Trust Based Efficiency for Cake Cutting Algorithms

Roie Zivan and Ella Segev,
Industrial Engineering and Management department,
Ben Gurion University of the Negev,
Beer-Sheva, Israel
{zivanr,ellasgv}@bgu.ac.il

Abstract. Fair division methods offer guarantees to agents of the proportional size or quality of their share in a division of a resource (cake). These guarantees come with a price. Standard fair division methods (or "cake cutting" algorithms) do not find efficient allocations (not Pareto optimal). The lack of efficiency of these methods makes them less attractive for solving multi-agent resource and task allocation.
Trust can be the foundation on which agents exchange information and enable the exploration of allocations that are beneficial for both sides. On the other hand, the willingness of agents to put themselves in a vulnerable position due to their trust in others, results in the loss of the fairness guarantees.
In this work we extend the study on fair and efficient cake cutting algorithms by proposing a new notion of *trust-based efficiency*, which formulates a relation between the level of trust between agents and the efficiency of the allocation. In addition, we propose a method for finding trust-based efficiency. The proposed method offers a balance between the guarantees that fair division methods offer to agents and the efficiency that can be achieved by exposing themselves to the actions of other agents. When the level of trust is the highest, the allocation produced by the method is globally optimal (social welfare). [1]

1 Introduction

One of the main challenges in multi-agent systems (MAS) is encouraging self-interested agents to cooperate. Fair division methods offer a possible solution to this challenge for resource and task allocation, by offering guarantees to agents of the quality or size of their share, as long as they are cooperative (follow the instructions of the method's protocol). Moreover, these guarantees hold for an agent, even if other agents choose an uncooperative strategy.

The classic problem that is considered in fair division studies is the division of a heterogeneous resource (a cake) for which agents have their private utility/preference function. Agents divide the cake among themselves by performing *cut and choose* operations. The most familiar cut and choose method is dividing a cake between two agents, so that each will consider her share as at least half of the cake (a proportional share). The method requires one of the agents to cut the cake into two pieces which she considers

[1] A preliminary version of this paper was accepted for publication as an extended abstract of two pages in the proceedings of AAMAS 2011. This submission followed a query to the program chair of IJCAI 2011 who gave his approval.

to be equal and the other to choose the piece she prefers. It is obvious that both agents would consider their share to be at least half of the value of the entire cake. However, this method also demonstrates the weakness of fair division methods. The resulting allocation is guaranteed to be fair but might not be efficient (not Pareto optimal). In other words, there can exist a different allocation that is preferred by both (or preferred by one and is equal in the eyes of the other). For example, consider a round cake, half chocolate and half vanilla, and one agent who strictly prefers vanilla while the other strictly prefers chocolate. The agent cutting the cake may cut the cake so that each piece would include an equal amount of vanilla and chocolate. However, both agents would benefit from an allocation in which each agent gets her preferred flavor.

Many applications of resource and task allocation among self-interested agents motivate the study of methods for fair and efficient allocations. Task allocation in an industrial environment is one example where both fairness and efficiency are required. If, in the name of fairness, we allow workers to perform tasks that they are less qualified for than other workers, we lose efficiency and the resulting revenue of the factory is smaller. Such applications motivate the study of methods, which besides fairness guarantees, would offer guarantees on the level of efficiency.

Previous attempts to introduce efficiency into a fair division method offered extensions of *Austin's method* [2, 5]. Austin's method is the only method for finding an *exact* allocation of a cake among two agents, i.e., it finds an allocation where both agents consider the two pieces to be exactly half of the cake [12]. A simple extension to Austin's procedure increases the efficiency of the allocation in an asymmetric manner. One of the agents selects the most beneficial piece for herself such that the other agent considers her share as exactly half of the cake. This method has the following obvious limitations: (1) Only allocations that include up to two cuts of the cake are considered. (2) The method does not consider allocations in which both agents value their share as strictly more than 50%. A method that achieves a similar asymmetric increase in efficiency by allowing one agent to exploit a model she holds of the other agent's preferences, was proposed in [13]. This method has the same limitations as the asymmetric extension of Austin's method with the addition of the dependency on the existence of an accurate model held by one agent of the other agent's preferences.

The possibility of finding solutions to negotiation problems that *expand the pie*, i.e., the sum of the benefit for the negotiating parties exceeds 100%, was acknowledged by social scientists [15]. This acknowledgment triggered studies that investigated the success of different strategies in producing such agreements.

Trust is a concept that has been intensively studied by social scientists and by the multi-agent systems community [10]. The common and accepted definition of trust is the willingness of an agent to put herself in a situation in which she is vulnerable to the actions of another (the party she *trusts*). The relation between trust and efficiency was also acknowledged by multi-agent system studies [10]. In a cake cutting algorithm, it is easy to see how trust can increase the efficiency of the allocation. If the agents would exchange information regarding their preferences, they can reach an agreement in which each agent is allocated the parts she values more. On the other hand, sharing such information can put an agent in a vulnerable position. The other agents can exploit this knowledge to increase their own benefit. Thus, trust can allow agents to find efficient allocations, but at the same time, expose agents to the exploitive actions of others.

