Journal of Information Processing Vol.20 No.3 686–693 (July 2012)

[DOI: 10.2197/ipsjjip.20.686]

Regular Paper

Meta-envy-free Cake-cutting and Pie-cutting Protocols

Yoshifumi Manabe[1,a)] Tatsuaki Okamoto[2,b)]

Received: August 23, 2011, Accepted: March 2, 2012

Abstract: This paper discusses cake-cutting protocols when the cake is a heterogeneous good, represented by an interval on the real line. We propose a new desirable property, the meta-envy-freeness of cake-cutting, which has not been formally considered before. Meta-envy-free means there is no envy on role assignments, that is, no party wants to exchange his/her role in the protocol with the one of any other party. If there is an envy on role assignments, the protocol cannot be actually executed because there is no settlement on which party plays which role in the protocol. A similar definition, envy-freeness, is widely discussed. Envy-free means that no player wants to exchange his/her part of the cake with that of any other player's. 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 to formalize this kind of envy. We propose that simultaneously achieving meta-envy-free and envy-free is desirable in cake-cutting. 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 cake-cutting protocols for two and three party cases that simultaneously achieves envy-free and meta-envy-free. Last, we show meta-envy-free pie-cutting protocols.

Keywords: game theory, cake-cutting, pie-cutting, envy-free, meta-envy-free

1. Introduction

Cake-cutting is an old problem in game theory [9], [16]. 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 $[0, 1]$ on 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. The 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 with that of another player's. The Divider must cut the cake into two equal size pieces (using the Divider's utility function), otherwise the Chooser might take the larger piece and the Divider will obtain less than half. Since the Divider cuts the cake into equal size pieces, she never envies the Chooser regardless of which piece the Chooser selects. The Chooser never envies the 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 the Divider and the Chooser. The Chooser is the better choice, as mentioned in several papers [5], [13]. If the utility functions of Alice and Bob are the same, the Divider and the 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 the 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 the 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, the Chooser is never worse off than the Divider, and the Chooser is better than the Divider if their utility functions differ. If both Alice and Bob know this fact, they both want to be the Chooser. Therefore, they must employ a method such as coin-flipping to decide who will be the Divider. If Alice is assigned the role of the Divider, she definitely *envies* Bob who is the Chooser. There is an envy on the assignment of roles in this protocol. Envy-freeness considers the assignment of cake, thus we need a new definition which deals the envy of roles in cake-cutting protocols. Although this type of envy is known, it has not been formally defined. We propose a new desirable property, the meta-envy-freeness of cake-cutting, which has not been formally considered before. Meta-envy-free means there is no envy on role assignments, that is, no party wants to exchange his/her role in the protocol with the one of any other party. If there is an envy on role assignments, the protocol cannot be actually executed because there is no settlement on which party plays which role in the protocol. Thus, we propose that simultaneously achieving meta-envy-free and envy-free is desirable in cake-cutting. This paper then proposes new protocols that simultaneously achieve meta-envy-free and envy-free for the two-party case and the three-party case.

Some readers might think that coin-flipping will result in a fair role assignment between Alice and Bob, and so it is not a prob-

[1] NTT Communication Science Laboratories, Kyoto 619–0237, Japan
[2] NTT Secure Platform Laboratories, Musasino, Tokyo 180–8585, Japan
[a)] manabe.yoshifumi@lab.ntt.co.jp
[b)] okamoto.tatsuaki@lab.ntt.co.jp

lem. If this supposition is accepted, the following protocol must be accepted as an envy-free protocol: 'Flip a coin and the winner takes the whole cake and the loser gets nothing.' This protocol is obviously unacceptable if we want to eliminate envy on the obtained portion of the cake. If we want to eliminate envy on role assignment, we must not accept such an envy role assignment using fair coin-flipping.

Previous studies defined stronger properties for the obtained portion such as strong envy-free, super envy-free, exact, and equitable [9], [16]. These properties, defined in Section 3, are hard to realize and do not directly consider this type of envy. We can obtain a three-party meta-envy-free and 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 can be used for such a case, but can only be allowed if its result causes no envy.

As an extension of the cake-cutting problem, a pie-cutting problem has been considered [12]. When the endpoints of a cake is connected to form a circle, it becomes a pie. All cuts are made between the center and a point on the circumference, so that each cut runs along a radius of the disk. We show a meta-envy-free pie-cutting protocol exists if a meta-envy-free cake-cutting protocol exists for any number of parties.

