Computing Optimal Equilibria in Repeated Games with Restarts

Ratip Emin Berker and **Vincent Conitzer**
Foundations of Cooperative AI Lab (FOCAL), Computer Science Department, Carnegie Mellon University
{rberker, conitzer}@cs.cmu.edu

Abstract

Infinitely repeated games can support cooperative outcomes that are not equilibria in the one-shot game. The idea is to make sure that any gains from deviating will be offset by retaliation in future rounds. However, this model of cooperation fails in anonymous settings with many strategic agents that interact in pairs. Here, a player can defect and then avoid penalization by immediately switching partners. In this paper, we focus on a specific set of equilibria that avoids this pitfall. In them, agents follow a designated sequence of actions, and *restart* if their opponent ever deviates. We show that the socially-optimal sequence of actions consists of an infinitely repeating goal value, preceded by a *hazing period*. We introduce an equivalence relation on sequences and prove that the computational problem of finding a representative from the optimal equivalence class is (weakly) NP-hard. Nevertheless, we present a pseudo-polynomial time dynamic program for this problem, as well as an integer linear program, and show they are efficient in practice. Lastly, we introduce a fully polynomial-time approximation scheme that outputs a hazing sequence with arbitrarily small approximation ratio.

1 Introduction

In social dilemmas, individual incentives hinder collective benefit: mutual cooperation is the best outcome for both players, but it is not a Nash equilibrium. Consider the symmetric two-player game in Table 1:

	D	C_1	C_2
D	4,4	11,0	14,0
C_1	0,11	5,5	0,0
C_2	0,14	0,0	8,8

Table 1: Payoffs of a two-player symmetric game (row, column)

Notice that for each player, D is the strictly dominant action, ensuring the maximum payoff against any fixed action of their opponent. This results in (D, D) being the only Nash equilibrium. In the infinitely repeated version of this game, however, mutual cooperation can be achieved: Consider the grim-trigger strategy where both players agree to play C_2 in every round, but if their opponent defects, they switch to playing D in all future rounds. During the game, a player can increase their payoff in a round by at most $14 - 8 = 6$ by defecting, but their payoff in all future rounds is now bound above by 4, as their opponent will switch to D, resulting in a per-round net loss of at least $8 - 4 = 4$ compared to if they had stuck to C_2. If the players value future rounds sufficiently (i.e., have a 'discount factor' close to 1, assumed true for the rest of this section), the gain of defecting (6, once) will be offset by loss due to the opponent retaliating (4, all future rounds), resulting in neither of the players defecting from C_2.

This model of cooperation, however, fails in anonymous settings with many players, in which players can simply find a new partner to play with. This setting can model any situation that involves two-player interactions within a larger pool, such as monogamous relationships, employer-employee interactions, and two-person research collaborations. Critically, each of these 'partnerships' can last for arbitrarily many rounds, but can be terminated at any point by one of the partners, who can then find themselves a new partner in the larger pool. In this game-theoretic setting, a player can defect and avoid retaliation, if they are able to switch partners immediately following the defection. If there is no way for a player to check their partner's history (that is, players are *anonymous*), this setting may result in the emergence of 'serial defectors,' who perpetually defect on a partner and move on the next, even if all their partners would follow the 'grim trigger' strategy if the relationship continued. This is especially relevant for settings where the players are AI agents (such as trading bots), who might more easily conceal their identity compared to traditional human players.

To avoid this pitfall, we turn to a specific type of equilibria in such infinitely repeated two-player games, those with *restarts*. Instead of the grim-trigger strategy of 'defect forever once the opponent deviates', consider instead a strategy profile where all players agree on a planned sequence of actions to follow, and they restart the same sequence with a new partner if their opponent ever deviates from it. (In the context of our paper, the punishment strategy of simply leaving the relationship, thereby forcing the partner to restart as well, is in fact without loss of generality: no other punishment strategy could be more effective, because if it were more effective,

the partner would simply leave the relationship.)[1]

For instance, going back to the game in Table 1, say the two players agree on the sequence $(D, C_1, C_2, C_2, C_2, \ldots)$. If the players stick to the plan, the sequence of payoffs they receive is $(4, 5, 8, 8, 8, \ldots)$. Neither will deviate from the plan in the first round (D), as doing so can only lower their payoff. In the second round (C_1), deviating brings a payoff of 11, but results in the opponent ending the relationship and having to go back to the start of the sequence with someone else. This results in a per-round average payoff of $(4 + 11)/2 = 7.5$ for the deviating player, which is less than the per-round payoff of 8 she can eventually receive by sticking to the planned sequence. Similarly, once the players get to the (C_2) portion of their sequence, any deviation can bring at most a payoff of 14 (an increase of 6), but results in a restart where the next two rounds (4,5) does a total damage of $(8-4)+(8-5) = 7$ when compared to sticking with the sequence, making the overall sequence stable. Conceptually, the planned sequence consists of a 'hazing' period (D, C_1), followed by lasting socially-optimal cooperation $(C_2, C_2, ...)$.

Not every planned sequence is stable: an alternative plan $(C_1, C_2, C_2, C_2, \ldots)$ would incentivize repeatedly deviating on the first step, to obtain 11 per round. Another alternative, $(D, C_2, C_2, C_2, \ldots)$, would see a player repeatedly deviating on the second step, ensuring a per round utility of $(4 + 14)/2 = 9 > 8$. Both of these alternatives suffer from under-hazing, as the cost of a restart for a defecting player fails to offset the gains from defection.

$(D, C_1, C_2, C_2, \ldots)$ is not the only stable sequence; so is $(D, C_1, C_1, \ldots, C_1, C_2, C_2, \ldots)$, for an arbitrarily large (positive) number of C_1s. However, this results in unnecessarily delaying the socially optimal outcome of C_2, i.e., over-hazing. Hence, in this paper we ask: how can we optimize the payoff of a planned sequence, while ensuring its stability?

1.1 Related Work

Repeated games (without restarts) have long been of interest to the AI community. In contrast to one-shot games, they introduce a temporal component to the game, and they allow modeling settings where cooperation can be sustained thanks to the threat of deviators being punished in future rounds. This is now of particular interest in the context of the nascent research area of *cooperative AI* [Dafoe *et al.*, 2021], especially for game-theoretic approaches to that [Conitzer and Oesterheld, 2023]. Thanks to the folk theorem, they also allow for more efficient computation of Nash equilibria than one-shot games, as was observed by [Littman and Stone, 2005] for two players. With three or more players the problem becomes hard again [Borgs *et al.*, 2010], though in practice it can often be solved fast [Andersen and Conitzer, 2013] and if correlated punishment is allowed then the problem becomes easy again [Kontogiannis and Spirakis, 2008].

Separately, the role of *anonymity* in game theory has long been of interest to the AI community. Perhaps most significantly, in mechanism design, there is a long line of work on *false-name-proof* mechanisms, in which agents cannot benefit from participating multiple times using fake identifiers [Yokoo *et al.*, 2001; Yokoo *et al.*, 2004; Conitzer and Yokoo, 2010]. This is conceptually related to the work in this paper, insofar as an agent that restarts with a different partner makes use of a degree of anonymity in the system. However, in our context an agent does not use multiple identifiers simultaneously, and the work seems technically quite distinct.