When examining realistic applications, situations in which there exists complete trust among self-interested agents are hard to find. It is common, for example, that people in a working environment would trust each other when they are working together on a project and share their ideas. However, rarely would an employee share her bank account details with her peers. Our approach towards trust is that there exists a scale on which the level of trust between agents can be measured and that the efficiency that can be reached by a cake cutting procedure can be incremented according to this level of trust. Notice, an agent may trust others to some extent to do the right thing in terms of global efficiency, even if it may result in her own loss. In other words, the agents do not trust one another to be fair but rather to be efficient.

In this paper we extend the research on fair and efficient cake cutting methods by:

1. Proposing a new notion of *trust-based efficiency*. It is a generalization of the concept of Pareto optimality, which reflects the level of trust between agents.
2. Proposing a method for finding *trust-based efficiency*. The method proposed allows agents to expose themselves with respect to the level of trust and make use of this exposure to increase efficiency while maintaining the guarantees on the fraction of the proportional share that the agents were not willing to risk. When the level of trust is maximal, the allocation found by the method is globally optimal (social welfare) [2].

Previous attempts to combine fairness and efficiency of a general cake considered the division of a cake between two agents (e.g., [13, 8]). This effort follows most studies on fair division, which attempt to solve the challenges such as proportionality, envy freeness, exact division, first for two agents and later propose a generalization to the case of n agents if possible [5, 12]. We follow this trend by formalizing the problem for the general case of n agents, proposing a solution for two agents and discussing the challenges that a generalized method will need to overcome.

2 Related work

Fair division is a well studied field that has drawn the attention of researchers for more than half a century [14, 5, 12]. The general aim of this field is to propose methods that allow agents with private preferences to divide a good among them. The methods offer guarantees to agents on the level of fairness, as long as they follow the protocol of the method. Standard fair division studies consider cake cutting algorithms in which agents perform the cut and choose operations to divide a heterogeneous resource (cake) among them [12].

Several studies acknowledged the existence of allocations that are both fair and efficient. Weller [16], and later Barbanel [3], prove the existence of envy-free Pareto

[2] Approximately fair protocols were suggested in previous studies, e.g., RobertsonW98. However, as far as we know, we are the first to present the relation between the level of guaranteed fairness and efficiency.

optimal allocations for a single heterogeneous good (cake) among n agents. However, there are very few studies on methods for finding fair and efficient allocations for a general cake. For multiple homogeneous divisible goods for which agents have linear utility functions, Reijnerse and Potters [11] propose an algorithm for finding an allocation among n agents, which is envy-free and Pareto optimal. Their solution is centralized, i.e., they assume that a central entity holds the true utility functions of all agents, and based on *market clearing* that is achieved using Fisher's model [7]. A later study [6] proposes a polynomial-time combinatorial algorithm for solving the same market clearing problem. The market clearing problem can also be solved using the *Eisenberg Gale* linear program [7].

Another attempt to apply a centralized algorithm for finding an efficient and fair division of multiple divisible homogeneous goods among two agents, was presented by Brams and Taylor [5]. They propose two methods that find an equitable allocation (an alternative notion of fairness in which both agents consider their allocation to be of the same value). One of the methods (the "adjusted winner") finds a Pareto optimal allocation while the other divides each of the goods proportionally between the two agents.

Another study, [8], proves that a division between two agents, which is fair and efficient with respect to a single planar cut of the cake, exists and offers a centralized method for finding it.

All of the above studies assume that a mediator holds the agents' preferences and computes the allocation. This is in contrast to standard cake cutting methods in which agents do not reveal their preferences to others [12].

The special case of an allocation of multiple indivisible goods has also drawn the attention of researchers. In this case a fair and efficient allocation does not always exist and thus, studies investigate the conditions for its existence and the complexity for finding it [9, 4].

We describe in Section 4 the method proposed by Sen and Biswas for increasing the efficiency of a division of a general cake between two agents via a cake cutting algorithm.

Trust is a concept whose different aspects are well studied by social scientists. These aspects include the development of trust [17] and the efficiency of teams with respect to the level of trust between their members [1]. The importance of the concept of trust in multi-agent systems was also acknowledged and drew extensive attention [10]. However, to the best of our knowledge, ours is the first attempt to increase the efficiency of resource allocation in multi-agent systems with respect to the level of trust, i.e., expose the agents partially with respect to the level of trust and use this exposure to increase efficiency.

