A Final Report for the Internship at Harvard

Jiale Chen

The Center on Frontiers of Computing Studies, Computer Science Dept., Peking University
jiale_chen@pku.edu.cn

August 16, 2021

1 Executive Summary

The research internship started from the end of January to the beginning of May in 2021. In the first two weeks, I was working on the envy-free cake-cutting problem. I read the paper of the currently best lower bound [1] and upper bound [2]. After understanding the construction of the lower bound, I gave an alternative proof of the $\Omega(n^2)$ bound but couldn't go further.

Then I switched to the Nash Welfare MDP problem and worked on it for a month. Specifically, I spent a week running simulations and finding counterexamples of the convergence of the algorithm in the proposal. Then I spent about three weeks proving the existence of a fixed point of the Bellman equation. I found an interesting explanation of the fixed point using the definition of equilibrium, but could only prove a mixed equilibrium exists. At the end of this project, I found a direction of multi-objective MDP that might be useful.

In the rest of the internship, I was working on the local fairness of the cake-cutting problem. I read several papers in details [2, 3, 4] and roughly understand the key ideas of current algorithms. Jamie gave a lower bound of local-proportional cake-cutting problem using the probabilistic method. After that, I was thinking about the similarity between these lower bound construction and found that the number of edges might play an important role. So I was trying to prove another $\Omega(n^2)$ lower bound for the star graph which contains $O(n)$ edges. I proved it in the discrete case and then was asked to think about the upper bound in tree cases. We found that even path-cases are hard to solve. I gave an algorithm for 4-path and 5-path. Jamie gave another solution on a 4-path case. Then we both found that it's hard to extend the algorithm into longer paths.

During the internship, I became more familiar with how to do research. I observed existing solutions and summarized the common points among them. I proposed research questions and partially solved them. Also, I learned a lot about those research topics and cooperated with others.

In this report, section 2 will present my work on policy aggregation via Nash welfare and section 3 will present my work on the local proportionality of cake-cutting.

2 Policy Aggregation via Nash Welfare

2.1 Background

Markov decision process (MDP) is a classical model in the decision-making topic. It consists of a 4-tuple (S, A, R, P) representing a finite state space, a finite action space, a reward function, and a transition matrix. Through inverse reinforcement learning, we can extract a reward function that is most likely to be the performed strategy of humans.

Here let's consider a setting where n people are making decisions. Their strategy is extracted to be n value functions $R_1, R_2, \ldots, R_n$. What we want to do is to build a representative agent such that its strategy is built upon those n people's strategies and it performs fairly.

Our method is to maximize the Nash welfare.

Question 1. *Find the policy that maximizes the Nash Welfare in each state. Formally, we want to find a randomized policy π^* that satisfies*

$$\forall s \in S,\ \pi^*(s) \in \text{argmax}_{q\in\Delta(A)} \prod_{i=1}^{n} \mathbb{E}_{a\sim q}\left[R_i(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)\cdot V_i^{\pi^*}(s')\right]$$

2.2 Algorithm in the proposal doesn't converge

First I tried to prove the convergence of algorithm in the proposal, which is described as follows.

- Given R,P,π and γ as inputs.
- In tth round,
  1. the algorithm first arbitrarily select a V and then update it using π_{t-1}. This step converges since it's contracting mapping in policy evaluation. After V converges, we call it V_{t-1}.
  2. We then calculate $\pi_t(s) = \arg\max_{q\in\Delta(A)} \prod_i E_{a\sim q}(R_i(s,a) + \gamma \sum_{s'} P(s' \mid s,a)V_{t-1}(s'))$

To prove the convergence, I tried the classical method, contracting mapping. Concretely, we define the operator $\phi(V)$ as follows:

1. Choose π according to $\pi(s) = \text{argmax}_{q\in\Delta(A)} \prod_{i=1}^{n} \mathbb{E}_{a\sim q}\left[R_i(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)\cdot V(s')\right]$
2. Update V according to π by policy evaluation.

