TEAM BELIEF DAG FORM: A CONCISE REPRESENTATION FOR TEAM-CORRELATED GAME-THEORETIC DECISION MAKING

Brian Hu Zhang, Gabriele Farina
Computer Science Department
Carnegie Mellon University
{bhzhang, gfarina}@cs.cmu.edu

Tuomas Sandholm
Computer Science Department, CMU
Strategic Machine, Inc.
Strategy Robot, Inc.
Optimized Markets, Inc.
sandholm@cs.cmu.edu

ABSTRACT

In this paper, we introduce a new representation for team-coordinated game-theoretic decision making, which we coin *team belief DAG form*. In our representation, at every timestep, a team coordinator observes the information that is public to all its members, and then decides on a prescription for all the possible states consistent with its observations. Our representation unifies and extends recent approaches to team coordination. Similar to the approach of Carminati et al. (2021), our team belief DAG form can be used to capture adversarial team games, and enables standard, out-of-the-box game-theoretic techniques including no-regret learning (e.g., CFR and its state-of-the-art modern variants such as DCFR and PCFR$^+$) and first-order methods. However, our representation can be exponentially smaller, and can be viewed as a lossless abstraction of theirs into a directed acyclic graph. In particular, like the LP-based algorithm of Zhang & Sandholm (2022), the size of our representation scales with the amount of information uncommon to the team; in fact, using linear programming on top of our team belief DAG form to solve for a team correlated equilibrium in an adversarial team games recovers almost exactly their algorithm. Unlike that paper, however, our representation explicitly exposes the structure of the decision space, which is what enables the aforementioned game-theoretic techniques.

1 INTRODUCTION

Decision making in teams has been a recent focus in computational game theory and reinforcement learning. Teams that cannot communicate perfectly have particular problems that single players and perfectly-communicating teams do not have. Examples include recreational games like Bridge (in which two teams compete adversarially), collusion in poker, military situations with restricted communications, various swindling settings, and many other real-world situations.

In general, computing optimal strategies in adversarial team games is NP-hard (Chu & Halpern, 2001), even given the extensive form and assuming there are only two players and no adversary. Despite this, there are cases that are tractable. For example, if both teams exhibit so-called *A-loss recall* (Kaneko & Kline, 1995) or *triangle-free interaction* between two players (Farina & Sandholm, 2020), then polynomial-time algorithms are known to exist.

Until recently, techniques for solving adversarial team games were focused largely on *column generation* (Farina et al., 2018; 2021a; Zhang et al., 2021), which works well in some games in practice but has no theoretical guarantees and scales poorly in most games. More recently, Zhang & Sandholm (2022) developed an algorithm for solving adversarial team games based on a novel *tree decomposition* of each player's strategy space, and use it to devise a linear program. They show parameterized complexity bounds on the runtime of the algorithm, and demonstrate strong practical performance. Simultaneously, Carminati et al. (2021) developed a conversion algorithm based on prior research in the multi-agent reinforcement learning community (e.g., Nayyar et al.

2013; Sokota et al. 2021). Their algorithm converts an extensive-form adversarial team game into an exponentially-larger *public team game*.

In this paper, we unify these last two lines of work. We give a representation of the decision problem faced by a team of players as a *sequential decision problem* (Farina et al., 2019a) on a directed acyclic graph (DAG), which we call the *team belief DAG form*.

Using a linear program to solve the resulting adversarial team game recovers almost exactly the algorithm of Zhang & Sandholm (2022). However, our representation has several advantages. First and most importantly, explicitly expressing the polytope as the set of flows over a DAG gives us a description of the set of strategies in terms of *scaled extensions*. This description allows us to apply the vast theoretical literature on sequence-form games, especially with respect to regret minimization, to team games as well. Without understanding the special structure of the DAG, this development is not possible. Second, it is slightly more concise, owing to the slight differences in definitions of public states and ease of optimization of our method. This will become clear in the experimental evaluation later in the paper. Finally, it is conceptually cleaner and easier to understand, not requiring the extra machinery of tree decompositions. The conceptual ease also allows easier theoretical statements and proofs with slightly tighter bounds.

Applying *safe imperfect-recall abstraction* (Lanctot et al., 2012) on top of the sequence form of the *folding representation* of Carminati et al. (2021) for a given team yields a decision space that is also basically equivalent to our team belief DAG form. However, this is a rather roundabout construction: their folding representation can have size exponentially larger than the team belief DAG, so our framework yields exponentially-faster algorithms by avoiding the construction of that larger representation.

In experiments, we demonstrate that the state-of-the-art variants of counterfactual regret minimization—namely DCFR (Brown & Sandholm, 2019) or PCFR$^+$ (Farina et al., 2021b)—applied on top of our team belief DAG form almost always outperform linear programming in team games. This represents the new practical state of the art for adversarial team games.

2 Preliminaries

We now introduce the paradigm of tree-form sequential decision making (Farina et al., 2019a), which we will use throughout the paper.

2.1 Tree-Form Sequential Decision Making

Definition 2.1. A *tree-form sequential decision-making problem* (TFSDP) $\mathcal{T}$ is a rooted tree with node set $\mathcal{S}$ and labelled edges, in which each node is a *decision node*, at which the player takes an action, or an *observation nodes*, at which the player receives an observation.

The edges out of decision nodes are called *actions*, and the edges out of observation nodes are called *observations*. The root node is denoted $\emptyset$. The set of leaf, or *terminal*, nodes is denoted $\mathcal{Z}_\mathcal{T}$.

If $s, s' \in \mathcal{S}$ are two nodes, $s \preceq s'$ means that there is a directed path from s to s'. If S, S' are set of nodes, $S \preceq S'$ means $s \preceq s'$ for some $s \in S, s' \in S'$. A_s is the set of labels on edges descending from node s (actions or observations). The node reached by following edge $a \in A_s$ out of node s is denoted sa. For two nodes $s, s' \in \mathcal{S}$, $s \wedge s'$ denotes the lowest common ancestor of s and s'.

A *pure strategy* is an assignment of one action to each decision node. The *sequence form*[1] $\boldsymbol{x} \in \{0,1\}^{\mathcal{Z}_\mathcal{T}}$ of a pure strategy is the vector for which $x[z] = 1$ if and only if the player plays all the actions on the root $\to z$ path. A *mixed strategy* is a distribution over pure strategies. The sequence form of a mixed strategy is the corresponding convex combination $\boldsymbol{x} \in [0,1]^{\mathcal{Z}_\mathcal{T}}$.

2.2 Online Convex Optimization

Online convex optimization (Zinkevich, 2003) is a framework for describing repeated interactions of a player with an arbitrary environment. At each timestep, the player selects a *strategy* $\boldsymbol{x}^t$ from a

[1]Differing from most literature, in this paper we define the sequence form to be over *terminal* sequences only.

convex, compact set $\mathcal{X} \subseteq \mathbb{R}^m$, and observes a (possibly adversarially chosen) *utility vector* $\boldsymbol{u}^t \in \mathbb{R}^m$. The goal of the player is to minimize the *regret* after T timesteps:

$$R^T := \max_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} \left\langle \boldsymbol{u}^t, \boldsymbol{x} - \boldsymbol{x}^t \right\rangle .$$

A *regret minimizer* is any algorithm for the player that guarantees that R^T is sublinear in T. In this paper, we will be concerned with regret minimizers over the decision space $\mathcal{X} \subseteq [0,1]^{\mathcal{Z}_\mathcal{T}}$ of sequence-form mixed strategies in various sorts of decision problems.

