, ex ante strong proportionality, and ex ante strong envy-freeness. As a simple but meaningful example, we identify the OneCut family of mechanisms that inherit these properties by virtue of being strict CCSRs.

Moving beyond fairness guarantees that only hold in expectation, we then develop the Seeded Probabilistic Allocation (SPA) mechanism, a strict CCSR that achieves both strong proportionality and strong envy-freeness *ex post*. These properties are non-trivial to achieve, and, before this work, no constructive mechanism was known that is both incentive compatible and even weakly proportional (let alone envy-free). Moreover, the result is tight in the sense that even for this weakened set of properties additionally achieving ex post incentive compatibility is impossible as it would violate a recent impossibility result (Bu et al. (2023)).

Closest to SPA in terms of satisfied properties are mechanisms predicated on being able to compute a perfect partition (Mossel and Tamuz (2010), Chen et al. (2013)). A perfect partition is an allocation for which every agent is allocated a piece of (commonly-agreed) value $1/n$, so they are trivially (weakly) proportional and (weakly) envy-free. However, while perfect partitions are known to exist, no algorithm is known to compute them. In fact, both OneCut and SPA can be interpreted as non-trivial generalizations of existing mechanisms in the literature, and it is insightful to make this connection explicit. Consider the weakly proper constant scoring rule that assigns the same constant score, independent of the report and outcome. When instantiating OneCut with the constant scoring rule, one obtains the naive mechanism that allocates the entire cake to a random agent. Similarly, when changing SPA's definition such that it uses the constant scoring rule instead of the continuous ranked probability score (CRPS), SPA's candidate allocations are converging to—but never reaching—a perfect partition.

APPENDIX A: PROOF OF THEOREM 2

To prove ex ante incentive compatibility, consider an agent i with valuation function v_i and report y_i. For all $\boldsymbol{y}$ consistent with y_i, her expected utility is

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i((y_1,\ldots,y_i,\ldots,y_n)),v_i)\big]\\
&=\int_0^1 s_i(\boldsymbol{y},x)v_i(x)\,\mathrm{d}x\\
&=\int_0^1\left(\frac{v_i(x)}{n}+\frac{1}{n}\left(R(y_i,x)v_i(x)-\frac{1}{n-1}\sum_{j\neq i}R(y_j,x)v_i(x)\right)\right)\,\mathrm{d}x\\
&=\frac{1}{n}+\frac{1}{n}\left(\int_0^1 R(y_i,x)v_i(x)\,\mathrm{d}x-\frac{1}{n-1}\sum_{j\neq i}\int_0^1 R(y_j,x)v_i(x)\,\mathrm{d}x\right)\\
&=\frac{1}{n}+\frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim v_i}R(y_i,X)-\frac{1}{n-1}\sum_{j\neq i}\int_0^1 R(y_j,x)v_i(x)\,\mathrm{d}x\right)\\
&\leq\frac{1}{n}+\frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim v_i}R(v_i,X)-\frac{1}{n-1}\sum_{j\neq i}\int_0^1 R(y_j,x)v_i(x)\,\mathrm{d}x\right)\\
&=\int_0^1 s_i((y_1,\ldots,v_i,\ldots,y_n),x)v_i(x)\,\mathrm{d}x=\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i((y_1,\ldots,v_i,\ldots,y_n)),v_i)\big]
\end{aligned}
$$

Note that if R is strictly proper and $y_i \neq v_i$, then the inequality is strict, yielding ex ante *strict* incentive compatibility.

To prove ex ante proportionality, we have, for any agent i and any profile $\boldsymbol{y}$,

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] &= \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X) - \frac{1}{n-1}\sum_{j\neq i}\int_0^1 R(y_j, x) y_i(x)\,\mathrm{d}x\right) \\
&= \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X) - \frac{1}{n-1}\sum_{j\neq i}\mathop{\mathbf{E}}_{X\sim y_i} R(y_j, X)\right) \\
&\geq \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X) - \frac{1}{n-1}\sum_{j\neq i}\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X)\right) = \frac{1}{n},
\end{aligned}$$

where the inequality follows from properness of R. Note that if R is strictly proper and there exists some agent j with $y_j \neq y_i$, then the inequality will be strict, yielding ex ante strong proportionality.

To prove ex ante envy-freeness, we have, for any pair of agents i, j and any profile $\boldsymbol{y}$,

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] &= \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X) - \frac{1}{n-1}\sum_{k\neq i}\int_0^1 R(y_k, x) y_i(x)\,\mathrm{d}x\right) \\
&= \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_i, X) - \frac{1}{n-1}\sum_{k\neq i}\mathop{\mathbf{E}}_{X\sim y_i} R(y_k, X)\right) \\
&\geq \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_j, X) - \frac{1}{n-1}\sum_{k\neq i}\mathop{\mathbf{E}}_{X\sim y_i} R(y_k, X)\right) \\
&\geq \frac{1}{n} + \frac{1}{n}\left(\mathop{\mathbf{E}}_{X\sim y_i} R(y_j, X) - \frac{1}{n-1}\sum_{k\neq j}\mathop{\mathbf{E}}_{X\sim y_i} R(y_k, X)\right) \\
&= \frac{1}{n} + \frac{1}{n}\left(\int_0^1 R(y_j, x) y_i(x)\,\mathrm{d}x - \frac{1}{n-1}\sum_{k\neq j}\int_0^1 R(y_k, x) y_i(x)\,\mathrm{d}x\right) \\
&= \int_0^1 y_i(x)\left(\frac{1}{n} + \frac{1}{n}\left(R(y_j, x) - \frac{1}{n-1}\sum_{k\neq j} R(y_k, x)\right)\right)\,\mathrm{d}x
\end{aligned}$$

$$= \int_0^1 y_i(x) s_j(\boldsymbol{y}, x)\, \mathrm{d}x = \mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)\big].$$

The inequalities follow from repeated application of the properness of R. Additionally, if R is strictly proper and $y_i \neq y_k$, then both inequalities are strict, which yields ex ante strong envy-freeness.

Finally, non-wastefulness follows from the crumb allocation probabilities summing to 1:

$$\begin{aligned}\sum_{i=1}^{n} \Pr_{\mathcal{D}\sim\mathcal{M}}(x \in \mathcal{D}_i(\boldsymbol{y})) = \sum_{i=1}^{n} s_i(\boldsymbol{y}, x) = 1 + \frac{1}{n}\sum_{i=1}^{n}\left(R\big(y_i, x\big) - \frac{1}{n-1}\sum_{j\neq i} R\big(y_j, x\big)\right) \\ = 1 + \frac{1}{n}\left(\sum_{i=1}^{n} R\big(y_i, x\big) - \frac{n-1}{n-1}\sum_{i=1}^{n} R\big(y_i, x\big)\right) = 1.\end{aligned}$$

APPENDIX B: PROOF OF THEOREM 4

We will show that every crumb is allocated according to Definition 8 for some continuous-outcome scoring rule R. To that end, consider a crumb x and a random cut point c. If $x < c$ then x is allocated to agent i with probability

$$\frac{1}{n} + \frac{1}{n}\left(R(V_i(c), 1) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c), 1)\right)$$

and if $x > c$ then x is allocated to agent i with probability

$$\frac{1}{n} + \frac{1}{n}\left(R(V_i(c), 0) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c), 0)\right).$$

Integrating over all possible values of c, each instantiating a deterministic mechanism, gives the overall probability that x is allocated to agent i:

$$\begin{aligned}\int_0^x \left(\frac{1}{n} + \frac{1}{n}\left(R(V_i(c), 0) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c), 0)\right)\right) \mathrm{d}c \\ + \int_x^1 \left(\frac{1}{n} + \frac{1}{n}\left(R(V_i(c), 1) - \frac{1}{n-1}\sum_{j\neq i} R(V_j(c), 1)\right)\right) \mathrm{d}c\end{aligned}$$

$$= \frac{1}{n} + \frac{1}{n}\left(\int_0^x R(V_i(c),0)\,\mathrm{d}c + \int_x^1 R(V_i(c),1)\,\mathrm{d}c\right.$$
$$\left. - \frac{1}{n-1}\sum_{j\neq i}\left(\int_0^x R(V_j(c),0)\,\mathrm{d}c + \int_x^1 R(V_j(c),1)\,\mathrm{d}c\right)\right).$$

Matheson and Winkler (1976) in their Equation 15 showed that the continuous-outcome scoring rule defined by $R^*(v_i,x) = \left(\int_0^x R(V_i(c),0)\,\mathrm{d}c + \int_x^1 R(V_i(c),1)\,\mathrm{d}c\right)$ is proper. Substituting R^* into the above expression yields the crumb allocation probabilities from Definition 8 as desired.