By running simulations, I found that the mapping ϕ is not a contracting mapping. Moreover, I also found a counterexample in which there are two πs that each one is the successor of the other one, showing that the algorithm in the proposal doesn't converge at all.

2.3 Finding a fixed point

After that, I first doubted whether a solution exists. I planned to link this problem to equilibrium finding. Regard each state s as a player who can choose his strategy π_s. The utility of player s is $u_s(\pi) = \prod_i V_i^\pi(s)$. And we are finding an equilibrium.

Remark:

1. This is slightly different from the Nash Equilibrium because the mixed strategy here doesn't bring a linear combination of utility.
2. This definition of utility implied different dynamics than the original bellman equations. The difference is q in the equation exists for only the first round or forever.

I was first trying to prove that a pure equilibrium exists. If we are going to prove the existence of a pure equilibrium using Kakutani's theorem, a crucial condition is that the best response function is convex-valued. However, this condition cannot be satisfied directly. Below is the detail of the attempt.

The existence of pure equilibrium (a failed attempt) We denote $\Pi = (\Delta A)^{|S|}$ and the best response correspondence $B : \Pi \rightrightarrows \Pi$. B is defined as $B(\pi) = B_1(\pi_{-1}) \times \cdots \times B_{|S|}(\pi_{-|S|})$ and $B_s(\pi_{-s}) = \arg\max_{q\in\Delta(A)} u_s(q,\pi_{-s})$, where $u_s(\pi) = \prod_i V_i^\pi(s)$. And we are trying to show that B satisfies the conditions of Kakutani's theorem.

First, we should show the utility function is continuous. If we denote

- $v_i^\pi = [V_i^\pi(s_1),\ldots,V_i^\pi(s_{|S|})]$

- $r_i^\pi = [E_{a\sim\pi_1} R_i(s_1, a), \ldots, E_{a\sim\pi_{|S|}} R_i(s_{|S|}, a)]$
- $(P^\pi)_{i,j} = E_{a\in\pi_i} P(s_j \mid s_i, a)$

then using bellman expectation equation, we have $v_i^\pi = r_i^\pi + \gamma P^\pi v_i^\pi$, i.e., $v_i^\pi = (I - \gamma P^\pi)^{-1} r_i^\pi$. Since P^π, r_i^π is continuous and the inverse matrix of nonsingular matrix is continuous too, we have v_i^π is continuous and thus u_s is continuous. Then we can check the conditions of Kakutani's theorem.

1. Π is compact, convex, and non-empty: it's straightforward.
2. $B(\pi)$ is non-empty: since $\Delta(A)$ is a non-empty, compact set, and u_s is continuous, it can always achieve its extreme value.
3. $B(\pi)$ is convex-valued: according to the proposal, it's a convex program, so the optimal set is convex. **(This might not be true for this problem. q here performs in every round in the future, so the optimization problem is not the original convex program anymore.)**
4. $B(\pi)$ has a closed graph: suppose to obtain a contradiction, that $B(\pi)$ doesn't have a close graph.
   - Then, there exists a sequence $(\pi^n, \hat{\pi}^n) \to (\pi, \hat{\pi})$ with $\hat{\pi}^n \in B(\pi^n)$, but $\hat{\pi} \notin B(\pi)$, i.e., there exists some s such that $\hat{\pi}_s \notin B_s(\pi_{-s})$. This implies that there exists some $\pi_s' \in \Delta(A)$ and some $\epsilon > 0$ s.t. $u_s(\pi_s', \pi_{-s}) > u_s(\hat{\pi}_s, \pi_{-s}) + 3\epsilon$.
   - By the continuity of u_s, and the fact that $\pi_{-s}^n \to \pi_{-s}$, we have for sufficient large n, $u_s(\pi_s', \pi_{-s}^n) \geq u_s(\pi_s', \pi_{-s}) - \epsilon$ and $u_s(\hat{\pi}_s, \pi_{-s}) \geq u_s(\hat{\pi}_s^n, \pi_{-s}^n) - \epsilon$.
   - Combining the previous relations, we have $u_s(\pi_s', \pi_{-s}^n) \geq u_s(\hat{\pi}_s, \pi_{-s}) + 2\epsilon \geq u_s(\hat{\pi}_s^n, \pi_{-s}^n) + \epsilon$. This contradicts the assumption that $\hat{\pi}_s^n \in B_s(\pi_{-s}^n)$.