3 Decision Making on DAGs

In this section, we discuss how techniques that apply to sequential decision making, for example, no-regret learning, can be used on a decision problem that is a DAG, which may be of independent interest beyond our interest in teams.

Definition 3.1. A *DAG-form sequential decision problem* (DFSDP) is an TFSDP on a DAG instead of a tree.

We will insist on the following technical conditions:

1. Observation nodes always have exactly one parent.
2. Nodes along every path alternate between decision nodes and observation nodes.
3. If p_1 and p_2 are two paths from the root ending at the same node, then the last node common to both p_1 and p_2 is a decision node.

The first two conditions are for expository simplicity and are without loss of generality; the final one is critical for our constructions to work.

The sequence form $\boldsymbol{x} \in \{0,1\}^{\mathcal{Z}_\mathcal{T}}$ of a pure strategy is the vector for which $x[z] = 1$ if and only if the player plays all the actions on *some* root $\to z$ path[2]. Mixed strategies and their sequence forms are defined as usual.

3.1 DFSDPs via Scaled Extensions

In this section, we show that the set of sequence-form strategies in a DFSDP can be expressed in terms of *scaled extensions* (Farina et al., 2019b).

Definition 3.2. Given two nonempty, compact, convex sets $\mathcal{X}, \mathcal{Y}$ and a linear map $h : \mathcal{X} \to \mathbb{R}_{\geq 0}$, the *scaled extension* of $\mathcal{X}$ with $\mathcal{Y}$ via h, is defined as

$$\mathcal{X} \overset{h}{\triangleleft} \mathcal{Y} = \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in h(\boldsymbol{x})\mathcal{Y}\}.$$

We now construct the set of sequence-form strategies in a given DFSDP. We begin with the set $\mathcal{X} \leftarrow \{1\}$. Then, for each decision node s, we perform the operation $\mathcal{X} \leftarrow \mathcal{X} \overset{x[s]}{\triangleleft} \Delta^{A_s}$ where

$$x[s] := \sum_{s' \text{ parent of } s} x[s'].$$

The technical condition ensures that $x[s] \in [0,1]$ for every s. The restriction of the resulting set $\mathcal{X}$ on the set of terminal states $\mathcal{Z}_\mathcal{D}$ is exactly the set of sequence-form mixed strategies. Thus, we have shown:

Theorem 3.3. *The set of sequence-form strategies on a DFSDP can be expressed by scaled extension operations with simplices via functions $h : \mathcal{X} \to [0,1]$.*

Corollary 3.4. *This set can be expressed as a polytope with $O(E)$ constraints, where E is the number of edges.*

[2]The final technical condition ensures that there is at most one such path.

3.2 Regret Minimization in DFSDPs

Any set that can be built from scaled extensions and simplexes admits a regret minimizer that can be constructed starting from any simplex regret minimizer (Farina et al., 2019b). This construction extends CFR (Zinkevich et al., 2007), and all its modern variants, to such sets. In particular, applying Proposition 1 of Farina et al. (2019b) on top of Theorem 3.3 gives us:

Corollary 3.5. *Deterministic variants of CFR (e.g., CFR$^+$, DCFR (Brown & Sandholm, 2019), PCFR$^+$ (Farina et al., 2021b)) can be run on a DFSDP, with regret bounded by $O(|\mathcal{S}|\sqrt{T})$ after T timesteps and iteration time $O(E)$ where E is the number of edges in the DAG.*

Pseudocode for running CFR on an arbitrary DFSDP can be found in Algorithm 1.

4 Team Sequential Decision Making

In this section, we define the *team TFSDP*, a generalization of the TFSDP to the setting where there are multiple collaborating players with uncommon information.

Definition 4.1. A team TFSDP is a TFSDP in which decision nodes are partitioned into *information sets*, or *infosets*.

We will insist that decision nodes in the same information set have the same action set. The action set in a given information set I will be denoted A_I.

To disambiguate from non-team TFSDPs and to be consistent with literature on extensive-form games, we will use $\mathcal{H}$ for the set of nodes in a team TFSDP instead of $\mathcal{S}$. For an infoset I and an action $a \in A_I$, we will use Ia to denote $\{ha : h \in I\}$.

The *team sequence* $\sigma(h)$ of a node h is the sequence of infosets reached and actions played by the team along the root $\to h$ path, including, if any, the infoset containing h itself. The *effective size* of a set of nodes $H \subseteq \mathcal{H}$ is the number of distinct team sequences among the nodes in H.

A team TFSDP is *timed* if no infoset spans multiple layers of the decision problem. That is, paths from the root to nodes in the same infoset must have the same length. In this paper, we will only work with timed team TFSDPs[3].

A *pure strategy* is an assignment of one action to each infoset. Insisting on choosing actions on the infoset level ensures that team members are not making decisions based on information that they do not know. A *correlated strategy* is a distribution over pure strategies[4]. The sequence form of a correlated strategy is called its *correlation form*.

It is known that the polytope of correlation plans cannot be represented by polynomially many constraints unless P = NP (Chu & Halpern, 2001).

It will occasionally be useful to us to distinguish the individual *players* on a team. Formally, if the team consists of n players, we partition the collection of infosets into n collections $\mathcal{I}_1, \dots, \mathcal{I}_n$, where $\mathcal{I}_i$ is the set of infosets at which player i plays. The *player sequence* $\sigma_i(h)$ of a node h for a player i is the sequence of infosets reached belonging to i, and actions played by i, on the root $\to h$ path, again including the infoset at h itself if i plays at h. We will assume that each individual player on a team has *perfect recall*: if $h, h' \in \mathcal{I}_i$, then $\sigma_i(h) = \sigma_i(h')$.

A team TFSDP can also be thought of as a (single-agent) TFSDP with *imperfect recall*. In this presentation, the team is represented as a single player whose memory is repeatedly updated to only contain the information of the team member who is currently playing. The two models are equivalent.

5 Public States and the Team Belief DAG Form

Let $\mathcal{T}$ be a team TFSDP. We will assume that $\mathcal{T}$ is *completely inflated* (Kaneko & Kline, 1995): there does not exist a partition $I_1 \sqcup I_2$ of any infoset I in $\mathcal{T}$ such that, for every $h_1 \in I_1$ and $h_2 \in I_2$, there

[3] As long as the relation $\preceq$ on infosets is a partial order, it is always possible to make a team TFSDP timed by adding dummy nodes.

[4] We use "correlated" instead of "mixed" because, unlike in the single-player case, we *cannot* assume without loss of generality that the decisions at each infoset are made independently.

is some infoset I' such that $I'a_1 \preceq h_1$ and $I'a_2 \preceq h_2$ for *different* actions $a_1 \neq a_2 \in A_{I'}$. This assumption assumptions can be satisfied WLOG by preprocessing the tree.