Cooperation in repeated games with the possibility of rematching with a new partner has been studied in the economic theory community, similarly identifying the importance of building up relations gradually. Our work differs from this literature in that we focus on the computational problem of optimizing the equilibrium, for arbitrary games, whereas the economics literature focuses on obtaining characterization results in specific settings such as lending and borrowing games [Datta, 1996; Wei, 2019], prisoner's dilemma [Fujiwara-Greve and Okuno-Fujiwara, 2009; Rob and Yang, 2010; Izquierdo *et al.*, 2014], and environments with agents of multiple types [Kranton, 1996; Ghosh and Ray, 1996; Rob and Yang, 2010].

1.2 Overview

In Section 2, we introduce the concepts and notation. In Section 3, we introduce optimal sequences and prove their various properties. In Section 4, we define an equivalence relation on sequences based on their total discounted utility with high discount factor. In Section 5, we formalize the computational problem of computing a representative of the optimal equivalence class of this relation. As our main results, we prove an NP-hardness result and present three algorithms for the problem: a pseudo-polynomial time dynamic program and an integer linear program that exactly solves it, as well as a fully polynomial time approximation scheme. In Section 6, we report runtimes from our experiments with these algorithms. We present directions for future research in Section 7.

2 Preliminaries

In this work, we restrict our attention to *symmetric games and strategies*, which allows us to condense the representation of a game and thereby simplify the presentation. We discuss moving beyond symmetry in Section 7.

2.1 Problem Instance

Given a two-player finite, symmetric normal form game Γ with actions $A = \{a^{(1)}, ..., a^{(n)}\}$ and integer payoffs, we define:

- The *cooperative payoff* function $p : A \to \mathbb{Z}$, which maps $a^{(j)}$ to the payoff that the two players receive in Γ if they both play $a^{(j)}$ for each $j \in [n]$.
- The *deviation payoff* function $p^* : A \to \mathbb{Z}$, which maps each $a^{(j)}$ to the max. payoff a player can achieve given they play any $A \setminus \{a^{(j)}\}$ and their opponent plays $a^{(j)}$.
- The *discount factor* $\beta \in [0, 1)$, such that if a player receives payoff p_i in round $i \in \mathbb{N}$, her total discounted

[1] The type of equilibrium we study remains an equilibrium in the standard repeated game setting (without restarts), and so is a *refinement* of the traditional concept. In this case, if one's partner deviates, one continues with the same partner but plays *as if* one were starting from the beginning, which is sufficient to deter deviation.

utility is $\sum_{i=0}^{\infty} \beta^i p_i$. To avoid confusion, we use subscript $i \in \mathbb{N} = \{0, 1, \ldots\}$ to iterate over the rounds, and superscript $j \in [n] = \{1, \ldots, n\}$ to iterate over actions.

Thus any finite symmetric game Γ can for our purposes be represented as $G = (\{(p^{(j)}, p^{*(j)})\}_{j\in[n]}, \beta)$, where $p^{(j)} := p(a^{(j)})$ and $p^{*(j)} := p^*(a^{(j)})$. For instance, the game in Table 1 is represented as $G_1 = (\{(4, 0), (5, 11), (8, 14)\}, \beta)$.

2.2 Equilibria with Restarts

In this paper, we focus on strategy profiles $\boldsymbol{\sigma}$ where each player plans to follow the sequence of moves $(a_0, a_1, ...) \in A^{\mathbb{N}}$, and will restart the sequence if the other player deviates from it. We define $p_i := p(a_i)$ and $p_i^* := p^*(a_i)$ for all $i \in \mathbb{N}$. Two forever-cooperating agents will achieve total discounted utility $\sum_{i=0}^{\infty} \beta^i p_i$, whereas an agent deviating on round k will achieve total discounted utility $\sum_{i=0}^{k-1} \beta^i p_i + \beta^k p_k^* + \sum_{i=0}^{\infty} \beta^{i+k+1} p_i$.[2] Accordingly, for $\boldsymbol{\sigma}$ to be stable (i.e., a Nash equilibrium), we need (for all $k \in \mathbb{N}$):

$$\sum_{i=0}^{k-1} \beta^i p_i + \beta^k p_k^* \leq (1 - \beta^{k+1}) \sum_{i=0}^{\infty} \beta^i p_i \qquad (1)$$

Since $\boldsymbol{\sigma}$ is completely defined by the planned sequence, we can succinctly represent it using its corresponding sequence of payoffs $(p_i)_{i\in\mathbb{N}} \in \{p^{(j)}\}_{j\in[n]}^{\mathbb{N}}$ and $(p_i^*)_{i\in\mathbb{N}} \in \{p^{*(j)}\}_{j\in[n]}^{\mathbb{N}}$.

3 Optimal Sequences

Given a game instance G, there can be infinitely many stable sequences $(p_i)_{i\in\mathbb{N}}$. For instance, given G_1, our representation of the game in Table 1, both of the following sequences are stable for $\beta = 0.9$, as they fulfill Equation (1) for all $k \in \mathbb{N}$:

$$p_i^a = \begin{cases} 4 & \text{if } i \leq 10 \\ 5 & \text{otherwise} \end{cases} \quad \text{and} \quad p_i^b = \begin{cases} 4 & \text{if } i \leq 3 \\ 8 & \text{if } i > 3, i \text{ even} \\ 5 & \text{otherwise} \end{cases}$$

The multitude of stable sequences for a given game raises two questions: how can we (i) determine which sequences are more desirable than others, and (ii) compute the most desirable sequences? The answer to (i) is straightforward in our context of symmetric games, where the payoffs of two cooperating players are the same. Thus, one can focus on maximizing total discounted utility without fairness concerns. For instance, $\sum_{i=0}^{\infty} \beta^i p_i^a \approx 43.1$ and $\sum_{i=0}^{\infty} \beta^i p_i^b \approx 56.9$, indicating $(p_i^b)_{i\in\mathbb{N}}$ to be 'the better plan'. More generally, we define:

Definition 1 (Optimal sequence)**.** Given any $G = (\{(p^{(j)}, p^{*(j)})\}_{j\in[n]}, \beta)$, a sequence of payoffs $(p_i)_{i\in\mathbb{N}}$ **surpasses** sequence $(p'_i)_{i\in\mathbb{N}}$ if $\sum_{i=0}^{\infty} \beta^i p_i > \sum_{i=0}^{\infty} \beta^i p'_i$. A sequence $(p_i)_{i\in\mathbb{N}}$ is an **optimal sequence** if:

1. it is stable, i.e., satisfies Equation (1) for all $k \in \mathbb{N}$, and
2. it is not surpassed by any other other stable sequence.

We first prove an existence result:

[2]Note that this assumes the deviating agent never deviates again; this is WLOG, since deviating repeatedly in this way is an improvement if and only if deviating once in this way is an improvement.

