Meta-Envy-Free Cake-Cutting Protocols

Yoshifumi Manabe[1] and Tatsuaki Okamoto[2]

[1] NTT Communication Science Laboratories, 3-1 Morinosato-Wakamitya, Atsugi, Kanagawa 243-0198 Japan manabe.yoshifumi@lab.ntt.co.jp
[2] NTT Information Sharing Platform Laboratories, 3-9-11, Midori-cho, Musashino-shi, 180-8585 Japan okamoto.tatsuaki@lab.ntt.co.jp

Abstract. This paper discusses cake-cutting protocols when the cake is a heterogeneous good that is represented by an interval in the real line. We propose a new desirable property, the meta-envy-freeness of cake-cutting, which has not been formally considered before. Though envy-freeness was considered to be one of the most important desirable properties, envy-freeness does not prevent envy about role assignment in the protocols. We define meta-envy-freeness that formalizes this kind of envy. We show that current envy-free cake-cutting protocols do not satisfy meta-envy-freeness. Formerly proposed properties such as strong envy-free, exact, and equitable do not directly consider this type of envy and these properties are very difficult to realize. This paper then shows meta-envy-free cake-cutting protocols for two and three party cases.

1 Introduction

Cake-cutting is an old problem in game theory. It can be employed for such purposes as dividing territory on a conquered island or assigning jobs to members of a group. This paper discusses the cake-cutting problem when the cake is a heterogeneous good that is represented by an interval $[0, 1]$ in the real line. The most famous cake-cutting protocol is 'divide-and-choose' for two players. Player 1 (Divider) cuts the cake into two equal size pieces. Player 2 (Chooser) takes the piece that she prefers. Divider takes the remaining piece. This protocol is proved to be envy-free. Envy-freeness is defined as: after the assignment is finished, no player wants to exchange his/her part for that of the other player. Divider must cut the cake into two equal size pieces (using Divider's utility function), otherwise Chooser might take the larger piece and Divider will obtain less than half. Since Divider cuts the cake into equal size pieces, she never envies Chooser whichever piece Chooser selects. Chooser never envies Divider because she chooses first.

Although it appears that the 'divide-and-choose' protocol is perfect, actually it is not, because it is not a complete protocol. When Alice and Bob execute this protocol, they must first decide who will be Divider and Chooser. Chooser is the better choice as mentioned in several papers [3][9]. If the utility functions of Alice and Bob are the same, Divider and Chooser obtain exactly half of the cake by using their utility function. Next we consider a case where the utility functions of Alice and Bob differ. Let us assume that Bob is Divider. Let us also assume that by using Bob's utility function, $[0, 1/4]$ and $[1/4, 1]$ is an exact division, because the cake is chocolate coated near 0 and Bob likes chocolate. Alice does not have such a preference, thus by choosing $[1/4, 1]$, Alice's utility is 3/4. If Alice is Divider, she cuts to $[0, 1/2]$ and $[1/2, 1]$. Then Bob chooses $[0, 1/2]$ and obtains more than half by his utility. Therefore, Chooser is never worse than Divider, and Chooser is properly better than Divider if their utility functions differ. If both Alice and Bob know this fact, they both want to be Chooser. Therefore, they must employ a method such as coin-flipping to decide who will be Divider. If Alice is assigned the role of Divider, she definitely *envies* Bob who is Chooser.

Some readers might think that coin-flipping will result in a fair decision between Alice and Bob, and so it is not a problem. If this supposition is accepted, the following protocol must also be accepted: 'Flip a coin and the winner takes the whole cake and the loser gets nothing.' This is an unfair (envy) assignment using fair coin-flipping. Game-theory researchers have discussed cake-cutting protocols where the unfairness (envy) is minimized. If there is the possibility of unfair assignment, we need to consider a better way that eliminates it. Now that we know 'divide-and-choose' is unfair, we must consider eliminating this kind of envy. Although this type of envy is known, it has not been formally defined. This paper defines this type of envy for the first time as meta-envy and proposes new protocols that eliminate it for two-party case and three-party case.

Previous studies defined stronger properties for the obtained portion such as strong envy-free, super envy-free, exact, and equitable [6][13]. These properties are hard to realize and do not directly consider this type of envy. We can obtain a three-party meta-envy-free protocol by modifying a three player envy-free protocol.

Note that we do not eliminate every coin-flip. For the above example of 'divide-and-choose', if Alice and Bob's utility functions are exactly the same, their cutting points are the same. Thus, both Alice and Bob think that the values of the two pieces are the same. To complete the protocol, we must assign each party either piece. Coin-flipping is necessary for such a case, but can only be allowed if its result causes no envy.

2 Preliminaries

Throughout this paper, the cake is a heterogeneous good that is represented by an interval $[0, 1]$ in the real line. Each party P_i has a utility function μ_i that has the following three properties. (1) For any non-empty $X \subseteq [0, 1]$, $\mu_i(X) > 0$. (2) For any X_1 and X_2 such that $X_1 \cap X_2 = \emptyset$, $\mu_i(X_1 \cup X_2) = \mu_i(X_1) + \mu_i(X_2)$. (3) $\mu_i([0, 1]) = 1$. The tuple of $P_i (i = 1, \ldots, n)$'s utility function is denoted by $(\mu_1, \ldots, \mu_n)$. Utility functions might differ among parties. No party has knowledge of the other parties' utility functions.

