Team Correlated Equilibria in Zero-Sum Extensive-Form Games via Tree Decompositions

Brian Hu Zhang,[1] **Tuomas Sandholm**[1,2,3,4]
[1]Computer Science Department, Carnegie Mellon University
[2]Strategic Machine, Inc.
[3]Strategy Robot, Inc.
[4]Optimized Markets, Inc.
{bhzhang, sandholm}@cs.cmu.edu

Abstract

Despite the many recent practical and theoretical breakthroughs in computational game theory, equilibrium finding in extensive-form team games remains a significant challenge. While NP-hard in the worst case, there are provably efficient algorithms for certain families of team game. In particular, if the game has *common external information*, also known as *A-loss recall*—informally, actions played by non-team members (i.e., the opposing team or nature) are either unknown to the entire team, or common knowledge within the team—then polynomial-time algorithms exist (Kaneko and Kline 1995). In this paper, we devise a completely new algorithm for solving team games. It uses a tree decomposition of the constraint system representing each team's strategy to reduce the number and degree of constraints required for correctness (tightness of the mathematical program). Our approach has the bags of the tree decomposition correspond to team-public states—that is, minimal sets of nodes (that is, states of the team) such that, upon reaching the set, it is common knowledge among the players on the team that the set has been reached. Our algorithm reduces the problem of solving team games to a linear program with at most $O(NW^{w+1})$ nonzero entries in the constraint matrix, where N is the size of the game tree, w is a parameter that depends on the amount of *uncommon* external information, and W is the treewidth of the tree decomposition. In *public-action games*, our program size is bounded by the tighter $2^{O(nt)}N$ for teams of n players with t types each. Our algorithm is based on a new way to write a custom, concise tree decomposition, and its fast run time does not assume that the decomposition has small treewidth. Since our algorithm describes the polytope of correlated strategies directly, we get equilibrium finding in correlated strategies for free—instead of, say, having to run a double oracle algorithm. We show via experiments on a standard suite of games that our algorithm achieves state-of-the-art performance on all benchmark game classes except one. We also present, to our knowledge, the first experiments for this setting where both teams have more than one member.

1 Introduction

Computational game solving in imperfect-information games has led to many recent superhuman breakthroughs in AI (e.g., Bowling et al. 2015; Brown and Sandholm 2018, 2019). Most of the literature on this topic focuses on *two-player zero-sum games with perfect recall*—that is, two players who *never forget any information during the game* face off in an adversarial manner. While this model is broad enough to encompass games such as poker, it breaks in the setting of *team games*, in which two *teams* of players face off against each other. While members of the team have perfect recall, the team as a whole may not, because different members of the team may know different pieces of information. Situations that fall into this category include recreational games such as contract bridge, Hanabi (in which there is no adversary), collusion in poker, and all sorts of real-world scenarios in which communication is limited. In this paper, we focus on such games.

We will assume that team members can coordinate before the game begins, including generating randomness that is shared within the team but hidden from the opposing team; in other words, they can *correlate* their strategies. Once the game begins, they can only exchange information by playing actions within the game. An alternative way of interpreting the setting is to consider zero-sum games with *imperfect recall* (e.g., Kaneko and Kline 1995), that is, games in which players may forget information that they once knew. In this interpretation, each team is represented by a single player whose "memory" is continuously refreshed to that of the acting team member. The two interpretations are equivalent.

In general, computing equilibria in such games is NP-hard (Chu and Halpern 2001). However, some subfamilies of games are efficiently solvable. For example, if both teams have *common external information*, also known as *A-loss recall* (Kaneko and Kline 1995) (in which all uncommon knowledge of the team can be attributed to team members not knowing about actions taken by other team members), or both teams have at most two players and the interaction between team members is *triangle-free* (Farina and Sandholm 2020) (which roughly means that the team's strategy tree can be re-ordered into a strategically equivalent tree obeying A-loss recall), then each player's strategy space can be described as the projection of a polytope with polynomially many constraints in the size of the game, and hence the game can be solved in polynomial time.

Practical algorithms for solving medium-sized instances of these games (Celli and Gatti 2018; Farina et al. 2018; Zhang, An, and Černỳ 2021; Farina et al. 2021) primarily focused on the case in which there is a team of players play-

---

ing against a single adversary. These algorithms are mostly based on *column generation*, or *single oracle*, and require a best-response oracle that is implemented in practice by an integer program. While they perform reasonably in practice, they lack theoretical guarantees on runtime. Although these techniques can be generalized naturally to the case of two teams using *double oracle* (McMahan, Gordon, and Blum 2003) in place of their column generation methods, we do not know of a paper that explores this more general case.

In this paper, we demonstrate a completely new approach to solving team games. From a game, we first construct a custom, concise *tree decomposition* (Robertson and Seymour 1986; for a textbook description, see Wainwright and Jordan 2008) for the constraint system that defines the strategy polytope of each player. Then, we bound the number of feasible solutions generated in each tree decomposition node, from which we derive a bound on the size of the representation of the whole strategy space. Our bound is linear in the size of the game tree, and exponential in a natural parameter w that measures the amount of uncommon external information. We also show a tighter bound in games with public actions (such as poker). Since our algorithm describes the polytope of correlated strategies directly, we get equilibrium finding in correlated strategies for free—instead of, say, having to run a double oracle algorithm. We show via experiments on a standard suite of games that our algorithm is state of the art in most game instances, with failure modes predicted by the theoretical bound. We also present, to our knowledge, the first experiments for the setting where both teams have more than one member.

Some papers (e.g., Daskalakis and Papadimitriou 2006) have explored the use of tree decompositions to solve *graphical games*, which are general-sum, normal-form games in which the interactions between *players* are described by a graph. Our setting, and thus the techniques required, are completely different from this line of work.

2 Preliminaries

In this section, we will introduce background information about extensive-form games and tree decompositions.

2.1 Extensive-Form Games

Definition 2.1. A *zero-sum extensive-form team game* (EFG) Γ, hereafter simply *game*, between two teams $\oplus$ and $\ominus$ consists of the following:

(1) A finite set H, of size $|H| := N$, of *histories* of vertices of a tree rooted at some *initial state* $\textsc{Root} \in H$. The set of leaves, or *terminal states*, in H will be denoted Z. The edges connecting any node $h \in H$ to its children are labeled with *actions*. The child node created by following action a at node h will be denoted ha.

(2) A map $P : (H \setminus Z) \to \{\oplus, \ominus, \textsc{Nature}\}$, where $P(h)$ is the team who acts at node h. A node at which team T acts is called an T*-node*.

(3) A *utility function* $u : Z \to \mathbb{R}$.

(4) For each team $T \in \{\oplus, \ominus\}$, a partition $\mathcal{I}_T$ of the set of T-nodes into *information sets*, or *infosets*. In each infoset $I \in \mathcal{I}_T$, every pair of nodes $h, h' \in I$ must have the same set of actions.

(5) For each nature node h, a distribution $p(\cdot|h)$ over the actions available to nature at node h.

It will sometimes be convenient to discuss the individual *players* on a team. In this context, for each team $T \in \{\oplus, \ominus\}$, we will assume that T itself is a set of distinct players, and there is a map $P_T : \mathcal{I}_T \to T$ denoting which member of the team plays at each infoset $I \in \mathcal{I}_T$.