APPENDIX C: PROOF OF LEMMA 6

Assume for now that the subroutine terminates. It is then easy to see that the output satisfies all three conditions of the lemma statement. In particular, note that each of the k pieces are mutually disjoint because z is updated to keep track of the subset of the cake already allocated, and every piece is explicitly defined to not intersect z. Given this fact, equal length (Condition 1) follows from the definition of function f, equal utility (Condition 2) follows from the stopping condition on x_i, and the partitioning of the interval $[\alpha,\beta]$ (Condition 3) immediately follows from Condition 1.

Therefore, all that remains to be shown is that the subroutine terminates. The only step containing more than a definition is Step 3a. To that end, suppose that we have reached the start of the ℓth iteration of Step 3 for some $\ell \in [k]$. Therefore, z is of length $(\ell-1)(\beta-\alpha)/k$ and utility $u(z,v) = (\ell-1)\cdot u([\alpha,\beta],v)/k$. To simplify notation, let $g(x,z) = u([x,f(x,z)]\setminus z, v)$, i.e., the utility of the “candidate piece” beginning at x. To show that Step 3a terminates it is sufficient to show that an x_i with $g(x_i,z) = u([\alpha,\beta],v)/k$ always exists. Note that g is continuous in x_i since it is defined by a difference of integrals and any integral is a continuous function of its limits. Therefore, by the intermediate value theorem, if there exist $\underline{x},\overline{x} \in [\alpha,\beta]$ with $g(\underline{x},z) \leq u([\alpha,\beta],v)/k$ and $g(\overline{x},z) \geq u([\alpha,\beta],v)/k$ then such an x_i must exist.

To see that such values $\underline{x},\overline{x} \in [\alpha,\beta]$ exist, suppose, for contradiction, that $g(x,z) < u([\alpha,\beta],v)/k$ for all $x \in [\alpha,\beta]$. (The opposite case where $g(x,z) > u([\alpha,\beta],v)/k$ for all $x \in [\alpha,\beta]$ is handled analogously.) The utility of the unallocated parts of $[\alpha,\beta]$ is

$u([\alpha,\beta],v)-u(z,v)=(k-(\ell-1))\cdot u([\alpha,\beta],v)/k$. Imagine "stitching together" these unallocated parts and partitioning them into $k-(\ell-1)$ contiguous (in this stitched-together space) pieces of length $(\beta-\alpha)/k$. The average utility of these pieces is $u([\alpha,\beta],v)/k$ and thus at least one of them needs to have utility at least $u([\alpha,\beta],v)/k$ — a contradiction.

APPENDIX D: PROOF OF THEOREM 7

The proof consists of three main parts. The first part shows that SPA is well defined, in that it assigns a piece of cake (i.e., a finite union of subintervals) to every agent. The second part is to show that SPA is a CCSR, which implies non-wastefulness and ex ante incentive compatibility. The third, and most intricate, part is to show that SPA satisfies ex post strong proportionality and ex post strong envy-freeness.

Part 1: SPA returns finite union of subintervals.

For now, assume that the algorithm terminates (shown in Part 3). In this part, we prove the implicit and explicit assumptions made in the algorithm description. Throughout the algorithm description, it is implicitly assumed that every agent receives a piece of cake (a *finite* set of subintervals) in all candidate allocations. We will refer to such allocations as *valid*. Here we show that this is indeed the case. First note that, if the candidate allocation at the end of Step 5 was valid, the repeated calling of the Common Valuation Subroutine in Step 7 will also be valid. To see this, note that in the final loop, for each group G_k and each subinterval A_k, the Common Valuation Subroutine is called t times and that the Common Valuation Subroutine returns one piece of cake per agent. Second, to apply the method from Step 5, it needs to be ensured that the number of cut points $(\ell-1)$ is finite. For example, a situation that needs to be avoided is that one of the S_i lines coincides with $\text{frac}^+(mx+b)$ for some subinterval of $[0,1]$. As we will show in this part, this cannot happen as long as $m\geq 2$, because then the line $\text{frac}^+(mx+b)$ is always "strictly steeper" than $S_i(\boldsymbol{y},x)$, which bounds the number of times that the two can intersect. Third, it is explicitly assumed that all crumbs between two adjacent cut points are allocated to the same agent. This, along with the finite number of cut points, guarantees that the candidate allocation created in Step 5 is valid. We now give the formal proof of the second and third points, with the first having been proven above.

We first show that for all $x_1, x_2 \in [0,1]$ and all $y_i \in \mathcal{P}$, it holds that $|R_{CRPS}(y_i, x_1) - R_{CRPS}(y_i, x_2)| \leq |x_1 - x_2|$. To see this, suppose without loss of generality that $x_1 \leq x_2$. We have

$$
\begin{aligned}
& |R_{CRPS}(y_i, x_1) - R_{CRPS}(y_i, x_2)| \\
&= \left| \left(1 - \int_0^1 \big(Y_i(w) - \mathbb{1}\{w \geq x_1\}\big)^2 \,\mathrm{d}w\right) - \left(1 - \int_0^1 \big(Y_i(w) - \mathbb{1}\{w \geq x_2\}\big)^2 \,\mathrm{d}w\right) \right| \\
&= \left| \int_0^1 \left(\big(Y_i(w) - \mathbb{1}\{w \geq x_2\}\big)^2 - \big(Y_i(w) - \mathbb{1}\{w \geq x_1\}\big)^2 \right) \mathrm{d}w \right| \\
&= \left| \int_{x_1}^{x_2} \left(\big(Y_i(w) - \mathbb{1}\{w \geq x_2\}\big)^2 - \big(Y_i(w) - \mathbb{1}\{w \geq x_1\}\big)^2 \right) \mathrm{d}w \right| \\
&\leq |x_2 - x_1| = |x_1 - x_2| \,.
\end{aligned}
\tag{2}
$$

The first and second equalities follow from simple algebra. The third equality holds because $x_1 \leq x_2$ and thus $\big(Y_i(w) - \mathbb{1}\{w \geq x_2\}\big)^2 - \big(Y_i(w) - \mathbb{1}\{w \geq x_1\}\big)^2 = 0$ for all $w \notin [x_1, x_2]$. Finally, for the inequality, first observe that, for all $w, x \in [0,1]$, we have $-1 \leq Y_i(w) - \mathbb{1}\{w \geq x\} \leq 1$, and therefore $0 \leq \big(Y_i(w) - \mathbb{1}\{w \geq x\}\big)^2 \leq 1$. From this it then follows that $-1 \leq \left(\big(Y_i(w) - \mathbb{1}\{w \geq x_2\}\big)^2 - \big(Y_i(w) - \mathbb{1}\{w \geq x_1\}\big)^2\right) \leq 1$ for all $w \in [0,1]$, which implies the inequality.

We next show that for all $x_1, x_2 \in [0,1]$ and all profiles of reports $\boldsymbol{y}$, it holds that $|s_i(\boldsymbol{y}, x_1) - s_i(\boldsymbol{y}, x_2)| \leq |x_1 - x_2|$ for all $i \in [n]$. To see this, we have

$$
\begin{aligned}
& |s_i(\boldsymbol{y}, x_1) - s_i(\boldsymbol{y}, x_2)| \\
&= \Bigg| \left(\frac{1}{n} + \frac{1}{n}\Big(R_{CRPS}(y_i, x_1) - \frac{1}{n-1} \sum_{j \neq i} R_{CRPS}(y_j, x_1) \Big) \right) \\
&\quad - \left(\frac{1}{n} + \frac{1}{n}\Big(R_{CRPS}(y_i, x_2) - \frac{1}{n-1} \sum_{j \neq i} R_{CRPS}(y_j, x_2) \Big) \right) \Bigg| \\
&= \frac{1}{n} \left| \Big(R_{CRPS}(y_i, x_1) - R_{CRPS}(y_i, x_2) \Big) + \frac{1}{n-1} \sum_{j \neq i} \Big(R_{CRPS}(y_j, x_2) - R_{CRPS}(y_j, x_1) \Big) \right| \\
&\leq \frac{1}{n} \left(\Big| R_{CRPS}(y_i, x_1) - R_{CRPS}(y_i, x_2) \Big| + \left| \frac{1}{n-1} \sum_{j \neq i} \Big(R_{CRPS}(y_j, x_2) - R_{CRPS}(y_j, x_1) \Big) \right| \right)
\end{aligned}
$$

$$\leq \frac{1}{n}\left(\left|R_{CRPS}(y_i,x_1)-R_{CRPS}(y_i,x_2)\right|+\frac{1}{n-1}\sum_{j\neq i}\left|R_{CRPS}(y_j,x_2)-R_{CRPS}(y_j,x_1)\right|\right)$$

$$\leq \frac{1}{n}\left(|x_1-x_2|+\frac{1}{n-1}(n-1)|x_1-x_2|\right)=\frac{2}{n}|x_1-x_2|\leq |x_1-x_2|, \tag{3}$$

where the first two inequalities are applications of the triangle inequality and the third inequality follows from Equation 2.