In this paper, 'party' indicates a person such as Alice, Bob, etc. and is denoted by P. 'Player' is a role in a protocol and is denoted by p. We sometimes state that 'party X is assigned to player y' if a person X executes the role of player y in the protocol.

An n-player cake-cutting protocol f assigns several portions of $[0, 1]$ to the players such that every portion of $[0, 1]$ is assigned to one player. We denote $f_i(\mu_1, \ldots, \mu_n)$ as the set of portions assigned to player p_i by f, when party $P_i(i = 1, \ldots, n)$ is assigned to player $p_i(i = 1, \ldots, n)$ in f. When f is a randomized algorithm, let us denote $f_i(\mu_1, \ldots, \mu_n; r)$ as the assignment to p_i when the sequence of random values used in f is r.

All parties are risk averse, namely they avoid gambling. They try to maximize the worst case utility they can obtain.

A desirable property for cake-cutting protocols is strategy-proofness [6]. A protocol is strategy-proof if there is no incentive for any player to lie about his utility function. A protocol defines what to do for each player p_i according to its utility function μ_i. Since μ_i is unknown to any other player, p_i can execute some action that differs from the protocol's definition (by pretending that p_i's utility function is $\mu'_i(\neq \mu_i)$). If p_i obtains more utility by lying about his utility function, the protocol is not strategy-proof. If a protocol is not strategy-proof, each player has to consider what to do and the result might differ from the intended result. If a protocol is strategy-proof, the best policy for each player is simply observing the rule of the protocol. Thus strategy-proofness is very important. As for 'divide-and-choose', the protocol requires Divider to cut the cake in half by using Divider's true utility function. Divider can cut the cake other than in half. However, if Divider does so, Chooser might take the larger portion and Divider might obtain less than half. Thus a risk averse party honestly executes the protocol, and 'divide-and-choose' is strategy-proof.

3 Meta-envy-freeness

This section provides the definition of meta-envy-freeness. We offer two definitions and show that they are equivalent.

Definition 1. *A cake-cutting protocol f is meta-envy-free if for any $(\mu_1, \ldots, \mu_n)$, i, j, and r,*

$$\mu_i(f_i(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n; r)) \geq \mu_i(f_j(\mu_1, \ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_n; r)) \quad (1)$$

This definition considers the following two executions of f. (1) party P_i (whose utility function is μ_i) plays the role of player p_i and party P_j (whose utility function is μ_j) plays the role of player p_j in f. (2) party P_i plays the role of player p_j and party P_j plays the role of player p_i in f, that is, P_i and P_j swap role assignments. If the swap does not increase the utility of the obtained portions, P_i will not want to swap the role assignment, thus the protocol is envy-free as regards the role assignment.

Next we show a stronger definition.

Definition 2. *A cake-cutting protocol f is meta-envy-free if for any $(\mu_1, \ldots, \mu_n)$, permutation $\pi : \{1, \ldots, n\} \to \{1 \ldots, n\}$, i, and r,*

$$\mu_i(f_i(\mu_1, \ldots, \mu_n; r)) = \mu_i(f_{\pi^{-1}(i)}(\mu_{\pi(1)}, \ldots, \mu_{\pi(n)}; r)) \tag{2}$$

This definition allows any permutation of the role assignment, which includes the case where P_i's role is unchanged. In addition, the utility must be unchanged for any permutation.

Theorem 1. *Definition 1 and Definition 2 are equivalent.*

Proof. If the condition of Definition 2 is satisfied, the condition of Definition 1 is obviously satisfied. Thus we prove the opposite direction.

Suppose that f satisfies the condition of Definition 1 and for some $(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n)$, i, j, and r,

$$\mu_i(f_i(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n; r)) > \mu_i(f_j(\mu_1, \ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_n; r)) \tag{3}$$

is satisfied. Then consider another execution of f with $(\mu_1, \ldots, \mu_j, \ldots, , \mu_i, \ldots, \mu_n)$, that is, P_i's utility function is μ_j and P_j's utility function is μ_i. Since the condition of Definition 1 is satisfied, swapping the roles of P_i and P_j does not increase P_j's utility, that is,

$$\mu_i(f_j(\mu_1, \ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_n; r)) \geq \mu_i(f_i(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n; r)) \tag{4}$$

This contradicts Eq. (3). Thus, for any $(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n)$, i, j, and r,

$$\mu_i(f_i(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n; r)) = \mu_i(f_j(\mu_1, \ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_n; r)) \tag{5}$$

is satisfied.

Next we consider a general permutation of the role assignment. Any permutation π can be realized by a sequence in which two elements are swapped. As shown above, P_i's utility is unchanged when the swap involves P_i, thus we discuss P_i's utility when there is a swap between the other parties. Consider two utilities $\mu_i(f_i(\ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_k, \ldots; r))$ and $\mu_i(f_i(\ldots, \mu_i, \ldots, \mu_k, \ldots, \mu_j, \ldots; r))$.