We will use the following notational conventions: $A(h)$ or $A(I)$ denotes the set of available actions at a node h or infoset I. $\preceq$ denotes the partial order created by the tree: if h, h' are nodes, infosets or sequences, $h \preceq h'$ means h is an ancestor of h' or $h' = h$. If S is a set of nodes, $h \succeq S$ (resp. $h \preceq S$) means $h \succeq h'$ (resp. $h \preceq h'$) for some $h' \in S$. If I is an infoset and a is an action at I, then $Ia = \{ha : h \in I\}$. $\wedge$ denotes the lowest common ancestor relation: $h \wedge h'$ is the deepest node h^* of the tree for which $h^* \preceq h, h'$. At a given node h, the *sequence* $s_i(h)$ for a team or player i is the list of infosets reached and actions played by i up to node h, including the infoset at h itself if i plays at h.

We will assume that each individual player on a team has perfect recall, that is, for each player $i \in T$, each infoset I with $P_T(I) = i$, and each pair of nodes $h, h' \in I$, we must have $s_i(h) = s_i(h')$. Of course, the team as a whole may not have perfect recall. We will also assume that the game is *timeable*, *i.e.*,, every node in a given infoset is at the same depth of the game tree. While this assumption is not without loss of generality, most practical games satisfy it.

A *pure strategy* σ for a team $T \in \{\oplus, \ominus\}$ is a selection of one action from the action space $A(I)$ at every infoset $I \in \mathcal{I}_T$. The *realization plan* (Farina et al. 2018), or simply *plan*, $\boldsymbol{x}_\sigma$ corresponding to σ is the vector $\boldsymbol{x}^\sigma \in \{0,1\}^H$ where $x^\sigma_h = 1$ if $\sigma(I) = a$ for every $Ia \preceq h$. A *correlated strategy* $\tilde{\sigma}$ is a distribution over pure strategies. The plan $\boldsymbol{x}$ corresponding to $\tilde{\sigma}$ is the vector $\boldsymbol{x} \in [0,1]^H$ where $x_h = \mathbb{E}_{\sigma \sim \tilde{\sigma}}\, x^\sigma_h$. The spaces of plans for teams $\oplus$ and $\ominus$ will be denoted $\mathcal{X}$ and $\mathcal{Y}$ respectively. Both $\mathcal{X}$ and $\mathcal{Y}$ are compact, convex polytopes.

A *strategy profile* is a pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$. The *value* of the strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ for $\oplus$ is $u(\boldsymbol{x}, \boldsymbol{y}) := \sum_{z \in Z} u_z p_z x_z y_z$, where p_z is the probability that nature plays all the actions needed to reach z: $p_z := \prod_{ha \preceq z : P(h) = \textsc{Nature}} p(a|h)$. Since the game is zero-sum, the payoff for $\ominus$ is $-u(\boldsymbol{x}, \boldsymbol{y})$. The *best response value* for $\ominus$ (resp. $\oplus$) to a strategy $\boldsymbol{x} \in \mathcal{X}$ (resp. $\boldsymbol{y} \in \mathcal{Y}$) is $u^*(\boldsymbol{x}) := \min_{\boldsymbol{y} \in \mathcal{Y}} u(\boldsymbol{x}, \boldsymbol{y})$ (resp. $u^*(\boldsymbol{y}) := \max_{\boldsymbol{x} \in \mathcal{X}} u(\boldsymbol{x}, \boldsymbol{y})$). A strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ is a *team correlated equilibrium*, or simply *equilibrium*, if $u^*(\boldsymbol{x}) = u(\boldsymbol{x}, \boldsymbol{y}) = u^*(\boldsymbol{y})$. Every equilibrium of a given game has the same value, which we call the *value of the game*. Equilibria in zero-sum team games can be computed by solving the bilinear saddle-point problem

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{y} \in \mathcal{Y}} u(\boldsymbol{x}, \boldsymbol{y}), \tag{2.2}$$

where $u(\boldsymbol{x}, \boldsymbol{y})$ can be expressed as a bilinear form $\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$ in which $\boldsymbol{A}$ has $O(N)$ nonzero entries.

One may wonder why we insist on allowing players to correlate. Indeed, alternatively, we could have defined *un-*

correlated strategies and equilibria, in which the distribution over pure strategies of each team is forced to be a product distribution over the strategy spaces of each player. However, in this case, the strategy space for both players becomes nonconvex, and therefore we may not even have equilibria at all! Indeed, if $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are the spaces of plans for uncorrelated strategies for each team, Theorem 7 in Basilico et al. (2017) implies that it is possible for

$$\max_{\boldsymbol{x}\in\tilde{\mathcal{X}}} \min_{\boldsymbol{y}\in\tilde{\mathcal{Y}}} u(\boldsymbol{x},\boldsymbol{y}) \neq \min_{\boldsymbol{x}\in\tilde{\mathcal{Y}}} \max_{\boldsymbol{x}\in\tilde{\mathcal{X}}} u(\boldsymbol{x},\boldsymbol{y}),$$

which makes it difficult to *define* the problem, much less solve it. Some authors (e.g., Basilico et al. 2017) have defined "team maxmin equilibria" in these games for the case where there is one opponent, i.e., $|\ominus| = 1$, by assuming that the team $\oplus$ plays first. In the present paper, we study zero-sum games between two *teams*—as such, due to symmetry between the teams, we have no reason to favor one team over the other, and thus cannot make such an assumption.

If $\mathcal{X}$ and $\mathcal{Y}$ can be described succinctly by linear constraints, the problem (2.2) can be solved by taking the dual of the inner minimization and solving the resulting linear program (Luce and Raiffa 1957). Unfortunately, such a description cannot exist in the general case unless $\mathsf{P} = \mathsf{NP}$:

Theorem 2.3 (Chu and Halpern 2001)**.** *Given a zero-sum team game and a threshold value u^*, determining whether the game's value is at least u^* is NP-hard, even when $\oplus$ has two players and there is no adversary.*

For completeness, we include proofs of results in this section in the appendix. Despite the hardness result, it is sometimes possible to express $\mathcal{X}$ and $\mathcal{Y}$ using polynomially many constraints, and hence solve (2.2) efficiently. One of these cases is the following:

Definition 2.4. A team T in a game has *common external information*, also known as *A-loss recall* (Kaneko and Kline 1995), if, for any two nodes h, h' in the same infoset I of team T, either $h_T = h'_T$, or there exists an infoset I' and two actions $a \neq a'$ at I' such that $h \succeq I'a$ and $h' \succeq I'a'$.

Informally, if a team has common external information, each member of that team has the same knowledge about actions taken by non-team members (i.e., the opponent and nature). In this case, the *perfect-recall refinement* of that team's strategy space, which is created by splitting infosets to achieve perfect recall, is strategically equivalent with respect to correlation plans. Thus, equilibria can be found efficiently when both teams have it.

In this paper, we expand and refine these complexity results. In particular, we show that the polytope of plans for a team can be expressed as a projection of a polytope with $O(NW^{w+1})$ constraints and variables, where w is a parameter describing the amount of information *generated by non-team members* that is known by *at least one member, but not all members*, of the team, and W is the treewidth of a particular tree decomposition. In particular, $w = 1$ for games with common external information, so our result is equivalent to the known result in that case.

2.2 Tree Decompositions, Integer Hulls, and Dependency Graphs

We now review tree decompositions and their use in parameterizing integer hulls. We refer the reader to Wainwright and Jordan (2008) for further reading on these topics.