So there exists a fixed-point, i.e., $\forall s,\ \pi_s^* \in B_s(\pi_{-s}^*)$. $\square$

After that, I was looking for other theorems that might be helpful.

Theorem 2 (Debreu, Glicksberg, Fan)**.** *Consider an infinite strategic form game $\langle \mathcal{I}, (S_i)_{i\in\mathcal{I}}, (u_i)_{i\in\mathcal{I}} \rangle$ such that for each $i \in \mathcal{I}$*

1. *S_i is compact and convex.*
2. *$u_i(s_i, s_{-i})$ is continuous in s_{-i}*
3. *$u_i(s_i, s_{-i})$ is continuous and concave in s_i [in fact quasi-concavity suffices].*

Then a pure strategy Nash equilibrium exists.

The third condition of theorem 2 is exactly a sufficient condition for the breaking point in the previous proof. However, the concavity probably doesn't exist.

There is also another more powerful theorem.

Theorem 3 (Glicksberg)**.** *Consider an infinite strategic form game $\langle \mathcal{I}, (S_i)_{i\in\mathcal{I}}, (u_i)_{i\in\mathcal{I}} \rangle$ such that for each $i \in \mathcal{I}$*

1. *S_i is nonempty and compact.*
2. *$u_i(s_i, s_{-i})$ is continuous in s_i and s_{-i}*

Then a mixed strategy Nash equilibrium exists.

In our problem, all conditions of theorem 3 are satisfied, so a mixed equilibrium exists. But the mixed equilibrium corresponds to a distribution of distribution in our problem, which is not what we want. So if we still want to directly prove the existence of a pure equilibrium, more careful analysis of the utility function is needed. Then some variant of the fixed point theorems can be applied.

2.4 A possible direction

In paper[5], they have considered the problem Markov decision processes with multiple discounted reward objectives. Specifically, there are n different reward functions $R_1, R_2, \dots, R_k$. Given an initial state s, a strategy π, and a discount factor $0 \leq \gamma \leq 1$, the discounted reward value vector, or payoff profile is defined as $\langle V_1^\pi(s,\gamma), V_2^\pi(s,\gamma), \dots, V_n^\pi(s,\gamma)\rangle$. What might be useful is that they are solving MDPs by linear programming.

They fix a starting state s and define variables $x(t,a)$ representing the discounted frequency of the state-action pair (t,a) when the starting state is s. The constraints of the linear programming over $x(t,a)$s are then directly given by the transition matrix. Also the objective function o_i of one reward function can be written as the inner product of x and r_i. Since there are n reward function, what they have done is to approximate the Pareto Curve of $\langle V_1^\pi(s,\gamma), V_2^\pi(s,\gamma), \dots, V_n^\pi(s,\gamma)\rangle$.

Then consider our setting. In each state, the optimization goal becomes $O_s = o_1 \times o_2 \times \cdots \times o_n$. Solving this programming provides us a Nash welfare maximal solution on state s. And if we want to find a globally optimal solution on every state, it might be useful to check the Pareto curve of $\langle O_1, O_2, \dots, O_{|S|}\rangle$. In this step, we can follow their method dealing with multiple-objective linear programming. Also, it might be possible that there is a globally optimal solution that is optimal in each state. If this solution exists, then the intersection of the solution sets for the mathematical programming w.r.t. each state is nonempty. Maybe this is another direction.

As for solving the mathematical programming, if we consider two R_is and want a locally optimal solution, it's an convex quadratic programming with a objective function like $(a_1^T x)(a_2^T x)$. To go further,

1. multiple R_is lead to objective functions in the form of $(a_1^T x)(a_2^T x)\cdots(a_n^T x)$ where a_i, x are non-negtive vectors;

2. if there isn't a globally optimal solution, some pure equilibrium might also exists and can be found on the PO curve.