2. Preliminaries

Throughout this paper, the cake is a heterogeneous good that is represented by an interval $[0, 1]$ on the real line. Each party P_i has a utility function μ_i that has the following three properties, which is the same as the definition of a measure. (1) For any $X \subseteq [0, 1]$ whose size is non-zero, $\mu_i(X) > 0$. (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 (or truthfulness) [9]. 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 to simply observe the rule of the protocol. Thus strategy-proofness is very important. As for 'divide-and-choose', the protocol requires the Divider to cut the cake in half by using the Divider's true utility function. The Divider can cut the cake other than in half. However, if the Divider does so, the Chooser might take the larger portion and the 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,*

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

From the symmetry of Definition 1, the following lemma is obviously derived.

Lemma 1. *If a cake-cutting protocol is meta-envy-free, then for any $(\mu_1, \ldots, \mu_n)$, i, j, and r,*

$$\mu_i(f_i(\mu_1, \ldots, \mu_i, \ldots, \mu_j, \ldots, \mu_n; r)) = \mu_i(f_j(\mu_1, \ldots, \mu_j, \ldots, \mu_i, \ldots, \mu_n; r)) \quad (2)$$

Proof. 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,

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

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,

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

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

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

is satisfied. □

This definition considers the following two executions of f. (A) 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 and random value r is used. (B) Party P_i plays the role of player p_j and party P_j plays the role of player p_i in f with the same random value r, that is, P_i and P_j swap role assignments.

The intuitive explanation of this definition is as follows. After f is executed with random value r, P_i thinks that the role of player p_j was better than p_i in f with the current random value r (that is, p_j obtained extra benefit from f with current random value r). Then P_i can execute f again with the same random value r when P_i plays the role of p_j and P_j plays the role of p_i. If P_i cannot obtain more utility with the latter execution, P_i does not want to exchange the roles of f, that is, P_i has no envy in terms of role assignment.

For the example of 'Divide-and-Choose,' assume that the role of the Divider/Chooser is decided by coin-flipping and P_2 becomes the Chooser when the random value is r_0. P_1 swaps roles with P_2, uses the same r_0, and obtains more utility by becoming the Chooser. Such a protocol is not meta-envy-free according to the definition.

Let us consider another example. There are two pieces of the cake, X_1, X_2 that satisfy $\mu_1(X_1) = \mu_2(X_1) = \mu_1(X_2) = \mu_2(X_2)$. Coin-flipping is used to assign one piece of X_1, X_2 to each of p_1, p_2. Now P_1 plays the role of p_1 and P_2 plays the role of p_2. Assume that X_1 is assigned to p_1 and X_2 is assigned to p_2 when the random value is r_0. In this case, swapping roles with P_2 and using the same random value r_0 results in assigning X_2 to P_1, but this does not change the utility of P_1. Thus P_1 does not want to swap roles in this example.

Though it might be natural to consider distribution of obtained utilities for randomized algorithms, we do not discuss distribution in our definition. Consideration of the distribution hides the effect of unfair role assignments. For the above 'Divide-and-Choose with the the Divider/Chooser role assignment by coin-flipping' protocol, each player becomes the Divider with probability 1/2, thus both players' distributions are the same. However, the protocol has an envy on role assignment for each random value r. Thus we do not consider the distribution according to the definition of meta-envy-free.

Note that meta-envy-freeness is independent of envy-freeness. There can be meta-envy-free protocols that are not envy-free. Let us consider the following artificial protocol f. Protocol f assigns the whole cake to the party whose utility of $[0, 0.1]$ is the largest among the parties. The result does not change even if some parties exchange their roles in the protocol, thus f is meta-envy-free. The party whose utility of $[0, 0.1]$ is not the largest envies the party who obtains the whole cake. We propose that simultaneously achieving meta-envy-free and envy-free is necessary in cake-cutting protocols.

Next we show a stronger definition of meta-envy-freeness.

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,*

$$\mu_i(f_i(\mu_1, \ldots, \mu_n; r)) = \mu_i(f_{\pi^{-1}(i)}(\mu_{\pi(1)}, \ldots, \mu_{\pi(n)}; r)) \qquad (6)$$

This definition allows any permutation of the role assignment, which includes the case where P_i's role is unchanged.