We next show that for every $i \in [n]$, every $x_1, x_2 \in [0,1]$ with $x_2 \neq x_1$, and all profiles of reports $\boldsymbol{y}$, it holds that $|S_i(\boldsymbol{y},x_1)-S_i(\boldsymbol{y},x_2)| < 2|x_1-x_2|$. To see this, first observe that for the special case of S_n, it holds that $S_n(\boldsymbol{y},x_1)-S_n(\boldsymbol{y},x_2)=1-1=0$. Furthermore, for all $i<n$, we have

$$\begin{aligned}
&|S_i(\boldsymbol{y},x_1)-S_i(\boldsymbol{y},x_2)|\\
=&\Bigg|\sum_{j=1}^{i}\left(\frac{1}{n}+\frac{1}{n}\Big(R_{CRPS}(y_j,x_1)-\frac{1}{n-1}\sum_{k\neq j}R_{CRPS}(y_k,x_1)\Big)\right)\\
&-\sum_{j=1}^{i}\left(\frac{1}{n}+\frac{1}{n}\Big(R_{CRPS}(y_j,x_2)-\frac{1}{n-1}\sum_{k\neq j}R_{CRPS}(y_k,x_2)\Big)\right)\Bigg|\\
=&\frac{1}{n}\Bigg|\sum_{j=1}^{i}\big(R_{CRPS}(y_j,x_1)-R_{CRPS}(y_j,x_2)\big)\\
&-\frac{1}{n-1}\sum_{j=1}^{i}\sum_{k\neq j}\big(R_{CRPS}(y_k,x_1)-R_{CRPS}(y_k,x_2)\big)\Bigg|\\
\leq&\frac{1}{n}(i|x_1-x_2|+i|x_1-x_2|)<2|x_1-x_2|,
\end{aligned} \tag{4}$$

where the first inequality holds by the triangle inequality and Equation 2.

We now show that the number of cut points is finite. Fix $i \in [n]$. We show that for $m \geq 2$ there can be at most $m+1$ points such that $\text{frac}^+(mx+b)=S_i(\boldsymbol{y},x)$. Note that on the domain $[0,1]$, the function $\text{frac}^+(mx+b)$ consists of at most $m+1$ linear pieces, each with slope m. If there were more than $m+1$ points such that $\text{frac}^+(mx+b)=S_i(\boldsymbol{y},x)$, then there must exist x_1, x_2, both within a single linear piece, with $|S_i(\boldsymbol{y},x_1)-S_i(\boldsymbol{y},x_2)| = m|x_1-x_2| \geq 2|x_1-x_2|$, contradicting Equation 4. Summing across all choices of i, the number of cut points (not including the end points 0 and 1) is at most $n(m+1)$ which, in particular, is finite.

To complete the proof that every agent is allocated a finite union of subintervals, we prove the fact, already claimed in Step 5 of the algorithm, that, in the candidate allocation created in that step, all crumbs $x \in (c_{k-1}, c_k)$ are allocated to the same agent. Let $x_1, x_2 \in (c_{k-1}, c_k)$ and suppose that x_1 is allocated to agent i while x_2 is allocated to agent $j \neq i$. Suppose, without loss of generality, that $i < j$. From the definition of the candidate allocation, we have that $S_i(\boldsymbol{y}, x_1) > \text{frac}^+(mx_1 + b)$ but $S_i(\boldsymbol{y}, x_2) \leq \text{frac}^+(mx_2 + b)$. If $S_i(\boldsymbol{y}, x_2) = \text{frac}^+(mx_2 + b)$ then x_2 is a cut point, contradicting the assumption that x_2 lies strictly between two adjacent cut points c_{k-1} and c_k. So assume that $S_i(\boldsymbol{y}, x_2) < \text{frac}^+(mx_2 + b)$. If x_1 and x_2 lie on the same linear piece of $\text{frac}^+(mx + b)$, then, because S_i is also continuous (Equation 4), the intermediate value theorem implies the existence of a value $x' \in (x_1, x_2)$ with $S_i(\boldsymbol{y}, x') = \text{frac}^+(mx' + b)$, which implies that x' is a cut point, again contradicting that c_{k-1} and c_k are adjacent cut points. Finally, if x_1 and x_2 lie on different linear pieces then there exists a point $x' \in (x_1, x_2)$ with $\text{frac}^+(mx' + b) = 1$.

Part 2: SPA is a CCSR.

To show that the mechanism is a CCSR, we will show that $\Pr_{\mathcal{D}\sim\mathcal{M}}(x \in \mathcal{D}_i(\boldsymbol{y})) = s_i(\boldsymbol{y}, x) = \frac{1}{n} + \frac{1}{n}\left(R_{CRPS}(y_i, x) - \frac{1}{n-1}\sum_{j\neq i} R_{CRPS}(y_j, x)\right)$ for all profiles of reports $\boldsymbol{y}$, all crumbs $x \in [0,1]$, and all agents $i \in [n]$. First, note that if all agents make the same report, $s_i(\boldsymbol{y}, x) = \frac{1}{n}$ for all agents i. Step 1 of SPA implements these probabilities because matching agents to pieces uniformly at random guarantees that every crumb is equally likely to go to any agent. Second, if two or more agents make different reports, then the algorithm proceeds to Steps 2–6. Let x be a crumb, i be an agent, and $\boldsymbol{y}$ be a profile of reports. Given any $m \in \mathbb{N}$, the probability that crumb x is allocated to agent i is exactly $s_i(\boldsymbol{y}, x)$. To see this, note that crumb x is allocated to agent i if and only if $S_i(\boldsymbol{y}, x) > \text{frac}^+(mx + b) \geq S_{i-1}(\boldsymbol{y}, x)$. Because $\text{frac}^+(mx + b)$ is uniformly distributed on $(0, 1]$ for fixed m and x, this happens with probability equal to $S_i(\boldsymbol{y}, x) - S_{i-1}(\boldsymbol{y}, x) = s_i(\boldsymbol{y}, x)$. Note that this probability is independent of m, and in particular is unaffected by whichever value m takes in Step 6. Thus, Steps 2–6 yield candidate allocations in accordance with CCSR probabilities. Third, consider Step 7 and observe that the candidate allocation remains unchanged from Step 5 for all agents with distinct reports, i.e., for any agent i with $y_i \neq y_j$

for all $j \neq i$. Consider then an agent i who belongs to a group G_k. Crumb x belongs to this group G_k with probability $\sum_{j \in G_k} s_j(\boldsymbol{y}, x)$. Since all group members have identical reports and any crumb belonging to the group is allocated among the group's members uniformly at random, the probability of crumb x being allocated to agent i is $\frac{\sum_{j \in G_k} s_j(\boldsymbol{y},x)}{n_k} = s_i(\boldsymbol{y}, x)$.

Part 3: SPA satisfies ex post properties.

If $y_i = y_j$ for all $i, j \in [n]$, then, by definition of the Common Valuation subroutine, the algorithm outputs an allocation for which every agent receives a piece of cake worth exactly $1/n$ to both them and every other agent. Therefore, we focus on the case where at least two agents report distinct valuation functions. We break the remainder of the proof into stages to ease readability.

Stage 1: Define slack in every agents' expected utility such that if their ex post utility is within this slack, then the algorithm reaches Step 7. We must show that there exists a sufficiently large value M such that the two conditions in Step 6 hold for all $m > M$, which guarantees that Step 7 will eventually be reached by continuously doubling m. The two conditions in Step 6 are about the agents' realized (ex post) utilities. Because, for any fixed $m \geq 2$, the candidate allocations resulting from Steps 4 and 5 implement strict CCSR crumb allocation probabilities, it is the case that the two conditions hold in expectation (over draws of b).