3 Preliminaries

Our goal is to divide an infinitely divisible but bounded heterogeneous resource (cake) X between n agents. We assume that the cake has a rectangular shape with length L and width 1. We further assume that all cuts are planar. A piece of the cake x can be noted by an ordered pair $x = \langle x^l, x^r \rangle$, where $0 \leq x^l \leq x^r \leq L$. The numerical values of

x^l and x^r are their distances from the left edge of the cake (coordinates) and therefore the length of piece x is equal to $x^r - x^l$. Thus, $X = \langle 0, L\rangle$ and a result of a single cut at distance $c < L$ from the left edge of the cake is two pieces $\langle 0, c\rangle$ and $\langle c, L\rangle$. When $x^l = x^r$ the piece is empty (of size zero). For the operators $\subseteq$ and $\subset$ the standard definitions (for sets) apply for pieces as well.

The following operators are defined on pieces:

– $\tilde{\cap}$: assume without loss of generality that $m' \leq m$.

$$\langle m', n'\rangle \,\tilde{\cap}\, \langle m, n\rangle = \begin{cases} nil, & n' \leq m \\ \langle m, n'\rangle, & m < n' < n \\ \langle m, n\rangle, & n \leq n' \end{cases}$$

– $\tilde{\cup}$: assume without loss of generality that $m' < m$.

$$\langle m', n'\rangle \,\tilde{\cup}\, \langle m, n\rangle = \begin{cases} nil, & n' < m \\ \langle m', n\rangle, & m \leq n' < n \\ \langle m', n'\rangle, & n \leq n' \end{cases}$$

We will use the term sub-piece to describe a piece that is contained in another piece, i.e., $\langle m, n\rangle$ is a sub-piece of piece $\langle m', n'\rangle$ if and only if $\langle m, n\rangle \,\tilde{\subseteq}\, \langle m', n'\rangle$.

The operator $\tilde{\setminus}$ removes a sub-piece $\hat{x}$ from a piece x that contains it. The result is a set including the remaining two pieces to the left and right of the removed sub-piece. Formally:

$$\langle m, n\rangle \,\tilde{\setminus}\, \langle m', n'\rangle = \begin{cases} \{\langle m, m'\rangle, \langle n', n\rangle\}, & \langle m', n'\rangle \,\tilde{\subseteq}\, \langle m, n\rangle \\ \langle m, n\rangle, & \text{otherwise} \end{cases}$$

We define a *max-piece* in a set of pieces S as follows: $x \in S$ is a max-piece in S if there is no ordered subset $\{x_i, ..., x_k\}$ of S, for which $x \subset [x_i \tilde{\cup} ... \tilde{\cup} x_k]$. In other words, max pieces are obtained from a given set of pieces by applying the union operator on any two pieces that are not disjoint. Any set of pieces can be uniquely represented by a set of max pieces.

An allocation A is constructed of n disjoint finite sets of pieces, $X_1, ..., X_n$, such that if we order the pieces in the union of these sets according to their left coordinate (x^l) and apply the $\tilde{\cup}$ operator on all of them in this order (from left to right), we get the entire cake X. Furthermore, for any two pieces in this union, x and x', $x \,\tilde{\cap}\, x' = nil$. Intuitively, the entire cake is split between the agents and the cutting process does not decrease the quantities so the union of the agents' pieces is the entire cake.

We define a *max-allocation* to be an allocation in which all the pieces included in the sets X_a and X_b are max-pieces. Each allocation can be represented as a max allocation and this representation is unique. In the rest of this paper, when we discuss allocations, we will always refer to max-allocations unless we specifically say differently. Similarly, we will always refer to max-pieces when discussing pieces allocated to agents unless we specifically state differently

We further assume that for each agent i, $1 \leq i \leq n$ the function $F_i : \mathbb{R} \rightarrow \mathbb{R}$ defines for each point of the cake its value to agent i. We define $F_i(z) = 0$ for $z < 0$ and $z \geq L$. We assume that for $0 \leq z < L$, $F_i(z) > 0$ and that for $0 < z < L$, $F_i(z)$ is continuous and differentiable.

The utility function U_i defines the utility that agent i derives from a piece allocated to her, i.e., for $1 \leq i \leq n$:
$U_i(x) = \int_{x^l}^{x^r} F_i(z)\,dz$.

We assume that the utilities agents derive from an allocation of the entire cake are equal and we normalize them as follows:
$U_i(X) = \int_0^L F_i(z)\,dz = 1$.

We will use the notation $U_i(A)$ for the utility agent i derives from an allocation A, which is equal to the utility the agent derives from her allocated set of pieces in A,
$U_i(A) = \sum_{x \in X_i} \int_{x^l}^{x^r} F_i(z)\,dz$.