The roles of P_j and P_k can be swapped by the sequence of (S1) swapping P_i and P_j, (S2) swapping P_i (current role is p_j) and P_k, and (S3) swapping P_i (current role is p_k) and P_j (current role is p_i).

For each swap, Eq. (5) must be satisfied. From these equalities, we obtain

$$\begin{aligned}
\mu_i(f_i(\ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_k, \ldots; r)) &= \mu_i(f_j(\ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_k, \ldots; r)) \\
\mu_i(f_j(\ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_k, \ldots; r)) &= \mu_i(f_k(\ldots, \mu_j, \ldots, \mu_k, \ldots, \mu_i, \ldots; r)) \\
\mu_i(f_k(\ldots, \mu_j, \ldots, \mu_k, \ldots, \mu_i, \ldots; r)) &= \mu_i(f_i(\ldots, \mu_i, \ldots, \mu_k, \ldots, \mu_j, \ldots; r)).
\end{aligned}$$

From these equalities, we obtain

$$\mu_i(f_i(\ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_k, \ldots; r)) = \mu_i(f_i(\ldots, \mu_i, \ldots, \mu_k, \ldots, \mu_j, \ldots; r)).$$

Since this equality holds for any single swap, the equality holds for any permutation π. □

Several desirable properties have been defined as shown below [6][13], but these definitions do not take role assignment into consideration.

Simple fair For any i, $\mu_i(f_i(\mu_1,\ldots,\mu_n)) \geq 1/n$.
Strong fair For any i, $\mu_i(f_i(\mu_1,\ldots,\mu_n)) > 1/n$.
Envy-free For any $i,j(i \neq j)$, $\mu_i(f_i(\mu_1,\ldots,\mu_n)) \geq \mu_i(f_j(\mu_1,\ldots,\mu_n))$.
Strong envy-free For any $i,j(i \neq j)$, $\mu_i(f_i(\mu_1,\ldots,\mu_n)) > \mu_i(f_j(\mu_1,\ldots,\mu_n))$.
Super envy-free For any $i,j(i \neq j)$, $\mu_i(f_j(\mu_1,\ldots,\mu_n)) < 1/n$.
Exact For any i,j, $\mu_i(f_j(\mu_1,\ldots,\mu_n)) = 1/n$.
Equitable For any i,j, $\mu_i(f_i(\mu_1,\ldots,\mu_n)) = \mu_j(f_j(\mu_1,\ldots,\mu_n))$.

Simple fair division can be achieved for any number of parties by using the moving-knife protocol [8]. Strong fair division cannot be achieved if every party has an identical utility function μ. Woodall [14] proposed an algorithm for achieving strong fair division provided that there is a portion $X \subset [0,1]$ such that $\mu_1(X) \neq \mu_2(X)$, when $n = 2$. The algorithm for obtaining such a portion X is an open problem. Envy-free division can be achieved for any number of parties [5], however the protocol is very complicated.

As regards strong envy-free cake-cutting, the lower bound of the number of cuts is shown [10]. Super envy-free division can be achieved if utility functions $\mu_1,\ldots,\mu_n$ are linearly independent, however the algorithm for obtaining an actual assignment is not shown[2]. An exact division algorithm has been reported for two players using a moving knife method [1]. Though existence of exact division was proved [11], no algorithm has been shown for $n \geq 3$. An equitable division algorithm between two parties has been described [9]. The case where $n \geq 3$ is an open problem.

As shown above, stronger properties than envy-free such as strong-envy-free, super-envy-free, exact, and equitable are very hard to realize.

A definition, similar to ours, called 'anonymous,' is provided in [12]. A cake-cutting protocol is anonymous if for any $(\mu_1,\ldots,\mu_i,\ldots,\mu_j,\ldots,\mu_n)$, i, and j,

$$f_i(\mu_1,\ldots,\mu_i,\ldots,\mu_j,\ldots,\mu_n) = f_j(\mu_1,\ldots,\mu_j,\ldots,\mu_i,\ldots,\mu_n)$$

holds. This is a severe definition that requires the assigned portion to be identical for any role swapping. In meta-envy-freeness the assigned portions need not be identical but their utilities must be identical for any role swapping. In addition, randomization is not explicitly considered in the definition of anonymity.

Equitability does not imply meta-envy-freeness. There can be an (artificial) protocol that is equitable but not meta-envy-free. Party P_1's utility μ_1 satisfies $\mu_1([0,1/4]) = 0.3$, $\mu_1([1/4,1/2]) = 0.3$, $\mu_1([1/2,3/4]) = 0.2$, and $\mu_1([3/4,1]) = 0.2$. Party P_2's utility μ_2 satisfies $\mu_2([0,1/4]) = 0.2$, $\mu_2([1/4,1/2]) = 0.2$, $\mu_2([1/2,3/4]) = 0.3$, and $\mu_2([3/4,1]) = 0.3$. A protocol f initially assigns $[0,1/4]$ to the first player and $[3/4,1]$ to the second player. The result of $f(\mu_1,\mu_2)$ is $f_1(\mu_1,\mu_2) = [0,1/2]$ and $f_2(\mu_1,\mu_2) = [1/2,1]$ and the utilities are 0.6 for both parties. On the other hand, $f(\mu_2,\mu_1)$ might result in $f_1(\mu_2,\mu_1) = ([0,1/4],$ $[1/2,3/4])$ and $f_2(\mu_2,\mu_1) = ([3/4,1],[1/4,1/2])$, thus the utilities are 0.5 for both parties. Therefore this (artificial) protocol is equitable, but not meta-envy-free,