Definition 2.5. Let G be a graph. A *tree decomposition*, also known as a *junction tree* or *clique tree*, is a tree $\mathcal{J}$, whose vertices are subsets of vertices of G, called *bags*, such that

(1) for every edge (u, v) in G, there is a bag in $\mathcal{J}$ containing both u and v, and
(2) for every vertex[1] $u \in G$, the subset of bags $\{U \in \mathcal{J} : u \in U\}$ is nonempty and connected in $\mathcal{J}$.

The *width* of $\mathcal{J}$ is the size of the largest bag, minus one.[2] Now, consider a convex set of the form[3]

$$\begin{aligned}&\mathcal{X} = \operatorname{conv}(X), \quad \text{where}\\ &X = \{\boldsymbol{x} \in \{0,1\}^n : g_k(\boldsymbol{x}) = 0 \;\; \forall k \in [m]\}\end{aligned}$$

and the constraints $g_k : \{0,1\}^n \to \mathbb{R}$ are arbitrary polynomials. This formulation is very expressive; for example, it is possible to express realization plan polytopes in this form where X is the set of pure plans, as we will show later. For notation, in the rest of the paper, if $\boldsymbol{x} \in \mathbb{R}^n$ and $U \subseteq [n]$, we will use $\boldsymbol{x}_U$ to denote the subvector of $\boldsymbol{x}$ on the indices in U.

Definition 2.6. The *dependency graph* of $\mathcal{X}$ is the graph whose vertices are the indices $i \in [n]$, and for which an edge connects $i, j \in [n]$ if there is a constraint g_k in which both variables x_i and x_j appear.

Definition 2.7. Let $U \subseteq [n]$, and $\tilde{\boldsymbol{x}} \in \{0,1\}^U$. We say that $\tilde{\boldsymbol{x}}$ is *locally feasible* on U if $\tilde{\boldsymbol{x}} = \boldsymbol{x}_U$ for some $\boldsymbol{x} \in X$.

We will use X_U to denote the set of locally feasible vectors on U. Of course, $X_{[n]} = X$. The following result follows from the junction tree theorem (e.g., Wainwright and Jordan 2008):

Proposition 2.8. *Let $\mathcal{J}$ be a tree decomposition of the dependency graph of $\mathcal{X}$. Then $\boldsymbol{x} \in \mathcal{X}$ if and only if there exist vectors[4] $\boldsymbol{\lambda}^U \in \Delta^{X_U}$ for each $U \in \mathcal{J}$ satisfying*

$$\begin{aligned}\boldsymbol{x}_U &= \sum_{\tilde{\boldsymbol{x}}\in X_U} \lambda^U_{\tilde{\boldsymbol{x}}} \tilde{\boldsymbol{x}} && \forall \text{ bags } U \in \mathcal{J}\\ \sum_{\substack{\tilde{\boldsymbol{x}}\in X_U:\\ \tilde{\boldsymbol{x}}_{U\cap V}=\tilde{\boldsymbol{x}}^*}} \lambda^U_{\tilde{\boldsymbol{x}}} &= \sum_{\substack{\tilde{\boldsymbol{x}}\in X_V:\\ \tilde{\boldsymbol{x}}_{U\cap V}=\tilde{\boldsymbol{x}}^*}} \lambda^V_{\tilde{\boldsymbol{x}}} && \begin{array}{c}\forall \text{ edges } (U,V) \text{ in } \mathcal{J},\\ \text{locally feasible}\\ \tilde{\boldsymbol{x}}^* \in X_{U\cap V}.\end{array}\end{aligned} \tag{2.9}$$

The constraint system has size[5] $O((W + D)\sum_{U\in\mathcal{J}} |X_U|)$, where W is the treewidth of $\mathcal{J}$ and D is the maximum degree of any bag in J.

---

[1] We will use $u \in G$ to mean u is a vertex in G.

[2] The -1 is so that trees have treewidth 1.

[3] conv denotes the convex hull operator. $[m] = \{1, \dots, m\}$.

[4] Δ^S is the set of distributions on set S

[5] Throughout this paper, *size* of a program refers to the number of nonzero entries in its constraint matrix.

Proposition 2.8 establishes a method of parameterizing convex combinations $\boldsymbol{x}$ over a set $X \subseteq \{0,1\}^n$. First, write down a tree decomposition of the constraint system defining X. Then, for each bag U in the tree decomposition, parameterize a *local distribution* over the set of locally feasible solutions X_U, and insist that, for adjacent bags U, V in $\mathcal{J}$, the distributions $\boldsymbol{\lambda}^U$ and $\boldsymbol{\lambda}^V$ agree on the marginal on $U \cap V$. Thus, given an arbitrary set $\mathcal{X}$ of the given form, to construct a small constraint system that defines $\mathcal{X}$, it suffices to construct a tree decomposition of its constraint graph.

In most use cases of tree decompositions, the next step is to bound the treewidth, and then use the bound $|X_U| \le 2^{W+1}$ to derive the size of the constraint system (2.9). In our domain, it will turn out that W can be too large to be useful; hence, we will instead *directly* bound $|X_U|$.

3 Tree Decompositions for Team Games

We now move to presenting our results. We take the perspective of the maximizing team $\oplus$. All of the results generalize directly to the opposing team, $\ominus$, by swapping $\oplus$ for $\ominus$ and $X, \mathcal{X}$ for $Y, \mathcal{Y}$.

Our approach has the bags of the tree decomposition correspond to team-public states—that is, minimal sets of nodes (that is, states of the team) such that, upon reaching the set, it is common knowledge among the players on the team that the set has been reached. This is similar, but not identical, to the notion of public tree, which instead decomposes based on knowledge common to all players, including opponents. Next we will present our approach formally.

We will assume that no two nodes $h \neq h'$ at the same depth of the tree have $\mathsf{seq}_\oplus(h) = \mathsf{seq}_\oplus(h')$. That is, there is no information known by no members of the team. We will also assume that the information partition of $\oplus$ has been *completely inflated* (Kaneko and Kline 1995)—in particular, we will assume that, if $\oplus$ has common external information, then $\oplus$ has perfect recall. These assumptions can be satisfied without loss of generality by merging nodes with the same sequence in the game tree, and performing inflation operations as necessary, before the tree decomposition.

Before defining the tree decomposition, we first need to define a notion of *public state*. Let $\sim$ be the following relation on pairs of nonterminal nodes: $h_1 \sim h_2$ if h_1 and h_2 are at the same depth *and* there is an infoset $I \in \mathcal{I}_\oplus$ with $h_1, h_2 \preceq I$. Let $\mathcal{J}_\oplus^\blacktriangle$ be the set of equivalence classes of the transitive closure of $\sim$. The elements of $\mathcal{J}_\oplus^\blacktriangle$ are the *public states* of the team: each public state is a minimal set of nodes such that, upon reaching that set, it is common knowledge among the team that the team's true node lies within that set. That is, if the true history of the team is in some $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$, then this fact is common knowledge among the team. Our notion differs from the usual notion (e.g., Kovařík et al. 2021), in two ways. First, our public states here are *restricted to the team*: the fact that a team is in a public state may not be known or common knowledge among members of the opposing team. Second, our public states are constructed from the game tree rather than supplied as part of the game description—as such, they may capture common knowledge that is not captured by public observations.