Definition 5.1. Two nodes h, h' in the same layer of $\mathcal{T}$, and either both terminal or both nonterminal[5], are *connected* if they have the same team sequence, *or* there is an infoset I such that $h \preceq I$ and $h' \preceq I$.

Connectedness induces a graph whose nodes are the decision nodes, which we call the *connectivity graph*.

Definition 5.2. A *public state* is a connected component of the connectivity graph.

Intuitively, a public state is a set of nodes P such that whether P has been reached is common knowledge among the team. The public states themselves induce a tree, which is known as the *public tree*. We will be interested in two separate branching factors. First, the *team branching factor* b is the largest number of actions at any decision node in $\mathcal{T}$. Second, the *public branching factor* b' is the branching factor of the public tree[6].

The *team belief DAG form* of a team TFSDP $\mathcal{T}$ is a DFSDP $\mathcal{D}$ defined as follows. All nodes in the team belief DAG form are identified with *sets* of nodes in the team TFSDP.

1. The initial node of $\mathcal{D}$ is a decision node corresponding to the set $\{\emptyset\}$ containing only the root node of the original problem.
2. At a decision node B in $\mathcal{D}$, let $I_1, \dots, I_m$ be all the infosets with nonempty intersection with B, and let J be the set of observation nodes (of $\mathcal{T}$) in set B. The player selects a *prescription* $\boldsymbol{a} \in \bigtimes_{i\in[m]} A_{I_i}$, consisting of one action a_i in each infoset I_i. The decision problem then transitions to the observation node $B\boldsymbol{a}$ consisting of all children of nodes in B which are consistent with the actions of the player. Formally,
$$B\boldsymbol{a} = \{ha_i : h \in I_i \cap B\} \cup \{ha : h \in J, a \in A_h\}.$$
3. At an observation node O in $\mathcal{D}$, let $P_1, \dots, P_m$ be the public states with nonempty intersection with O. Then the player transitions to the decision node $O \cap P_i$ for some $i \in [m]$.

If a decision node B contains terminal nodes in $\mathcal{D}$, then it is terminal.

The sets B identified with *decision* nodes in $\mathcal{D}$ are called *beliefs*. In this construction, multiple decision nodes may be identified with the same belief. The subtrees induced by these decision nodes are copies of each other. Therefore, we merge them, forming a DAG. This DAG is the *team belief DAG*, and the DFSDP is the *team belief DAG form*. For an example, see Figure 1; for pseudocode, see Algorithm 2.

To be consistent with the previous two sections and to distinguish a team TFSDP from its team belief DAG form, we will use $\mathcal{H}$ to refer to the set of nodes in the former, and $\mathcal{S}$ to refer to the set of nodes in the latter.

Given a sequence-form mixed strategy $\boldsymbol{x}'$ in the team belief DAG form, we construct a correlation plan $\boldsymbol{x}$ in $\mathcal{T}$ as follows. For each terminal node z in $\boldsymbol{x}$, let $\Sigma(z)$ be the set of all terminal nodes in $\mathcal{H}$ sharing a team sequence and a level with z, that is,
$$\Sigma(z) = \{z' \in \mathcal{Z}_{\mathcal{T}} : \sigma(z) = \sigma(z'), z, z' \text{ are at the same level}\}.$$
Then $\Sigma(z)$ is a terminal decision node in $\mathcal{D}$ by construction. We set $\boldsymbol{x}[z] = \boldsymbol{x}'[\Sigma(z)]$.

Theorem 5.3. *The set of vectors $\boldsymbol{x}$ constructible in the above fashion is exactly the set of correlation plans of the team TFSDP.*

The proof of this theorem and the proofs of all following theorems are in the appendix.

[5]The inclusion of the careful edge case for terminal nodes is so that every terminal node in $\mathcal{T}$ maps onto a node in $\mathcal{D}$. Compare nodes F and L in Figure 1. There is a node in the DAG representing the reach probability of F, but not one for L. For the latter, one would have to sum the reach probabilities of nodes L and LM in the DAG.

[6]The public branching factor is usually *larger* than the branching factor of the original tree. For example, our notion of public tree has each terminal team sequence belonging to its own public state, and thus information that was previously private suddenly becomes public at the very end of the game.

6 THE SIZE OF A TEAM BELIEF DAG

Since all our theoretical results depend on the size of the team belief DAG, it is critical to analyze that size. The hardness result of Chu & Halpern (2001) means that our size bounds will not be polynomial. However, we can still bound the size relative to natural parameters related to the complexity of the game.

The correspondence between our construction and that of Zhang & Sandholm (2022) allows us to achieve similar theoretical guarantees to that paper. Here, we explicitly give such results in our language. Let $\mathcal{T}$ be a team TFSDP with node set $\mathcal{S}$ and public state set $\mathcal{P}$.

Theorem 6.1. *The team belief DAG of $\mathcal{T}$ has at most $\binom{p}{\le w} b^w b' |\mathcal{P}| \le (b(p+1))^w b' |\mathcal{P}|$ edges*[7]*, where p is the largest effective size of any public state and w is the largest effective size of any belief.*

The parameter w is identical to the same-named parameter in Zhang & Sandholm (2022). As discussed in that paper, w depends only on the amount of *uncommon external information*, that is, *observations* (as opposed to decisions) that are not common knowledge to the team.

In a certain family of team TFSDPs including those with team-public actions, we can do better.

Definition 6.2. An n-player team TFSDP is *k-private* if, in every public state, there are at most k distinct *player* sequences. That is, $|\{\sigma_i(h) : i \in [n], h \in P\}| \le k$ for every $P \in \mathcal{P}$.

This is distinct from the *effective size* p, which is the total number of *team* sequences in P. In particular, in games with *team-public actions* (such as poker), where each player has at most t private types, we have $k \le nt$.

In a k-private team TFSDP, it is possible that $w = (k/n)^n$, so Theorem 6.1 gives a bound of $(2bp)^{(k/n)^n} b' |\mathcal{P}|$, which is bad. However, we can improve upon this through a more careful analysis.

Theorem 6.3. *The team belief DAG of a k-private team TFSDP has at most $(b+1)^k b' |\mathcal{P}|$ edges.*

If necessary, it is possible to "mix and match" the analyses of Theorems 6.1 and 6.3 when some public states have low w and some have low t. To save the cumbersome notation, we will not do that here.

7 ADVERSARIAL TEAM GAMES

All our results so far have been given from the perspective of a single team. However, they generalize very naturally to the case of multiple teams competing in a game.

An *extensive-form adversarial team game* Γ consists of:

1. a tree of nodes $\mathcal{H}$, where the edges are labelled with actions. The set of terminal nodes in $\mathcal{H}$ will, as usual, be denoted $\mathcal{Z}_\Gamma$;
2. a partition $\mathcal{I}$ of nonterminal nodes in $\mathcal{H}$ into infosets, where every node in a given infoset must have the same action set;
3. a partition of $\mathcal{I}$ into three collections $\mathcal{I}^{\mathsf{C}} \sqcup \mathcal{I}^{\oplus} \sqcup \mathcal{I}^{\ominus}$, where $\mathcal{I}^{\mathsf{C}}$ is the collection of infosets at which *chance* acts, and $\mathcal{I}^\tau$ is the collection of infosets assigned to team $\tau \in \{\oplus, \ominus\}$;
4. for each infoset $I \in \mathcal{I}^{\mathsf{C}}$ assigned to chance, a fixed distribution over the actions available at I; and
5. a *utility function* $u : \mathcal{Z}_\Gamma \to \mathbb{R}$, where $u(z)$ denotes the utility for $\oplus$ in terminal node z. The utility for $\ominus$ is $-u(z)$.