1: **begin**
2: p_1 cuts into three pieces (so that p_1 considers their sizes are the same)
3: Let X_1, X_2, X_3 be the pieces where X_1 is the largest and X_3 is the smallest for p_2.
4: **if** X_1 is larger than X_2 for p_2 **then**
5: p_2 cuts L from X_1 so that $X_1' = X_1 - L$ is the same as X_2 for p_2.
6: p_3 selects the largest (for p_3) among X_1', X_2, and X_3.
7: **if** X_1' remains **then**
8: **begin**
9: p_2 must select X_1'.
10: Let (p_a, p_b) be (p_3, p_2).
11: **end**
12: **else**
13: **begin**
14: p_2 selects X_2 (the largest for p_2).
15: Let (p_a, p_b) be (p_2, p_3).
16: **end**
17: p_1 obtains the remaining piece.
18: **if** L is not empty **then**
19: p_a cuts L into three pieces (such that p_a considers their sizes are the same) and p_b, p_1, and p_a selects one piece in this order.
20: **end**.

Fig. 1. Three-player envy-free protocol.

since P_1 prefers the first player. On the other hand, the meta-envy-free protocols shown in the next section are not equitable. Note that meta-envy-freeness does not imply envy-freeness. As shown in the introduction, the following holds.

Theorem 2. *The 'divide-and-choose' protocol is not meta-envy-free.*

Next, we consider the envy-free cake-cutting protocol for three players, found independently by Selfridge and Conway (introduced in [6]), and shown in Fig. 1.

Note that without loss of envy-freeness, we assume that when a player cuts L from $X_1 = [x_1, x_2]$, L must be cut as $[x_1, x_3]$ for some x_3.

Theorem 3. *The protocol in Fig. 1 is not meta-envy-free.*

Proof. Let there be three parties P_x, P_y, and P_z whose utility functions are μ_x, μ_y, and μ_z, respectively.

We show that party P_x prefers the role of player p_3 to that of p_2 in this protocol. Let us consider the following two executions:
(Ex1) $(p_1, p_2, p_3) = (P_z, P_y, P_x)$ and (Ex2) $(p_1, p_2, p_3) = (P_z, P_x, P_y)$.

The result of the initial cut by P_z at line 2 is the same in (Ex1) and (Ex2). Let the three pieces be Z_1, Z_2, and Z_3. Without loss of generality, Z's are ordered from the largest to the smallest for P_y. All possible cases are categorized as follows.

(Case 1) P_y does not cut L in (Ex1).

(Case 1-1) P_x cuts L' from some piece Z in (Ex2).
(Case 1-2) P_x does not cut L in (Ex2).
(Case 2) P_y cuts L from Z_1 in (Ex1).
(Case 2-1) P_x also cuts L' from Z_1 in (Ex2).
(Case 2-1-1) L' is larger[3] than L. **(Case 2-1-2)** L' is smaller than L. **(Case 2-1-3)** $L' = L$.
(Case 2-2) P_x cuts L' from another piece Z in (Ex2).
(Case 2-3) P_x does not cut L' in (Ex2).

(Case 1-1) Let the largest piece for P_x be Z_1'. P_x selects Z_1' at line 6 in (Ex1) and obtains utility $\mu_x(Z_1')$. In contrast, at lines 7-16 of (Ex2), P_x obtains a piece whose utility equals $\mu_x(Z_1' - L')$, because there are two pieces with utility $\mu_x(Z_1' - L')$ after cutting L'. At line 19 of (Ex2), P_x obtains a cut of L' whose utility is smaller than $\mu_x(L')$. Thus, the total utility of P_x is smaller than $\mu_x(Z_1')$. Therefore, (Ex1) is better for P_x.

(Case 1-2) There are at least two largest pieces for P_x among Z_1, Z_2, and Z_3. P_x selects the largest piece at line 6 in (Ex1). In contrast, after P_y has selected Z_1 at line 6 in (Ex2), P_x can select one of the largest pieces at lines 7-16. Thus P_x obtains the same utility in (Ex1) and (Ex2).

(Case 2-1-1) At line 6 in (Ex1), the largest piece for P_x is $Z_1 - L$, since L' is larger than L. At line 19, P_x obtains at least $\mu_x(L)/3$. Thus, P_x obtains at least $\mu_x(Z_1) - 2\mu_x(L)/3$ in total. In contrast, P_y selects Z_2, which is larger than $Z_1 - L'$, at line 6 in (Ex2). Thus P_x selects $Z_1 - L'$ at line 9. In addition, P_x obtains at least $\mu_x(L')/3$. P_x obtains at least $\mu_x(Z_1) - 2\mu_x(L')/3$ in total. Thus, (Ex1) is better for risk averse party P_x.