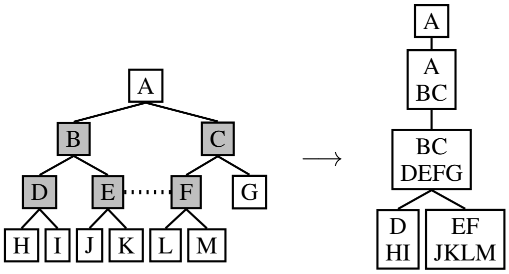

Figure 1: An example of a game with a single team, and its tree decomposition. Utilities are not shown as they are not relevant. Player nodes are shaded; the root (A) is a nature node. Information sets are joined by dotted lines. In each bag $U = U^\blacktriangle \cup U^\blacktriangledown$, the nodes in $U^\blacktriangle$ are shown in the first row, and the nodes in $U^\blacktriangledown$ are shown in the second row.

The space of feasible plans can be written as $\mathcal{X} = \operatorname{conv}(X)$, where X is the set of pure plans $\boldsymbol{x} \in \{0,1\}^H$, that is, plans satisfying:

$$\begin{aligned} x_h &= \sum_{a \in A(h)} x_{ha} && \forall \oplus\text{-nodes } h \\ x_h &= x_{ha} && \forall \text{ non-}\oplus\text{-nodes } h, \\ &&& \quad \text{actions } a \in A(h) \\ x_{ha} x_{h'} &= x_{h'a} x_h && \forall \text{ infosets } I \in \mathcal{I}_\oplus, \\ &&& \quad \text{nodes } h, h' \in I, \\ &&& \quad \text{actions } a \in A(I) \end{aligned} \tag{3.1}$$

We will never explicitly write a program using this constraint system; the only purpose of defining it is to be able to apply Proposition 2.8 to the resulting tree decomposition.

We now construct a tree decomposition of the dependency graph of (3.1). For a public state $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$, let $U^\blacktriangledown$ be the set of all children of nodes in $U^\blacktriangle$, that is, $U^\blacktriangledown := \{ha : h \in U, a \in A(h)\}$. Let $U = U^\blacktriangle \cup U^\blacktriangledown$.

Theorem 3.2. *The following is a tree decomposition $\mathcal{J}_\oplus$ of the constraint system* (3.1).

(1) The bags are the sets U for $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$, and the set $\{\text{ROOT}\}$ *containing only the root node.*

(2) There is an edge between bags U and V if $U \cap V \neq \emptyset$

Proof. We first check that $\mathcal{J}_\oplus$ is actually a tree. Since public states $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$ are disjoint sets, edges in $\mathcal{J}_\oplus$ must span across different depths. Therefore, it suffices to show that for every $U \in \mathcal{J}_\oplus$, there is only (at most) one edge from U to any bag at shallower depth. Let $u, v \in U^\blacktriangle$, and u' and v' be the parents of u and v respectively. It suffices to show that u' and v' are in the same public state. Since u and v are in the same public state, there is a sequence of nodes $v_0, \dots, v_k \in U^\blacktriangle$ such that $u = v_0 \sim v_1 \sim \dots \sim v_k = v$. Let v_i' be the parent of v_i. Then by definition of $\sim$, we have $u' = v_0' \sim v_1' \sim \dots \sim v_k' = v'$.

Now by definition of public state, any two nodes in the same infoset share the same public state, and for a nonterminal node $h \in U^\blacktriangle$, U by definition contains both h and all its children. Therefore, every constraint is contained in some bag. Finally, every node in the game tree appears in at most two bags U and U', where U' is the parent of U in the tree decomposition, and these are connected by an edge. We have thus checked all the required properties of a tree decomposition, so we are done. □

Therefore, it remains only to construct the sets X_U of locally feasible solutions on each U, and bound their sizes. The tree structure of $\mathcal{J}_\oplus$ is the *public tree* for the team. and an example can be found in Figure 1.

Algorithm 3.3: Constructing locally feasible sets

```
for each $U \in \mathcal{J}_\oplus$, in top-down order do
    if $U = \{\textsc{Root}\}$ then set $X_U \leftarrow \{1\}$ and continue
    let $U'$ be the parent of $U$ (by construction, $U^\blacktriangle \subset U'$)
    $X_{U^\blacktriangle} \leftarrow \{\tilde{\boldsymbol{x}}_{U^\blacktriangle} : \tilde{\boldsymbol{x}} \in X_{U'}\}$
    $X_U \leftarrow \emptyset$
    for each locally feasible solution $\tilde{\boldsymbol{x}} \in X_{U^\blacktriangle}$ do
        $\mathcal{I}_{\tilde{\boldsymbol{x}}} \leftarrow \{I \in \mathcal{I}_\oplus : I \subseteq U^\blacktriangle, I \cap \tilde{\boldsymbol{x}}^{-1}(1) \neq \emptyset\}$.
        for each partial strategy $\boldsymbol{a} \in \times_{I \in \mathcal{I}_{\tilde{\boldsymbol{x}}}} A(I)$ do
            $\tilde{\boldsymbol{x}}' \leftarrow \mathbf{0} \in \{0,1\}^U$
            for each $h \in U^\blacktriangle$ such that $\tilde{x}_h = 1$ do
                $\tilde{x}'_h \leftarrow 1$
                if $h \in I$ is a $\oplus$-node then $\tilde{x}'_{ha_I} \leftarrow 1$
                else for each $a \in A(h)$ do $\tilde{x}'_{ha} \leftarrow 1$
            add $\tilde{\boldsymbol{x}}'$ to $X_U$.
```

Algorithm 3.3 enumerates the locally feasible sets X_U in each bag $U \in \mathcal{J}_\oplus$ iteratively starting from the root. It has runtime $O(W \sum_{U \in \mathcal{J}_\oplus} |X_U|)$, where W is the treewidth[6] and a straightforward induction shows that it is correct. Therefore, chaining together Algorithm 3.3 and (2.9) to obtain a description of both players' polytopes $\mathcal{X}$ and $\mathcal{Y}$, and solving the resulting bilinear saddle-point problem by dualizing the inner minimization and using linear programming, we obtain:

Theorem 3.4 (Main Theorem)**.** *Team correlated equilibria in extensive-form team games can be computed via a linear program of size*

$$O\left(N + W \sum_{U \in \mathcal{J}_\oplus} |X_U| + W \sum_{U \in \mathcal{J}_\ominus} |Y_U|\right).$$

In Appendix C, we include the full constraint system for a player's strategy space in both the general case and, as an example, the game in Figure 1.

For intuition, we briefly discuss the special case where the team has perfect recall. In this case, after applying the assumptions without loss of generality, the team tree is (up to some redundancies) exactly the sequence-form tree for the team. Every public state $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$ has exactly one node, namely the information set node of the lone information set I at $U^\blacktriangle$, and the local feasible solutions $\tilde{\boldsymbol{x}} \in X_U$ correspond to sequences Ia for actions a at I. Thus, the LP given by Theorem 3.4 is (again up to some redundancies) exactly the sequence-form LP (Koller, Megiddo, and von Stengel 1994).

4 Bounding the Sizes of Locally Feasible Sets

On its own, Theorem 3.4 is not very useful: we still need to bound the sizes $|X_U|$ and $|Y_U|$. In the worst case, we will not be able to derive small bounds due to NP-hardness. However, we will now show some cases in which this result matches or tightens known results. In this section, we again take the perspective of a single team $\oplus$.