3 Local Proportionality of Cake Cutting

3.1 Background

The cake-cutting problem is about dividing a heterogeneous and divisible good among multiple agents with different preferences. And we care about the fairness of the allocation result.

In mathematical terms, there is a set of agent $N = \{1, 2, \dots, n\}$ and a cake $[0,1]$. Each agent i has a valuation function V_i for any subinterval in $[0,1]$. The valuation function is additive, divisible, non-negative and normalized. We will allocate a piece of cake X_i to agent i, where X_i is a finite union of disjoint intervals. Our goal is to design an algorithm that allocates a piece of cake X_i to each agent $i \in N$ such that $X_1, X_2, \dots, X_n$ forms a partition of $[0,1]$. Also, we require the allocation to satisfy some kind of fairness. Two important fairness criteria is

1. envy-freeness: for each agent, his value of his own piece of cake must be at least as large as his value for any other's piece of cake, i.e.,
$$\forall\ i, j, V_i(X_i) \geq V_i(X_j).$$

2. proportionality: each of n agents must receive a piece of cake with at least $\frac{1}{n}$ value, i.e.,
$$\forall\ i, V_i(X_i) \geq \frac{1}{n}.$$

What the algorithm can do is described by the Robertson-Webb model:

1. $\text{eval}_i(x_1, x_2)$: returns $V_i([x_i, x_2])$.

2. $\text{cut}_i(x_1, \alpha)$: returns $x_2 \in [0,1]$ such that $V_i([x_1, x_2]) = \alpha$.

And we are interested in the number of queries needed to derive a fair allocation in the cake-cutting problem.

Proportional cake cutting The Even–Paz protocol[6], based on recursively halving the cake and the group of agents, requires only $O(n \log n)$ actions. In each stage, we ask every agent's half value point and then sort those points from left to right. We choose the middle point and split the cake into two halves. The left cake and those agents with points on the left become the first subproblem. The right cake and those agents with points on the right become the second subproblem. Recursively solve those subproblems and then we get a proportional allocation.

Surprisingly, the recursive algorithm is proved to be the optimal solution up to a constant factor[4]. The proof of the lower bound itself is also interesting. First, they notice that in proportional cake cutting, the interaction between agents is small. They describe a 'Thin-Rich' game that involves only one agent. The algorithm is expected to find a piece of cake for this agent with width at most $\frac{2}{n}$ and value at least $\frac{1}{n}$. They find that if this game has a lower bound $T(n)$ then the proportional cake cutting has a lower bound $\Omega(nT(n))$. To bound $T(n)$ they introduce the definition value trees to help to construct the value distribution adversarially.

In detail, the value tree is a balanced 3-ary rooted tree, with $\frac{2}{n}$ leaves. Each node represents an interval which is the union of the intervals represented by its sons. The value of the node is the value of the corresponding interval. The construction of the distribution is done by assigning the weight on each edge of the tree. The edge weight represents the ratio of the son node's value to the parent node's value. Since the 'thin-rich' must intersects with a high-density leaf, the goal of the adversary is to find a way to assign edge weight such that heavy weight won't occur frequently along a path. Finally, it can be proved that after k queries, any root-leaf path contains at most $2k$ heavy edges.

Envy-free cake cutting Aziz and Mackenzie propose the first discrete and bounded protocol for envy-freeness[2]. The key idea of their protocol is to recursively use the Core Protocol to find a partial envy-free allocation and then to ensure that one set of agents will dominate others, so the problem can be reduced to a subproblem with fewer agents. Here the relationship that agent i dominates agent j means that even if all the remaining cake is allocated to j, agent i won't envy him.