(Case 2-1-2) At line 6 in (Ex1), P_x does not select $Z_1 - L$, since it is not greater than the second largest piece, whose utility is $\mu_x(Z_1 - L')$, for P_x. P_x chooses the piece and obtains $\mu_x(Z_1 - L')$. In addition, at line 19, P_x obtains $\mu_x(L)/3$ because P_x cuts L. P_x obtains $\mu_x(Z_1) - \mu_x(L') + \mu_x(L)/3$ in total. In contrast, at line 6 in (Ex2), P_y selects $Z_1 - L'$, which is the largest for P_y. Thus P_x selects Z_2 or Z_3 whose utility is $\mu_x(Z_1 - L')$. P_x then obtains $\mu_x(L')/3$ at line 19 because P_x cuts L'. P_x obtains $\mu_x(Z_1) - 2\mu_x(L')/3$ in total, which is smaller than that in (Ex1), since L' is smaller than L.

(Case 2-1-3) In both (Ex1) and (Ex2), P_x obtains a piece whose utility is $\mu_x(Z_1 - L)$. The only difference is who cuts L. As shown in the proof of 'divide-and-choose', being Chooser is the better than being Divider at line 19. In (Ex1), P_x can select $Z_1 - L$ and become Chooser. In (Ex2), if P_y selects $Z_1 - L$, P_x must become Divider. Thus (Ex1) is better than (Ex2).

(Case 2-2) In (Ex1), P_x selects the largest piece, which is not $Z_1 - L$, at line 6 and obtains $\mu_x(Z)$. At line 19, P_x obtains at least $\mu_x(L)/3$. In (Ex2), P_y selects Z_1 not $Z - L'$ at line 6. Thus P_x obtains $\mu_x(Z) - \mu_x(L')$ at line 9. At line 19, P_x obtains less than $\mu_x(L')$. P_x obtains less than $\mu_x(Z)$ in total, which is worse than in (Ex1).

[3] To compare the sizes of L and L', they must be cut in a canonical way. Thus the additional rule for cutting L is necessary.

begin
$P_i(i = 1, 2)$ simultaneously declare c_i that satisfies $\mu_i([0, c_i]) = 1/2$.
if $c_1 = c_2$ **then**
    Cut at c_1, coin-flip and decide which party obtains $[0, c_1]$ or $[c_1, 1]$.
else
    Cut as $[0, (c_1 + c_2)/2]$, $[(c_1 + c_2)/2, 1]$. P_i obtains the piece which contains c_i.
end.

Fig. 2. Two-party meta-envy-free protocol.

(Case 2-3) There are at least two largest pieces among Z_1, Z_2, and Z_3 for P_x. Let $\mu_x(Z)$ be the utility of the largest piece. In (Ex1), P_x can obtain $\mu_x(Z)$ at line 6. In addition, P_x obtains $\mu_x(L)/3$ at line 19. In contrast, in (Ex2), P_x obtains $\mu_x(Z)$. Thus (Ex1) is better than (Ex2) for P_x. □

4 Meta-envy-free protocols for two and three parties

This section shows meta-envy-free cake-cutting protocols for two and three parties. Note that the word 'party' is used in the descriptions in this section because every player's role is identical. When there are two parties, the protocol proposed in [4], shown in Fig. 2, is meta-envy-free.

The simultaneous declaration of values by multiple parties can be realized in several ways, (1) Trusted third party (TTP): P_i sends c_i to the TTP. After the TTP receives all the values, he broadcasts them to all parties. (2) Commitment scheme [7]: P_i first sends $com_i(c_i)$, which is a commitment of c_i. The other parties cannot obtain the value c_i from $com_i(c_i)$. After P_i has obtained the other parties' committed values, P_i opens its commitment (that is, sends c_i and a proof that $com_i(c_i)$ is really made by c_i). P_i cannot provide a false proof that $com_i(c_i)$ is made by $c'_i(\neq c_i)$.

Theorem 4. *The protocol in Figure 2 is meta-envy-free, envy-free, and strategy-proof.*

Proof. The cut point depends only on the parties' declared values. The result is independent of the role assignment or the order of declaration. Thus the protocol is meta-envy-free. The protocol is envy-free because both parties obtain at least half evaluated by their utility functions. The protocol is strategy-proof since if P_1 declares false cut point c'_1, P_2's true cut point c_2 might satisfy $c_2 = c'_1$ and P_1 might obtain less than half by coin-flipping. Thus, risk adverse parties obey the rule and declare their true cut points. □

There is another method for assigning portions when the declared values differ. Without loss of generality, assume that $c_1 < c_2$. Assign $[0, c_1]$ to P_1, $[c_2, 1]$ to P_2, and execute the same protocol again for the remaining piece $[c_1, c_2]$. Although this method might need an infinite number of declaration rounds and each party might obtain multiple fragments of the cake, the assignment guarantees $\mu_1(f_1(\mu_1, \mu_2)) = \mu_2(f_2(\mu_1, \mu_2))$.