4.1 Low Uncommon External Information

Call a set $S \subseteq U^\blacktriangle$ *reachable* if there is a pure plan $\boldsymbol{x} \in X$ such that S is exactly the set of nodes in $U^\blacktriangle$ reached by $\boldsymbol{x}$, i.e., $S = \{h \in U^\blacktriangle : x_h = 1\}$. Let $w(U)$ be the largest size of any reachable subset of $U^\blacktriangle$. If $w(U)$ is small for every U, then we can bound the size of the linear program:

Theorem 4.1. *Team correlated equilibria in extensive-form team games can be computed via a linear program of size*

$$O\left(N + W \sum_{U \in \mathcal{J}_\oplus \cup \mathcal{J}_\ominus} |U|^{w(U)}\right) \le O(NW^{w+1}),$$

where we call $w := \max_{U \in \mathcal{J}_\oplus \cup J_\ominus} w(U)$ *the* reachable width *of* $\mathcal{J}_\oplus$.

Proof. Let $\boldsymbol{x} \in X$, and let $U \in J_\oplus$. Let

$$U^* = \{ha \in U^\blacktriangledown : P(h) = \oplus\} \cup \{h \in U^\blacktriangle : P(h) \neq \oplus\}.$$

Then the following are true:

(1) $\boldsymbol{x}_{U^*}$ uniquely determines $\boldsymbol{x}_U$: for a $\oplus$-node $h \in U^\blacktriangle$, we have $x_h = \sum_a x_{ha}$, and for a non-$\oplus$-node $h \in U^\blacktriangle$, we have $x_h = x_{ha}$ for all a.

(2) Let $h, h' \in U^*$, and suppose $x_h = x_{h'} = 1$. Then $h \wedge h'$ is not a $\oplus$-node: otherwise, a pure strategy playing to both h and h' would have to select two actions at $h \wedge h'$.

Thus, we have, $|X_U| \le \binom{|U^*|}{\le w(U)} \le |U|^{w(U)}$, where $\binom{n}{\le k} := \sum_{j=0}^{k} \binom{n}{k}$ is the number of ways to pick at most k items from a set of size n. The theorem follows. □

This bound is applicable in any game, but, again due to NP-hardness in the worst case, there will exist games in which $w = \Theta(N)$, in which case the bound will be exponentially bad and we would rather use the trivial bound $|X_U| \le 2^W$.

We now state some straightforward properties of reachable sets $S \subseteq U^\blacktriangle$.

(1) Every pair of nodes $h \neq h' \in S$ has a different sequence $s_\oplus(h)$. That is, information distinguishing nodes in S is known to at least one player on the team.

(2) S is a subset of a public state. That is, information distinguishing nodes in S is not common knowledge for the team.

---

[6] The degree of any bag U in the decomposition is at most $|U| \le W + 1$, because each child of U must contain a disjoint subset of $U^\blacktriangledown$, so in Proposition 2.8 we have $D = O(W)$ and we can therefore ignore D.

(3) For every pair of nodes $h \neq h' \in S$, and every infoset $I \prec h, h'$, the two nodes h and h' must follow from the same action a at I, that is, $Ia \prec h, h'$. That is, the information distinguishing nodes in S was not generated by players on the team.

(4) If the team has common external information, then $U^\blacktriangle$ has size 1 (and its single element is a $\oplus$-node by assumption), and thus S also can also have size at most 1.

Conditions 1 and 3 are effectively the negation of the definition of common external information, with the role of infosets I in that definition taken by public states U. Thus, in some sense, the reachable width measures the *amount of uncommon external information* in the game.

Therefore, Theorem 4.1 interpolates nicely between the extremes of common external information (which, by Property 4, has reachable width 1), and the game used in the NP-hardness reduction (Theorem 2.3), which can have reachable width $\Theta(N)$.

Using reachable sets, as opposed to merely arguing about the treewidth W and bounding $|X_U| \le 2^{W+1}$, is crucial in this argument: while Items 1, 2, and 4 in the above discussion follow for unreachable sets as well, Item 3 is false for unreachable sets. Thus, the treewidth W cannot be interpreted as the amount of uncommon *external* information, while the reachable width w can. In Appendix B, we show an example family of games in which the tree decomposition has $O(1)$ reachable width (and thus low uncommon external information), but $\Theta(N)$ treewidth.

4.2 Public Actions

Suppose that our game has the following structure for $\oplus$: $\oplus$ has n players. Nature begins by picking *private types* $t_i \in [t]$ for each player $i \in \oplus$, and informs each player i of t_i. From that point forward, all actions are public, the player to act is also public, and no further information is shared between teams. We call such games *public action*. For example, poker has this structure.

Assume, again without loss of generality, that the branching factor of the game is at most 2—this assumption can be satisfied by splitting decision nodes as necessary, and increases the number of public states by only a constant factor.

Consider a public state $U^\blacktriangle \in \mathcal{J}^\blacktriangle_\oplus$ at which a player $i \in \oplus$ picks one of two actions L or R. Since all actions are public, the set of reachable subsets of U can be described as follows: for each type $t_i \in [t]$, i chooses to either play L in $U^\blacktriangle$, play R in $U^\blacktriangle$, or not play to reach $U^\blacktriangle$ at all. For each other player $i' \neq i \in \oplus$, i' choosees, for each type $t_i \in [t]$, whether or not to play to reach $U^\blacktriangle$. There are a total of $3^t 2^{(n-1)t}$ choices that can be made in this fashion, so we have $|X_U| \le 3^t 2^{t(n-1)}$. Thus, we have shown:

Corollary 4.2. *Team correlated equilibria in extensive-form team public-action games with at most t types per player, and n players on each team can be computed via a linear program of size $O(3^t 2^{t(n-1)} NW) \le 2^{O(tn)} N$.*

This bound is much tighter than the bound given by the previous section—we have $W, w = O(t^n)$, so Theorem 4.1 is subsumed by the trivial bound $|X_U| \le 2^{W+1} = 2^{O(t^n)}$. It is also again in some sense tight: the game used in Theorem 2.3 has public actions and $t = \Theta(N)$ types for both players.

5 Experiments

We conducted experiments to compare our algorithm to the prior state of the art, namely the algorithms of Farina et al. (2021) ("FCGS-21" in the table) and **?** ("ZAC-20" in the table). Experimental results can be found in Table 1. The experiments table has the following syntax for identifying games: mnGp, where m and n are the sizes of teams $\oplus$ and $\ominus$ respectively, G is a letter representing the game, and p represents parameters of the game, described below. All games described are variants of common multi-player games in which teams are formed by colluding players, who we assume will evenly split any reward. For example, "31K5" is Kuhn poker (K) with $|\oplus| = 3, |\ominus| = 1$, and 5 ranks. Where relevant, $\oplus$ consists of the first m players, and $\ominus$ consists of the remaining n players.

- mnKr is *Kuhn poker* with r ranks.
- mnLbrc is *Leduc poker*. b is the maximum number of bets allowed in each betting round. r is the number of ranks. c is the number of suits (suits are indistinguishable). In variant L', team $\oplus$ is not allowed to raise.
- mnDn is *Liar's Dice* with one n-sided die per player. In Liar's Dice, if both teams have consecutive players, then the game value is trivially 0. Therefore, in these instances, instead of $\ominus$ being the last n players, the last $2n$ players alternate teams—for example, in 42D, the teams are $\oplus = \{1, 2, 3, 5\}$ and $\ominus = \{4, 6\}$.
- mnG and mnGL are *Goofspiel* with 3 ranks. GL is the limited-information variant.

