Theoretical and Practical Advances on Smoothing for Extensive-Form Games

Christian Kroer*
Computer Science Department
Carnegie Mellon University
ckroer@cs.cmu.edu

Kevin Waugh
Department of Computing Science
University of Alberta
kevin.waugh@gmail.com

Fatma Kılınç-Karzan†
Tepper School of Business
Carnegie Mellon University
fkilinc@andrew.cmu.edu

Tuomas Sandholm‡
Computer Science Department
Carnegie Mellon University
sandholm@cs.cmu.edu

February 14, 2017

Abstract

Sparse iterative methods, in particular first-order methods, are known to be among the most effective in solving large-scale two-player zero-sum extensive-form games. The convergence rates of these methods depend heavily on the properties of the distance-generating function that they are based on. We investigate the acceleration of first-order methods for solving extensive-form games through better design of the dilated entropy function—a class of distance-generating functions related to the domains associated with the extensive-form games. By introducing a new weighting scheme for the dilated entropy function, we develop the first distance-generating function for the strategy spaces of sequential games that has no dependence on the branching factor of the player. This result improves the convergence rate of several first-order methods by a factor of $\Omega(b^d d)$, where b is the branching factor of the player, and d is the depth of the game tree.

Thus far, counterfactual regret minimization methods have been faster in practice, and more popular, than first-order methods despite their theoretically inferior convergence rates. Using our new weighting scheme and practical tuning we show that, for the first time, the excessive gap technique can be made faster than the fastest counterfactual regret minimization algorithm, CFR+, in practice.

1 Introduction

Extensive-form games (EFGs) are a broad class of games; they model sequential interaction, imperfect information, and outcome uncertainty. Nash equilibria prescribe a particular notion of rational behavior in such games. In the specific case of two-player zero-sum EFGs with perfect recall, an exact Nash equilibrium can be computed in polynomial time using a Linear Program (LP) whose

*Supported by a Facebook Fellowship, the NSF under grants IIS-1617590, IIS-1320620, and IIS-1546752, and the ARO under award W911NF-16-1-0061.

†Supported by NSF grant CMMI 1454548.

‡Supported by the NSF under grants IIS-1617590, IIS-1320620, and IIS-1546752, the ARO under award W911NF-16-1-0061.

size is linear in the size of the game tree [von Stengel, 1996]. However, in practice the LP approach has two major drawbacks limiting its applicability. First, the LP may be prohibitively large and may not fit in memory. Second, even when it does, the iterations of interior-point methods or the simplex algorithm are prohibitively expensive [Sandholm, 2010]. Practical methods for EFG solving tackle this issue through two complementary approaches: Abstraction and iterative game solvers with low memory requirements [Sandholm, 2010]. In this paper we focus on the second approach. Iterative game solvers mainly fall in two categories: (i) counterfactual-regret-based methods [Zinkevich et al., 2007, Lanctot et al., 2009] achieving a convergence rate on the order of $O(\frac{1}{\epsilon^2})$, and (ii) first-order methods (FOMs) [Hoda et al., 2010, Kroer et al., 2015] achieving a convergence rate of $O(\frac{1}{\epsilon})$. The better convergence rate of FOMs makes them more attractive from a theoretical viewpoint. This paper investigates the acceleration of such FOMs for EFGs, from both a theoretical and a numerical perspective.

Nash equilibrium computation of a two-player zero-sum EFG with perfect recall admits a Bilinear Saddle Point Problem (BSPP) formulation where the domains are given by the polytopes that encode strategy spaces of the players. The most efficient FOMs are designed to solve this BSPP. The classical FOMs to solve BSPPs such as *mirror prox* (MP) [Nemirovski, 2004] or the *excessive gap technique* (EGT) [Nesterov, 2005a] utilize *distance-generating functions* (DGFs) to measure appropriate notions of distances over the domains. Then the convergence rate of these FOMs relies on the DGFs and their relation to the domains in three critical ways: Through the strong convexity parameters of the DGFs, the norm associated with the strong convexity parameter, and set widths of the domains as measured by the DGFs.

Hoda et al. [2010] introduced a general framework for constructing DGFs for *treeplexes*—a class of convex polytopes that generalize the domains associated with the strategy spaces of an EFG. While they also established bounds on the strong convexity parameter for their DGFs in some special cases, these lead to very weak bounds and result in slow convergence rates. Kroer et al. [2015] developed explicit strong convexity-parameter bounds for entropy-based DGFs (a particular subclass of DGFs) for general EFGs, and improved the bounds for the special cases considered by Hoda et al. [2010]. These bounds from Kroer et al. [2015] generate the current state-of-the-art parameters associated with the convergence rate for FOMs with $O(\frac{1}{\epsilon})$ convergence.

In this paper we construct a new weighting scheme for such entropy-based DGFs. This weighting scheme leads to new and improved bounds on the strong convexity parameter associated with general treeplex domains. In particular, our new bounds are first-of-their kind as they have no dependence on the branching operation of the treeplex. Informally, our strong convexity result allows us to improve the convergence rate of FOMs by a factor of $\Omega(b^d d)$ (where b is the average branching factor for a player and d is the depth of the EFG) compared to the prior state-of-the-art results from Kroer et al. [2015]. Our bounds parallel the simplex case for matrix games where the entropy function achieves a logarithmic dependence on the dimension of the simplex domain.