Avoiding multiple declaration is possible if P_i simultaneously declares the utility density function u_i. Utility density function u_i satisfies $u_i(z) > 0$ for $[0, 1]$ and $\int_0^1 u_i(z)dz = 1$.

When the remaining piece is $[l^{(j)}, r^{(j)}]$ at round j ($l^{(1)} = 0$ and $r^{(1)} = 1$), The cut point declaration at round j is the point $c_i^{(j)}$ that satisfies

$$\int_{l^{(j)}}^{c_i^{(j)}} u_i(z)dz = \int_{c_i^{(j)}}^{r^{(j)}} u_i(z)dz. \tag{6}$$

If $c_1^{(j)} \neq c_2^{(j)}$, let $l^{(j+1)} = min(c_1^{(j)}, c_2^{(j)})$, $r^{(j+1)} = max(c_1^{(j)}, c_2^{(j)})$, and execute next round.

A protocol that uses a utility density function is also proposed in [3]. Here the cake is cut into two pieces. However, the protocol has the disadvantage that it is not strategic-proof, that is, a party can obtain more utility by declaring a false utility density function.

Next we show a protocol for a three-party case in Fig. 3. The protocol is outlined as follows. First, each party P_i simultaneously declares cut point l_i such that $[0, l_i]$ is 1/3 for P_i. Cases are switched according to how many of l_1, l_2, and l_3 are the same. If at least two of them are the same, the parties with the same value simultaneously declare cut point r_i such that $[r_i, 1]$ is 1/3 for P_i. Envy-free assignment can be easily obtained using the declared values when at least two of l_1, l_2, and l_3 are the same. The remaining case is when l_1, l_2, and l_3 are all different (without loss of generality, assume that $l_1 < l_2 < l_3$). Here, we execute the three-player envy-free protocol in Fig. 1 with the role assignment (p_1, p_2, p_3)=(P_3, P_2, P_1), that is, P_3 plays the role of p_1 in the protocol, and so on, with the restriction that P_3 must use l_3 as a cut. Note that this role assignment is executed by the declared value l_i, thus the protocol is meta-envy-free.

Although (p_1, p_2, p_3)=(P_3, P_2, P_1) is not a unique acceptable role assignment, there are unacceptable role assignments. Let us consider the following role assignment: (p_1, p_2, p_3)=(P_2, P_1, P_3), namely, the cake is cut at l_2, r_2 and P_1 cuts L from the largest piece. Suppose that $[0, l_2]$ is the largest for P_1. P_1 cuts L from $[0, l_2]$. In this case, $[0, l_2]$ is less than 1/3 for P_3 because $l_3 > l_2$. After P_1 cuts L from $[0, l_2]$, P_3 will never select $[0, l_2] - L$ as the largest piece for P_3. P_1 knows this fact from $l_3 > l_2$, thus P_1 will not cut L honestly from $[0, l_2]$. In this case, P_3 will select some piece other than $[0, l_2]$. P_1 then selects $[0, l_2]$ and obtains more utility than when honestly cutting L. Thus, the protocol is not strategy-proof.

Theorem 5. *The protocol in Fig. 3 is meta-envy-free, envy-free, and strategy-proof.*

Proof. The protocol is meta-envy-free because the role is decided solely by the declared values. Next let us consider envy-freeness. All possible cases are categorized as follows. **(Case 1)** $l_1 = l_2 = l_3$ and $r_1 = r_2 = r_3$. **(Case 2)** $l_1 = l_2 = l_3$, $r_1 = r_2$, and $r_3 > r_1$. **(Case 3)** $l_1 = l_2 = l_3$, $r_1 = r_2$, and $r_1 > r_3$. **(Case 4)** $l_1 = l_2 = l_3$ and $r_1 < r_2 < r_3$. **(Case 5)** $l_1 = l_2 (\neq l_3)$ and $r_1 = r_2$. **(Case 6)** $l_1 = l_2 (\neq l_3)$ and $r_1 < r_2$. **(Case 7)** $l_1 < l_2 < l_3$.