For every agent i, define the *slack* of agent i, denoted slack_i, as the minimum of two values: (1) the difference between agent i's expected utility for her piece and $1/n$, and (2) half the difference between her expected utility for her own piece and the maximum of her expected utility for any other agent j's piece with $y_j \neq y_i$. That is,

$$\begin{aligned}\text{slack}_i = \min\Bigg\{ &\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] - \frac{1}{n}, \\ &\frac{1}{2}\left(\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y}), y_i)\big] - \max_{j\in[n], y_j \neq y_i}\left\{\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)\big]\right\}\right)\Bigg\},\end{aligned}$$

where $\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y}), y_i)\big] = \int_0^1 y_i(x) s_j(\boldsymbol{y}, x)\, \mathrm{d}x$ is the expected utility that agent i has for agent j's piece. Note that, due to ex ante strong proportionality and ex ante strong envy-freeness, $\text{slack}_i > 0$ for all $i \in [n]$. Condition C1 of Step 6 is satisfied if, for all $i \in [n]$, the difference between i's ex ante and ex post utility for her own piece is less than slack_i (by the first part of the slack definition). Condition C2 of Step 6 is satisfied whenever, for all

$i, j \in [n]$ with $y_j \neq y_i$, both the difference between i's ex ante and ex post utility for her own piece and the difference between i's ex ante and ex post utility for j's piece are less than slack_i (by the second part of the slack definition).

Stage 2: Use half of each agent's slack to bound the difference between ex ante and ex post utility from regions that may contain Lipschitz violations. Fix $i \in [n]$ and let $\epsilon = \frac{\text{slack}_i}{2}$. Then, by the assumption that y_i is relaxed Lipschitz, there exists a piece of cake $z = z_1 \cup \ldots \cup z_r$ such that $u_i(z, y_i) \geq 1 - \epsilon$ and that y_i is Lipschitz on each subinterval of z. Denote by L the maximum Lipschitz constant for y_i over all subintervals. Because $u_i([0,1] \backslash z, y_i) \leq \epsilon$, the difference between ex ante and ex post utility on $[0,1] \backslash z$ is at most ϵ. We use the other half of each agent's slack to bound the difference between ex ante and ex post utility from z. In particular, if we can show that, for sufficiently large m and all $j \in [n]$, the difference between i's ex ante and ex post utility for $z \cap a_j$ is at most ϵ, then the total utility difference is less than slack_i.

Stage 3: Divide the cake into phases that start and end every time frac$^+(mx + b) = 1$. Consider some arbitrary $m \geq 2$. We begin by analyzing a single "phase," defined as an interval of width $1/m$ between any two neighboring solutions to $\text{frac}^+(mx + b) = 1$, i.e., $[x', x' + \frac{1}{m}]$ with $0 \leq x' < x' + \frac{1}{m} \leq 1$ such that $mx' + b$ is an integer. Note that dividing the cake into phases in this way might ignore pieces of cake with length less than $1/m$ at the beginning and the end of the cake, which we will account for later. Additionally, we first analyze only phases that are subsets of z, so that y_i is Lipschitz on the entire phase, and account for the remaining phases later.

Stage 4: Fix a phase $\Pi \subset z$. Every agent receives a subinterval from the phase. Lower bound the (ex post) utility that an agent i receives from that subinterval and upper bound her (ex post) utility for another agent j's subinterval. Let $\Pi \subset z$ be a single phase. By definition, each agent is allocated exactly one subinterval from Π. Denote agent i's subinterval from Π as $\Pi_i = \Pi \cap a_i$. For $i = 1$, that subinterval is $\Pi_i = [\min(\Pi_i), \max(\Pi_i)] = [x', c_k]$ for some $k \in [\ell]$, with $\text{frac}^+(mx' + b) = 1$ and $\text{frac}^+(mc_k + b) = S_1(\boldsymbol{y}, c_k)$. Additionally, x' is the cut point immediately to the left of c_k, i.e., $x' = c_{k-1}$. We therefore have that $mc_{k-1} + b = t$ for some integer t, and that $mc_k + b = t + S_1(\boldsymbol{y}, c_k)$. From this it immediately follows that $mc_k - mc_{k-1} = S_1(\boldsymbol{y}, c_k) = s_1(\boldsymbol{y}, c_k)$, and therefore that $c_k - c_{k-1} = \frac{s_1(\boldsymbol{y}, c_k)}{m}$.

For $i \geq 2$, the corresponding subinterval is $\Pi_i = [c_{k-1}, c_k]$, where $\text{frac}^+(mc_{k-1} + b) = S_{i-1}(\boldsymbol{y}, c_{k-1})$ and $\text{frac}^+(mc_k + b) = S_i(\boldsymbol{y}, c_k)$ for some $k \in [\ell]$. We need upper and lower bounds on the length of Π_i, i.e., on $c_k - c_{k-1}$. Note that since c_{k-1} and c_k both lie in the same phase Π, it holds that $\text{frac}^+(mc_k + b) - \text{frac}^+(mc_{k-1} + b) = mc_k + b - (mc_{k-1} + b) = m(c_k - c_{k-1})$, i.e., $c_k - c_{k-1} = \frac{\text{frac}^+(mc_k+b) - \text{frac}^+(mc_{k-1}+b)}{m}$. We have

$$
\begin{aligned}
c_k - c_{k-1} &= \frac{\text{frac}^+(mc_k + b) - \text{frac}^+(mc_{k-1} + b)}{m} \\
&= \frac{S_i(\boldsymbol{y}, c_k) - S_{i-1}(\boldsymbol{y}, c_{k-1})}{m} \\
&= \frac{S_i(\boldsymbol{y}, c_k) - S_{i-1}(\boldsymbol{y}, c_k) + S_{i-1}(\boldsymbol{y}, c_k) - S_{i-1}(\boldsymbol{y}, c_{k-1})}{m} \\
&= \frac{s_i(\boldsymbol{y}, c_k) + S_{i-1}(\boldsymbol{y}, c_k) - S_{i-1}(\boldsymbol{y}, c_{k-1})}{m}
\end{aligned}
$$

where the last equality holds because, by definition, $S_i(\boldsymbol{y}, x) = S_{i-1}(\boldsymbol{y}, x) + s_i(\boldsymbol{y}, x)$. We can now use the fact that $|S_{i-1}(\boldsymbol{y}, c_k) - S_{i-1}(\boldsymbol{y}, c_{k-1})| \leq 2|c_k - c_{k-1}|$ for all $i \in \{2, \ldots, n\}$ (Equation 4), along with $c_k - c_{k-1} \geq 0$, to obtain the following bounds:

$$
\frac{s_i(\boldsymbol{y}, c_k) - 2(c_k - c_{k-1})}{m} \leq c_k - c_{k-1} \leq \frac{s_i(\boldsymbol{y}, c_k) + 2(c_k - c_{k-1})}{m}.
$$

Rearranging and substituting $c_k = \max(\Pi_i)$ and $c_{k-1} = \min(\Pi_i)$ yields

$$
\frac{s_i(\boldsymbol{y}, \max(\Pi_i))}{m+2} \leq \max(\Pi_i) - \min(\Pi_i) \leq \frac{s_i(\boldsymbol{y}, \max(\Pi_i))}{m-2},
$$

for $m > 2$, which is sufficient because we only need to show that there exists a sufficiently large value M such that the two conditions in Step 6 hold for all $m > M$. From this it follows immediately that the (realized) utility that agent i gets from Π_i is at least

$$
u_i(\Pi_i, y_i) \geq \min_{x \in \Pi} y_i(x) \cdot \frac{s_i(\boldsymbol{y}, \max(\Pi_i))}{m+2}. \tag{5}
$$

Moreover, the utility that agent i assigns to the subinterval of Π that agent $j \in [n]$ receives is at most

$$
u_i(\Pi_j, y_i) \leq \max_{x \in \Pi} y_i(x) \cdot \frac{s_j(\boldsymbol{y}, \max(\Pi_j))}{m-2}. \tag{6}
$$

Note that these bounds are derived from examining the length of Π_j for $j \geq 2$, but observe that the exact length of Π_1 derived earlier is consistent with both bounds.

Stage 5: Upper bound the expected utility that agent i obtains from phase Π and lower bound the expected utility that agent i has for the subinterval of Π that j receives. The expected utility $\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y})\cap\Pi, y_i)\big]$ that agent i obtains from phase Π is upper bounded by the length of Π multiplied by the maximum value of agent i's valuation function on Π, multiplied by the maximum marginal probability that any given crumb in Π is allocated to i. That is, $\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y})\cap\Pi, y_i)\big] \le \frac{1}{m}\cdot\max_{x\in\Pi} y_i(x)\cdot\max_{x\in\Pi} s_i(\boldsymbol{y},x)$. Intuitively, $\frac{1}{m}\cdot\max_{x\in\Pi} s_i(\boldsymbol{y},x)$ bounds the expected length of Π_i.

We also lower bound the expected utility $\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y})\cap\Pi, y_i)\big]$ that agent i assigns to the subinterval of Π that agent $j\in[n]$ receives. We can establish a lower bound analogously to the preceding upper bound: by multiplying the length of Π by the minimum value of agent i's valuation function on Π, multiplied by the minimum marginal probability that any given crumb in Π is allocated to j. That is, $\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y})\cap\Pi, y_i)\big] \ge \frac{1}{m}\cdot\min_{x\in\Pi} y_i(x)\cdot \min_{x\in\Pi} s_j(\boldsymbol{y},x)$.