These are the same games used by Farina et al. (2021) in their work; we refer the interested reader to that paper for detailed descriptions of all the games. In many cases, teams either have size 1 or have common external information (the latter is always true in Goofspiel). In these cases, it would suffice, after inflation, to use the standard sequence-form representation of the player's strategy space (Koller, Megiddo, and von Stengel 1996). However, we run our technique anyway, to demonstrate how it works in such settings.

Our experiments show clear state-of-the-art performance in all tested cases in which comparisons could be made, except Kuhn Poker. In Kuhn Poker, the number of types t is relatively large compared to the game size, so our technique scales poorly compared to prior techniques.

6 Conclusions

In this paper, we devised a completely new algorithm for solving team games that uses tree decomposition of the constraints representing each team's strategy to reduce the number and degree of constraints required for correctness. Our approach has the bags of the tree decomposition correspond to team-public states—that is, minimal sets of nodes (that is, states of the team) such that, upon reaching the set, it is common knowledge among the players on the team that the set has been reached. Our algorithm reduces the problem of

		Team $\oplus$			Team $\ominus$					Runtime		
Game	$\lvert Z\rvert$	#Seq	$\sum\lvert X_C\rvert$	Ratio	#Seq	$\sum\lvert X_C\rvert$	Ratio	Value	LP Size	Ours	FCGS-21	ZAC-20
21K3	78	91	351	3.9	25	25	1	.0000	2386	**0.01s**	**0.01s**	<0.7s
21K4	312	177	1749	9.9	33	33	1	-.0417	18810	0.02s	**0.01s**	0.7s
21K6	1560	433	52669	122	49	49	1	-.0236	1150838	1.37s	—	2s
21K8	4368	801	1777061	2218	65	65	1	-.0193	62574750	2m24s	<4.96s	4s
21K12	17160	1873			97	97	1			oom	**4.96s**	10s
22K5	3960	611	23711	39	3083	521	5.9	-.0368	426297	1.27s	—	—
31K5	3960	2611	974470	373	81	81	1	-.0300	21106658	3m20s	—	—
21L133*	6477	2725	17718	6.5	457	703	1.5	.2148	126075	**0.49s**	2m23s	1h22m
21L143	20856	6377	115281	18	801	1225	1.5	.1073	1195766	**7.66s**	1h07m	>1h22m
21L151*	10020	6051	130359	22	1001	1531	1.5	-.0192	1425583	8.02s	—	4h24m
21L153	51215	12361	757884	61	1241	1891	1.5	.0240	11234573	**3m50s**	>1h07m	>4h24m
21L223	8762	5765	21729	3.8	1443	3123	2.2	.5155	112305	**0.23s**	3m29s	—
21L523	775148	492605	2042641	4.1	123153	305835	2.5	.9520	10507398	**3m27s**	>3m29s	—
22L133	80322	9781	55788	5.7	9745	52053	5.3	.1470	785032	**9.26s**	>2m23s	>1h22m
31L'132*	3834	3991	46122	12	151	235	1.6	-.1333	369673	2.97s	—	1m08s
31L'133*	8898	5644	69642	12	151	235	1.6	-.1457	577942	8.88s	—	21m04s
31L133	80322	42361	703390	17	1633	2479	1.5	.3470	6326658	**1m52s**	>2m23s	>1h22m
21D3	13797	6085	74635	12	1021	2686	2.6	.2840	500665	**2.03s**	1m12s	—
21D4	262080	87217	3521398	40	10921	29749	2.7	.2843	32755273	**9m59s**	>6h	—
22D3	331695	36784	402669	11	36388	381218	11	.2000	5275196	**1m00s**	>6h	—
33D2	262080	31545	178123	5.6	29433	152286	5.2	.0721	1862387	15.7s	—	—
42D2	262080	83969	761600	9.1	9491	30150	3.2	.2647	4751391	1m24s	—	—
51D2	262080	185905	2537927	14	3459	7207	2.1	.3333	17635669	8m21s	—	—
21GL	1296	2713	8572	3.2	934	2158	2.3	.2524	34310	**0.06s**	0.33s	—
21G	1296	3601	11344	3.2	1630	3766	2.3	.2534	47810	**0.08s**	1.18s	—
31GL	7776	21082	71278	3.4	3502	7582	2.2	.2803	249374	0.66s	—	—
31G	7776	30250	102118	3.4	8758	18994	2.2	.2803	378410	**0.79s**	>1.18s	—
32GL	46656	71722	241030	3.4	34393	104908	3.1	.0000	1095542	3.19s	—	—
32G	46656	160498	538546	3.4	102097	310720	3.0	.0000	2620502	7.86s	—	—

Table 1: Experiments. "oom" indicates that our algorithm exhausted the memory limit of 16GB. "—" means that the respective paper did not report a runtime for that game. Our runtimes list only the time taken by the LP solver; the time to construct the LP itself is smaller in all instances. LPs are solved with the barrier algorithm in Gurobi 9.1 with crossover and presolver both off (the latter was needed for numerical stability), using 4 CPU cores. The hardware is comparable to that used by FCGS-21, and weaker than that used by ZAC-20. *: The Leduc games in ZAC-20 were constructed using a different implementation than ours and FCGS-21, and thus the reported game sizes differ despite the underlying games being the same. Our implementation matches FCGS-21.

solving team games to a linear program of size NW^{w+1}, where w is a parameter that depends on the amount of uncommon external information and W is the treewidth. In *public-action games*, we achieve a tighter bound $2^{O(nt)}N$ for teams of n players with t types each. Our algorithm is based on a new way to write a custom, concise tree decomposition, and its fast run time does not rely on low treewidth. We show via experiments on a standard suite of games that our algorithm achieves state-of-the-art performance on all benchmark games except Kuhn poker. We also present, to our knowledge, the first experiments for this setting where more than one team has more than one member.

Compared to the techniques of past papers (Celli and Gatti 2018; **?**; Farina et al. 2021), our technique has certain clear advantages and disadvantages.

(1) *Advantage*: Unlike prior techniques, ours does not require solving integer programs for best responses. While integer programming and normal-form double oracle can have reasonable practical performance, neither has worst-case performance bounds. In contrast, we are able to derive worst-case performance bounds.

(2) *Advantage*: Since our algorithm describes the polytope of correlated strategies directly, we get equilibrium finding in correlated strategies for free—instead of, say, having to run a double oracle algorithm, which, like integer programming, has no known polynomial convergence bound despite reasonable practical performance in some cases.

(3) *Advantage*: In domains where there is not much uncommon external information (i.e., $w(U)$ or t is small), our program size scales basically linearly in the game size. Thus, our algorithm is able to tackle some games with 10^5 sequences for both players.

(4) *Disadvantage*: Our algorithm scales poorly in games with high uncommon external information. In the experiments, this can be seen in the Kuhn poker instances.

Acknowledgements

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. We also thank Gabriele Farina, Andrea Celli, and our anonymous reviewers at AAAI and AAAI-RLG for suggesting improvements to the writing.

References

Basilico, N.; Celli, A.; De Nittis, G.; and Gatti, N. 2017. Team-maxmin equilibrium: efficiency bounds and algorithms. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Bowling, M.; Burch, N.; Johanson, M.; and Tammelin, O. 2015. Heads-up Limit Hold'em Poker is Solved. *Science*, 347(6218).

Brown, N.; and Sandholm, T. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374): 418–424.

Brown, N.; and Sandholm, T. 2019. Superhuman AI for multiplayer poker. *Science*, 365(6456): 885–890.