We assume that an agent i can compute accurately for any $0 \leq m \leq n \leq L$ the integral: $\int_m^n F_i(z)\,dz$ and that agents can perform cuts accurately, i.e., if an agent cuts a piece $\langle m, n\rangle$ at point k, $m \leq k \leq n$, the result are two pieces $\langle m, k\rangle$ and $\langle k, n\rangle$.

We will call an allocation A *efficient*, if it is Pareto optimal, i.e., there is no other allocation A' so that for some $j \in \{1, ..., n\}$: $U_j(A') > U_j(A)$ and for all $1 \leq i \leq n, i \neq j$: $U_i(A') \geq U_i(A)$.

The *social welfare value* of an allocation A is the summation of utilities $U_1(A) + ... + U_n(A)$. An allocation A has an optimal social welfare value (SW_{opt}) when there is no A' with $U_1(A') + ... + U_n(A') > U_1(A) + ... + U_n(A)$.

An allocation A is *proportional* if for every agent $1 \leq i \leq n$, $U_i(A) \geq \frac{1}{n}$.

4 Austin's method and asymmetric extensions

Austin's moving knife procedure can find an *exact* division of a heterogeneous cake between two agents, in which both agents consider their share as exactly $\frac{1}{2}$ [2, 12]. One agent (without loss of generality we will assume that this is agent a) holds two parallel knives. In the initial state, the left knife is placed at point zero and the right knife at point r so that $\int_0^r F_a(z)\,dz = \frac{1}{2}$. Agent a moves both knives to the right so that for every location of the left knife, the right knife is placed so the piece between the knives is worth exactly $\frac{1}{2}$ to her (we will refer to the piece between the knives as P and to the remainder of the cake as P'. P_{ll} and $\bar{P_{ll}}$ will be used to note the piece between the knives and the remainder when the left knife location is ll. The initial location of the left knife is 0, in which agent a puts the left knife on the left edge of the cake. The final location in which the right knife reaches the right edge will be noted by $\hat{ll}$. When at some location of the left knife ll, $0 \leq ll \leq \hat{ll}$, $U_b(P_{ll}) = \frac{1}{2}$, agent b calls stop, she gets P_{ll} while agent a gets $\bar{P_{ll}}$. Notice that if both agents followed the protocol, $U_a(P_{ll}) = U_b(\bar{P_{ll}}) = \frac{1}{2}$. Such an allocation can always be found by Austin's procedure due to the continuous nature of the scan of the two knives by agent a.

A small adjustment to Austin's procedure can result in increased efficiency. Notice that while $U_a(P_{ll}) = \frac{1}{2}$ for any location of the left knife ll, $0 \leq ll \leq \hat{ll}$, $U_b(P_{ll})$ may be changing. Thus, if we would allow agent b to observe the full process in which agent a moves the knives from the initial position to the final complementary position, and then choose the piece $P_{ll}, 0 \leq ll \leq \hat{ll}$, we can increase the efficiency of the method, since for the resulting allocation A', $U_b(A') \geq \frac{1}{2}$ while $U_a(A') = \frac{1}{2}$. However, it is clear that this increment in efficiency is one-sided (agent a would never derive more utility than $\frac{1}{2}$).

A different extension to Austin's method, which increases its efficiency, was proposed by Sen and Biswas [13]. This method is also asymmetric, only in contrast to the asymmetric extension of Austin's method described above, here, the advantage is to the cutting agent (a). The advantage for agent a is derived from the assumption that she is holding a model of the utility function of agent b, $\hat{U}_b$. As before, we assume that agent b is allowed to observe the entire process in which the knives are moved by agent a across the cake and select the position of the left knife ll, in which $U_b(P_{ll})$ is maximal.

In order to increase the efficiency of the allocation, agent a selects a piece $\hat{P}$, for which $\hat{U}_b(\hat{P}) = \frac{1}{2}$ and $U_a(\hat{P})$ is minimal. Then, she makes sure that agent b will prefer this piece $\hat{P}$ over any other P_{ll} by keeping the knives so that $\hat{U}_b(P_{ll'}) < \frac{1}{2}$ for $ll' \neq ll$. Thus, agent b selects $\hat{P}$ and if $\hat{U}_b$ is accurate, the utility for agent a is the greatest possible among the allocations with two cuts in which agent b receives a proportional share.

The two methods described above are both asymmetric. Both give an advantage to one of the agents over the other. This advantage allows the agent to choose the allocation that maximizes her gain, given that the allocation does not require more than two cuts and that the utility the other agent derives from it is exactly $\frac{1}{2}$. However, the utility derived from the allocation to the agent who does not have the advantage can never be larger than $\frac{1}{2}$. Therefore, allocations that increment the benefit for both agents are not considered.