Stage 6: Show that, in any given phase, agent i's ex post and ex ante utilities are close to each other (both for her own subinterval and for the subinterval of another agent j). The intuition for this stage is that because neither the reported valuation functions nor the functions s_i are "too steep," they can be treated as approximately flat for large enough m. If they were exactly flat then expected and ex post utilities would exactly match; since they are not exactly flat there is a gap, but this gap is not too large. Formally, we have

$$u_i(\Pi_i, y_i) \ge \min_{x\in\Pi} y_i(x)\cdot\frac{s_i(\boldsymbol{y},\max(\Pi_i))}{m+2} \ge \left(\max_{x\in\Pi} y_i(x) - \frac{L}{m}\right)\cdot\left(\frac{\max_{x\in\Pi} s_i(\boldsymbol{y},x) - \frac{1}{m}}{m+2}\right),$$

where the first inequality follows from Equation 5, the first half of the product on each side of the second inequality from the fact that y_i is Lipschitz on Π with Lipschitz bound L, and the second half of the product on each side of the second inequality from the fact that the rate of change of s_i is at most 1 (Equation 3).

Let $h = \max_{x\in\Pi} y_i(x)$. Note that h exists because y_i is Lipschitz on Π. Taking the difference between ex ante and ex post utility yields

$$\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y})\cap\Pi, y_i)\big] - u_i(\Pi_i, y_i)$$

$$
\begin{aligned}
&\leq \frac{1}{m} \cdot \max_{x \in \Pi} y_i(x) \cdot \max_{x \in \Pi} s_i(\boldsymbol{y}, x) - \left(\max_{x \in \Pi} y_i(x) - \frac{L}{m} \right) \cdot \left(\frac{\max_{x \in \Pi} s_i(\boldsymbol{y}, x) - \frac{1}{m}}{m+2} \right) \\
&= \frac{2h \max_{x \in \Pi} s_i(\boldsymbol{y}, x)}{m(m+2)} + \frac{L \max_{x \in \Pi} s_i(\boldsymbol{y}, x)}{m(m+2)} + \frac{h}{m(m+2)} - \frac{L}{m^2(m+2)} \\
&\leq \frac{2h}{m(m+2)} + \frac{L}{m(m+2)} + \frac{h}{m(m+2)} = \frac{3h+L}{m(m+2)},
\end{aligned}
$$

where the first inequality follows from the bounds we derived, the equality follows from simple algebra and the definition of h, the second inequality follows from the fact that $s_i(\boldsymbol{y}, x) \leq 1$ for all $\boldsymbol{y}, x$ and from dropping the negative term.

For agent i's utility for the part of Π assigned to agent j, we can argue analogously, and we obtain

$$
u_i(\Pi_j, y_i) \leq \max_{x \in \Pi} y_i(x) \cdot \frac{s_j(\boldsymbol{y}, \max(\Pi_i))}{m-2} \leq \left(\min_{x \in \Pi} y_i(x) + \frac{L}{m} \right) \cdot \left(\frac{\min_{x \in \Pi} s_j(\boldsymbol{y}, x) + \frac{1}{m}}{m-2} \right),
$$

where the first inequality follows from Equation 6, the first half of the product on each side of the second inequality from the fact that y_i is Lipschitz on Π with Lipschitz bound L, and the second half of the product on each side of the second inequality from the fact that the rate of change of s_i is at most 1 (Equation 3).

Let $l = \min_{x \in \Pi} y_i(x)$. Taking the difference between ex post and ex ante utility gives us

$$
\begin{aligned}
&u_i(\Pi_j, y_i) - \mathop{\mathbf{E}}_{\mathcal{D} \sim \mathcal{M}} \left[u_i(\mathcal{D}_j(\boldsymbol{y}) \cap \Pi, y_i) \right] \\
&\leq \left(\min_{x \in \Pi} y_i(x) + \frac{L}{m} \right) \cdot \left(\frac{\min_{x \in \Pi} s_j(\boldsymbol{y}, x) + \frac{1}{m}}{m-2} \right) - \frac{1}{m} \cdot \min_{x \in \Pi} y_i(x) \cdot \min_{x \in \Pi} s_j(\boldsymbol{y}, x) \\
&= \frac{2l \min_{x \in \Pi} s_j(\boldsymbol{y}, x)}{m(m-2)} + \frac{L \min_{x \in \Pi} s_j(\boldsymbol{y}, x)}{m(m-2)} + \frac{l}{m(m-2)} + \frac{L}{m^2(m-2)} \\
&\leq \frac{2l}{m(m-2)} + \frac{L}{m(m-2)} + \frac{l}{m(m-2)} + \frac{L}{m^2(m-2)} \leq \frac{3l+2L}{m(m-2)},
\end{aligned}
$$

where the first inequality follows from the bounds we derived, the equality follows from simple algebra and the definition of l, and the second inequality follows from the fact that $s_i(\boldsymbol{y}, x) \leq 1$ for all $\boldsymbol{y}, x$.

Stage 7: Consider phases that intersect but are not fully contained in z and partial phases at the start and end of the cake. Bound agent i's expected and ex post utility for her own subintervals of these phases as well as those of any other agent j. Thus far, we have only considered phases Π with $\Pi \subset z$. We have also implicitly accounted for phases Π with $\Pi \cap z = \emptyset$ (low value by definition) using the first half of the slack. It remains to account for (1) the phases that are neither fully contained in nor fully separate from z (i.e., those Π with $\Pi \cap z \subsetneq \Pi$ and $\Pi \cap z \neq \emptyset$) as well as (2) the potential "partial phases" at the very beginning and end of the cake with length less than $1/m$. The number of both of these types of phases is bounded and independent of m, so that, as m increases and the width of each phase decreases, the total value from these two types of phases approaches zero. In particular, since z consists of r subintervals, there are at most $2r$ phases of the first type and at most two partial phases. Let $h = \max_{x \in \Pi \cap z} y_i(x)$; note that this is consistent with our previous definition of h for Π with $\Pi \subset z$. First consider a phase Π with $\Pi \cap z \subsetneq \Pi$ and $\Pi \cap z \neq \emptyset$; since we have already accounted for agent i's utility from $[0, 1] \setminus z$, we only need to consider her utility from $\Pi \cap z$. Agent i obtains at least 0 realized utility and at most $\frac{h}{m}$ expected utility from $\Pi \cap z$ (since the length of $\Pi \cap z$ is at most $1/m$ and y_i is upper bounded by h on $\Pi \cap z$). By identical reasoning, the same bounds apply for agent i's expected and realized utility for each of the two partial phases at the beginning and end of the cake. Taking care of the utility that agent i assigns to agent j's piece uses the same logic with the arguments reversed. That is, for any phase Π with $\Pi \cap z \subsetneq \Pi$, agent i has at least 0 expected utility for the part of $\Pi \cap z$ allocated to any agent j, and would assign at most $\frac{h}{m}$ utility to the part of $\Pi \cap z$ allocated to j, with the same bounds holding for the two partial phases at the start and the end of the cake.

Stage 8: All necessary bounds have been derived. Put them together to bound (1) the difference between agent i's expected and realized utility for her own piece and (2) the difference between agent i's ex post and expected utility for any agent j's piece with $y_j \neq y_i$. Taking into account at most $2r$ phases Π with $\Pi \cap z \subsetneq \Pi$ and $\Pi \cap z \neq \emptyset$, at most two partial phases at the start and end of the cake, and as many as m full phases, the difference between agent i's expected and realized utility from z is at most

$$\mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_i(\boldsymbol{y})\cap z, y_i)\big] - u_i(a_i\cap z, y_i) \le (2r+2)\frac{h}{m} + m\frac{3h+L}{m(m+2)} = \frac{(2r+2)h}{m} + \frac{3h+L}{m+2}.$$

For the difference between agent i's ex post and expected utility for the part of z assigned to agent j with $y_j \neq y_i$, we have

$$u_i(a_j\cap z, y_i) - \mathop{\mathbf{E}}_{\mathcal{D}\sim\mathcal{M}}\big[u_i(\mathcal{D}_j(\boldsymbol{y})\cap z, y_i)\big] \le (2r+2)\frac{h}{m} + m\frac{3l+2L}{m(m-2)} = \frac{(2r+2)h}{m} + \frac{3l+2L}{m-2}.$$

Let M denote the value of m so that the larger of the two differences is exactly ϵ. Then, for any $m > M$, both differences will be strictly less than ϵ, thus guaranteeing that the algorithm reaches Step 7.