Proposition 1. *For any $G = (\{(p^{(j)}, p^{*(j)})\}_{j\in[n]}, \beta)$, if there is any stable sequence, then an optimal sequence exists. This sequence is not necessarily unique.*

Proof. The proof follows from the following claim. To save space, we present the proof of the claim as well as an example for non-uniqueness in Appendix A.1 of the full version of the paper.

Claim 1. Given any instance G, the set of achieveable total utilities is a closed set, where we say total $t \in \mathbb{R}$ is achievable if there exists a stable sequence $(p_i)_{i\in\mathbb{N}}$ with $\sum_{i=0}^{\infty} \beta^i p_i = t$.

Since the set of achievable total discounted utilities is closed and bounded, it contains its supremum [Rudin, 1976, Thm. 2.28], proving an optimal stable sequence exists. □

Having proven the existence of optimal sequences, we now present two lemmas about their properties.

Lemma 1. *Any optimal sequence will reach a step after which a single payoff (the highest in the sequence) is repeated forever. We call this payoff the **goal value** of the sequence.*

Intuitively, Lemma 1 implies that any optimal sequence consists of an infinitely repeating goal value, the highest payoff, as well as a preceding *hazing period*, a finite sequence of non-maximal payoffs. These two stages are interdependent: an agent cooperates during the hazing period due to the promise of the goal value, whereas any defection after reaching the goal value is avoided by the threat of facing the hazing period again.

Lemma 2. *For large enough β, the goal value of any optimal sequence is $p_\Omega := \max_{j\in[n]} p^{(j)}$.*

Together, Lemmas 1 and 2 (proven in Appendix A.2) allow easily ruling out many sequences as non-optimal for sufficiently high values of β, including $(p_i^a)_{i\in\mathbb{N}}$ and $(p_i^b)_{i\in\mathbb{N}}$ from above, the latter since it never converges to a goal value, and the former since it converges to one that is not the highest in the game. As we have seen in Lemma 2, the large β setting achieves the largest goal values, and accordingly, the largest gap between achieving cooperation and failing to do so, adding to the significance of computing stable and optimal sequences. Indeed, in much of the literature on repeated games in general (including folk theorems), the focus is on the limit case where $\beta \to 1$. Accordingly, in the next section, we study the optimality of sequences in the $\beta \to 1$ limit.

4 Limit-Utility Equivalence Classes

We say that a sequence $(p_i)_{i\in\mathbb{N}}$ is stable in the $\beta \to 1$ limit if there exists a β' such that $(p_i)_{i\in\mathbb{N}}$ is stable for all $\beta > \beta'$. Of course, as $\beta \to 1$, the total discounted utility diverges. Hence, we now introduce an equivalence relation that allows us to compare the total discounted utilities in this limit:

Definition 2 (Limit-utility equivalence)**.** Given a game $G = \{(p^{(j)}, p^{*(j)})\}_{j\in[n]}$, two stable sequences $(p_i)_{i\in\mathbb{N}}$ and $(p'_i)_{i\in\mathbb{N}}$ are **limit-utility equivalent** if $\lim_{\beta\to 1} \sum_{i=0}^{\infty} \beta^i (p_i - p'_i) = 0$.

Indeed, this relationship is symmetric, reflexive, and transitive. We now introduce the optimal equivalence class:

Proposition 2. *There exists a well-defined optimal limit-utility equivalence class: that is, an equivalence class such that for any member sequence $(p_i)_{i\in\mathbb{N}}$ and any other sequence $(p'_i)_{i\in\mathbb{N}}$, we have $\lim_{\beta\to 1}\sum_{i=0}^{\infty}\beta^i(p_i - p'_i) \geq 0$.*

Proof. By Lemmas 1 and 2, any $(p_i)_{i\in\mathbb{N}}$ that converges to $p_\Omega = \max_{j\in[n]} p^{(j)}$ will surpass any $(p'_i)_{i\in\mathbb{N}}$ that does not, implying the equivalence class of the latter cannot be optimal. Ruling such sequences out allows us to define for any sequence of payoffs $(p_i)_{i\in\mathbb{N}}$ a corresponding sequence of hazings $(h_i)_{i\in N}$, where $h_i = p_\Omega - p_i$. For any two sequences that converge to p_Ω, we have $\lim_{\beta\to 1}\sum_{i=0}^{\infty}\beta^i(p_i - p'_i) = \sum_{i=0}^{\infty} h'_i - \sum_{i=0}^{\infty} h_i$, where neither of the sums diverge since $h_i \neq 0$ or $h'_i \neq 0$ for only finitely many i. Hence, the equivalence class of a sequence with goal value p_Ω is entirely determined by the (non-discounted) sum of its per-round hazings. By Lemma 2, there exists at least one such stable sequence $(h_i)_{i\in\mathbb{N}}$, say with total hazing $H = \sum_{i=0}^{\infty} h_i$. Since the total non-discounted hazing for any sequence is an integer and bounded below by 0, there can only be finitely many improvements over H, implying that there is a well-defined minimum total hazing H_{min} with at least one corresponding stable sequence. Hence, any stable sequence with total hazing H_{min} cannot be surpassed (as $\beta \to 1$) by any other stable sequence, either converging to p_Ω or not, proving its equivalence class is optimal. □

5 Computational Problem: OptRep

Having proven the existence of the optimal equivalence class, the natural next question becomes: how do we compute a (representative) sequence from this class? Before formalizing this as computational problem, we investigate the stability condition given in Equation (1) in the $\beta \to 1$ limit:

Proposition 3. *A sequence $(p_i)_{i\in\mathbb{N}}$ with goal value $p_\Omega = \max_{j\in[n]} p^{(j)}$ is stable in the $\beta \to 1$ limit if and only if the following holds for all $k \in \mathbb{N}$:*

$$\sum_{i=0}^{k-1}(p_\Omega - p_i) > p^*_k - p_\Omega \tag{2}$$

The proof is given in Appendix A.3. Intuitively, the left-hand side of (2) is the cost of restarting, as repeating each step i results in a loss of $p_\Omega - p_i$ (the so-called hazing cost) compared to the goal value, whereas the right-hand side is the gains from deviating, represented by the advantage over the goal value one can gain by deviating now. The inequality needs to be strict because, intuitively, for any $\beta < 1$, a "tie" between the two sides of the equation would be broken towards the side of deviation, as that deviation payoff comes earlier.