5 Trust Efficient Allocations

The shortcoming of asymmetric methods in finding allocations that extend the benefit for both agents beyond their proportional share, motivates the development of a model or method that will enable such allocations. As mentioned in Section 1, the relation between trust and efficiency in applications such as multi-agent negotiation, has been acknowledged in previous studies. However, besides the potential for cooperation between agents that will result in efficiency, by definition, trust includes the willingness of agents to become vulnerable to the manipulations of other agents. Such vulnerability somewhat contradicts the motivation for fair division methods that offer guarantees to agents regardless of the actions of others. In reality, this trust is rarely a "take it or leave it" (binary) choice. While it would not be realistic to assume that an agent would trust another enough to risk her entire share, commonly, some level of trust between agents does exist. In other words, in many cases agents would be willing to expose themselves partially in order to increase the efficiency of the result. The amount of risk they will be willing to take (the level of trust) is determined by many elements and has been studied by social scientists [17]. Our goal is to introduce trust into the existing efficiency formalization of cake allocations. This formalism will set lower bounds on the efficiency of allocations dependent on the level of trust between agents.

We propose a novel approach to efficiency in cake cutting algorithms depending on the level of trust between the agents. To this end, we make the following innovative definitions:

Definition 1 *l-trust: given l, the symmetric level of trust among agents $1, ..., n$, l-trust is the fraction of the proportional share that the agents are willing to risk.*

Following this definition is an incentive participation constraint for each agent $1 \leq j \leq n$, where for any possible resulting allocation A, $U_j(A) \geq \frac{1-l}{n}$.

Definition 2 *l-trust-efficiency: An allocation A is l trust efficient if there does not exist an ordered set of agents $1, ..., k$ each holding max-pieces $x_j, 1 \leq j \leq k$, respectively, such that $U_j(x_{j-1}) \geq U_j(x_j)$ for $j = i+1, ..., k$ and $U_i(x_k) > U_i(x_i)$. Furthermore, $U_j(x_{j-1}) \geq \frac{1-l}{n}$, $j = i+1, ..., k$ and $U_i(x_k) \geq \frac{1-l}{n}$.*

Intuitively an l-trust-efficient allocation does not include a *Pareto improvement exchange cycle*, which is a cycle in which each of the participating agents gives a piece to another agent and receives a different piece, and for one agent this exchange increases her utility while for all others the utility does not decrease [3]. For l-trust-efficiency we add another constraint, that the derived utility for each agent from the piece she is receiving is at least $\frac{1-l}{n}$,

The definition of l-trust-efficient allocations is inspired by the definition of Pareto optimal allocations in which no exchange that is strictly beneficial to one agent and weakly beneficial to all others is possible [3]. Intuitively, l-trust-efficiency is the resolution in which the value of pieces to different agents can be identified.

6 Finding l-trust-efficient allocations between two agents

For two players we start by extending the problem definition from Section 3 to include l-trust. Formally, we assume that besides the cake X and the functions F_a and F_b, the input of the problem includes the symmetric level of trust, l, $0 \leq l < 1$. Our aim is to find an l-trust-efficient allocation A, in which $U_a(A) \geq \frac{1-l}{2}$ and $U_b(A) \geq \frac{1-l}{2}$.

6.1 LTE

We propose the following method for finding l-trust-efficient allocations, LTE:

1. Agent a places the left knife on the left edge of the cake and the right knife so that $U_a(P_0) = \frac{1-l}{2}$.
2. Agent a moves the knives to the right, keeping $U_a(P_{ll}) = \frac{1-l}{2}$ until the right knife reaches the right edge of the cake.
3. Agent b decides which pieces of the cake to allocate to agent a and which parts to herself, cuts the cake and makes the allocation accordingly.

It remains to describe how agent b decides on the allocation at the third step. Notice that, like in Austin's procedure, while $U_a(P_{ll})$ remains the same for each ll, $0 \leq ll \leq \hat{ll}$, $U_b(P_{ll})$ may be changing with the movement of the knives. The function $U_b(P_{ll})$ is observed and analyzed by agent b, in order to produce the allocation.

If possible, agent b selects a value $v \leq \frac{1-l}{2}$ and selects a set of pieces X_a where:

1. $x \in X_a \Rightarrow x = P_{ll}, 0 \leq ll \leq \hat{ll}$.
2. $\forall x, x' \in X_a, x \neq x' \Rightarrow x \tilde{\cap} x' = nil$.
3. $x \in X_a \Rightarrow U_b(x) \leq v$.

4. $\exists \hat{x} \subseteq x \in X_b$ s.t. $\hat{x} = P_{ll}, 0 \leq ll \leq \hat{ll} \Rightarrow U_b(\hat{x}) > v$.
5. $U_b(X_b) \geq \frac{1-l}{2}$.
6. $X_a \neq \emptyset$.

Notice that the conditions above do not necessarily define a max-allocation since the pieces in X_a can be adjacent. However, as always the resulting allocation has an equivalent max-allocation.