Celli, A.; and Gatti, N. 2018. Computational results for extensive-form adversarial team games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

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

Daskalakis, C.; and Papadimitriou, C. 2006. Computing pure Nash equilibria in graphical games via Markov random fields. In *ACM Conference on Electronic Commerce (ACM-EC)*, 91–99. Ann Arbor, MI.

Farina, G.; Celli, A.; Gatti, N.; and Sandholm, T. 2018. Ex Ante Coordination and Collusion in Zero-Sum Multi-Player Extensive-Form Games. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Farina, G.; Celli, A.; Gatti, N.; and Sandholm, T. 2021. Connecting Optimal Ex-Ante Collusion in Teams to Extensive-Form Correlation: Faster Algorithms and Positive Complexity Results. In *International Conference on Machine Learning*.

Farina, G.; and Sandholm, T. 2020. 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)*.

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

Koller, D.; Megiddo, N.; and von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *ACM Symposium on Theory of Computing (STOC)*.

Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient Computation of Equilibria for Extensive Two-Person Games. *Games and Economic Behavior*, 14(2).

Kovařík, V.; Schmid, M.; Burch, N.; Bowling, M.; and Lisý, V. 2021. Rethinking formal models of partially observable multiagent decision making. *Artificial Intelligence*, 103645.

Luce, R. D.; and Raiffa, H. 1957. *Games and Decisions*. New York: John Wiley and Sons. Dover republication 1989.

McMahan, H. B.; Gordon, G. J.; and Blum, A. 2003. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning (ICML)*, 536–543.

Robertson, N.; and Seymour, P. D. 1986. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3): 309–322.

Wainwright, M. J.; and Jordan, M. I. 2008. *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Zhang, Y.; An, B.; and Černý, J. 2021. Computing Ex Ante Coordinated Team-Maxmin Equilibria in Zero-Sum Multiplayer Extensive-Form Games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

A Proofs

A.1 Theorem 2.3

Proof. We reduce from 3-SAT. Let ϕ be a 3-SAT instance, and consider the following game for a single team $\oplus = \{\mathrm{P1}, \mathrm{P2}\}$. First, nature chooses a clause uniformly at random. Then, P1 is told the clause and chooses a variable within that clause. Finally, P2 is told the variable that P1 chose, but *not* the clause, and chooses an assignment (true or false) to that literal. The team wins if P2's assignment satisfies the clause.

Clearly, the game has size linear in the size of the instance (six terminal nodes per clause). If ϕ is satisfiable, the team can always win by having P2 play a satisfying assignment and P1 pick a positive literal in each clause. If ϕ is not satisfiable, then no matter what P2's strategy is, the team must lose with probability at least $1/m$ where m is the number of clauses. Therefore, this game has value at least $1 - 1/2m$ if and only if ϕ is satisfiable. $\square$

A.2 Proposition 2.8

Proof. We repeat the constraint system here for clarity.

$$\boldsymbol{x}_U = \sum_{\tilde{\boldsymbol{x}} \in X_U} \lambda^U_{\tilde{\boldsymbol{x}}} \tilde{\boldsymbol{x}} \qquad \forall \text{ bags } U \in \mathcal{J} \tag{A.1}$$

$$\sum_{\substack{\tilde{\boldsymbol{x}} \in X_U: \\ \tilde{\boldsymbol{x}}_{U \cap V} = \tilde{\boldsymbol{x}}^*}} \lambda^U_{\tilde{\boldsymbol{x}}} = \sum_{\substack{\tilde{\boldsymbol{x}} \in X_V: \\ \tilde{\boldsymbol{x}}_{U \cap V} = \tilde{\boldsymbol{x}}^*}} \lambda^V_{\tilde{\boldsymbol{x}}} \qquad \forall \text{ edges } (U, V) \text{ in } \mathcal{J}, \text{ locally feasible } \tilde{\boldsymbol{x}}^* \in X_{U \cap V} \tag{A.2}$$

Since the $\boldsymbol{\lambda}^U$s are local distributions with consistent marginals across the edges of the tree decomposition, the junction tree theorem gives a distribution $\boldsymbol{\Lambda}$ on $\{0,1\}^n$ such that, for every bag U, the marginal of $\boldsymbol{\Lambda}$ on U is exactly $\boldsymbol{\lambda}^U$. By definition, we have $\boldsymbol{x} = \mathbb{E}_{\hat{\boldsymbol{x}} \sim \boldsymbol{\Lambda}} \hat{\boldsymbol{x}}$. Consider some $\hat{\boldsymbol{x}} \in \{0,1\}^n$ in the support of $\boldsymbol{\Lambda}$. Then $\hat{\boldsymbol{x}}_U$ is in the support of $\boldsymbol{\lambda}^U$, which in particular means that $\hat{\boldsymbol{x}}$ satisfies all constraints whose variables are fully contained in U. Since this is true of every bag U, and every constraint has its variables fully contained in some bag U, we have $\hat{\boldsymbol{x}} \in X$, and thus $\boldsymbol{x} \in \mathcal{X}$, as needed.

We now bound the size (number of total terms) of the system.

1. The number of appearances of each x_i is at most the total sum of all bag sizes, which is $O(W|\mathcal{J}|)$.
2. Each variable $\lambda^U_{\tilde{\boldsymbol{x}}}$ appears at most W times in constraints of the form (A.1), one for each i such that $\tilde{x}_i = 1$. This accounts for $O(W \sum_{U \in \mathcal{J}} |X_U|)$ terms.
3. In each edge (U, V), each $\lambda^U_{\tilde{\boldsymbol{x}}}$ and $\lambda^V_{\tilde{\boldsymbol{x}}}$ appears exactly once in constraints of the form (A.2). Thus, each $\lambda^U_{\tilde{\boldsymbol{x}}}$ appears an additional D times, where D is the maximum degree of any bag U of $\mathcal{J}$. This accounts for $O(D \sum_{U \in \mathcal{J}} |X_U|)$ terms.

Adding these up gives the stated result. $\square$

B Example Game Where Reachable Width is Much Smaller than Width

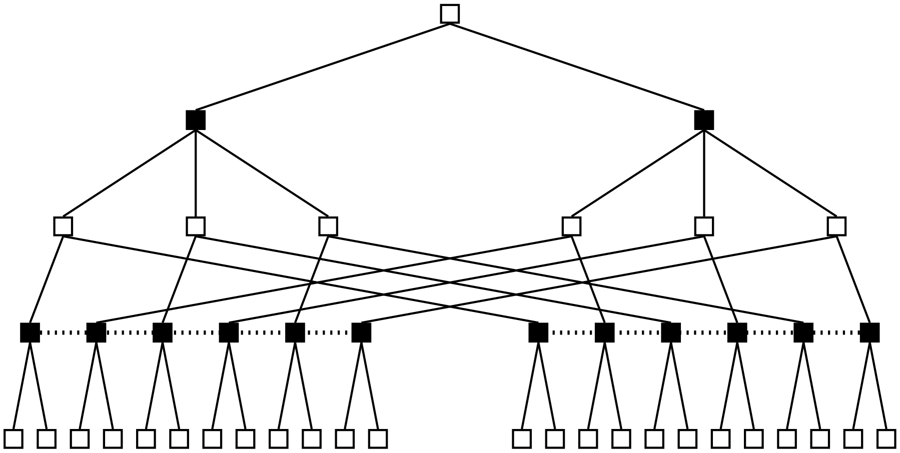

Figure 2: The game tree of the family of games in Appendix B, for $k = 3$. Information sets are connected by dotted lines. Team nodes are black, and nature and terminal nodes are white.