Each party P_i simultaneously declares l_i such that $[0, l_i]$ is 1/3 for P_i.
if $l_1 = l_2 = l_3$ **then**
  begin
  Each party P_i simultaneously declares r_i such that $[r_i, 1]$ is 1/3 for P_i.
  if $r_1 = r_2 = r_3$ **then**
    Cut at l_1 and r_1. Coin-flip and assign $[0, l_1], [l_1, r_1], [r_1, 1]$ to the parties.
  else
    if two of r_1, r_2, r_3 are the same **then**
      begin /* Without loss of generality, let $r_1 = r_2$. */
      Cut at l_1 and r_1.
      if $r_3 > r_1$ **then** Assign $[r_1, 1]$ to P_3.
      else /* $r_3 < r_1$ */
        Assign $[l_1, r_1]$ to P_3.
      Coin-flip and assign the remaining two pieces to P_1 and P_2.
      end
    else /* Without loss of generality, let $r_1 < r_2 < r_3$. */
      Cut at l_1 and r_2. Assign $[0, l_1]$ to P_2, $[l_1, r_2]$ to P_1, and $[r_2, 1]$ to P_3.
  end /* end of case $l_1 = l_2 = l_3$. */
else
  if two among l_1, l_2, and l_3 are the same **then**
    begin /* Without loss of generality, let $l_1 = l_2$. */
    P_1 and P_2 simultaneously declare r_i such that $[r_i, 1]$ is 1/3 for P_i.
    if $r_1 = r_2$ **then**
      begin
      Cut at l_1 and r_1. P_3 selects one piece among $[0, l_1], [l_1, r_1]$, and $[r_1, 1]$.
      Coin-flip and assign the remaining two pieces to P_1 and P_2.
      end
    else /* $r_1 \neq r_2$. */
      begin /* Without loss of generality, let $r_1 < r_2$. */
      Cut at l_1, r_1, r_2. $L \leftarrow [r_1, r_2]$. P_3 selects one among $[0, l_1]$, $[l_1, r_1]$, $[r_2, 1]$.
      if P_3 selects $[0, l_1]$ **then**
        begin
        Assign $[l_1, r_1]$ and $[r_2, 1]$ to P_1 and P_2, respectively.
        P_3 cuts L into three pieces. P_1, P_2, P_3 selects one in this order.
        end
      else
        if P_3 selects $[l_1, r_1]$ **then**
          begin
          Assign $[0, l_1]$ and $[r_2, 1]$ to P_1 and P_2, respectively.
          P_3 cuts L into three pieces. P_2, P_1, P_3 selects one in this order.
          end
        else /* P_3 selects $[r_2, 1]$. */
          begin
          Assign $[l_1, r_1]$ and $[0, l_1]$ to P_1 and P_2, respectively.
          P_3 cuts L into three pieces. P_1, P_2, P_3 selects one in this order.
          end
      end
    end
  else /* l_1, l_2 and l_3 are different. Without loss of generality, let $l_1 < l_2 < l_3$. */
    Execute Fig. 1 with (p_1, p_2, p_3)=(P_3, P_2, P_1) and l_3 is used as a cut.

Fig. 3. Three party meta-envy-free protocol.

(Case 1) Since the utilities of $[0.l_1]$, $[l_1, r_1]$, and $[r_1, 1]$ are 1/3 for all parties, no assignment causes envy.

(Case 2) The utilities of $[0, l_1]$, $[l_1, r_1]$, and $[r_1, 1]$ are the same for P_1 and P_2. $[r_1, 1]$ is the largest for P_3 since $r_3 > r_1$ and $l_3 = l_1$. Thus assigning $[r_1, 1]$ does not cause any party envy. Assigning the remaining pieces to P_1 and P_2 can be arbitrary.

(Case 3) The utilities of $[0, l_1]$, $[l_1, r_1]$, and $[r_1, 1]$ are the same for P_1 and P_2. $[l_1, r_1]$ is the largest for P_3 since $r_3 < r_1$ and $l_3 = l_1$. Thus assigning $[l_1, r_1]$ does not cause any party envy. Assigning the remaining pieces to P_1 and P_2 can be arbitrary.

(Case 4) Among $[0, l_1]$, $[l_1, r_2]$, and $[r_2, 1]$, $[l_1, r_2]$ is the largest for P_1 since $r_1 < r_2$. $[r_2, 1]$ is the largest for P_3 since $r_2 < r_3$ and $l_1 = l_3$. P_2 feels the three pieces are the same size, thus assigning $[0, l_1]$ to P_2 does not cause envy.

(Case 5) The utilities of $[0, l_1]$, $[l_1, r_1]$, and $[r_1, 1]$ are the same for P_1 and P_2. Thus, P_3's selection from these pieces does not cause envy.

(Case 6) The utilities of $[0, l_1]$, $[l_1, r_1]$, and $[r_1, 1]$ are the same for P_1. The utilities of $[0, l_1]$, $[l_1, r_2]$, and $[r_2, 1]$ are the same for P_2. Cutting the cake into four pieces, $[0, l_1]$, $[l_1, r_1]$, $[r_2, 1]$, and $L = [r_1, r_2]$ is exactly the same situation as during three-player envy-free cutting (Case 6-1) P_1 executes the initial cut ($[0, l_1]$, $[l_1, r_1]$, and $[r_1, 1]$) and P_2 cuts L from the largest piece $[r_1, 1]$ so that its size becomes that of the second largest piece $[0, l_1]$ and (Case 6-2) P_2 executes the initial cut ($[0, l_1]$, $[l_1, r_2]$, and $[r_2, 1]$) and P_1 cuts L from the largest piece $[l_1, r_2]$ so that its size becomes that of the second largest piece $[0, l_1]$.

When P_3 selects $[0, l_1]$ from the three pieces, we can regard this as (Case 6-2) being executed. With the three-player envy-free protocol, next P_1 must select $[l_1, r_1]$ and P_2 selects the remaining piece $[r_2, 1]$. P_3 cuts L into three pieces. P_1, P_2, and P_3 each select one piece in this order. Because of the envy-freeness of the three-player protocol, the result is envy-free.