As usual, we will *not* demand that the teams have perfect recall. An extensive-form game immediately defines a team TFSDP for each team τ, with information partition $\mathcal{I}^\tau$. For consistency with our definition of the team belief DAG form, we insist that non-chance nodes all have exactly 2 children.

A *team correlated equilibrium* (TMECor) in an adversarial team game is a Nash equilibrium of the game viewed as a two-player zero-sum game between the two teams. A TMECor can be expressed as

[7] $\binom{p}{\le w} := \sum_{i=1}^{w} \binom{p}{i}$.

the solution to a convex bilinear saddle-point problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle \tag{1}$$

where $\boldsymbol{A}$ is the *payoff matrix*, whose bilinear form is the expected utility for $\oplus$ if $\oplus$ plays correlation plan $\boldsymbol{x}$ and $\ominus$ plays correlation plan $\boldsymbol{y}$.

Let S and E be the total number of nodes and edges respectively across the team belief DAGs of both teams the following results follow from standard results on linear programming for saddle-point problems and regret minimization, and Corollary 3.5:

Corollary 7.1. *Nash equilibria in adversarial team games can be found via a linear program of size* $O(E + |\mathcal{Z}_\Gamma|)$.

Corollary 7.2. *In adversarial team games, after T iterations of a CFR variant, the average strategy for both players is an $O(S/\sqrt{T})$-Nash equilibrium. Each iteration takes time $O(E)$.*

The former result is analogous to the results shown in Zhang & Sandholm (2022). The latter, to our knowledge, is new.

In light of Corollary 3.5 and Theorems 6.1 and 6.3, Corollaries 7.1 and 7.2 immediately imply fixed-parameter runtime and convergence bounds for both the CFR family of algorithms and linear programming.

8 Experiments

We experimentally investigate solving adversarial team games using the team belief DAG form. Since all our experiments are in games with public actions, we preprocess with branching factor reduction in all cases.

8.1 Equilibrium-Finding Algorithms

We investigate the application of the following two categories of game-solving algorithm on the belief DAG of the team:

1. CFR-like algorithms obtained from the scaled extension decomposition of the team belief DAG (Section 3.2). We implemented the following state-of-the-art variants of CFR: *Predictive CFR$^+$ (PCFR$^+$)* (Farina et al., 2021b), *Discounted CFR (DCFR)* (Brown & Sandholm, 2019), and *Linear CFR (LCFR)* (Brown & Sandholm, 2019). PCFR$^+$ and DCFR use quadratic averaging of iterates, while LCFR uses linear averaging. PCFR$^+$ is a predictive regret minimization algorithm. At each time t, we use the previous utility vector for each time as prediction for the next. Each implementation is single-threaded.
2. Linear programming (LP) solvers applied directly on the bilinear saddle point formulation of the equilibrium problem (1). To solve the linear program, we used the commercial solver Gurobi. We investigated using both the barrier algorithm, and the (concurrent primal-dual) simplex algorithm. For each algorithm, we experimented leaving the presolver on or off. We allowed Gurobi to use up to four threads.

We compare solving an adversarial team games in team belief DAG form against two prior state-of-the-art algorithms:

1. The tree-decomposition-based LP solver proposed by Zhang & Sandholm (2022) (henceforth "**ZS22**"), which has already discussed at length in this paper. We used the original implementation of the authors, which internally uses the barrier algorithm implemented by the commercial solver Gurobi. As recommended by the authors, we turned Gurobi's presolver off to avoid numerical instability and increase speed. We allowed Gurobi to use up to four threads.
2. The single-oracle algorithm of Farina et al. (2021a) (henceforth "**FCGS21**"). FCGS21 iteratively refines the strategy of each team by solving best-response problems using a tight integer program derived from the theory of extensive-form correlation (von Stengel & Forges, 2008; Farina et al., 2021b). We used the original code by the authors, which was implemented for three-player games in which a team of two players faces an opponent. Like ZS22 and our LP-based solver, FCGS21 uses the commercial solver Gurobi to solve linear and integer linear programs. We allowed Gurobi to use up to four threads.

All algorithms were run on a 64-core AMD EPYC 7282 processor. Each algorithm was allocated a maximum of 4 threads, 60GBs of RAM, and a time limit of 6 hours.

8.2 Game Instances

We ran experiments on the following standard, parametric benchmark games:

- ^{n}Kr: n-player Kuhn poker with r ranks (Kuhn, 1950).
- ^{n}Lbrs: n-player Leduc poker with a b-bet maximum in each betting round, r ranks, and s suits (Southey et al., 2005).
- ^{n}Dd: n-player Liar's Dice with one d-sided die for each player (Lisý et al., 2015).
- ^{n}G and ^{n}GL: n-player Goofspiel with 3 ranks. GL is the imperfect-information variant.

These are the same games used by Zhang & Sandholm (2022) and Farina et al. (2021a) in their experimental evaluations. We refer the reader to the latter paper for detailed descriptions of the games. The size of each game, measured in terms of number of terminal states (leaves), is reported in the second column of Table 1.

Experimental results are summarized in Table 1. Overall, we observe that our algorithms based on the team belief DAG form are generally 2-3 orders of magnitude faster than ZS22. In games with low parameters p and k, our algorithms are also several orders of magnitude faster than FCGS21, validating the conclusion of Zhang & Sandholm (2022). The difference in runtime between our algorithms and FCGS21 is especially dramatic in Leduc games. In games with high parameters (*e.g.*, ^{3}K8 and ^{3}K12), on the other hand, FCGS21 is significantly more scalable, as it avoids the exponential dependence in the parameters at the cost of requiring the solution to integer programs, for which runtime guarantees are hard to give.

9 Conclusion and Future Research

We have given a new representation, the *team belief DAG form*, for the decision problem faced by a team of correlating players, which we have used to develop new algorithms for solving adversarial team games. Our method enjoys the parameterized complexity bounds of Zhang & Sandholm (2022), and the extensibility and interpretability of Carminati et al. (2021). Experiments show that modern variants of CFR applied with our team belief DAG form give state-of-the-art performance across multiple domains. Possible directions for future research include:

1. devising a technique to allow the use of *Monte Carlo CFR (MCCFR)* (Lanctot et al., 2009) in DFSDPs, and in particular in the team belief DAG form;
2. finding theoretically sound techniques for mitigating the exponential blowup in parameters w and k;
3. finding a "best-of-both-worlds" algorithm that combines the strengths of our approach and the single-oracle-based methods;
4. investigating extensions of our technique to correlated equilibrium in general-sum games;
5. relaxing the assumption of timeability;
6. devising a construction that additionally generalizes the *triangle-free interaction* (Farina & Sandholm, 2020), a known polynomially-solvable subclass of the problem; and
7. investigating the use of other standard game-theoretic techniques in two-player zero-sum games, such as abstraction, dynamic pruning, subgame solving, etc., in team games.

Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1901403, and CCF-1733556, and the ARO under award W911NF2010081.

REFERENCES

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Luca Carminati, Federico Cacciamani, Marco Ciccone, and Nicola Gatti. Public information representation for adversarial team games. In *NeurIPS Cooperative AI Workshop*, 2021.

Francis Chu and Joseph Halpern. On the NP-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 2001.

Gabriele Farina and Tuomas Sandholm. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Ex ante coordination and collusion in zero-sum multi-player extensive-form games. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Online convex optimization for sequential decision processes and extensive-form games. In *AAAI Conference on Artificial Intelligence*, 2019a.

Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Efficient regret minimization algorithm for extensive-form correlated equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. In *International Conference on Machine Learning*, 2021a.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021b.

Mamoru Kaneko and J Jude Kline. Behavior strategies, mixed strategies and perfect recall. *International Journal of Game Theory*, 24(2):127–145, 1995.

H. W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pp. 97–103. Princeton University Press, Princeton, New Jersey, 1950.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2009.

Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. In *International Conference on Machine Learning (ICML)*, 2012.

Viliam Lisỳ, Marc Lanctot, and Michael H Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. In *Autonomous Agents and Multi-Agent Systems*, 2015.

Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

Samuel Sokota, Edward Lockhart, Finbarr Timbers, Elnaz Davoodi, Ryan D'Orazio, Neil Burch, Martin Schmid, Michael Bowling, and Marc Lanctot. Solving common-payoff games with approximate policy iteration. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 9695–9703, 2021.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

Brian Hu Zhang and Tuomas Sandholm. Team correlated equilibria in zero-sum extensive-form games via tree decompositions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Youzhi Zhang, Bo An, and Jakub Černỳ. Computing ex ante coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pp. 928–936, Washington, DC, USA, 2003.

Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2007.

Algorithm 1 CFR on an arbitrary DFSDP

function $\mathcal{D}$.NEXTSTRATEGY
  ▷ $\boldsymbol{x}'$ *will store the* unscaled *probabilities*
  $\boldsymbol{x}, \boldsymbol{x}' \leftarrow \mathbf{1}_{\in \mathbb{R}^{\mathcal{S}}}$
  for each decision node s in $\mathcal{D}$ (top down) **do**
    if s is not the root **then**
      $x[s] \leftarrow \sum_{s' \text{ parent of } s} x[s']$
    $S \leftarrow \sum_{a \in A_s} R[sa]^+$
    for each action $a \in A_s$ **do**
      if $S = 0$ **then** $x'[sa] = 1/|A_s|$
      else $x'[sa] \leftarrow R[sa]^+/S$
      $x[sa] \leftarrow x'[sa]x[s]$
  return $\boldsymbol{x}$
function $\mathcal{D}$.OBSERVEUTILITY($\boldsymbol{u} \in \mathbb{R}^{\mathcal{Z}_\mathcal{D}}$)
  for each $s \in \mathcal{S} \setminus \mathcal{Z}_\mathcal{D}$ **do** $u[s] \leftarrow 0$
  for each decision node s in $\mathcal{D}$ (bottom up) **do**
    $u[s] \leftarrow u[s] + \sum_{a' \in A_s} u[sa']x'[sa']$
    for each action $a \in A_s$ **do**
      $R[sa] \leftarrow R[sa] + u[sa] - u[s]$
    for each parent s' of s **do**
      $u[s'] \leftarrow u[s'] + u[s]$

Algorithm 2 Constructing the team belief DAG form.

function $\mathcal{D}$.MAKEDECISIONNODE($B \subseteq \mathcal{H}$)
  if $\mathcal{D}$ has a decision node s with belief B **then**
    return s
  $s \leftarrow$ new decision node in $\mathcal{D}$ with belief B
  if B contains terminal nodes in $\mathcal{T}$ **then**
    for each $z \in B$ **do** $\mathcal{D}.\Sigma(zs) \leftarrow s$
    add sh to $\mathcal{Z}_\mathcal{D}$
    return s
  $\{I_1, \dots, I_m\} \leftarrow \{I \ni h : h \in B, I \in \mathcal{I}\}$
  $J \leftarrow \{h \in B : h$ is an observation node$\}$
  for each prescription $\boldsymbol{a} \in \bigtimes_{i \in [m]} A_{I_i}$ **do**
    $B\boldsymbol{a} \leftarrow \{ha_i : h {\in} I_i {\cap} B\} \cup \{ha : h {\in} J, a {\in} A_h\}$
    $s' \leftarrow \mathcal{D}$.MAKEOBSERVATIONNODE($B\boldsymbol{a}$)
    add edge $s \rightarrow s'$
  return s
function $\mathcal{D}$.MAKEOBSERVATIONNODE($O \subseteq \mathcal{H}$)
  $s \leftarrow$ new observation node in $\mathcal{D}$
  $\{P_1, \dots, P_m\} \leftarrow \{P \ni h : h \in O, P \in \mathcal{P}\}$
  for $i \in [m]$ **do**
    $s' \leftarrow \mathcal{D}$.MAKEDECISIONNODE($O \cap P_i$)
    add edge $s \rightarrow s'$
  return s
function MAKETEAMBELIEFDAGFORM(
    team TFSDP $\mathcal{T}$ with node set $\mathcal{H}$)
  $\mathcal{D} \leftarrow$ new decision problem
  $\mathcal{D}$.MAKEDECISIONNODE($\{\emptyset_{\in \mathcal{H}}\}$)

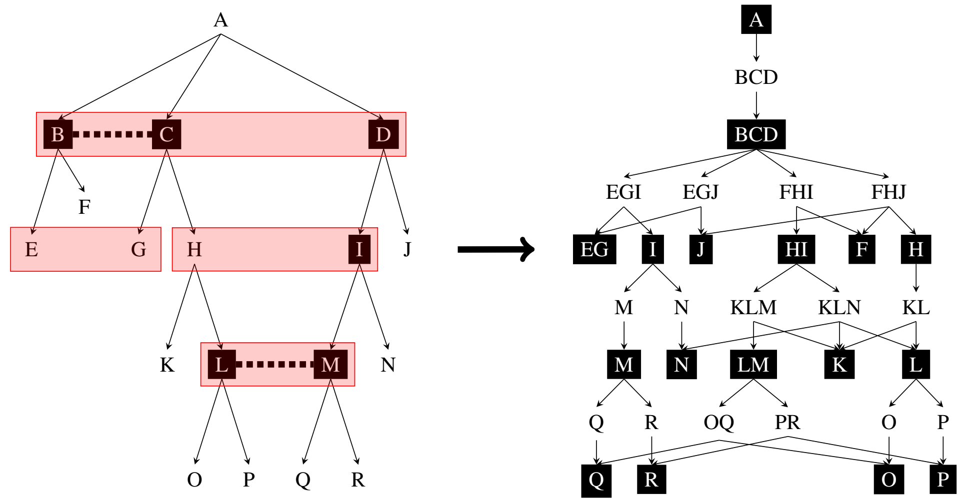


Figure 1: An example decision problem of a team (left) and its team belief DAG form (right). Decision nodes have light text on a dark background; observation nodes have the opposite. Dotted lines connect decision nodes in the same infoset. Red shaded regions connect nodes in the same public state. The team decision problem can be viewed as having two (BCDI and LM), three (e.g., BCD, I, and LM), or four (one at each infoset) players on the team. Which view is chosen is not relevant to our algorithms.