If no such value v can be found, then agent b selects any location ll, $0 \leq ll \leq \hat{ll}$ for which $U_b(P_{ll})$ is minimal and allocates P_{ll} to a, leaving the rest of the cake for herself.

The conditions listed above for selecting the value v offer some degree of freedom for agent b in selecting an l-trust-efficient allocation. We will call the method in which the maximal possible value for v is selected *LTE-max*. We will prove in Section 6.2 that the selection of the maximal value for v maximizes the social welfare value of the resulting allocation. We note that selection of a value v can result in a number of possible allocations. In order to establish determinism we further assume that the following two ordering decisions are used when selecting the pieces that will be added to X_a:

1. A piece x will always be added to X_a before a piece x' if $U_b(x) < U_b(x')$.
2. If $U_b(x) = U_b(x')$, then the piece with the smaller left coordinate will be selected first.

When there is a limit to the number of cuts that can be made when performing the allocation, the LTE method can be adjusted by adding an additional constraint to the conditions for the selection of value v. The cuts should be made to generate max-pieces only after the max-allocation is identified. Thus, assigning a set of consecutive pieces to a single agent would result in two cuts at most. The smallest number of cuts that allows LTE to find an l-trust-efficient allocation is 2. These two cuts are required so that at least one piece x with $U_a(x) = \frac{1-l}{2}$ and with minimal value b, can be allocated to agent a.

6.2 Properties

The first property is concerned with the guarantees provided to agents by the LTE method.

Theorem 1 *For any allocation A found by LTE, $U_a(A) \geq \frac{1-l}{2}$ and $U_b(A) \geq \frac{1-l}{2}$.*

Proof: Immediate by construction.

Next, we prove that the LTE method proposed above indeed finds an l-trust-efficient allocation.

Theorem 2 *Any allocation found by LTE is l-trust-efficient.*

Proof: The case where no value v that satisfies the conditions described in Section 6.1 exists is trivial; therefore we will only prove the case in which such a value v was found.

Assume that there exists a piece $x \in X_a$ and a piece $x' \in X_b$ so that (reminder, we are considering max-allocations):

1. $U_a(x') \geq \frac{1-l}{2}$.
2. $U_b(x) \geq \frac{1-l}{2}$.
3. $U_a(x) < U_a(x')$.
4. $U_b(x) \geq U_b(x')$.

By construction, $U_a(x) = k\frac{1-l}{2}, k \in \mathbb{N}$, and $U_b(x) < kv$. Therefore, according to the assumption, $U_a(x') > k\frac{1-l}{2}$ and thus, x' can be divided into $k+1$ consecutive sub-pieces, where for each sub-piece $\hat{x}'$ among the first k, $U_a(\hat{x}') = \frac{1-l}{2}$ and $U_b(\hat{x}') \geq v$. Thus, $U_b(x') \geq kv > U_b(x)$ in contrast to our assumption.

Notice that in the last expression we do not use the additional $k+1$ sub-piece. Therefore, the same proof holds for the case where:

1. $U_a(x') \geq \frac{1-l}{2}$.
2. $U_b(x) \geq \frac{1-l}{2}$.
3. $U_a(x) \leq U_a(x')$.
4. $U_b(x) > U_b(x')$.

□

The selection of v can affect the social welfare value. Therefore, we prove the following property:

Theorem 3 *For two allocations A and A' found by LTE with the corresponding values v and v', $v \geq v' \Rightarrow U_a(A) + U_b(A) \geq U_a(A') + U_b(A')$.*

Proof: Since $v \geq v'$, either $A = A'$, or there exists a piece (without loss of generality we assume there is exactly one such piece) x with the following properties:

1. $U_a(x) = \frac{1-l}{2}$.
2. $U_b(x) \leq v \leq \frac{1-l}{2}$.
3. in A, $x \in X_a$.
4. in A', $x \in X_b$.

Therefore, due to the deterministic manner in which the allocation to agent a by agent b is determined in LTE, both allocations are identical for $X \tilde{\setminus} x$ and $U_a(x) > U_b(x)$. Thus, $U_a(A) + U_b(A) \geq U_a(A') + U_b(A')$. □

Last, we prove the strong relation between our proposed notations and method to global efficiency (social welfare value). We start by defining a *flip point*: $\hat{z}$ is a flip point if $F_j(\hat{z}) = F_i(\hat{z})$ and $F'_j(\hat{z}) \neq F'_i(\hat{z})$.