When P_3 selects $[l_1, r_1]$ from the three pieces, we can regard this as (Case 6-1) being executed. With the three-player envy-free protocol, next P_2 must select $[r_2, 1]$ and P_1 selects the remaining piece $[0, l_1]$. P_3 cuts L into three pieces. P_2, P_1, and P_3 each select one piece in this order. Because of the envy-freeness of the three-player protocol, the result is envy-free.

Lastly, when P_3 selects $[r_2, 1]$ from the three pieces, we can regard this as (Case 6-2) being executed. With the three-player envy-free protocol, next P_1 must select $[l_1, r_1]$ and P_2 selects the remaining piece $[0, l_1]$. P_3 cuts L into three pieces. P_1, P_2, and P_3 each select one piece in this order. Because of the envy-freeness of the three-player protocol, the result is envy-free.

(Case 7) Since the players execute the three-player envy-free protocol, the result is envy-free.

Lastly, let us discuss strategy-proofness. When P_i declares a cut point l_i (or r_i) simultaneously with some other process P_j, declaring a false value l'_i (or r'_i) might result in a worse utility, since P_j's true value l_j (or r_j) might satisfy $l_j = l'_i$ (or $r_j = r'_i$) and P_i might obtain a smaller piece by coin-flipping.

When P_3 selects one piece at line 30, a false selection results in a worse utility for P_3. Note that this selection does not affect who will be the divider of L.

Next, consider the execution of the three-player envy-free protocol with extra information $l_1 < l_2 < l_3$. When P_3 cuts as $[0, l_3], [l_3, r_3]$, and $[r_3, 1]$, a false cut r'_3 might result in P_3 obtaining less than 1/3. When P_2 cuts L from the largest piece, information of l_1 does not help P_2 to obtain greater utility with false cut L' even if P_2 cuts L from $[0, l_3]$. The reason is as follows. For any true cut L, either of the two cases can happen according to P_1's utility (that is unknown to P_2): (1) $[l_3, r_3]$ or $[r_3, 1]$ is the largest for P_1 or (2) $[0, l_3] - L$ is the largest for P_1. Thus, if P_2 cuts L' that is smaller than L, P_1 might select $[0, l_3] - L'$ and P_2's utility might become worse. If P_2 cuts L'' that is larger than L, P_1 might select $[l_3, r_3]$ and P_2's utility might become worse. With respect to cutting L into three pieces, the strategy-proofness is exactly the same as that of the original three-player envy-free protocol. Therefore, the protocol is strategy-proof. □

Acknowledgment We thank Dr. Hiro Ito for his valuable comments.

References

1. Austin, A.K.: "Sharing a Cake," Mathematical Gazette, Vol. 66, No. 437, pp. 212-215(1982).
2. Barbanel, J.B.: "Super Envy-Free Cake Division and Independence of Measures," J. of Mathematical Analysis and Applications, Vol. 197, No. 1, pp. 54-60 (1996).
3. Brams, S. J., Jones, M. A., and Klamler, C.: "Better Ways to Cut a Cake," Notices of the AMS, Vol. 53, No. 11, pp. 1314-1321 (2006).
4. Brams, S. J., Jones, M. A., and Klamler, C.: "Divide-and-Conquer: A Proportional, Minimal-Envy Cake-Cutting Procedure," Proc. of Dagstuhl Seminar (2007).
5. Brams, S.J. and Taylor, A.D.: "An Envy-Free Cake Division Protocol," American Mathematical Monthly, Vol. 102, No. 1, pp. 9-18 (1995).
6. Brams, S. J. and Taylor, A. D.: "Fair Division: From Cake-Cutting to Dispute Resolution," Cambridge University Press (1996).
7. Brassard, G., Chaum, D., and Crépeau, C.: "Minimum Disclosure Proofs of Knowledge," Journal of Computer and System Sciences, Vol. 37, No. 2, pp. 156-189 (1988).
8. Dubins, L. E. and Spanier, E. H.: "How to Cut a Cake Fairly," American Mathematical Monthly, Vol. 85, No. 1, pp. 1-17 (1961).
9. Jones, M. A.: "Equitable, Envy-free, and Efficient Cake Cutting for Two People and its Application to Divisible Goods." Mathematics Magazine Vol. 75, No. 4, pp. 275-283 (2002).
10. Magdon-Ismail, M., Busch, C., and Krishnamoorthy, M.S.: " Cake Cutting is Not a Piece of Cake," Proc. of the 20th STACS, LNCS 2607, pp. 596-607 (2003).
11. Neyman, J.: "Un theoreme d'existence," C. R. Acad. Sci. Paris Vol. 222 pp. 843-845 (1946).
12. Nicolò, A. and Yu, Y.: "Strategic Divide and Choose," Games and Economic Behavior, Vol. 64, No. 1, pp. 268-289 (2008).
13. Robertson, J. and Webb, W.: "Cake-Cutting Algorithms: Be Fair If You Can," A K Peters (1998).
14. Woodall, D.R.: "A Note on the Cake-Division Problem," J. of Combbinatorial Theory, A, Vol. 42, No. 2, pp. 300-301 (1986).