Consider the following family of games involving a team of two players $\oplus = \{\mathrm{P1}, \mathrm{P2}\}$. Nature picks a random type t_1 for P1. P1 learns t_1 and picks an action $a \in [k]$. Nature then picks a type $t_2 \in \{0, 1\}$ for P2, possibly conditional on P1's type and/or action. P2 learns t_2, but *not* the action a played by P1, and must then play one of two actions. We do not specify the rewards as they are not relevant. The game tree for $k = 3$ is depicted in Figure 2.

In this game, the two P2 infosets, each of size $2k$, are public states. Therefore, the tree decomposition defined in this paper has treewidth $6k - 1$ (when $k = 3$, the treewidth is 17). However, there are only two bits of external information, namely the two types $t_1, t_2 \in \{0, 1\}$. Indeed, using reachable width instead of treewidth fixes this issue: the reachable width of each public state in this game is only 2 (regardless of the value of k), which accurately reflects the amount of uncommon external information.

We could have written this counterexample without nature picking the type t_2, but then the game tree would be rewriteable by switching the order of P1 and P2's decisions, and having P2 decide her move at the root of the tree, upon which the game would become perfect information. Adding the second layer of nature nodes prevents such a straightforward rewriting.

C Full Constraint System and Example

In this section, we state the constraint system for the team's decision space induced by chaining Algorithm 3.3 with (2.9), and give an example using the game in Figure 1.

C.1 General Constraint System

For each public state $U^\blacktriangle \in \mathcal{J}_\oplus^\blacktriangle$, and for each reachable subset $S \subseteq U^\blacktriangle$, let I_S be the set of $\oplus$-infosets intersecting S. For each joint action $\boldsymbol{a} \in \bigtimes_{I \in \mathcal{I}_S} A(I)$, let $S\boldsymbol{a} \subseteq U^\blacktriangledown$ be the set of children of S reached if the team plays joint action $\boldsymbol{a}$. That is,

$$S\boldsymbol{a} = \{ha_I : h \in S, h \in I \in \mathcal{I}_S\} \cup \{ha : h \in S, P(h) \neq \oplus, a \in A(h)\}.$$

Let $\tilde{\boldsymbol{x}}^{S\boldsymbol{a}} \in \{0,1\}^U$ be the indicator function on $S \cup S\boldsymbol{a}$, i.e., $\tilde{\boldsymbol{x}}_i^{S\boldsymbol{a}} = \mathbf{1}\{i \in S \cup S\boldsymbol{a}\}$. For each such set $S \cup S\boldsymbol{a} \subseteq U$, we introduce a variable $\lambda_{S\boldsymbol{a}}$ representing the local probability mass on $\tilde{\boldsymbol{x}}^{S\boldsymbol{a}}$.

In the root bag[7] $\{\textsc{Root}\}$, we introduce the constraint that $\boldsymbol{\lambda}$ is a feasible local distribution in this bag and $x_{\textsc{Root}}$ is consistent with it, which simply means that $x_{\textsc{Root}} = \lambda_{\{\textsc{Root}\}} = 1$.

In every bag, we introduce the constraint that $\boldsymbol{x}$ is consistent with the choice of the local distributions[8]:

$$x_h = \sum_{\substack{\text{reachable } S\boldsymbol{a} \subseteq U \\ h \in S\boldsymbol{a}}} \lambda_{S\boldsymbol{a}} \quad \forall\, h \in U^\blacktriangledown.$$

Finally, if U is not the root bag, for each nonempty[9] reachable subset $S \subseteq U^\blacktriangle$, we introduce the constraint

$$\sum_{\substack{\text{reachable } S'\boldsymbol{a}' \subseteq U' \\ S = S'\boldsymbol{a}' \cap U}} \lambda_{S'\boldsymbol{a}'} = \sum_{\boldsymbol{a} \in \bigtimes_{I \in \mathcal{I}_S} A(I)} \lambda_{S\boldsymbol{a}}$$

where U' is the parent of U. This constraint states that, for each reachable subset $S \subseteq U^\blacktriangle$, the "marginal" onto S in both directions is consistent: from above, this marginal is constructed by summing the probabilities of all $S'\boldsymbol{a}'$ in the parent node such that the marginal of $S'\boldsymbol{a}'$ onto U is exactly S; from below, it is constructed by summing over all joint actions $\boldsymbol{a}$ available to the team at S. As stated in the main paper, in the case where the team has perfect recall, the left-hand side of this constraint will have exactly one term, and we recover, up to some redundancies, the sequence-form polytope of the team.

C.2 Example: Constraint System for the Game in Figure 1

$$\boxed{\text{A}} \text{ and } \boxed{\begin{matrix}\text{A}\\ \text{BC}\end{matrix}} \qquad \lambda_{\text{A}} = \lambda_{\text{ABC}} = x_{\text{A}} = x_{\text{B}} = x_{\text{C}} = 1$$

$$\boxed{\begin{matrix}\text{BC}\\ \text{DEFG}\end{matrix}} \left\{\begin{aligned} \lambda_{\text{ABC}} &= \lambda_{\text{BCDF}} + \lambda_{\text{BCDG}} + \lambda_{\text{BCEF}} + \lambda_{\text{BCEG}} \\ x_{\text{D}} &= \lambda_{\text{BCDF}} + \lambda_{\text{BCDG}} \\ x_{\text{E}} &= \lambda_{\text{BCEF}} + \lambda_{\text{BCEG}} \\ x_{\text{F}} &= \lambda_{\text{BCDF}} + \lambda_{\text{BCEF}} \\ x_{\text{G}} &= \lambda_{\text{BCDG}} + \lambda_{\text{BCEG}} \end{aligned}\right.$$

$$\boxed{\begin{matrix}\text{D}\\ \text{HI}\end{matrix}} \left\{\begin{aligned} \lambda_{\text{BCDF}} + \lambda_{\text{BCDG}} &= \lambda_{\text{DH}} + \lambda_{\text{DI}} \\ x_{\text{H}} &= \lambda_{\text{DH}} \\ x_{\text{I}} &= \lambda_{\text{DI}} \end{aligned}\right.$$

$$\boxed{\begin{matrix}\text{EF}\\ \text{JKLM}\end{matrix}} \left\{\begin{aligned} \lambda_{\text{BCDF}} &= \lambda_{\text{FL}} + \lambda_{\text{FM}} \\ \lambda_{\text{BCEF}} &= \lambda_{\text{EFJL}} + \lambda_{\text{EFKM}} \\ \lambda_{\text{BCEG}} &= \lambda_{\text{EJ}} + \lambda_{\text{EK}} \\ x_{\text{J}} &= \lambda_{\text{EFJL}} + \lambda_{\text{EJ}} \\ x_{\text{K}} &= \lambda_{\text{EFKM}} + \lambda_{\text{EK}} \\ x_{\text{L}} &= \lambda_{\text{EFJL}} + \lambda_{\text{FL}} \\ x_{\text{M}} &= \lambda_{\text{EFKM}} + \lambda_{\text{FM}} \end{aligned}\right.$$

---

[7] The condition that $\boldsymbol{\lambda}$ is a local distribution in all other bags will follow from the marginalization constraints between bags.

[8] The constraints for $h \in U^\blacktriangle$ again follow from marginalization constraints.

[9] The constraints for empty S are redundant.