Motivated by this way of expressing the stability condition, we modify the representation of a game to fit our problem: Given $G = \{(p^{(j)}, p^{*(j)})\}_{j\in[n]}$ and $p_\Omega = \max_{j\in[n]} p^{(j)}$, for each $j \in [n]$ we define a corresponding ***hazing cost*** $h^{(j)} = p_\Omega - p^{(j)}$ and a ***threshold*** $t^{(j)} = p^{*(j)} - p_\Omega$. We now represent the game as $G = \{(h^{(j)}, t^{(j)})\}_{j\in[n]}$ and any sequence of payoffs $(p_i)_{i\in\mathbb{N}}$ with the corresponding sequence of hazing costs $(h_i)_{i\in\mathbb{N}}$. The stability condition from (2) then becomes:

$$\sum_{i=0}^{k-1} h_i > t_k \tag{3}$$

As shown in the proof of Prop 2, all members of the optimal limit-utility equivalence class necessarily have p_Ω as their goal value, and hence $h_i \neq 0$ for finitely many i, and the equivalence class of any such sequences is entirely determined by $\sum_{i=0}^{\infty} h_i$. Any such stable sequence must satisfy $\sum_{i=0}^{\infty} h_i > p^*_\Omega - p_\Omega \equiv \Delta$, to ensure (3) is fulfilled for the steps after reaching the goal value. Thus, the goal of finding a representative of the optimal equivalence class reduces to:

Definition 3 (OptRep)**.** Given $\{(h^{(j)}, t^{(j)})\}_{j\in[n]} \subset \mathbb{Z}_+ \times \mathbb{Z}$ and $\Delta \in \mathbb{Z}_+$, **OptRep** asks to find a finite sequence $(h_i)_{i\in\{0\}\cup[\ell]} \in \{h^{(j)}\}_{j\in[n]}^{\ell+1}$ for some $\ell \in \mathbb{N}$ such that $\sum_{i=0}^{\ell} h_i$ is minimized, subject to:

$$(\forall k \in \{0\} \cup [\ell]) : \textstyle\sum_{i=0}^{k-1} h_i > t_k \textbf{ and } \sum_{i=0}^{\ell} h_i > \Delta$$

We now prove a hardness result for OptRep:

Theorem 1. *OptRep is (weakly) NP-hard, and the corresponding decision problem is NP-complete.*

Proof. We will prove the theorem by reducing the Unbounded Subset-Sum Problem (USSP) to OptRep. USSP is NP-complete [Lueker, 1975] and asks:

Given positive integers $\{a_i\}_{i\in[m]}$ and A, do there exist non-negative integers $\{r_i\}_{i\in[m]}$ such that $\sum_{i=1}^{m} r_i \cdot a_i = A$?

We now present our reduction: For each $j \in [m]$, define $h^{(j)} = a_j$ and $t^{(j)} = -1$ (in the payoff representation, this simply implies $p^{*(j)} = p_\Omega - 1$). Lastly, set $\Delta = A - 1$. We claim that the answer to the Unbounded Subset-Sum Problem is yes if and only if the total hazing of the optimal sequence computed by OptRep on input $G = (\{(h^{(j)}, t^{(j)})\}_{j\in[m]}, \Delta)$ is A. The backwards direction is obvious. For the forward direction, say that the answer to USSP is yes, with coefficents $(r_j)_{j\in[m]}$. We can construct a finite hazing sequence $(h_i)_{i\in\{0\}\cup[R]}$ that has each $h^{(j)}$ repeating r_j times, where $R = \sum_{j=1}^{m} r_j$. Notice that (3) is fulfilled since $t_i = -1$ for all $i \in \{0\} \cup [R]$. As for the final hazing threshold, we have $\sum_{i=0}^{R} h_i = \sum_{j=1}^{m} r_j a_j = A = \Delta + 1 > \Delta$, so the overall sequence is stable. Since the total hazing will be integral, any sequence with less total hazing will fail to meet the threshold set by Δ. Hence, the total hazing of the optimal sequence computed by OptRep on input $G = (\{(h^{(j)}, t^{(j)})\}_{j\in m}, \Delta)$ is A.

The reduction proves that OptRep is NP-hard. The corresponding decision problem (of whether a stable sequence with total hazing H in the $\beta \to 1$ limit exists) is NP-complete, since, as we will see in Lemma 3, every sequence has a polynomial-size representation that does not change its total hazing cost, which can be computed in polynomial time. □

As shown by [Wojtczak, 2018, Theorem 4], USSP becomes *strongly* NP-complete when the inputs are rational numbers rather than integers. Due to our reduction above, the same result readily translates to OptRep:

Corollary 1. *OptRep with rational inputs is strongly NP-hard, and the corresponding decision problem is strongly NP-complete.*

Considering the similarities between OptRep and USSP presented in the proof of Theorem 1, one might wonder if a reduction from OptRep to USSP is also possible. However, OptRep has additional challenges: instead of a set of items, we need to pick a sequence and ensure that every step of it meets the requirement in (3), as opposed to only a final capacity requirement.

Despite these challenges, we can exploit the structural properties of OptRep to develop techniques similar to those employed for knapsack problem variants such as USSP. One key observation is that the stability of a given sequence at any step solely depends on the hazing that has already occurred. This enables us to solve the problem with a dynamic program with a single state variable (the hazing so far), leading to:

Theorem 2. *OptRep is solvable in pseudo-polynomial time.*

Proof. We prove the theorem by showing the correctness, time complexity, and space complexity of Algorithm 1.
(a) Correctness: Due to integral hazing costs, the minimum hazing cost for any stable sequence is $\Delta + 1$. We prove (by strong induction on decreasing h) that once Algorithm 1 is complete, $D[h]$ contains the minimum overshoot over this lower bound one can achieve starting from a hazing cost of h. As a base case, for any $h \geq \Delta + 1$, the hazing goal has been met and the best we can do is an overshoot of $h - \Delta - 1$ (Line 4). For any $h < \Delta + 1$, at least one more action needs to be added to meet the hazing goal. In Line 7, the algorithm chooses the next eligible action that minimizes the overshoot, where action $j \in [n]$ is eligible if the hazing so far, h, has met its threshold, $t^{(j)}$. Inductively assuming that $D[h']$ is correct for all $h' > h$ ensures that $D[h]$ is correctly set. Accordingly, $D[0]$ gives the overshoot of a sequence from the optimal equivalence class, and the list A keeps track of the next steps to later reconstruct the sequence.
(b) Time complexity: Line 4 takes constant time and is implemented $h_{\max} = \max_{j\in[n]} h^{(j)}$ times. Lines 5-8 take $O(n)$ time and are executed $O(\Delta)$ times. Since the sequence returned has at most $\Delta + 1$ actions, Lines 10-12 take $O(\Delta)$ time. All other steps take constant time. Overall, the algorithm runs in $O(n\Delta + h_{max})$ time, which can be improved to $O(n\Delta)$ by feeding Line 4 into the definition of $D[h]$ in Lines 5-8. This is polynomial in the input but, since input is represented in binary, exponential in the problem size.
(c) Space complexity: D and A both have Δ variable entries, giving the algorithm a space complexity of $O(\Delta)$. If we are not interested in reconstructing the representative sequence, the space complexity improves to $O(\min\{h_{max}, \Delta\})$, since we may only store the most recent h_{max} entries of D, which are sufficient for Line 7. □

Alternatively, one can formulate OptRep as an Integer Linear Program (ILP), using binary variables for whether action (j) is the ith action in the sequence. But the hazing period can have as many as $\Delta + 1$ actions, resulting in $O(n\Delta)$ variables, exponentially many. Instead, we ask if one can impose some structure on the output sequence without losing generality.

Algorithm 1 Dynamic Program for OptRep

Input: Final hazing goal: $\Delta > 0$, tuples of (hazing, thresholds) values for each action: $\{(h^{(j)}, t^{(j)})\}_{j\in[n]} \subset \mathbb{Z}_+ \times \mathbb{Z}$
Output: a list S_o representing a sequence from the optimal equivalence class, along its hazing cost $H_o \in \mathbb{Z}_+$.