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.

Any permutation π can be realized by a sequence in which two elements are swapped. From Lemma 1, 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. (2) must be satisfied. From these equalities, we obtain

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

From these equalities, we obtain

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

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

Several desirable properties have been defined as shown below [9], [16], 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 players by using the moving-knife protocol [11]. Strong fair division cannot be achieved if every player has an identical utility function μ. Woodall [18] 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 players [8], however the protocol is very complicated.

Regarding strong envy-free cake-cutting, the lower bound of the number of cuts has already been shown [14]. Super envy-free division can be achieved if the utility functions $\mu_1, \ldots, \mu_n$ are linearly independent. However the algorithm for obtaining an actual assignment has not been shown [2]. An exact division algorithm has been reported for two players using a moving knife method [1]. An equitable division algorithm between two players has been also described [13]. The case where $n \geq 3$ remains 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 Ref. [15]. 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. For $n = 2$, an anonymous single-cut cake-cutting is obtained [15]. 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, since P_1 prefers the first player. On the other hand, the meta-envy-free protocols shown in the next section are not equitable.

As shown in the introduction, the following holds.

Observation 1. *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 Ref. [9]), 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.

Instead of showing that the protocol in Fig. 1 is not meta-envy-free, we show a stronger statement that any party prefers the role of player p_3 to that of p_2 in this protocol. The statement shows that this protocol has a serious envy on role assignment.

Theorem 2. *Any party prefers the role of player p_3 to that of p_2 in the protocol of Fig. 1.*

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 P_y prefers the role of p_3 to that of p_2.

Let us consider the following two executions:

(Ex1) $(p_1, p_2, p_3) = (P_z, P_y, P_x)$,

(Ex2) $(p_1, p_2, p_3) = (P_z, P_x, P_y)$.

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 strictly larger than X_2 for p_2 **then**
5: $\quad$ p_2 cuts L from X_1 so that $X'_1 = X_1 - L$ is the same as X_2 for p_2.
6: **else**
7: $\quad$ /* Do nothing. Let L be empty and $X'_1 = X_1$. */
8: p_3 selects the largest (for p_3) among X'_1, X_2, and X_3.
9: **if** X'_1 remains **then**
10: $\quad$ **begin**
11: $\quad$ p_2 must select X'_1.
12: $\quad$ Let (p_a, p_b) be (p_3, p_2).
13: $\quad$ **end**
14: **else**
15: $\quad$ **begin**
16: $\quad$ p_2 selects X_2 (the largest for p_2 among X_2 and X_3).
17: $\quad$ Let (p_a, p_b) be (p_2, p_3).
18: $\quad$ **end**
19: p_1 obtains the remaining piece among X_2 and X_3.
20: **if** L is not empty **then**
21: $\quad$ 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.
22: **end**.

Fig. 1 Three-player envy-free protocol.

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, the 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 [*1] 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 8 during (Ex1) and obtains utility $\mu_x(Z'_1)$. In contrast, at lines 9–18 during (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 21 during (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 8 during (Ex1). In contrast, after P_y has selected Z_1 at line 8 during (Ex2), P_x can select one of the largest pieces at lines 9–18. Thus P_x obtains the same utility in (Ex1) and (Ex2).

(Case 2-1-1) At line 8 during (Ex1), the largest piece for P_x is $Z_1 - L$, since L' is larger than L. At line 21, 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 8 dur-

[*1] 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.

ing (Ex2). Thus P_x selects $Z_1 - L'$ at line 11. 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 8 during (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 21, 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 8 during (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 21 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 the Chooser is the better than being the Divider at line 21. In (Ex1), P_x can select $Z_1 - L$ and become the Chooser. In (Ex2), if P_y selects $Z_1 - L$, P_x must become the 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 8 and obtains $\mu_x(Z)$. At line 21, P_x obtains at least $\mu_x(L)/3$. In (Ex2), P_y selects Z_1 not $Z - L'$ at line 8. Thus P_x obtains $\mu_x(Z) - \mu_x(L')$ at line 11. At line 21, P_x obtains less than $\mu_x(L')$. P_x obtains less than $\mu_x(Z)$ in total, which is worse than in (Ex1).