A Experiments Table

Experimental results are summarized in Table 1.

Column '**Game**' indicates the game, and the set of players on Team $\ominus$. Columns '**Team $\oplus$'s DAG**' and '**Team $\ominus$'s DAG**' report the total number of vertices and edges in the team belief DAG for teams $\oplus$ and $\ominus$ respectively. Column '**Team $\oplus$ value**' reports the utility that team $\oplus$ can expect to gain at equilibrium. Column '**P. S. Size**' reports the largest effective size p of any public state. Column '***k***' reports the value of k for which both teams are k-private.

Column '**This paper, CFR**' of Table 1 reports the time to convergence of the *best* CFR variant to an average team exploitability of less than 10^{-3} times the range of payoffs of the game. Convergence plots for all CFR variants on all games can be found in the appendix. Column '**This paper, LP**' reports the runtime of the *best* of the LP algorithms (see Section 8.1) operating on the belief DAG form of the game, solving to to Gurobi's default precision.[8]

Column '**ZS22**' reports the time it took ZS22 to compute an equilibrium strategy for team $\oplus$, again to Gurobi's default precision. Finally, column '**FCGS21**' reports the time it took FCGS21 to compute an equilibrium strategy for team $\oplus$ with exploitability of less than 10^{-3} times the range of payoffs of the game. The missing values in that column are due to the fact that the implementation of FCGS21 by the original authors only supported 3-player games.

B Branching Factor Reduction

Since the branching factor b appears as the base of an exponential in Theorems 6.1 and 6.3, it is natural to ask whether it can reduced without affecting the other parameters. This turns out to be true assuming *team-public actions*, which we now formalize.

[8]LP-based algorithms generally do not produce feasible iterates, and therefore they are not anytime algorithms for the purposes of computing strong team strategies, unlike our CFR-based algorithms and FCGS21. Thus, we report only the runtime to default precision.

Game {⊖}	**Leaves**	**Team ⊕'s DAG** Vertices	Edges	**Team ⊖'s DAG** Vertices	Edges	**Team ⊕ Value**	**P. S. Size**	***k***	**This paper** CFR	LP	**ZS22**	**FCGS21**
^{3}K3 {3}	78	502	957	37	36	0.000	6	6	0.00s	0.00s	0.01s	0.00s
^{3}K4 {3}	312	2,130	6,789	49	48	−0.042	12	8	0.00s	0.01s	0.02s	0.04s
^{3}K6 {3}	1,560	54,330	337,139	73	72	−0.024	30	12	0.03s	0.48s	1.12s	0.22s
^{3}K8 {3}	4,368	1,784,066	15,565,129	97	96	−0.019	56	16	4.27s	44.74s	3m 40s	0.75s
^{3}K12 {3}	17,160	—	—	—	—	−0.014	132	24	oom	oom	oom	1.76s
^{4}K5 {3,4}	3,960	26,716	125,265	4,621	15,415	−0.037	20	10	0.03s	0.65s	2.01s	—
^{4}K5 {4}	3,960	1,005,711	4,673,340	121	120	−0.030	60	15	1.92s	1m 31s	4m 5s	—
^{3}L133 {3}	6,477	23,983	49,005	685	684	0.215	9	6	0.02s	0.23s	0.60s	2m 24s
^{3}L143 {3}	20,856	139,964	417,027	1,201	1,200	0.107	16	8	0.09s	4.71s	9.64s	48m 1s
^{3}L151 {3}	10,020	153,607	501,036	1,501	1,500	−0.019	20	10	0.18s	6.26s	11.14s	14.37s
^{3}L153 {3}	51,215	855,397	3,486,091	1,861	1,860	0.024	25	10	1.45s	3m 46s	4m 19s	> 6h
^{3}L223 {3}	8,762	32,750	45,913	2,437	2,436	0.516	4	4	0.03s	0.13s	0.28s	51.94s
^{3}L523 {3}	775,148	2,911,352	4,183,685	220,705	220,704	0.953	4	4	10.28s	1m 19s	3m 56s	> 6h
^{4}L133 {3,4}	80,322	79,351	158,058	75,157	155,475	0.147	9	6	0.24s	5.58s	12.07s	—
^{3}D3 {3}	13,797	91,858	215,967	1,522	1,521	0.284	9	6	0.10s	0.74s	2.28s	5m 59s
^{3}D4 {3}	262,080	4,043,377	13,749,608	16,381	16,380	0.284	16	8	26.48s	3m 47s	9m 38s	> 6h
^{4}D3 {2,4}	331,695	514,072	1,232,775	485,986	1,168,029	0.200	9	6	2.18s	7.52s	1m 22s	—
^{6}D2 {2,4,6}	262,080	253,778	459,259	216,046	387,883	0.072	8	6	1.81s	2.76s	19.84s	—
^{6}D2 {4,6}	262,080	988,801	2,025,010	46,592	61,563	0.265	16	8	4.00s	14.14s	2m 11s	—
^{6}D2 {6}	262,080	3,131,832	7,356,087	5,551	5,550	0.333	32	10	27.54s	25.80s	8m 20s	—
^{3}G {3}	1,296	1,531	1,530	505	504	1.253	1	2	0.00s	0.00s	0.08s	1.02s
^{4}G {4}	7,776	10,441	10,440	1,477	1,476	1.780	1	3	0.01s	0.01s	0.85s	—
^{5}G {4,5}	46,656	30,853	30,852	13,195	13,194	0.600	1	3	0.01s	0.06s	7.04s	—
^{3}GL {3}	1,296	1,207	1,206	289	288	1.252	1	2	0.00s	0.00s	0.06s	0.53s
^{4}GL {4}	7,776	7,921	7,920	577	576	1.780	1	3	0.00s	0.01s	0.55s	—
^{5}GL {4,5}	46,656	16,489	16,488	5,095	5,094	0.600	1	3	0.01s	0.04s	3.72s	—

Table 1: Experimental investigation of the runtime of our CFR- and LP-based algorithms (columns '**This paper**') using the team belief DAG form, compared to the prior state-of-the-art algorithms by Zhang & Sandholm (2022) ('**ZS22**') and Farina et al. (2021a) ('**FCGS21**'), on several standard parametric benchmark games. See the beginning of Appendix A for a detailed description of the meaning of each column, and Section 8.2 for a description of the games. Missing or unknown values are denoted with '—'. For each row (benchmark game), the background color of each runtime column is set proportionally to the ratio R with the best runtime for the row, according to the logarithmic color scale 1 10 ≥ 100.

Definition B.1. A team TFSDP has *team-public actions* if, for all public states P containing decision nodes, for all edge labels (*i.e.*, actions or observations) $a \in \bigcup_{h \in P} A_h$, the set $\{ha : h \in P, a \in A_h\}$ is a union of public states.