1: Let $\ell = \Delta + \max_{j\in[n]} h^{(j)}$.
2: $D \leftarrow [\mathit{inf}]^\ell, \quad A \leftarrow [\mathit{inf}]^\ell$
3: **for** $h = \ell, \ell - 1, \ldots, 1, 0$ **do**
4: **if** $h \geq \Delta + 1$ **then** $D[h] \leftarrow h - \Delta - 1$
5: **else**
6: $S_h \leftarrow \{j \in [n] : h > t^{(j)}\}$
7: $D[h] \leftarrow \min_{j\in S_h} D[h + h^{(j)}]$
8: $A[h] \leftarrow \arg\min_{j\in S_h} D[h + h^{(j)}]$
9: $H_o \leftarrow D[0] + \Delta + 1, \quad S_o \leftarrow [], \quad i \leftarrow 0$
10: **while** $i \neq \mathit{inf}$ **do**
11: $S_o \leftarrow S_o + [A[i]], \quad i \leftarrow A[i]$
12: **end while**
13: **return** S_o, H_o

One intuitive candidate is monotonicity: does the optimal equivalence class always have a sequence with non-increasing hazing costs (non-decreasing payoffs), considering the goal value is the highest payoff? The answer, it turns out, is no. Consider: $\Delta = 10$ with $(h^{(1)}, t^{(1)}) = (5, -1), (h^{(2)}, t^{(2)}) = (6, 4)$. The sequence $(h^{(1)}, h^{(2)})$ is stable and achieves a hazing of 11. The minimum hazing from any non-increasing sequence is 15, by $(h^{(1)}, h^{(1)}, h^{(1)})$.

There is, however, a property that can be imposed on the sequences without ruling out the optimal equivalence class:

Lemma 3. *Any stable sequence of the optimal equivalence class can be converted to a stable sequence in the same class where (a) all appearances of any action are adjacent and (b) actions appear in order $j_1, j_2, ..., j_\ell$ where $t^{(j_1)} \leq t^{(j_2)} \leq \ldots \leq t^{(j_\ell)}$. We call such sequences **threshold-monotonic**.*

Proof. For a given action j, say i_j is its first appearance in the sequence. Move all appearances of $h^{(j)}$ to appear immediately after i_j, pushing all the actions that were previously there to later steps, without changing their order. The total hazing (and hence the equivalence class) has not changed. The stability of all $h^{(j)}$ follows from its stability at step i_j: once its threshold is met, this action will always be stable. All other actions are still stable since the hazing that precede them could have only increased. Repeating this for every $j \in [n]$ achieves (a).

Say the actions now appear in order $j_1, j_2, \ldots, j_n$. Assume the first k of them are the lowest k threshold actions (with the base case $k = 0$). Then the action with $(k+1)^{\text{th}}$ lowest threshold is j_z for some $z \in \{k+1, \ldots n\}$. Move every occurrence of action j_z to the starting point of action j_{k+1}, sliding every other action to the right. The occurrences of action j_z will be stable by the stability of j_{k+1} prior to the shift, which has an equal or greater threshold. All other actions are stable as their preceding hazing can only increase. Repeating this inductively results in the overall sequence fulfilling (b). □

Algorithm 2 Integer Linear Program for OptRep

Input: Final hazing threshold: $\Delta > 0$, tuples of (hazing, thresholds) values for each action: $\{(h^{(j)}, t^{(j)})\}_{j\in[n]} \subset \mathbb{Z}_+ \times \mathbb{Z}$, with $t^{(1)} \leq t^{(2)} \leq \ldots \leq t^{(n)} \leq \Delta$

Parameters: $(\forall j \in [n])$: $r^{(j)}$ indicating how many times action j is repeated in the output sequence

Output: Final values of $[r^{(j)}]_{j\in[n]}$

1: **if** $t^{(1)} \geq 0$ **then raise** *error*
2: **minimize** $\sum_{j=1}^{n} r^{(j)} h^{(j)}$
3: **subject to**
4: $\quad (\forall j \in [n] \setminus \{1\}) \sum_{j'=1}^{j-1} r^{(j')} h^{(j')} \geq t^{(j)} + 1$
5: $\quad \sum_{j=1}^{n} r^{(j)} h^{(j)} \geq \Delta + 1$
6: **return** $[r^{(j)}]_{j\in[n]}$

By Lemma 3, we can restrict our attention to threshold-monotonic sequences while solving OptRep, leading to an ILP with $O(n)$ variables, presented in Algorithm 2. Note that ordering actions by thresholds takes an additional $O(n \log n)$ preprocessing time. The assumption $t^{(n)} \leq \Delta$ is WLOG, since any action j with $t^{(j)} \geq \Delta + 1$ cannot be used until the final hazing threshold is met. The ILP returns $[r^{(j)}]_{j\in[n]}$, the number of times each action is repeated, as this is sufficient to represent a threshold-monotonic sequence. As seen in our experiments in Section 6, Algo. 2 is efficient in practice.

Lemma 3 can also improve the space complexity of Algorithm 1: each $A[h]$ can now store a size-n vector corresponding to $[r_j]_{j\in[n]}$, with Line 8 replaced with $A[h][j^*] \leftarrow A[h][j^*] + 1$ where $j^* = \arg\min_{j\in S_h} D[h + h^{(j)}]$. Thus, it now suffices to store only h_{max} entries of A and D. Call Algorithm 1 with this modification (as well as those mentioned in the proof of Theorem 2(b-c)) Algorithm 1^*. This leads to the below strengthening of Theorem 2:

Theorem 3. *Algorithm* 1^* *solves OptRep in* $O(n\Delta)$ *time using* $O(\min\{\Delta, n \cdot h_{max}\})$ *space. The space complexity is* $O(\min\{\Delta, h_{max}\})$ *for the corresponding decision problem.*

While efficient in practice, both Algorithms 1 and 2 are exponential in the worst case, which motivates the question of whether OptRep can be approximated with a Fully Polynomial Time Approximation Scheme (FPTAS), which runs in polynomial in the problem size and in $1/\varepsilon$, where ε is the approximation ratio, given as an input. While we have given a pseudopolynomial-time algorithm for our problem, doing so in general does not guarantee the existence of an FPTAS: for instance, knapsack with multiple constraints is pseudo-polynomial-time solvable but does not have an FPTAS unless $P = NP$ [Magazine and Chern, 1984]. However, for OptRep there is in fact an FPTAS, given as Algorithm 3. The algorithm is a modification of Ibarra and Kim's original FPTAS for unbounded knapsack [1975], with additions that address the differences between OptRep and unbounded knapsack.

Theorem 4. *Algorithm 3 is an FPTAS for OptRep, and runs in* $O(n \log n + \frac{n}{\varepsilon^2})$ *time.*

Proof. (a) Correctness: To prove correctness, we first present several claims, the proofs of which are in Appendix A.4

Algorithm 3 FPTAS for OptRep

Input: Final hazing threshold: $\Delta > 0$, tuples of (hazing, thresholds) values for each action: $\{(h^{(j)}, t^{(j)})\}_{j\in[n]} \subset \mathbb{Z}_+ \times \mathbb{Z}$, with $t^{(1)} \leq t^{(2)} \leq \ldots \leq t^{(n)} \leq \Delta$

Output: $\hat{H}, \hat{L}$, where $\hat{L}$ is a list of actions representing the output sequence, and $\hat{H}$ is the corresponding hazing cost.