The best lower bound for this problem is derived by Procaccia[1]. The idea is also to deal with the agents separately and design the valuation function. He regards both query operations w.r.t. agent i as adding two points on the $[0,1]$ and split the interval into small subintervals. Standing in agent i's view, the adversary should minimize the value i has and maximize the value other agents have. That is, if the allocation of i does not contain an entire small segment, i has no value for this part. But if agent j's allocation has a non-trivial intersection with a segment, j can have the value of the entire segments in i's view. When the algorithm outputs, the allocation must be envy-free even in the adversarial setting. A necessary condition for it is that i's value for his allocation should be at least the value of his favorite segment. Otherwise, some agent j may contain some part of this segment which leads to envy. To deriving the concrete lower bound, he chooses the "uniform-like case" to link the value and the length of the segments.

The proof can be slightly simplified that, in this "uniform-like case", in agent i's view, i's length $\geq$ i's value $\geq$ j's value $\geq$ j's length. So every agent must have a $\frac{1}{n}$ length of $[0,1]$. In i's view, i's value is at most $\frac{1}{n}$ while others' value is at least $\frac{1}{n}$. Thus the equal sign must be taken, which means that each agent's allocation should be exactly some segments in i's segment set and have a total length of $\frac{1}{n}$. That requires $\Omega(n)$ points in i's segment sets. Totally, we have an $\Omega(n^2)$ lower bound.

Locally fair cake cutting The cake cutting problem can be extended to a graph-based model. Suppose there is a graph structure of agents $G(V, E)$. Each node in V represents an agent. Different from the previous setting, the agent here only cares about the allocation results in his neighborhood. Thus, the fairness criteria become a local property.

1. For local proportionality, we are considering
$$\forall\ i \in V, \quad V_i(X_i) \geq \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} V_i(X_j).$$
2. For local envy-freeness, we are considering
$$\forall\ i \in V, j \in \mathcal{N}(i), \quad V_i(X_i) \geq V_i(X_j).$$

Bei et al.[3] design a protocol for the locally proportional cake cutting. There are two parts in their protocol: creating a dominant point and then spreading the dominance. The dominant here is defined as even if all the remaining cake is allocated to his neighbors, he remains proportional. They heavily rely on the Core Protocol in Aziz and Mackenzie's protocol[2]. More concretely, they choose a cutter point and run the Core Protocol to create enough insignificant pieces while maintaining the global envy-freeness. Afterward, they swap at least one insignificant piece to the neighborhood of the cutter point while maintaining local proportionality. As soon as this is done, the cutter point becomes a dominant point. There are two main properties of Core Protocol they use. Global envy-freeness is helpful and somehow necessary for maintaining local proportionality while swapping. The definition of the insignificant piece helps create dominance.

Our goal was to explore the locally fair cake-cutting problem. Jamie has proved that on some bipartite graph, there is an $\Omega(n^2)$ lower bound for algorithms that output a locally proportional allocation. Using the same method in [1], we can easily prove that for any graph with m edges, there is an $\Omega(m)$ lower bound for algorithms that output a locally envy-free protocol.

One of my observations is that the number of edges seems to play an important role in the proof structure. A question is can we achieve a $\Omega(n^2)$ lower bound in graphs with $o(n^2)$ edges? I'd like to first try the star graph.

3.2 Star Graph

An $O(n^2)$ locally proportional protocol The TreeCore protocol in [3] obtains a locally envy-free partial allocation on trees. But implementing it on a star graph can obtain a complete locally envy-free allocation. I present its simplified version on the star graph, the StarCore Protocol.

Algorithm 1 StarCore Protocol

```
Require: A star graph $S$ with center $r$, $|S| = n$, and the cake $[0, 1]$.
Ensure: A locally envy-free allocation on $S$.
1: Center r cuts the cake into $n$ equally preferred pieces in his own measure.
2: for all agent $u \in S \setminus \{r\}$ in an arbitrarily order do
3:     Agent $u$ takes the piece that she values the highest in the remaining pieces.
4: The last piece is allocated to $r$.
```

An $\Omega(n^2)$ lower bound in the discrete case When I was trying to prove a lower bound, I first consider the case that we fix the center as the cutter point and each agent gets one piece. The algorithm could only query the value of each part.