(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 8. In addition, P_x obtains $\mu_x(L)/3$ at line 21. In contrast, in (Ex2), P_x obtains $\mu_x(Z)$. Thus (Ex1) is better than (Ex2) for P_x. □

Envy-free protocol for any number of players is shown in Ref. [8]. The outline of their protocol is shown in **Fig. 2**. We denote that "p_i has IA (irrevocable advantage) over p_j" when p_i thinks p_i gets a larger piece than p_j even if p_j obtains all of the rest of the cake L.

```
1: begin
2: L ← [0, 1].
3: Let N be the least common multiple of {2, 3, ..., n}.
4: p_1 cuts L into N pieces of the same size.
5: The players are divided into two groups A and D.
6: A: The player feels that the values of all N pieces are the same.
7: D: The player feels some of the values are not the same.
8: if a pair(p_i, p_j) exists such that p_i ∈ A, p_j ∈ D and p_j does not have IA
   over p_i then
9:     begin
10:    Execute IA-Subgame(L, p_i, p_j).
11:    L ← L'
12:    Goto line 4
13:    end
14: else /* D = ∅ or every p_j ∈ D has IA over every p_i ∈ A. */
15:    N pieces are divided by the members of A (Each member of A gets
   the same number of pieces).
16: end.
17:
18: Procedure IA-Subgame(L, p_i, p_j)
19: /* Assign some part of L envy-free among all players. */
20: /* L' ← the remaining cake after the assignment. */
21: /* After the assignment, p_i has IA over p_j and p_j has IA over p_i. */
```

Fig. 2 Envy-free protocol for any number of players.

Since the protocol is very complicated and the detail of the protocol is unnecessary to show meta-envy, we do not state the detail of subroutine "IA-Subgame." IA-Subgame increases the pair of players that have IA over each other. Thus, after a finite number of rounds, any player has IA over any other player. Thus eventually, the "else" condition at line 14 is satisfied and the algorithm terminates.

Theorem 3. *The protocol in Fig. 2 is not meta-envy-free.*

Proof. Let us consider the case $n = 4$. The cases when $n > 4$ can be similarly considered. In this case, $N = 12$. Let us assume P_i's utility function μ_i is as follows. Utility $\mu_i (i = 1, 2, 3)$ are uniform, that is, for any $[a, b]$, $\mu_i([a, b]) = b - a$. Utility μ_4 satisfies the following. $\mu_4([j/12, (j+1)/12]) = 1/12 (j = 0, \ldots, 9)$, $\mu_4([10/12, 21/24]) = 1/24 - \epsilon$, $\mu_4([21/24, 11/12]) = 1/24 + \epsilon$, $\mu_4([11/12, 23/24]) = 1/24 - \epsilon$, and $\mu_4([23/24, 1]) = 1/24 + \epsilon$ (thus, $\mu_4([10/12, 11/12]) = \mu_4([11/12, 1]) = 1/12$).

Consider the following two executions, (A) $(P_1, P_2, P_3, P_4) = (p_1, p_2, p_3, p_4)$ and (B) $(P_1, P_2, P_3, P_4) = (p_2, p_1, p_3, p_4)$.

The execution of (A) is as follows. Party P_1 can arbitrarily cut the cake into $N = 12$ pieces of the same size (for P_1). P_1 cuts the cake into $([j/12, (j+1)/12]) (j = 0, \ldots, 11)$. $A = \{p_1, p_2, p_3, p_4\}$. Thus, the procedure ends with the division of the pieces among all players.

The execution of (B) is as follows. Party P_2 can arbitrarily cut the cake into N pieces of the same size (for P_2). P_2 cuts the cake into $([j/12, (j+1)/12]) (j = 0, \ldots, 9)$, $([10/12, 21/24], [11/12, 23/24])$, and $([21/24, 11/12], [23/24, 1])$. Since the utilities of the last two pieces are not $1/12$ for P_4, $A = \{p_1, p_2, p_3\}$ and $D = \{p_4\}$.

Player p_4 does not have irrevocable advantage over p_1 thus IA-Subgame is executed. In the second round, P_2 cuts the rest of the cake L' and the protocol terminates after some number of rounds.

In execution (A), P_3 obtains $1/4$. In execution (B), P_3 might obtain more than $1/4$ by P_3's utility, because the protocol is envy-free.

P_3's obtained utility depends on the role assignment, thus the protocol is not meta-envy-free. □

4. Meta-envy-free Protocols for Two and Three Parties

This section shows meta-envy-free and 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 Ref. [6], shown in **Fig. 3**, 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 [10]: 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

Journal of Information Processing Vol.20 No.3 686–693 (July 2012)

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. 3 Two-party meta-envy-free protocol.