Intuitively, this means that any action taken by a team member becomes common knowledge for the team. The definition also allows for information other than the action to become common knowledge, and for some public states to give the team members private information.

Theorem B.2. *Given a team TFSDP $\mathcal{T}$ with public actions, there exists another realization-equivalent team decision problem $\mathcal{T}'$ such that the branching factor of $\mathcal{T}'$ is at most 2 at each decision node, the parameters p, n, k, w in $\mathcal{T}'$ are the same as in $\mathcal{T}$, and $|\mathcal{P}|$ has increased by at most a constant factor.*

Corollary B.3. *In a team TFSDP $\mathcal{T}$ with public actions, it is possible to create a team DAG for $\mathcal{T}$ with $O((2p+2)^w b' |\mathcal{P}|)$ or $O(3^k b' |\mathcal{P}|)$ edges.*

C PROOFS

C.1 THEOREM 5.3

We will show the claim for pure strategies, which is enough since mixed and correlated strategies come from taking convex combinations of pure strategies.

($\Rightarrow$) Consider a pure strategy π in $\mathcal{T}$, inducing a correlation plan $\boldsymbol{x}$. Consider the pure strategy in the team belief DAG form in which the team chooses the prescription in each belief consistent with π, inducing a sequence form $\boldsymbol{x}'$.

Let z be a terminal node in $\mathcal{T}$, and suppose $x[z] = 1$. We need to demonstrate a path through $\mathcal{D}$ leading to $\Sigma(s)$ such that π plays every action prescribed along that path. Consider the path through $\mathcal{D}$ defined by following the prescriptions of π, and always selecting the public state that leads to z. By construction of the team belief DAG form, this path must end exactly at z, so $x'[\Sigma(z)] = 1$.

Conversely, suppose that such a path exists. Then, every infoset $I \preceq z$ must have appeared in some belief node B along the path, and, at that belief node, in order for $\Sigma(z)$ to still have been reachable, the team must have chosen the action at I leading to z. Thus, the team plays all actions on the path from the root to z, so $x[z] = 1$.

($\Leftarrow$) Consider a pure strategy π' in $\mathcal{D}$, and let $\boldsymbol{x}'$ be its sequence form. Define the pure strategy π in $\mathcal{T}$ as follows. In each public state P, π' induces a unique belief $B \subseteq P$. Let $\boldsymbol{a}$ be the prescription in π' at B. At every infoset $I \subseteq P$, define $\pi(I) = a_I$ if I intersects B, and arbitrarily otherwise. We claim that π defined in this way is realization-equivalent to π'. As before, let $\boldsymbol{x}$ be the correlation plan of π.

Suppose $x'[\Sigma(z)] = 1$; that is, there is a path through $\mathcal{D}$ ending at $\Sigma(z)$ at which π' plays every prescription. Then, at each public state P along this path, the belief $B \subseteq P$ induced by π' must be exactly the unique belief used to construct π at P. Thus, in particular, if P contains an infoset $I \preceq z$, then $I \cap B \neq \emptyset$. Thus, the team plays all actions on the root $\to z$ path, so $x[z] = 1$.

Conversely, suppose $x[z] = 1$. Then, a straightforward induction shows that, at every public state P on the root $\to z$ path, the belief at B induced by π' must contain an ancestor of z and thus must be played to. This completes the proof.

C.2 Theorem 6.1

A belief $B \subseteq P$ of a public state $P \in \mathcal{P}$ is uniquely identified by its sequence set $\sigma(B)$. We have $|\sigma(B)| \le w$ by definition. Hence, there are at most $\binom{p}{\le w}$ such beliefs, and at most b^w prescriptions $\boldsymbol{a}$ at B. For each prescription $\boldsymbol{a}$, the children of $B\boldsymbol{a}$ must be public state children of P; in particular, there are at most b' of them. Multiplying these gives the desired result.

C.3 Theorem 6.3

Consider a public state P. Each of the k player sequences corresponds to at most one information set in P. Thus, to specify that player sequence's contribution to a belief-prescription pair $B\boldsymbol{a}$, it suffices to specify one of: either the player does not play to the sequence, or the player chooses one of her b possible actions at the sequence. Thus, there are at most $(b+1)^k$ possible belief-prescription pairs $B\boldsymbol{a}$. Each one has at most b' children as argued before. The bound follows.

C.4 Theorem B.2

Consider a public state P of $\mathcal{T}$. If $\mathcal{P}$ contains no decision nodes, we leave it alone. Otherwise, let $\mathcal{B}$ be an arbitrary binary tree with leaf set $A := \bigcup_{h \in P} A_h$. The internal nodes of $\mathcal{B}$ will be labelled with *partial actions* $\tilde{a}$, which we can think of as partial bitstrings of indices of actions in A. For each node $h \in P$, we replace h with a modified copy of $\mathcal{B}$ wherein subtrees containing no nodes in A_h have been pruned. If h and h' are in the same infoset in P, then for every partial action (i.e., nonterminal node) $\tilde{a} \in \mathcal{B}$ we connect $h\tilde{a}$ and $h'\tilde{a}$ in an infoset. This creates a new public tree $\mathcal{T}'$, whose parameters we must now analyze.

For each public state P of $\mathcal{T}$, the construction creates the internal nodes of a public subtree in $\mathcal{T}'$ with leaves corresponding to the child public states of P. This adds at most as many public states as there are children of P in $\mathcal{T}$, so the number of public states in $\mathcal{T}'$ is at most twice that of $\mathcal{T}$.