1: **if** $t^{(1)} \geq 0$ **then raise** *error*
2: $F \leftarrow \{j \in [n] : t_i < 0, h_i \leq \Delta\}$
3: **if** $F = \emptyset$ **then**
4: $\quad \hat{L} \leftarrow [\arg\min_{j\in[n]:t_i<0} h^{(j)}], \quad \hat{H} \leftarrow h^{(\hat{L}[0])}$
5: $\quad$ **return** $\hat{H}, \hat{L}$
6: $j^* \leftarrow \min F, \quad \tilde{H} \leftarrow k \cdot h^{(j^*)}$ where k is the min. integer s.t. $k \cdot h^{(j^*)} > \Delta$
7: $\delta \leftarrow \tilde{H}\left(\frac{\varepsilon}{3}\right)^2, g \leftarrow \left\lfloor \frac{\tilde{H}}{\delta} \right\rfloor = \left\lfloor \left(\frac{3}{\varepsilon}\right)^2 \right\rfloor, S \leftarrow \emptyset$
8: $T \leftarrow [\mathit{inf}]^{g+1}, L \leftarrow [\mathit{inf}]^{g+1}, T[0] \leftarrow 0, L[0] \leftarrow [\,]$
9: **for** $j = 1, \ldots, n$ **do**
10: $\quad$ **if** $h^{(j)} \leq \frac{\varepsilon}{3}\tilde{H}$ **then** $S \leftarrow S + [j]$
11: $\quad$ **else**
12: $\quad\quad f^{(j)} \leftarrow \left\lfloor \frac{h^{(j)}}{\delta} \right\rfloor$
13: $\quad\quad$ **for** $k = 0, ..., g - f^{(j)}$ **do**
14: $\quad\quad\quad$ **if** $T[k] \neq \mathit{inf}$ **and** $T[k] > t^{(j)}$ **and** $(T[k + f^{(j)}] = \mathit{inf}$ **or** $T[k] + h^{(j)} > T[k + f^{(j)}])$ **then**
15: $\quad\quad\quad\quad T[k + f^{(j)}] \leftarrow T[k] + h^{(j)}$
16: $\quad\quad\quad\quad L[k + f^{(j)}] \leftarrow L[k] + [j]$
17: $j_s \leftarrow \arg\min_{j\in S} t^{(j)}$
18: $k^* \leftarrow \arg\min_{k\in\{0\}\cup[g]:T[k]>t^{(j_s)}} T[k] + n_k \cdot h^{(j_s)}$ where n_k is the min. integer s.t. $T[k] + n_k \cdot h^{(j_s)} > \Delta$
19: $\hat{H} \leftarrow T[k^*] + n_{k^*} \cdot h^{(j_s)}, \quad \hat{L} \leftarrow L[k^*] + [j_s]^{n_{k^*}}$
20: **return** $\hat{H}, \hat{L}$

Claim 2. Say H^* is the optimal hazing. If the algo. returns on Line 5, $\hat{H} = H^*$. Else, $\tilde{H} \geq H^* \geq \frac{1}{2}\tilde{H}$ after Line 6.

Note that Line 7 sets a 'normalizing factor' of $\delta = \tilde{H}\left(\frac{\varepsilon}{3}\right)^2$, later used for computing normalized hazing costs $f^{(j)} = \left\lfloor h^{(j)}/\delta \right\rfloor$ and a normalized upper bound for the optimal hazing $g = \lfloor \tilde{H}/\delta \rfloor$.

Claim 3. Any $f^{(j)}$ set on Line 12 satisfies $f^{(j)} \cdot \delta \leq h^{(j)} \leq f^{(j)} \cdot \delta \cdot \left(1 + \frac{\varepsilon}{2}\right)$

Claim 4. At any point during the execution starting from Line 8, for any $k \in \{0, ..., g\}$, either $T[k] = L[k] = \mathit{inf}$ or $L[k]$ contains a list $[j_0, j_1, \ldots, j_\ell]$ with $\sum_{i=0}^{\ell} f^{(j_i)} = k$ and $\sum_{i=0}^{\ell} h^{(j_i)} = T[k]$.

Claim 5. Say $(h_i)_{i\in\{0\}\cup[\ell]}$ is any stable threshold-monotonic *sub*sequence (i.e., it fulfills (3) for all $k \leq \ell$, but does not necessarily meet the final hazing goal Δ), where the order of actions is consistent with the labeling, with $h_i > \frac{\varepsilon}{3}\tilde{H}$ for all i and total hazing at most $\tilde{H}$. Defining $f_i = \lfloor h_i/g \rfloor$ and $F = \sum_{i=0}^{\ell} f_i$, we must have $T[F] \geq \sum_{i=0}^{\ell} h_i$ in the end of the execution.

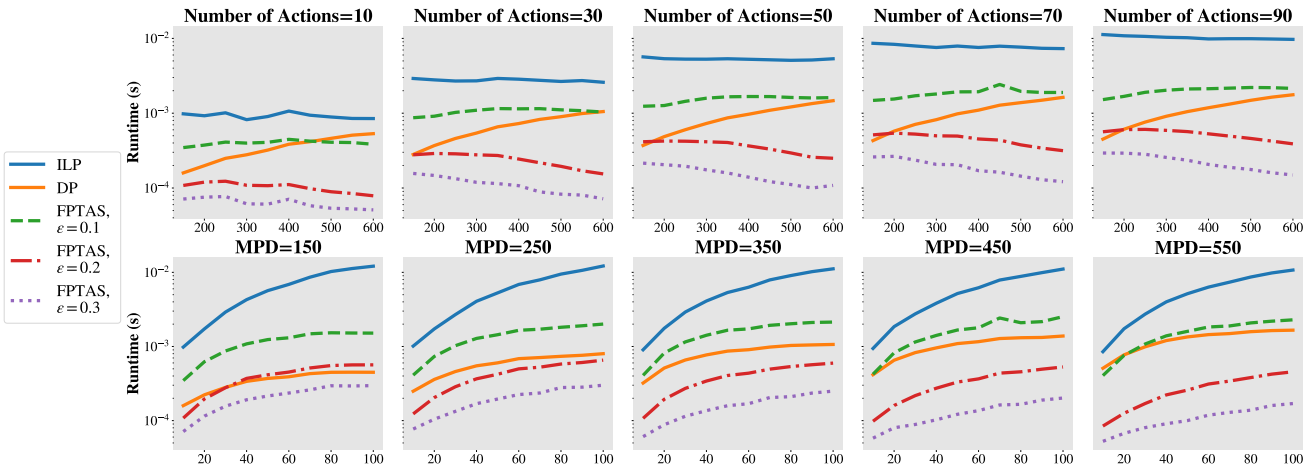

Figure 1: The semi-log plots of runtimes of Algorithms 1,2, and 3 (with $\varepsilon = 0.3, 0.2$, and 0.1). Each data point is averaged over 5000 trials. **Top Row:** Fixed number of actions (n); x-axis shows Maximum Payoff from Deviation (MPD). **Bottom Row:** Fixed MPD; x-axis shows n.