In this way, I called this problem the discrete case of star graph.

- There are n items and n people.
- Each agent i has a value v_{ij} on each item j.
- An allocation is called valid iff each agent is allocated one item and for any agent $i \in \{2, 3, \ldots, n\}$, she will not envy agent 1. Formally, a valid allocation is a permutation σ such that
$$\forall\ 2 \leq i \leq n, v_{i\sigma_i} \geq v_{i\sigma_1}.$$

- The algorithm doesn't know $\{v_{ij}\}$ but it can query for one of them each time.

Note that a valid allocation always exists by running the round-robin algorithm in the order of $2, 3, \ldots, n, 1$. The goal is to prove that any algorithm that always outputs a valid allocation needs $\Omega(n^2)$ queries.

As an adversary, we are going to assign values to $\{v_{ij}\}$ such that

- For any agent $2 \leq i \leq n$ and items $j \neq k$ $v_{ij} \neq v_{ik}$
- For any two agents $2 \leq i \neq j \leq n$, their favorite items $f_i \neq f_j$.
- For the remaining item r^* that no agent $2 \leq i \leq n$ values the highest, it's their second favorite.

If $\{v_{ij}\}$ satisfies the above conditions, then there is only one valid allocation, $\sigma_1 = r^*$ and for the rest agents, $\sigma_i = f_i$. So intuitively, the algorithm must find the favorite piece of each agent.

Theorem 4. *Any algorithm cannot output a valid allocation with less than $\frac{1}{2}(n-1)(n-2)$ queries.*

Proof Without loss of generality, we can fix $r^* = 1$ and tell the algorithm about it. Also, for each query, we will return the rank in that agent's view rather than the absolute value. These would both help the algorithm. I'm going to prove that the algorithm still needs $\frac{1}{2}(n-1)(n-2)$ queries to find f_is.

Consider a bipartite graph G with $V(G) = L \cup R$, where each node in L represents an agent and each node in R represents an item. Since the algorithm already knows that $r^* = 1$, we can assume that $|L| = |R| = n-1$. Each edge between agent i and item j represents that it's possible for $f_i = j$. Initially, each edge between L and R exists. A valid allocation exists if and only if there is a perfect matching, while an algorithm could successfully output a valid allocation if and only if there is a unique perfect matching.

Each time the algorithm queries v_{ij},

1. If v_{ij} is queried before, return the same answer.
2. If there is a perfect matching after deleting the edge (i, j), return a rank larger than 1, i.e., $f_i \neq j$, and delete the edge (i, j) from G.
3. Otherwise, return a rank 1, i.e., $f_i = j$.

It can be seen that in each step, the algorithm can delete at most one edge and it's guaranteed that a perfect matching exists. Thus we only need to bound the number of edges in a graph with unique perfect matching. Consider 2 edges in the perfect matching and the corresponding 4 vertices. For these 4 vertices, we can add at most one more edge, otherwise, there will be another perfect matching. Thus the graph can have no more than $(n-1) + \binom{n-1}{2} = \frac{1}{2}n(n-1)$ edges. Therefore, there should be at least $\frac{1}{2}(n-1)(n-2)$ queries. $\square$

The discrete case corresponds to the situation in the StarCore protocol after step 1. And I believe it somehow describes the difficulty of the envy-freeness towards a single agent. One future direction is to convert it into a continuous case and completely solve the star graph case.

3.3 4-path

After having this result, we were trying to find a polynomial protocol of local proportionality on the tree. As a start point, we were surprised that the 4-path case is already non-trivial. For convenience, I will call the points in path agent 1,2,3,4 in order. I give an algorithm for the 4-path case below and the algorithm can be easily extended to the 5-path case.