Stage 9: Show that Step 7 of the SPA mechanism removes envy from agents with identical reports without reintroducing proportionality violations or envy between agents with different reports. We denote agent i's candidate allocation at the end of Step 7ii with its respective value of t as a superscript, i.e., a_i^t, and the candidate allocation at the end of Step 6 by a_i^0. At the end of Step 6, the candidate allocation is such that every agent receives strictly more than $1/n$ utility, and every agent strictly prefers her own piece to the piece of any agent with a different report. We will show that both of these properties are preserved by Step 7. Note that at the end of Step 6, it is possible that some agent i envies some other agent j with $y_i = y_j$. Step 7ii removes this envy, by definition, so that retaining Conditions C1 and C2 from Step 6 is sufficient to prove the theorem.

We begin by showing that Condition C1 is retained by Step 7. Let $i \in G_{k'}$ for some $k' \in [q]$. Observe that, as Step 7 progresses, no part of the cake ever leaves or is added to a group of agents with the same report; the redistribution only takes places within groups. Furthermore, by definition of the Common Valuation Subroutine, envy between agents with the same report (i.e., within the same group) is eliminated after even a single iteration of Step 7ii. Therefore, $u_i(a_i^t, y_i) = \frac{1}{n_{k'}}\sum_{i'\in G_{k'}} u_i(a_{i'}^0, y_i)$ for all $t \ge 1$. Since $y_i = y_{i'}$ for all $i' \in G_{k'}$, it is also the case that $u_i(a_{i'}^0, y_i) = u_{i'}(a_{i'}^0, y_{i'}) > \frac{1}{n}$, where the inequality follows from the fact that Condition C1 held at the end of Step 6. Combining these two facts yields $u_i(a_i^t, y_i) > \frac{1}{n}$ for all $t \ge 1$. That is, Condition C1 is retained throughout the progression of Step 7.

The remainder of the proof establishes that Condition C2 is retained for high enough t, i.e., that envy is removed between pairs of agents in different groups. To that end, consider a group of n_k agents G_k who all make the same report, and another agent $i \in G_{k'} \neq G_k$. For every $j \in G_k$, it follows from Condition C2 that $u_i(a_i^0, y_i) > u_i(a_j^0, y_i)$. Furthermore, for every $i' \in G_{k'}$, it holds that $u_i(a_{i'}^0, y_i) = u_{i'}(a_{i'}^0, y_{i'}) > u_{i'}(a_j^0, y_{i'}) = u_i(a_j^0, y_i)$, with both equalities following from $y_i = y_{i'}$ and the inequality again from Condition C2. Thus, agent i's average utility for the pieces of agents in $G_{k'}$ is greater than her average utility for the pieces of agents in G_k, i.e., $\frac{1}{n_{k'}} \sum_{i' \in G_{k'}} u_i(a_{i'}^0, y_i) > \frac{1}{n_k} \sum_{j \in G_k} u_i(a_j^0, y_i)$. Let

$$\text{slack}_{k',k} = u_i(a_i^t, y_i) - \frac{1}{n_k} \sum_{j \in G_k} u_i(a_j^0, y_i) = \frac{1}{n_{k'}} \sum_{i' \in G_{k'}} u_i(a_{i'}^0, y_i) - \frac{1}{n_k} \sum_{j \in G_k} u_i(a_j^0, y_i) > 0$$

denote this gap. Therefore, the termination condition in Step 7iii is satisfied for $i \in G_{k'}$ and $j \in G_k$ for any t such that $u_i(a_j^t, y_i) - \frac{1}{n_k} \sum_{j' \in G_k} u_i(a_{j'}^0, y_i) < \text{slack}_{k',k}$. That is, the utility assigned to any agent j's piece is less than $\text{slack}_{k',k}$ higher than the average utility i assigns to j's group. We will show that there exists a T such that the termination condition is satisfied for all $t > T$. Taking the maximum such T over all k, k' guarantees termination.

The remainder of the proof proceeds similarly to the earlier part, where we bounded the differences between expected and realized utilities. For clarity, we reuse much of the same notation from that part. We also follow the logic of that part in the sense that we divide the slack into two equal-sized parts. The first half is used to account for the part of the cake where y_i is not Lipschitz and the second half is used to account for increasingly small differences between the utility that i assigns to j's piece and the average utility of G_k. Let $\epsilon = \frac{\text{slack}_{k',k}}{2}$. Then, since y_i is relaxed Lipschitz, there exists a piece of cake $z = z_1 \cup \ldots \cup z_r$ such that $u_i(z, y_i) \geq 1 - \epsilon$ and that y_i is Lipschitz on each subinterval of z. Denote by L the maximum Lipschitz constant for y_i over all subintervals.

For the first half of the slack, because i values the non-Lipschitz parts of the cake by at most ϵ, i.e., $u_i([0,1] \backslash z, y_i) \leq \epsilon$, even adding all of these parts to agent j's piece would only increase i's utility for j's piece by at most ϵ. For the second half of the slack, we need to show that there exists a T such that for all $t > T$, agent i's utility for agent j's piece is within ϵ of agent i's average utility for the pieces of agents in G_k when restricted to the Lipschitz portion of y_i, i.e., $u_i(a_j^t \cap z, y_i) < \frac{1}{n_k} \sum_{j' \in G_k} u_i(a_{j'}^0 \cap z, y_i) + \epsilon$ for all $t > T$. Once this part for the second half of the slack is also shown, we will have

$$u_i(a_j^t, y_i) = u_i(a_j^t \cap z, y_i) + u_i(a_j^t \cap ([0,1] \backslash z), y_i) \leq u_i(a_j^t \cap z, y_i) + u_i([0,1] \backslash z, y_i)$$

$$\leq u_i(a_j^t \cap z, y_i) + \epsilon < \frac{1}{n_k} \sum_{j' \in G_k} u_i(a_{j'}^0 \cap z, y_i) + 2\epsilon \leq u_i(a_i^t, y_i)$$

To that end, for the second part of the slack, consider a single subinterval $A_k = [\underline{d}, \overline{d}]$ of $\cup_{j' \in G_k} a_{j'}$ and a single bin Π of length $\frac{\overline{d}-\underline{d}}{t}$, one of the t equal-length subintervals of A_k created in Step 7ii. As part of her piece a_j^t, agent $j \in G_k$ receives a "sub piece" of length $\frac{\overline{d}-\underline{d}}{tn_k}$ from that bin.

Consider a bin Π that is fully contained in z, i.e., $\Pi \subset z$. Recall that agent $i \in G_{k'}$ is in a different group than agent $j \in G_k$. Agent i's utility for the whole bin, which is allocated to group G_k, is $u_i(\Pi, y_i) \geq (\min_{x \in \Pi} y_i(x)) \frac{\overline{d}-\underline{d}}{t}$ and her average utility for the pieces of all agents in G_k is $\frac{1}{n_k} u_i(\Pi, y_i) \geq (\min_{x \in \Pi} y_i(x)) \frac{\overline{d}-\underline{d}}{tn_k}$. Furthermore, her utility for j's sub piece of the bin is at most

$$u_i(a_j^t \cap \Pi, y_i) \leq (\max_{x \in \Pi} y_i(x)) \frac{\overline{d}-\underline{d}}{tn_k} \leq \left(\min_{x \in \Pi} y_i(x) + L \frac{\overline{d}-\underline{d}}{t} \right) \frac{\overline{d}-\underline{d}}{tn_k}.$$

Therefore, the difference between i's utility for j's sub piece of Π and i's average utility from bin Π over all agents in G_k is at most (the inequality in fact holds strictly but the argument is more intricate and the weak version is sufficient for our purposes)

$$\left(\min_{x \in \Pi} y_i(x) + L \frac{\overline{d}-\underline{d}}{t} \right) \frac{\overline{d}-\underline{d}}{tn_k} - (\min_{x \in \Pi} y_i(x)) \frac{\overline{d}-\underline{d}}{tn_k} = L \frac{(\overline{d}-\underline{d})^2}{t^2 n_k} \leq \frac{L}{t^2 n_k}.$$

We must also account for bins Π that are not fully contained in z, i.e., $\Pi \cap z \subsetneq \Pi$ and $\Pi \cap z \neq \emptyset$ (recall that bins that do not intersect z are already accounted for above as non-Lipschitz parts of the cake). Similar to the earlier part of the proof, there can be as many as $2r$ bins within A_k that are not fully contained within z. (There can be $2r$ intersecting bins across the whole $[0, 1]$ interval and here we are only considering $A_k \subseteq [0, 1]$, so the bound still applies.) For each of these bins Π not fully contained in z, we are still interested in agent i's utility only for those parts that intersect with z. Her utility for agent j's sub piece from $\Pi \cap z$ is at most $u_i(a_j^t \cap \Pi \cap z, y_i) \leq h \frac{\overline{d}-\underline{d}}{t} \leq \frac{h}{t}$, where $h = \max_{x \in \Pi \cap z} y_i(x)$.