We state the following Lemmas:

Lemma 1 *In LTE, a piece x for which $U_a(x) \geq \frac{1-l}{2}$ or $U_b(x) \geq \frac{1-l}{2}$, and x does not contain a flip point, is allocated to agent a if and only if $U_a(x) \geq U_b(x)$.*

Lemma 2 *For a piece of the cake $\langle z^l, z^r \rangle$, which does not contain a flip point, and $F_a(z) > F_b(z)$ for each $z^l \leq z \leq z^r$, the number of sub-pieces that are not allocated to agent a is equal to the number of extreme points of F_b in $\langle z^l, z^r \rangle$.*

Now we can state and prove the following theorem:

Theorem 4 *When the level of trust between agents is the highest, LTE-max finds an allocation that is optimal in terms of social welfare, i.e.,* $[U_a(A_l) + U_b(A_l)]_{l\to 1} = SW_{opt}$ [3].

Proof: Note by A^* the allocation that maximizes social welfare. Assume there are k flip points in X. According to Lemma 1, if piece x does not contain a flip point and one of the agents values it at least as $\frac{1-l}{2}$, then LTE-max allocates x to the agent who values it more. Thus, all P_{ll} pieces, $0 \le ll \le \hat{l}l$, in A (an allocation found by LTE-max) that do not contain flip points are allocated as in A^* except for the k pieces, which include the flip points and the pieces in set M from which agent a derives less utility than $\frac{1-l}{2}$, which are allocated to agent b. According to Lemma 2 this number is bounded by the number of extreme points in F_b. Thus, the following holds: $U_a(A_l) + U_b(A_l) \ge U_a(A^*) + U_b(A^*) - k\frac{1-l}{2} - |M|\frac{1-l}{2} = U_a(A^*) + U_b(A^*) - (k + |M|)\frac{1-l}{2}$.
Since $lim_{l\to 1}\frac{1-l}{2} = 0$, we get that when $l \to 1$, $U_a(A_l)+U_b(A_l) \ge U_a(A^*)+U_b(A^*)$.
□

Notice that the scale of the level of trust has social welfare on one side (when $l \to 1$); on the other side, when $l = 0$ we get the asymmetric extension of Austin's method.

6.3 Example

Consider the example depicted in Figure 1. The agents have contradicting preferences for the left side of the cake, while having similar preferences for its right side. We evaluated two levels of trust, $l = 0.4$ and $l = 0.8$, and had agent a move the knives accordingly. The functions $U_b(P_{ll})$, which agent b generates for the two different levels of trust, are depicted in Figure 2. The utilities derived by the agents from the resulting allocations, are depicted in Table 1.

The sum of the resulting utilities that the agents derive from the allocation ($U_a = 0.3$ and $U_b = 0.8$) expands the benefit beyond 100% as derived in Austin's procedure. However, agent a received the minimal value according to the l-trust guarantees.

l-trust	0.4	0.8
U_a	0.3	0.7
U_b	0.8	0.719
SW	1.1	1.419

Table 1. Utilities derived by agents and the social welfare value in the different example scenarios

If the trust level between the agents is greater, e.g., $l = 0.8$, agent b can be much more specific and expressive regarding her preferences. The resulting utilities are $U_a(X_a) = 0.7$ and $U_b(X_b) = 0.719$. Thus, the greater level of trust not only enabled an allocation with greater social welfare value (1.419), but also, both agents derived utilities beyond a 50% allocation.

[3] A_l is defined as before, an allocation found by LTE when the level of trust is equal to l.

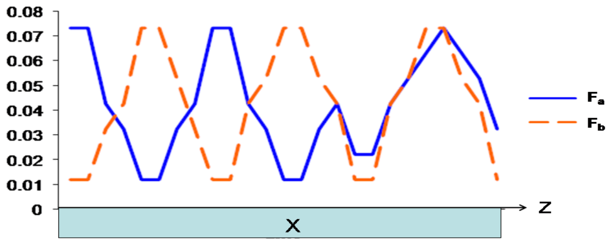

Fig. 1. Example of LTE

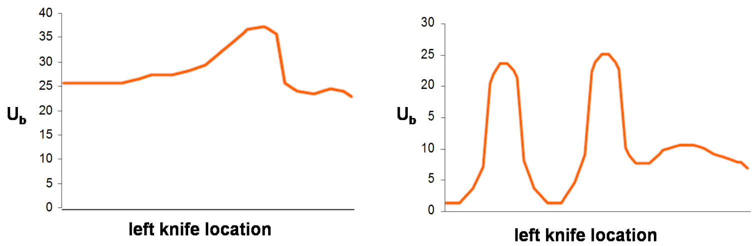

Fig. 2. $U_b(P_{ll})$ for $l = 0.4$ (left) and $l = 0.8$ (right)

7 Discussion of the general case

An l-trust-efficient allocation can be found using the following extension of the proposed LTE method for two agents. We order the agents $1, ..., n$ and let the first among them, agent 1, move the knives in the same manner as agent a did in the two agents method. Each agent $1 \leq i \leq n-1$ generates the function $U_i(ll)$ as done by agent b in the two agents version. Then, all functions $U_i(ll)$ are passed to agent n, which generates the function $U_{max}(ll) = maxU_i(ll), 1 \leq i \leq n-1$. Agent n adds to X_n disjoint pieces so that $x \in X_n \Rightarrow U_i(x_l) = U_{max}(x_l)$ and $U_n(x) > \frac{1-l}{n}$. The pieces are added in a deterministic order beginning with the piece from which agent n derives the most utility to the piece which she derives the least from. This process repeats with agent $n-1$ selecting pieces not yet allocated according to $U_{max}(ll)$, and so on until finally agent $n-1$ splits what ever is left with agent n.

