Bounded and Envy-free Cake Cutting

HARIS AZIZ, DATA61 and UNSW
and
SIMON MACKENZIE, Carnegie Mellon University

Although a finite envy-free cake-cutting protocol has been known for more than twenty years, it had been open whether a protocol exists in which the number of steps taken by the protocol is bounded by a function of the number of agents. In this letter, we report on our recent results on discrete, bounded, and envy-free cake-cutting protocols.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms: Theory, Algorithms, Economics

Additional Key Words and Phrases: Fair allocation, Solutions Concepts, Multiagent resource allocation, Computational Complexity

1. INTRODUCTION

The cake cutting problem is a fascinating and fundamental mathematical problem in which the cake is a heterogeneous divisible resource represented by the unit interval [Brams and Taylor, 1996, Robertson and Webb, 1998]. Each of the n agents have additive and non-negative valuations over segments of the interval. The challenge is to query agents about their valuations in an efficient way to find a fair allocation. Originally formalized by Polish mathematician Hugo Steinhaus in the 1940's, the problem has attracted considerable attention in mathematics, computer science and economics. One of the most important criteria for fairness is *envy-freeness*. An allocation is envy-free if no agent would prefer replacing her allocation with another agent's. A cake cutting protocol is termed envy-free if each agent is guaranteed to be non-envious if she reports her real valuations. If a protocol is envy-free, then an honest agent will not be envious even if other agents misreport their valuations.

The most famous envy-free cake cutting protocol is *Divide and Choose* for two agents: one agent is asked to divide the cake into two equally preferred pieces. The other agent is then asked to pick her most preferred piece whereas the cutter gets the remaining piece. In the 1960's, John Selfridge and John Conway independently proposed an envy-free protocol for three agents that requires at most five cuts on the cake. In the early 1990's, Steven Brams and Alan Taylor invented a general finite envy-free cake cutting protocol [Brams and Taylor, 1995]. The protocol can require arbitrary number of steps and cuts on the cake even for four agents. It has been an open problem whether a four-agent bounded envy-free protocol exists or not [Brams and Taylor, 1995, Procaccia, 2013, 2016].

This year we presented a four agent envy-free protocol that requires 203 cuts on

Authors' addresses: haris.aziz@data61.csiro.au, simonm@andrew.cmu.edu

the cake and a total of 584 queries [Aziz and Mackenzie, 2016a].[1] We have now generalized the protocol to any number of agents [Aziz and Mackenzie, 2016b,c]. In this letter, we give an overview of some of the ideas and building blocks of the general protocol.

2. A BIRD'S EYE VIEW OF THE PROTOCOL

In a nutshell, the protocol allocates a large enough portion of the cake in an envy-free manner. After that, it tries to add some small portions of the unallocated cake to the allocated part in a structured and envy-free manner with the goal to reduce the problem to envy-free allocation for a smaller number of agents.

A crucial building block of our protocol is the Core Protocol. The Core Protocol asks one of the n agents—the cutter—to divide the cake into n equally preferred pieces. It then uses the recursive SubCore Protocol to obtain a *neat* allocation for the other agents. In a neat allocation, each agent gets a part of exactly one of the pieces, one agent gets a full piece, and the agents are not envious of each other or of the unallocated pieces. The cutter then gets one of the unallocated and untrimmed pieces.

The SubCore is a general protocol that takes as input agents and pieces of cake where the number of agents is at most the number of pieces. We order the agents and find a neat envy-free allocation for an expanding set of agents. Say that we have found a neat allocation for $m-1$ agents. In that case, $m-2$ pieces could be partially allocated but the other pieces are untrimmed. If the m-th agent thinks that one of the unallocated full pieces is her most preferred, then she is simply given such a piece. Otherwise, we have a situation where m agents are interested in $m-1$ '*contested*' pieces. In such a situation one of the m agents has to be given a most preferred uncontested piece. In order to find such an agent as well as reallocate the $m-1$ contested pieces, we have to do more work and may have to call SubCore recursively for a smaller number of agents.

In the Core Protocol, the cutter agent gets a full piece. Another agent also gets a full piece. So from the cutter's perspective at least $2/n$ of the cake is allocated by one call of the Core Protocol. If we call the Core Protocol with a different cutter each time to further allocate the unallocated cake, we just need n calls of the Core Protocol to obtain an envy-free allocation in which each agent thinks she gets $1/n$ value of the whole cake. This answers an open problem posed by Segal-Halevi et al. [2015] which asks whether there exists a bounded algorithm that returns envy-free partial allocation that is proportional (gives each agent value at least $1/n$ of the whole cake).

Continuing to call the Core Protocol on the updated remaining cake gives no guarantee that the cake will be allocated fully even in finite time. Hence, we need to use other protocols. Throughout the overall protocol, we maintain an envy-free allocation as well as keep track of the updated unallocated cake.

Since the Core Protocol by itself is not powerful enough to allocate all the cake in bounded time, we rely on the idea of *domination.* An agent i *dominates* another agent j if she is not envious of j even if the unallocated cake is given to j. The

[1] Walter Stromquist observed that both the number of cuts and queries can be halved by a simple adjustment.

other protocols are used with the following objective in mind: find a set of agents $N \setminus A \subset N$ such that each agent in the set dominates each agent in $A \subset N$. In order to ensure that each agent in some set $N \setminus A$ dominates each agent in A requires changing the current allocations of the agents as well as the unallocated cake. While we make changes to the allocation, we ensure that the current partial allocation remains envy-free. By identifying such a set $N \setminus A$, we reduce the problem to envy-free allocation for a smaller number of agents. The agents in $N \setminus A$ are not envious whatever the unallocated cake is allocated among agents in N.

3. CONCLUSIONS AND OPEN PROBLEMS

In this letter, we provided a very high-level overview of our bounded envy-free protocol. The protocol has an upper bound that is a power tower of six n's. In the other direction, any envy-free protocol requires at least $\Omega(n^2)$ queries [Procaccia, 2016].[2] There is a lot of work to be done to close the gap between the current upper and lower bound.

We additionally show that even if we do not run our protocol to completion, it can find in at most n calls of the Core Protocol a partial allocation of the cake that achieves proportionality (each agent gets at least $1/n$ of the value of the whole cake) and envy-freeness. It also adds further evidence to the idea popularized by Segal-Halevi et al. [2015] that wasting some resource can lead to much faster fair division algorithms. If we allow for partial allocations, an interesting open problem is the following one: can envy-freeness and proportionality can be achieved in polynomial number of steps? Finally we mention that it is still open whether a bounded and envy-free cake cutting protocol exists for the case of where agents have negative valuations.

Acknowledgments

We thank Hu Fu, editor SIGEcom Exchanges, for inviting us to write this letter.

REFERENCES

H. Aziz and S. Mackenzie. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 454–464. ACM Press, 2016a.

H. Aziz and S. Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. Technical Report arXiv:1604.03655, arXiv.org, 2016b.

H. Aziz and S. Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 2016c.

S. J. Brams and A. D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

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

A. D. Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56 (7):78–87, 2013.

[2] In the second author's PhD thesis, this lower bound has been improved to $\Omega(n^3)$.

A. D. Procaccia. Cake cutting algorithms. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 13. Cambridge University Press, 2016.

J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can.* A. K. Peters, 1998.

E. Segal-Halevi, A. Hassidim, and Y. Aumann. Waste makes haste: Bounded time protocols for envy-free cake cutting with free disposal. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 901–908. IFAAMAS, 2015.