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 Fig. 3 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 respective utility functions. The protocol is strategy-proof since if P_1 declares a 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

$$\int_{l^{(j)}}^{c_i^{(j)}} u_i(z)dz = \int_{c_i^{(j)}}^{r^{(j)}} u_i(z)dz. \tag{7}$$

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 the next round.

A protocol that uses a utility density function is also proposed in Ref. [5]. 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. 4**.

The protocol is outlined as follows. First, each party P_i simultaneously declares the 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

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**
begin
Cut at l_1 and r_1.
Coin-flip and assign $[0, l_1]$, $[l_1, r_1]$, $[r_1, 1]$ to the parties.
end
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 $r_1 < r_2 < r_3$. */
begin
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 /* 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 piece 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 piece 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 piece 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 piece in this order.
end
end /* end of the case $r_1 \neq r_2$. */
end /* end of the case when two among l_1, l_2, and l_3 are the same */
else /* l_i are different. Without loss of generality, let $l_1 < l_2 < l_3$. */
Execute the algorithm in Fig. 1 with (p_1, p_2, p_3)=(P_3, P_2, P_1) and l_3 is used as a cut.

Fig. 4 Three party meta-envy-free protocol.

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. 4 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) $l_1 < l_2 < l_3$.

(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 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, P_1 next 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, P_2 next 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, P_1 next 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 38 of the protocol, 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 $l_1 < l_2 < l_3$. Note that when l_i are all different, declaration of r_i is not executed, thus the extra apparent information in the three-player envy-free protocol is l_1 and l_2.

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 a 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. □

5. Pie-cutting Problem

When the endpoints of a cake is connected to form a circle, it becomes a pie. In pie-cutting, all cuts are made between the center and a point on the circumference, so that each cut runs along a radius of the disk. Several results were shown about pie-cutting protocols that differ from cake-cutting protocols [3], [4], [7], [12], [17]. Pie-cutting is more difficult than

begin
Decide an initial diameter (northward, for example).
$P_i (1 \le i \le n)$ randomly selects degree d_i from $[0, 360)$ and simultaneously declares d_i.
Calculate $D = \sum_{i=1}^{n} d_i \mod 360$.
Cut the pie at degree D from the initial diameter.
/* Consider $[D, D + 360°]$ as [0,1] of a cake */
Execute meta-envy-free cake-cutting protocol for the pie $[D, D + 360°]$.
end.

Fig. 5 Meta-envy-free pie-cutting protocol.

cake-cutting because there is more flexibility in cutting than cake-cutting.

One example of the difference is about the domination of assignments. An assignment is undominated if no other assignment gives each player at least as much value according to his or her measure as he or she had in the original assignment, and no single player has strictly more value. There is a tuple of utility functions (μ_1, μ_2, μ_3) for a pie such that there is no assignment that satisfies envy-free and undominated and each player obtains one piece. On the other hand, for any tuple of utility functions, there is an envy-free and undominated assignment for three-player cake-cutting such that each player obtains one piece [4].

For meta-envy-free pie-cutting, the following theorem shows the existence of protocols.

Theorem 6. *For any number of parties, there is a meta-envy-free pie-cutting protocol if there is a meta-envy-free cake-cutting protocol.*

Figure 5 shows a meta-envy-free pie-cutting protocol using any meta-envy-free cake-cutting protocol. When a cut is made between the center and a point on the circumference, the pie becomes a cake. The cut must be made randomly.

Proof. Consider the case when party P_i wants to set the cut diameter to some specific point C (or any point in some specific set of points). Since the honest party P_j randomly selects d_j, there is no way for P_i to select his value d_i to set $D = C$. Even if P_j does not observe the rule and select some specific value d'_j by P_j's utility function, d'_j is just the same as a random value for P_i since P_j's utility function is unknown to P_i. Thus, D is random for all parties. The procedure of setting D is obviously meta-envy-free. □

6. Conclusion

This paper proposed meta-envy-free cake-cutting and pie-cutting protocols. The remaining problem involves obtaining a meta-envy-free cake-cutting protocol for $n \geq 4$.

Acknowledgments We thank Dr. Hiro Ito and anonymous referees for their valuable comments.