To obtain the difference between i's utility for j's sub piece of A_k and i's average utility from A_k over all agents in G_k, we sum over as many as t bins fully contained within z and as many as $2r$ intersecting but not fully contained in z. Doing so, we get the following upper bound:

$$u_i(a_j^t \cap z, y_i) - \frac{1}{n_k} \sum_{j' \in G_k} u_i(a_{j'}^0 \cap z, y_i) \leq t \left(\frac{L}{t^2 n_k} \right) + 2r\frac{h}{t} = \frac{L}{tn_k} + \frac{2rh}{t}.$$

Finally we must sum over all subintervals A_k of $\cup_{j' \in G_k} a_{j'}$. Note that since $\cup_{j' \in G_k} a_{j'}$ is a union of pieces of cake, the number of subintervals A_k is finite. Since, at the beginning of Step 7, a large enough m has been chosen and every agent's piece in the candidate allocation consists of at most $m+1$ subintervals (at most 1 per phase), there are at most $n_k(m+1)$ subintervals A_k. Set T so that

$$n_k(m+1) \left(\frac{L}{Tn_k} + \frac{2rh}{T} \right) \leq \epsilon.$$

Then for any $t > T$, the difference between i's utility for j's piece and i's average utility for the pieces of all agents in G_k is at most ϵ.

REFERENCES

AZIZ, HARIS AND CHUN YE (2014): "Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations," in *Web and Internet Economics: 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings 10*, Springer, 1–14. [2, 4, 5, 24]

BARBANEL, JULIUS B (1996): "Super Envy-Free Cake Division and Independence of Measures," *Journal of Mathematical Analysis and Applications*, 197 (1), 54–60. [8, 11, 12, 23]

BEI, XIAOHUI, NING CHEN, GUANGDA HUZHANG, BIAOSHUAI TAO, AND JIAJUN WU (2017): "Cake Cutting: Envy and Truth." in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 3625–3631. [2]

BEI, XIAOHUI, GUANGDA HUZHANG, AND WARUT SUKSOMPONG (2020): "Truthful Fair Division without Free Disposal," *Social Choice and Welfare*, 55, 523–545. [3]

BRAMS, STEVEN J AND ALAN D TAYLOR (1996): *Fair Division: From Cake-Cutting to Dispute Resolution*, Cambridge University Press. [2]

BRÂNZEI, SIMINA AND PETER BRO MILTERSEN (2015): "A Dictatorship Theorem for Cake Cutting," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. [22]

BRIER, GLENN W. (1950): "Verification of Forecasts Expressed in Terms of Probability," *Monthly Weather Review*, 78 (1), 1–3. [6, 13, 16]

BU, XIAOLIN, JIAXIN SONG, AND BIAOSHUAI TAO (2023): "On existence of truthful fair cake cutting mechanisms," *Artificial Intelligence*, 319, 103904. [2, 4, 5, 6, 25]

BUDISH, ERIC, YEON-KOO CHE, FUHITO KOJIMA, AND PAUL MILGROM (2013): "Designing Random Allocation Mechanisms: Theory and Applications," *American Economic Review*, 103 (2), 585–623. [4]

CHAMBERS, CHRISTOPHER P (2005): "Allocation rules for land division," *Journal of Economic Theory*, 121 (2), 236–258. [2]

CHEN, YILING, JOHN K LAI, DAVID C PARKES, AND ARIEL D PROCACCIA (2013): "Truth, justice, and cake cutting," *Games and Economic Behavior*, 77 (1), 284–297. [3, 4, 5, 14, 25]

COHLER, YUGA J., JOHN K. LAI, DAVID C. PARKES, AND ARIEL D. PROCACCIA (2011): "Optimal envy-free cake cutting," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 626–631. [22]

DUBINS, LESTER E AND EDWIN H SPANIER (1961): "How to Cut a Cake Fairly," *The American Mathematical Monthly*, 68 (1P1), 1–17. [11, 23]

FOLEY, DUNCAN (1967): "Resource Allocation and the Public Sector," *Yale Economics Essays*, 7, 45–98. [2, 12]

FREEMAN, RUPERT, JENS WITKOWSKI, JENNIFER WORTMAN VAUGHAN, AND DAVID M PENNOCK (2023): "An Equivalence Between Fair Division and Wagering Mechanisms," *Management Science*, 70 (10), 6704–6723. [6]

GNEITING, TILMANN AND ADRIAN E. RAFTERY (2007): "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, 102, 359–378. [6, 13]

GOOD, I. J. (1952): "Rational Decisions," *Journal of the Royal Statistical Society. Series B*, 14 (1), 107–114. [6, 13]

KILGOUR, D. MARC AND YIGAL GERCHAK (2004): "Elicitation of Probabilities Using Competitive Scoring Rules," *Decision Analysis*, 1 (2), 108–113. [6]

KUROKAWA, DAVID, JOHN K. LAI, AND ARIEL D. PROCACCIA (2013): "How to Cut a Cake Before the Party Ends," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 555–561. [22]

LAMBERT, NICOLAS, JOHN LANGFORD, JENNIFER WORTMAN, YILING CHEN, DANIEL REEVES, YOAV SHOHAM, AND DAVID M. PENNOCK (2008): "Self-Financed Wagering

Mechanisms for Forecasting," in *Proceedings of the 9th ACM Conference on Electronic Commerce*, ACM, 170–179. [6]

LAMBERT, NICOLAS S., JOHN LANGFORD, JENNIFER WORTMAN VAUGHAN, YILING CHEN, DANIEL M. REEVES, YOAV SHOHAM, AND DAVID M. PENNOCK (2015): "An axiomatic characterization of wagering mechanisms," *Journal of Economic Theory*, 156, 389–416. [6]

MATHESON, JAMES E. AND ROBERT L. WINKLER (1976): "Scoring Rules for Continuous Probability Distributions," *Management Science*, 22 (10), 1087–1096. [6, 13, 14, 16, 17, 29]

MENON, VIJAY AND KATE LARSON (2017): "Deterministic, Strategyproof, and Fair Cake Cutting," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 352–358. [2]

MOSSEL, ELCHANAN AND OMER TAMUZ (2010): "Truthful Fair Division," in *Algorithmic Game Theory: Third International Symposium, SAGT 2010, Athens, Greece, October 18-20, 2010. Proceedings 3*, Springer, 288–299. [3, 5, 11, 14, 25]

NEYMAN, JERZY (1946): "Un théorème d'existence," *Comptes Rendus de l'Académie des Sciences, Paris*, 222, 843–845. [3]

PROCACCIA, ARIEL D. (2013): "Cake Cutting: Not Just Child's Play," *Communications of the ACM*, 56 (7), 78–87. [2, 8]

——— (2016): "Cake Cutting Algorithms," in *Handbook of Computational Social Choice*, ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, Cambridge: Cambridge University Press, 311–330. [2, 22]

ROBERTSON, JACK AND WILLIAM WEBB (1998): *Cake-Cutting Algorithms: Be Fair if You Can*, AK Peters/CRC Press. [22]

SEGAL-HALEVI, EREL AND BALÁZS R SZIKLAI (2019): "Monotonicity and competitive equilibrium in cake-cutting," *Economic Theory*, 68 (2), 363–401. [23]

STEINHAUS, HUGO (1948): "The problem of fair division," *Econometrica*, 16, 101–104. [2, 11]

WEBB, WILLIAM A (1999): "An Algorithm For Super Envy-Free Cake Division," *Journal of Mathematical Analysis and Applications*, 239 (1), 175–179. [23]

WELLER, DIETRICH (1985): "Fair division of a measurable space," *Journal of Mathematical Economics*, 14 (1), 5–17. [23]

WOODALL, DOUGLAS R (1986): "A note on the cake-division problem." *Journal of Combinatorial Theory, Series A*, 42 (2), 300–301. [23]