While this method would result in an l-trust efficient allocation of pieces x for which $U_1(x) = \frac{1-l}{n}$, the following issues need to be solved so the generalized method will apply to the properties achieved by the two agent method:

1. There can exist a piece x', for which $U_1(x') < \frac{1-l}{n}$ but for some other agent $j > 1$, $U_j(x') = \frac{1-l}{n}$. We need to be careful when allocating such pieces in order not to lose l-trust-efficiency.
2. The allocation may not satisfy the trust guarantees for some agents, i.e., when agent i is considering her share, there might be not enough left so that $U_i(X_i) \geq \frac{1-l}{n}$. We

will need to propose some initial phase in which each agent receives her guarantee before applying the method above.

8 Conclusion

In this paper we proposed the use of *trust in cake cutting algorithms*. We defined the level of trust between agents as the proportional quantity of their fair share that they are willing to expose to the actions of other agents, and risk losing. We further defined a new concept, l-trust-efficiency, which generalizes the Pareto efficiency concept. When an allocation is l-trust-efficient, there does not exist any other allocation that can be derived from the current allocation by exchanging pieces that are worth at least $\frac{1-l}{n}$ to the agents between them and is strictly better for at least one agent and at least as beneficial to all other agents as the current allocation.

We proposed a method for finding l-trust-efficient allocations between two agents. The method allows agents to achieve this kind of efficiency with respect to the level of trust between them, but at the same time, guarantees the allocation of the quantity that they were not willing to risk. The method allows the agents to divide the cake between them according to the utility they derive from allocations of the different parts of the cake (the one who values it more gets the share) and, as a result, achieve not only the efficiency we defined but also increased the social welfare of the allocation.

We discussed the challenges in proposing a method that finds l-trust-efficient allocations between n agents. In future work we intend to find solutions to these challenges and propose a method for the general case.

References

1. B. E. Ashforth and R. T. Lee. Defensive behavior in organizations: A preliminary model. *Math. Gazett*, 43:621–648, 1990.
2. A. K. Austin. Sharing a cake. *Math. Gazett*, 66:212–215, 1982.
3. J. Barbanel. Partition ratios, pareto optimal cake division, and related notions. *Journal of Mathematical Economics*, 32:401–428, 1999.
4. S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research (JAIR)*, 32:525–564, 2008.
5. S. J. Brams and A. D. Taylor. *Fair Division. From cake-cutting to dispute resolution*. Cambridge University press, 1996.
6. N. R. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani. Market equilibrium via a primal-dual-type algorithm. In *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 389–395, Washington, DC, USA, 2002.
7. D. Gale. *The Theory of Linear Economic Models*. McGraw-Hill, 1960.
8. M. A. Jones. Equitable, envy-free, and efficient cake cutting for two people and its application to divisible goods. *Mathematics Magazine*, 75(4):275–283, 2002.
9. M. Meertens, J. A. M. Potters, and J. H. Reijnierse. Envy-free and pareto efficient allocations in economies with indivisible goods and money. *Mathematical Social Sciences*, 44(3):223–233, 2003.
10. S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

11. J. H. Reijnierse and J. A. M. Potters. On finding an envy-free pareto-optimal division. *Mathematical Programming*, 83:291–311, 1998.
12. Y. J. Robertson and W. Webb. *Cake-Cutting Algorithms, Be Fair If You Can*. A K Peters, Ltd, 1998.
13. S. Sen and A. Biswas. More than envy-free. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, page 433, Washington, DC, USA, 2000. IEEE Computer Society.
14. H. Steinhaus. The problem of fair division. *Econometrica*, 16:101 – 104, 1948.
15. L. R. Weingart and J. M. Brett. Tactical behavior and negotiation outcomes. *International Journalof Conflict Management*, 1:7–31, 1990.
16. D. Weller. Fair division of a measurable space. *Journal of Mathematical Economics*, 14(1):5–17, February 1985.
17. J. M. Wilson, S. G. S. b, and B. McEvily. All in due time: The development of trust in computer-mediated and face-to-face teams. *Organizational Behavior and Human Decision Processes.*, 99(1):16–33, 2002.