References

[1] Austin, A.: Sharing a Cake, *The Mathematical Gazette*, Vol.66, No.437, pp.212–215 (online), DOI: 10.2307/3616548 (1982).

[2] Barbanel, J.B.: Super Envy-Free Cake Division and Independence of Measures, *Journal of Mathematical Analysis and Applications*, Vol.197, No.1, pp.54–60 (online), DOI: 10.1006/S0022-247X(96)90006-2 (1996).

[3] Barbanel, J.B. and Brams, S.J.: Two-person Pie-Cutting: The Fairest Cuts, *The College Mathematics Journal*, Vol.42, No.1, pp.25–32 (2011).

[4] Barbanel, J.B., Brams, S.J. and Stromquist, W.: Cutting a Pie is not a Piece of Cake, *American Mathematical Monthly*, Vol.116, No.6, pp.496–514 (online), DOI: 10.4169/193009709X470407 (2009).

[5] 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).

[6] Brams, S.J., Jones, M.A. and Klamler, C.: Divide-and-Conquer: A Proportional, Minimal-Envy Cake-Cutting Procedure, *Dagstuhl Seminars* (2007).

[7] Brams, S.J., Jones, M.A. and Klamler, C.: Proportional Pie-Cutting, *International Journal of Game Theory*, Vol.36, pp.353–367 (online), DOI: 10.1007/s00182-007-0108-z (2008).

[8] Brams, S.J. and Taylor, A.D.: An Envy-Free Cake Division Protocol, *The American Mathematical Monthly*, Vol.102, No.1, pp.9–18 (online), DOI: 10.2307/2974850 (1995).

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

[10] Brassard, G., Chaum, D. and Crépeau, C.: Minimum Disclosure Proofs of Knowledge, *J. Comput. Syst. Sci.*, Vol.37, pp.156–189 (online), DOI: 10.1016/0022-0000(88)90005-0 (1988).

[11] Dubins, L.E. and Spanier, E.H.: How to Cut a Cake Fairly, *The American Mathematical Monthly*, Vol.68, No.1, pp.1–17 (online), DOI: 10.2307/2311357 (1961).

[12] Gale, D.: Mathematical Entertainments, *Mathematical Intelligencer*, Vol.15, pp.48–52 (online), DOI: 10.1007/BF03025257 (1993).

[13] 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 (online), DOI: 10.2307/3219163 (2002).

[14] Magdon-ismail, M., Busch, C. and Krishnamoorthy, M.S.: Cake-Cutting Is Not a Piece of Cake, *Symposium on Theoretical Aspects of Computer Science*, pp.596–607 (online), DOI: 10.1007/3-540-36494-3_52 (2003).

[15] Nicolò, A. and Yu, Y.: Strategic Divide and Choose, *Games and Economic Behavior*, Vol.64, pp.268–289 (online), DOI: 10.1016/j.geb.2008.01.006 (2008).

[16] Robertson, J. and Webb, W.: *Cake-Cutting Algorithms: Be Fair If You Can*, Ak Peters Series, A.K. Peters (1998).

[17] Thomson, W.: Children Crying at Birthday Parties. Why?, *Economic Theory*, Vol.31, pp.501–521 (online), DOI: 10.1007/s00199-006-0109-3 (2007).

[18] Woodall, D.R.: A Note on the Cake-Division Problem, *Journal of Combinatorial Theory*, Vol.42, pp.300–301 (online), DOI: 10.1016/0097-3165(86)90101-9 (1986).

Yoshifumi Manabe was born in 1960. He received his B.E., M.E., and Dr.E. degrees from Osaka University, Osaka, Japan, in 1983, 1985, and 1993, respectively. In 1985, he joined Nippon Telegraph and Telephone Corporation. Currently, he is a senior research scientist, supervisor of NTT Communication Science Laboratories. His research interests include distributed algorithms, cryptography, game theory, and graph theory. He has been a guest associate professor of Kyoto University since 2001. He is a member of IEICE, JSIAM, ACM, and IEEE.

Tatsuaki Okamoto received his B.E., M.E., and Dr.E. degrees from the University of Tokyo, Tokyo, Japan, in 1976, 1978, and 1988, respectively. He is a Fellow of NTT Secure Platform Laboratories. He is presently engaged in research on cryptography and information security. Dr. Okamoto is a guest professor of Kyoto University.