Numerical Details of Example

This example instance has $n = 2$ agents with reported valuations corresponding to beta distributions $y_1 = \text{Beta}(2,5)$ and $y_2 = \text{Beta}(3,4)$. To implement SPA we must first compute $R_{\text{CRPS}}(y_i, x)$ for $i \in \{1,2\}$. It will be helpful to work with the cumulative distribution functions of the reports, which are given by

$$Y_1(w) = 15\,w^2 - 40\,w^3 + 45\,w^4 - 24\,w^5 + 5\,w^6,$$

$$Y_2(w) = 20\,w^3 - 45\,w^4 + 36\,w^5 - 10\,w^6.$$

Observe that $\int_0^1 \mathbf{1}\{w \geq x\}\,dw = 1 - x$, and therefore

$$\begin{aligned} R_{\text{CRPS}}(y_i, x) &= 1 - \int_0^1 \big(Y_i(w) - \mathbf{1}\{w \geq x\}\big)^2\,dw. \\ &= 1 - \left[\int_0^1 Y_i(w)^2\,dw - 2\int_x^1 Y_i(w)\,dw + \int_0^1 \mathbf{1}\{w \geq x\}\,dw\right] \\ &= 1 - \left[\int_0^1 Y_i(w)^2\,dw - 2\int_x^1 Y_i(w)\,dw + (1-x)\right] \\ &= x + 2\int_x^1 Y_i(w)\,dw - \int_0^1 Y_i(w)^2\,dw. \end{aligned}$$

For agent 1, we have

$$\int_x^1 Y_1(w)\,dw = \frac{5}{7} - \Big(5x^3 - 10x^4 + 9x^5 - 4x^6 + \tfrac{5}{7}x^7\Big)$$

and

$$\int_0^1 Y_1(w)^2\,dw = \frac{625}{1001},$$

so that

$$R_{\text{CRPS}}(y_1, x) = -\frac{10}{7}x^7 + 8x^6 - 18x^5 + 20x^4 - 10x^3 + x \;+\; \frac{115}{143}.$$

Analogously, for agent 2, we obtain

$$R_{\text{CRPS}}(y_2,x)=\frac{20}{7}x^7-12x^6+18x^5-10x^4+x\;+\;\frac{96}{143}.$$

Hence, the CCSR marginals are given by

$$\begin{aligned}s_1(\boldsymbol{y},x)&=\frac{1}{2}+\frac{1}{2}\big(R_{\text{CRPS}}(y_1,x)-R_{\text{CRPS}}(y_2,x)\big)\\&=-\frac{15}{7}\,x^7+10\,x^6-18\,x^5+15\,x^4-5\,x^3+\frac{81}{143}\end{aligned}$$

and $s_2(\boldsymbol{y},x)=1-s_1(\boldsymbol{y},x)$.

For this example run, let the random seed be $b=0.12$. Moreover, the mechanism begins by setting $m=2$, so that the cut points are the union of $\{0,1\}$, those x values for which $\text{frac}^+(2x+0.12)=1$, and those x values for which $s_1(\boldsymbol{y},x)=-\frac{15}{7}\,x^7+10\,x^6-18\,x^5+15\,x^4-5\,x^3+\frac{81}{143}=\text{frac}^+(2x+0.12)$. The x values for which $\text{frac}^+(2x+0.12)=1$ are 0.44 and 0.94. The x values for which $s_1(\boldsymbol{y},x)=-\frac{15}{7}\,x^7+10\,x^6-18\,x^5+15\,x^4-5\,x^3+\frac{81}{143}=\text{frac}^+(2x+0.12)$ are 0.2112258473 and 0.6555044112. That is, there are a total of five subintervals, with agent 1 and agent 2 receiving pieces [0, 0.2112] ∪ [0.44, 0.6555] ∪ [0.94, 1] and [0.2112, 0.44] ∪ [0.6555, 0.94], respectively. The agents' utilities for their own pieces are

$$\begin{aligned}u_1&=\big[Y_1(0.2112)-Y_1(0)\big]+\big[Y_1(0.6555)-Y_1(0.44)\big]+\big[Y_1(1)-Y_1(0.94)\big]\\&\approx 0.3722109\;+\;0.1554797\;+\;0.0000044\;=\;0.5276951,\end{aligned}$$

$$\begin{aligned}u_2&=\big[Y_2(0.44)-Y_2(0.2112)\big]+\big[Y_2(0.94)-Y_2(0.6555)\big]\\&\approx 0.4250186\;+\;0.1113555\;=\;0.5363741.\end{aligned}$$

Common Valuation Subroutine Example

EXAMPLE: Consider $k=3$ agents with valuation function given by

$$v(x)=\begin{cases}\frac{4}{9}, & \text{for } 0\le x\le\frac{1}{4}\\ \frac{16}{9}x+\frac{2}{9}, & \text{for } \frac{1}{4}<x\le\frac{1}{2}\\ \frac{4}{3}, & \text{for } \frac{1}{2}<x\le 1.\end{cases}$$

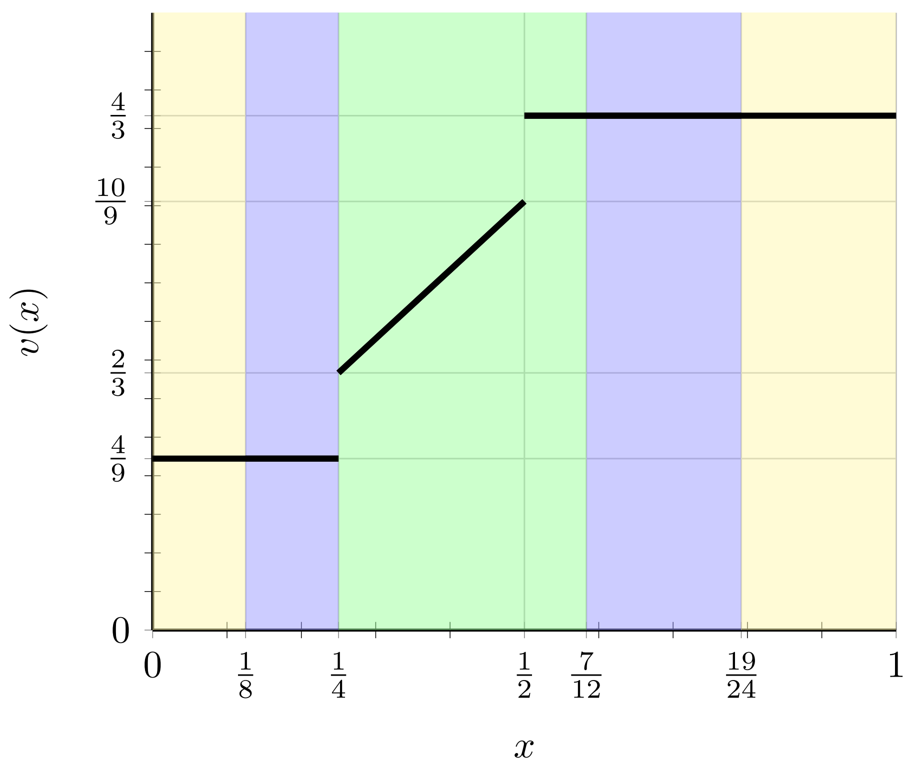

FIGURE 3.— The output of the Common Valuation Subroutine applied to valuation function v from Example 6. The first piece is shown in green, the second piece is shown in blue, and the third piece is shown in yellow.

We apply the Common Valuation subroutine to the entire cake, i.e., $[\alpha, \beta] = [0, 1]$. Figure 3 provides a visual representation of the subroutine's behavior on this example.

Begin by setting $z = \emptyset$. For the first piece, the subroutine reaches $x_1 = \frac{1}{4}$ with $f(x_1, z) = \frac{1}{4} + \frac{1}{k} = \frac{7}{12}$, satisfying the desired condition that $u([x_1, f(x_1, z)] \backslash z, v) = u([\frac{1}{4}, \frac{7}{12}], v) = \frac{1}{3} = u([\alpha, \beta], v)/k$. For the second piece, the subroutine updates z to consist of the allocated portion of the cake, i.e., $z = [\frac{1}{4}, \frac{7}{12}]$, and searches for an appropriate value of x_2. In this iteration, it reaches $x_2 = \frac{1}{8}$ with $f(x_2, z) = \frac{19}{24}$, returning $[x_2, f(x_2, z)] \setminus z = [\frac{1}{8}, \frac{19}{24}] \setminus [\frac{1}{4}, \frac{7}{12}] = [\frac{1}{8}, \frac{1}{4}] \cup [\frac{7}{12}, \frac{19}{24}]$. The length requirement is satisfied because $\text{len}([x_2, f(x_2, z)] \setminus z) = \text{len}([\frac{1}{8}, \frac{1}{4}] \cup [\frac{7}{12}, \frac{19}{24}]) = \frac{1}{3}$ and the utility requirement because $u([x_2, f(x_2, z)] \backslash z, v) = u([\frac{1}{8}, \frac{1}{4}] \cup [\frac{7}{12}, \frac{19}{24}], v) = \frac{1}{3}$. The third piece consists of the remainder of the interval.

Co-editor [Name Surname; will be inserted later] handled this manuscript.