R	the remaining cake
$\mathcal{A}_i$	the ith allocation
$\mathcal{A}_{ij}$	agent j's part in the ith allocation
V_i	the valuation function of agent j
Π_{ijk}	$V_j(\mathcal{A}_{ij}) - V_j(\mathcal{A}_{ik})$

Table 1: Notations

Algorithm 2 4-Path-Core

```
Require: A 4-path $P$, and a cake.
Ensure: A locally envy-free allocation on $P$, and the remaining cake.
1: Agent 2 cuts the cake into 4 equally preferred pieces in his own measure.
2: Agent 1 chooses his favorite piece.
3: Agent 3 chooses two remaining pieces that she values the highest. She chops the higher ones so
   that she values them equally. Let agent 4 chooses his favorite one. The other piece is allocated to
   agent 3.
4: The last piece is allocated to 2.
5: return the allocation and the chopped part.
```

Algorithm 3 4-Path Protocol

```
Require: A 4-path $P$, and a cake.
Ensure: A locally envy-free allocation on $P$ with agent 2 dominating agent 3, and the remaining cake
   $R$.
1: Let $R$ be the entire cake initially.
2: for t from 1 to 2 do
3:     $\mathcal{A}_t, R \leftarrow$ 4-Path-Core$(P, R)$
4:     if $R = \emptyset$ or Agent 3 receives an insignificant piece then
5:         return the combination of $\mathcal{A}_t$, and $R$.
6: Let $S = \{\mathcal{A}_1, \mathcal{A}_2\}$.
7: Let $\mathcal{A}_p \in S$ be the allocation such that $\Pi_{p43}$ is the higher. $S = S \setminus \{\mathcal{A}_p\}$.
8: Let $\mathcal{A}_q \in S$ be the remaining allocation in $S$. Swap $\mathcal{A}_{q3}$ and $\mathcal{A}_{q4}$.
9: return the combination of modified $\mathcal{A}_t$ and $R$.
```

Correctness Proof There is only one chopped piece in each allocation of 4-Path-Core and can only be given to agent 3 or 4. The only case we need to be careful of is that agent 4 always gets it. Intuitively, as long as we can swap the chopped piece in some allocation from agent 4 to 3 while maintaining the locally envy-freeness, we're done.

First of all, agent 1 never envies other agents. Secondly, in 4-Path-Core, we can see that agent 3 values the pieces for agents 3 and 4 the same. Therefore, the swap will not make agent 3 envious. So we only need to deal with the envy-freeness of agent 4.

To deal with that, we should first see that envy-freeness leads to $\Pi_{p43} \geq \Pi_{q43} \geq 0$. If we swap the pieces of agents 3 and 4 in $\mathcal{A}_q$, the total advantageous of agent 4 will be $\Pi_{p43} - \Pi_{q43} \geq 0$, which means that agent 4 will not envy agent 3. □

Remark Regarding the Dominance Creating and Spreading framework in [3], they are trying to move an insignificant piece to one neighbor of the cutter. The pieces are exchanged along the shortest path. To maintain local proportionality of the points outside of the path, envy-freeness from the outside points to the points in the path is critical. That means a local envy-free partial allocation is not enough for this method. So although there is a partial envy-free protocol on the tree, it cannot be the driving horse of the whole protocol. We need a partial protocol with the stronger property as our core.

One sufficient condition in the tree case is that any point doesn't envy its parent's parent and the descendant nodes of its parent. Further, a sufficient condition in a general graph would be a similar condition on a layer diagram generated by the shortest path algorithm. This might be a future direction, but the complexity of the whole algorithm will probably remain exponential.

References

[1] Ariel D Procaccia. Thou shalt covet thy neighbor's cake. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[2] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–427. IEEE, 2016.

[3] Xiaohui Bei, Xiaoming Sun, Hao Wu, Jialin Zhang, Zhijie Zhang, and Wei Zi. Cake cutting on graphs: A discrete and bounded proportional protocol, 2019.

[4] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. In *SODA*, volume 6, pages 271–278, 2006.

[5] Krishnendu Chatterjee, Rupak Majumdar, and Thomas A. Henzinger. Markov decision processes with multiple objectives. In Bruno Durand and Wolfgang Thomas, editors, *STACS 2006*, pages 325–336, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[6] Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984.