We now complete the proof of the theorem: Say $(h_i^\star)_{i\in\{0\}\cup[\ell]}$ is a member of the optimal equivalence class ($\sum_{i=0}^{\ell} h_i^\star = H^*$), with corresponding normalized hazing costs $(f_i^\star)_{i\in\{0\}\cup[\ell]}$ and $F^* = \sum_{i=0}^{\ell} f_i^\star$. By Lemma 3, we can assume WLOG that $(h_i^\star)_{i\in\{0\}\cup[\ell]}$ is threshold-monotonic, with respect to the current labeling of the actions.

Say $\hat{H} = T[k^*] + n_{k^*} \cdot h^{j_s}$ and $\hat{L} = L[k^*] + [j_s]^{n_{k^*}}$ are the final outputs of the algorithm, as set in Line 19, and S is the set of actions j with $h^{(j)} \le (\varepsilon/3)\tilde{H}$, as filled in Line 10. By Claim 4, the total hazing of the sequence represented by $\hat{L}$ is indeed $\hat{H}$. The stability of every step in $L[k^*]$ in ensured by the if statement in Line 14. The stability of the final n_{k^*} steps is ensured by the $T[k^*] > t^{(j_s)}$ condition in Line 18.

Case 1: $(h_i^\star)_{i\in\{0\}\cup[\ell]}$ has no element from S. Then it fulfills the assumptions of Claim 5 (as $H^* \le \tilde{H}$ by Claim 2), implying $T[F^*] \ge H^* \ge \Delta + 1 > t^{(j_s)}$. Since $H^* \le \tilde{H}$, we have $F^* \le g$, so $T[F^*]$ is one of the options considered in Line 18 (with $n_{F^*} = 0$), implying $\hat{H} \le T[F^*]$. By Claim 4, we get $\sum_{j\in L[F^*]} f^{(j)} = F^* = \sum_{i=0}^{\ell} f_i^\star$. Using Claim 3 twice:

$$T[F^*] = \sum_{j\in L[F^*]} h^{(j)} \le \left(1+\frac{\varepsilon}{2}\right)\delta F^* \le \left(1+\frac{\varepsilon}{2}\right)\left(\sum_{i=0}^{\ell} h_i^\star\right)$$

Hence, $\hat{H} \le T[F^*] \le \left(1+\frac{\varepsilon}{2}\right)H^*$, fulfilling the bound.

Case 2: $(h_i^\star)_{i\in\{0\}\cup[\ell]}$ does have an action from S. Say i^* is the first index where such an action appears and $F' = \sum_{i=0}^{i^*-1} f_i^\star$. Then $(h_i^\star)_{i\in\{0\}\cup[i^*-1]}$ satisfies the assumptions of Claim 5, so we have $T[F'] \ge \sum_{i=0}^{i^*-1} h_i^\star > t_{i^*}^\star \ge t^{(j_s)}$, where the inequalities follow from Claim 5, stability of h_{i^*} and the minimality of $t^{(j_s)}$, respectively. Then $T[F'] + n_{F'}h^{(j_s)}$ is considered in Line 18, implying $\hat{H} \le T[F'] + n_{F'}h^{(j_s)}$. Say $n_{F'} \neq 0$: since $n_{F'}$ is the smallest integer such that $T[F'] + n_{F'}h^{(j_s)} > \Delta$, we have:

$$\hat{H} \le T[F'] + (n_{F'}-1)h^{(j_s)} + h^{(j_s)} \le \Delta + h^{(j_s)} < H^* + \frac{\varepsilon}{3}\tilde{H} \le H^* + \frac{2\varepsilon}{3}H^* \le (1+\varepsilon)H^*$$

fulfilling the bound. Otherwise, say $n_{F'} = 0$, implying $T[F'] > \Delta$. By the same argument as Case 1, using Claims 4 and 3, we have $T[F'] \le \left(1+\frac{\epsilon}{2}\right)\left(\sum_{i=0}^{i^*-1} h_i\right)$. However, we have $\sum_{i=0}^{i^*-1} h_i \le \Delta < H^*$ as otherwise item i^* would be unnecessary to reach the goal value. This gives us: $\hat{H} \le T[F'] \le \left(1+\frac{\varepsilon}{2}\right)H^*$, fulfilling the bounds.

Overall, we have shown that in all cases, the output $\hat{L}$ contains a stable sequence with total hazing cost $\hat{H} < (1+\varepsilon)H^*$. (b) *Time complexity:* Ordering actions by thresholds takes $O(n \log n)$ time. Lines 1-8 take $O(n)$ time. The outer loop in Line 9 gets executed $O(n)$ times, each running a constant number of operations plus the inner loop in Line 13, which gets executed at most $g = O(1/\varepsilon^2)$ times with constant time per iteration. Thus, Lines 9-16 take $O\left(\frac{n}{\varepsilon^2}\right)$ time. Lines 30, 31 take $O(n)$ and $O(g)$ time, respectively. Overall, Algo. 3 takes $O\left(n \log n + \frac{n}{\varepsilon^2}\right)$ time, polynomial in both n and $\frac{1}{\varepsilon}$. □

6 Experiments

We present the semi-log plots for the runtimes of Algorithms 1-3 in Figure 1. Given the number of actions n and a positive integer Maximum Payoff of Deviation (MPD), we generated a game by (for each action $j \in [n]$) uniformly choosing $p^{(j)}$ from $\{i \in \mathbb{Z} : 0 \le i \le 30\}$ and uniformly choosing $p^{*(j)}$ from $\{i \in \mathbb{Z} : p^{(j)} \le i \le \text{MPD}\}$.[3] As expected, the runtimes of all algorithms increase with increasing n, and the runtime of FPTAS increases with decreasing ε. Similarly, consistent with our runtime analysis, Algorithm 1 (DP) increases with increasing MPD, since MPD and Δ are positively correlated, whereas the other algorithms do not exhibit

[3] Imposing $p^{*(j)} \ge p^{(j)}$ ensures $\Delta > 0$ for each game.

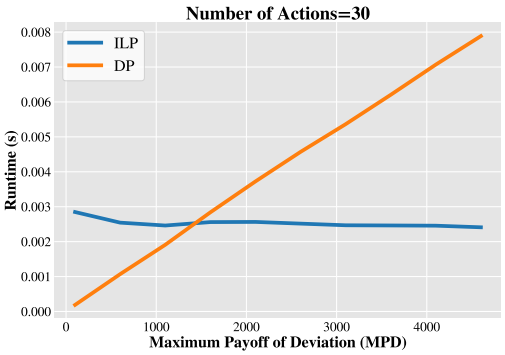

Figure 2: Runtimes of Algorithms 1 and 2 for a fixed number of actions (30) as a function of Maximum Payoff of Deviation (MDP). Each data point is averaged over 5000 trials.

such an increase, as their runtimes are independent of the actual payoffs.[4] While the DP consistently outperforms the ILP in the plots, this trend is naturally reversed for high Δ, which we demonstrate by rerunning the same runtime experiment for these algorithms for larger values of MPD. Our results are presented in Figure 2.