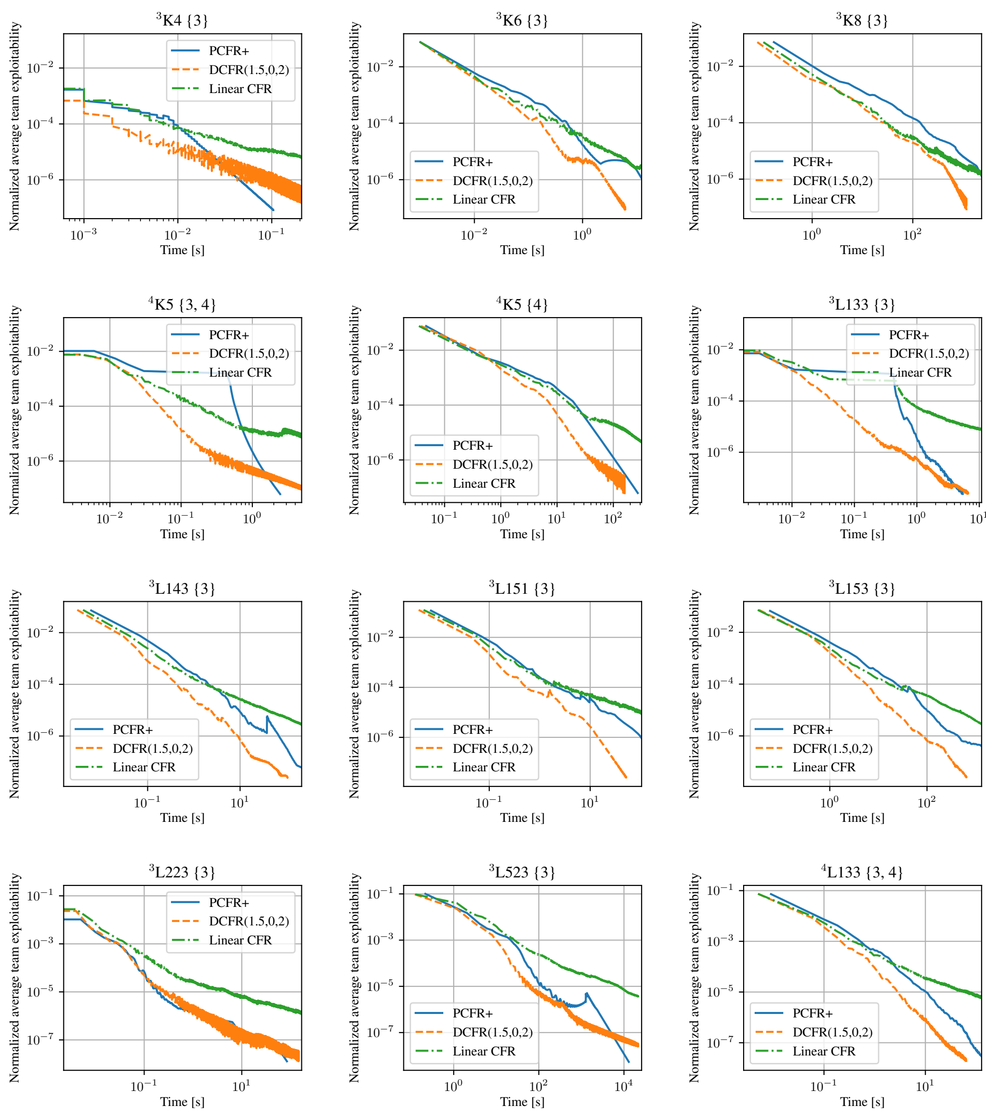
The number of players n remains the same.

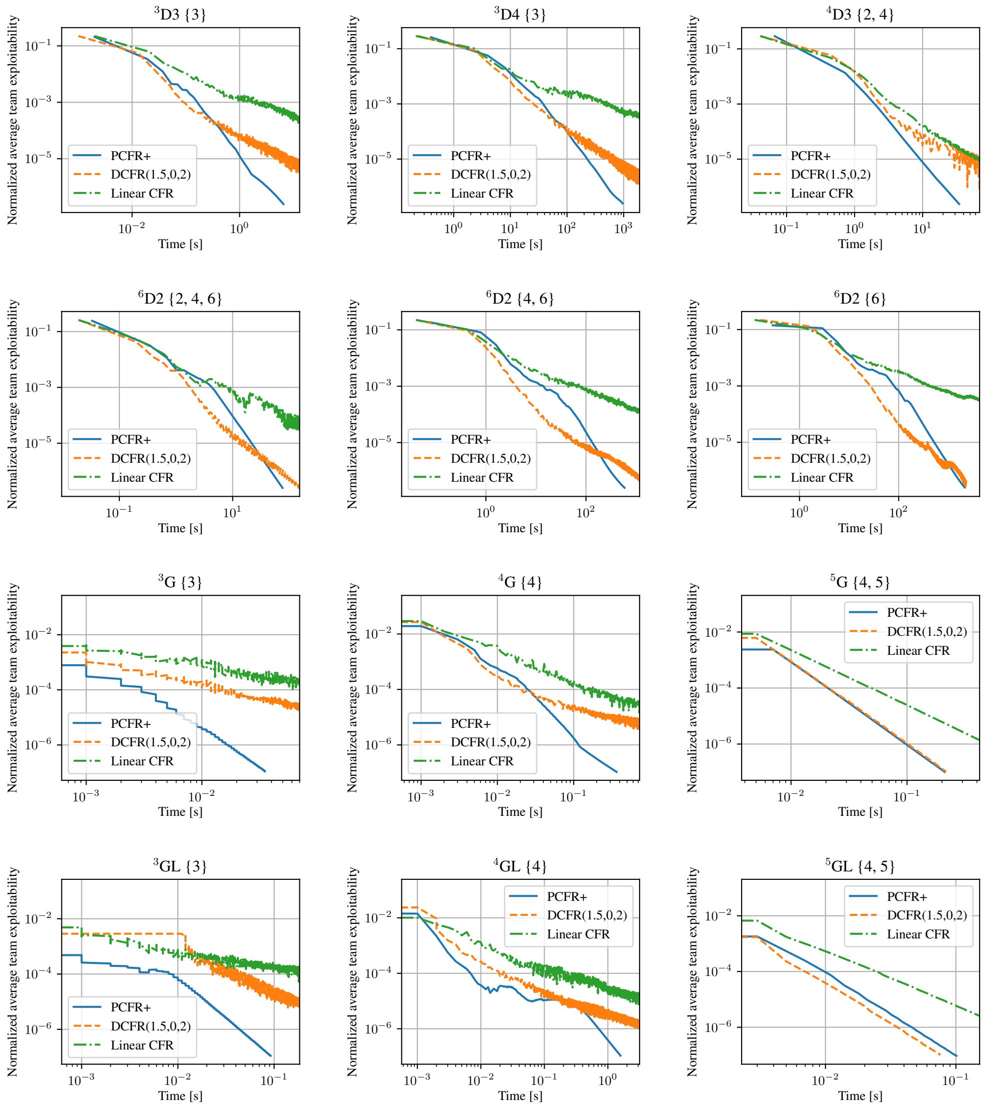
For each new public state P^* constructed in this process, we have $P^* \subseteq P\tilde{a} := \{h\tilde{a} : h \in P, \tilde{a} \preceq a \in A_h\}$ for some $\tilde{a}$, where $\preceq$ denotes precedence in $\mathcal{B}$ (the subset may not be the whole set, because it is possible for the partial action $\tilde{a}$ to have already revealed further common knowledge that was not available at P). Thus, every team or player sequence $\sigma(h)$ in P identifies at most one unique team or player sequence in P^*—namely $\sigma(h\tilde{a})$, if present—and thus p and k have not increased.

Finally, for each belief $B \subseteq P$, the largest belief in $P\tilde{a}$ induced by B is $B\tilde{a}$, which has no larger effective size. Hence, w has not increased. This completes the proof.

D CFR Convergence Plots

In this section, we show the performance of each of the three CFR variants that we implemented to perform no-regret learning on the team belief DAG. As a rule of thumb, the predictive algorithm PCFR+ (Farina et al., 2021b) is fastest when high precision (low team exploitability) is necessary. For low precision, DCFR (Brown & Sandholm, 2019) is often the fastest algorithm in practice, especially in certain variants of Kuhn poker.




^{3}D3 {3}
^{3}D4 {3}
^{4}D3 {2, 4}
^{6}D2 {2, 4, 6}
^{6}D2 {4, 6}
^{6}D2 {6}
^{3}G {3}
^{4}G {4}
^{5}G {4, 5}
^{3}GL {3}
^{4}GL {4}
^{5}GL {4, 5}
Normalized average team exploitability
Time [s]
PCFR+
DCFR(1.5,0,2)
Linear CFR