Notice that for $n = 30$, we see that the ILP (the runtime of which is robust to increases in the payoffs) starts to outperform the DP (the runtime of which steadily increases with increasing MPD) when the MPD reaches approximately 1500. The figure demonstrates the better performance of the ILP in the high Δ limit, which corresponds to games requiring longer (in terms of the number of rounds) hazing periods before the goal value is reached.

7 Future Work

As mentioned in Section 5, Algorithm 3 is inspired by the original FPTAS for the unbounded knapsack problem [Ibarra and Kim, 1975]. Since then, faster schemes for the same problem have been developed [Lawler, 1979; Jansen and Kraft, 2018]. It is possible that similar modifications can be made to these algorithms to enable them to solve OptRep, improving the runtime of its FPTAS.

A natural next step following our work is considering asymmetric games and strategies. While some of our results generalize to this setting, there are important differences. For example, goal values must be generalized to goal sequences to maximize payoff (e.g., if (C_2, C_2) from Table 1 instead had payoff (6,6), it would be better to forever alternate between (D, C_2) and (C_2, D)); hence, computing the "optimal goal sequence" is a separate computational problem on its own. A two-variable generalization of Algorithm 1 can compute sequences reaching a goal sequence that maximizes the total payoff, but fails for any other goal sequence. However, since fairness concerns come into play with asymmetric payoffs, maximizing the total payoff may not be sufficient, requiring novel algorithmic techniques.

Another possible extension is to consider *Bayesian* games [Harsanyi, 1967], where agents have private information about how they value certain outcomes. In such games, play might differ depending on an agent's type, and we may even have defection and restarting on the path of play in equilibrium. For example, if a small fraction of the population obtains great benefit from deviating in the first round, it may be better to tolerate those few agents repeatedly taking advantage for one round and then finding another partner, than to add significant hazing for the entire population to prevent this. In this context, *partial* anonymity could also be of interest, for example where one can *choose* to reveal one's history to a new partner, for example to show that one behaved properly in a previous relationship but the partner unfairly defected.

[4] In fact, as seen in Fig.1(top), the runtime for Algo. 3 sometimes decreases with increasing MPD. This is likely because increasing MPD increases the upper bound for the deviation payoffs but not for the cooperation payoffs, resulting in Δ increasing while $h^{(j)}$s stay in the same range. Thus, more actions are placed in S and skip the inner loop in Line 13, resulting in less computation by the FPTAS.

Acknowledgments

We thank the Cooperative AI Foundation, Polaris Ventures (formerly the Center for Emerging Risk Research), and Jaan Tallinn's donor-advised fund at Founders Pledge for financial support. We are grateful to Matt Rognlie for his valuable contributions to the conceptualization of and preliminary work on this project. We thank Anupam Gupta for his helpful comments on FPTAS literature. Lastly, we thank the members of Foundations of Cooperative AI Lab (FOCAL) for comments and suggestions on earlier drafts of this work.

References

[Andersen and Conitzer, 2013] Garrett Andersen and Vincent Conitzer. Fast equilibrium computation for infinitely repeated games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 53–59, Bellevue, WA, USA, 2013.

[Borgs *et al.*, 2010] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. The myth of the Folk Theorem. *Games and Economic Behavior*, 70(1):34–43, 2010.

[Conitzer and Oesterheld, 2023] Vincent Conitzer and Caspar Oesterheld. Foundations of cooperative AI. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 15359–15367, Washington, DC, USA, 2023.

[Conitzer and Yokoo, 2010] Vincent Conitzer and Makoto Yokoo. Using mechanism design to prevent false-name manipulations. *AI Magazine*, 31(4):65–77, 2010.

[Dafoe *et al.*, 2021] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. Cooperative AI: machines must learn to find common ground. *Nature*, 593(7857):33–36, 2021.

[Datta, 1996] Saikat Datta. Building trust. Sticerd - theoretical economics paper series, Suntory and Toyota International Centres for Economics and Related Disciplines, LSE, 1996.

[Fujiwara-Greve and Okuno-Fujiwara, 2009] Takako Fujiwara-Greve and Masahiro Okuno-Fujiwara. Voluntarily separable repeated prisoner's dilemma. *The Review of Economic Studies*, 76(3):993–1021, 2009.

[Ghosh and Ray, 1996] Parikshit Ghosh and Debraj Ray. Cooperation in community interaction without information flows. *The Review of Economic Studies*, 63(3):491–519, 1996.

[Harsanyi, 1967] John C. Harsanyi. Games with incomplete information played by "Bayesian" players, i-iii. part i. the basic model. *Management Science*, 14(3):159–182, 1967.

[Ibarra and Kim, 1975] Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, oct 1975.

[Izquierdo *et al.*, 2014] Luis R. Izquierdo, Segismundo S. Izquierdo, and Fernando Vega-Redondo. Leave and let leave: A sufficient condition to explain the evolutionary emergence of cooperation. *Journal of Economic Dynamics and Control*, 46:91–113, 2014.

[Jansen and Kraft, 2018] Klaus Jansen and Stefan E.J. Kraft. A faster FPTAS for the unbounded knapsack problem. *European Journal of Combinatorics*, 68:148–174, 2018. Combinatorial Algorithms, Dedicated to the Memory of Mirka Miller.

[Kontogiannis and Spirakis, 2008] Spyros C. Kontogiannis and Paul G. Spirakis. Equilibrium points in fear of correlated threats. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 210–221, Shanghai, China, 2008.

[Kranton, 1996] Rachel E Kranton. The Formation of Cooperative Relationships. *The Journal of Law, Economics, and Organization*, 12(1):214–233, April 1996.

[Lawler, 1979] Eugene L. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, 4(4):339–356, 1979.

[Littman and Stone, 2005] Michael L. Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39:55–66, 2005.

[Lueker, 1975] G.S. Lueker. Two NP-complete problems in nonnegative integer programming. Technical Report 178, Computer Science Laboratory, Princeton University, 1975.

[Magazine and Chern, 1984] Michael J. Magazine and Maw-Sheng Chern. A note on approximation schemes for multidimensional knapsack problems. *Mathematics of Operations Research*, 9(2):244–247, 1984.

[Rob and Yang, 2010] Rafael Rob and Huanxing Yang. Long-term relationships as safeguards. *Economic Theory*, 43(2):143–166, 2010.

[Rudin, 1976] Walter Rudin. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

[Wei, 2019] Dong Wei. A model of trust building with anonymous re-matching. *Journal of Economic Behavior & Organization*, 158:311–327, 2019.

[Wojtczak, 2018] Dominik Wojtczak. On strong NP-completeness of rational problems. In Fedor V. Fomin and Vladimir V. Podolskii, editors, *Computer Science – Theory and Applications*, pages 308–320, Cham, 2018. Springer International Publishing.

[Yokoo *et al.*, 2001] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167–181, 2001.

[Yokoo *et al.*, 2004] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.