Finally, we complement our theoretical results with numerical experiments to investigate the speed up of FOMs with convergence rate $O(\frac{1}{\epsilon})$ and compare the performance of these algorithms with the premier regret-based methods CFR and CFR+ [Tammelin et al., 2015]. CFR+ is the fastest prior algorithm for computing Nash equilibria in EFGs when the entire tree can be traversed (rather than sampled). Bowling et al. [2015] used it to essentially solve the game limit Texas hold'em.

CFR+ is also the algorithm used to accurately solve endgames in the Libratus agent, which showed superhuman performance against a team of top Heads-Up No-Limit Texas hold'em poker

specialist professional players in the Brains vs AI event [1]. A slight variation[2] of CFR+ was used in the DeepStack agent Moravčík et al. [2017], which beat a group of professional players. Our experiments show that FOMs are substantially faster than both CFR algorithms when using a practically tuned variant of our DGF. We also test the impact of stronger bounds on the strong convexity parameter: we instantiate EGT with the parameters developed in this paper, and compare the performance to the parameters developed by Kroer et al. [2015]. These experiments illustrate that the tighter parameters developed here lead to better practical convergence rate.

The rest of the paper is organized as follows. Section 2 discusses related research. We present the general class of problems that we address—bilinear saddle-point problems—and describe how they relate to EFGs in Section 3. Then Section 4 describes our optimization framework. Section 5 introduces treeplexes, the class of convex polytopes that define our domains of the optimization problems. Our focus is on dilated entropy-based DGFs; we introduce these in Section 6 and present our main results—bounds on the associated strong convexity parameter and treeplex diameter. In Section 7 we demonstrate the use of our results on instantiating EGT. We compare our approach with the current state-of-art in EFG solving and discuss the extent of theoretical improvements achievable via our approach in Section 7.1. Section 8 presents numerical experiments testing the effect of various parameters on the performance of our approach as well as comparing the performance of our approach to CFR and CFR+. We close with a summary of our results and a few compelling further research directions in Section 9.

2 Related work

Nash equilibrium computation has received extensive attention in the literature [Littman and Stone, 2003, Lipton et al., 2003, Gilpin and Sandholm, 2007, Zinkevich et al., 2007, Daskalakis et al., 2009, Jiang and Leyton-Brown, 2011, Kroer and Sandholm, 2014, Daskalakis et al., 2015]. The equilibrium-finding problems vary quite a bit based on their characteristics; here we restrict our attention to two-player zero-sum sequential games.

Koller et al. [1996] present an LP whose size is linear in the size of the game tree. This approach, coupled with lossless abstraction techniques, was used to solve Rhode-Island hold'em [Shi and Littman, 2002, Gilpin and Sandholm, 2007], a game with 3.1 billion nodes (roughly size $5 \cdot 10^7$ after lossless abstraction). However, for games larger than this, the resulting LPs tend to not fit in the computer memory thus requiring approximate solution techniques. These techniques fall into two categories: iterative ϵ-Nash equilibrium-finding algorithms and game abstraction techniques [Sandholm, 2010].

The most popular iterative Nash equilibrium algorithm is the counterfactual-regret-minimization framework instantiated with regret matching (CFR) [Zinkevich et al., 2007], its sampling-based variant monte-carlo CFR (MCCFR) [Lanctot et al., 2009], and CFR instantitated with a new regret minimization technique called regret matching plus (CFR+). These regret-minimization algorithms perform local regret-based updates at each information set. Despite their slow convergence rate of $O(\frac{1}{\epsilon^2})$, they perform very well in pratice, especially CFR+. Recently, Waugh and Bagnell [2015] showed, with some caveats, an interpretation of CFR as a FOM with $O(\frac{1}{\epsilon^2})$ rate. Nonetheless, in this paper we make a distinction between regret-based methods and $O(\frac{1}{\epsilon})$ FOMs for ease of exposition.

Hoda et al. [2010] initially proposed DGFs for EFGs leading to $O(\frac{1}{\epsilon})$ convergence rate when used with EGT. Kroer et al. [2015] improved these result for the dilated entropy function. Gilpin

[1] Confirmed through author communication

[2] This variation was chosen for implementation reasons, though, and has inferior practical iteration complexity.

et al. [2012] give an algorithm with convergence rate $O(\ln(\frac{1}{\epsilon}))$. Their bound has a dependence on a certain condition number of the payoff matrix, which is difficult to estimate; and as a result they show a bound of $O(\frac{1}{\epsilon})$ which is independent of the condition number. Detailed comparisons to all three algorithms discussed here are given in Section 7.1.

Finally, Bosansky et al. [2014] develop an iterative double-oracle algorithm for exact equilibrium computation. This algorithm only scales for games where it can identify an equilibrium of small support, and thus suffers from the same performance issues as the general LP approach.

In addition to equilibrium-finding algorithms, another central topic in large-scale game solving has been automated abstraction [Sandholm, 2010, 2015]. Initially, this was used mostly for information abstraction [Gilpin and Sandholm, 2007, Shi and Littman, 2002, Zinkevich et al., 2007]. Lately, action abstraction approaches have gained considerable interest [Hawkin et al., 2011, 2012, Brown and Sandholm, 2014, Kroer and Sandholm, 2014, 2016]. Sequential game abstraction approaches with solution quality bounds have also emerged for stochastic [Sandholm and Singh, 2012] and extensive-form [Lanctot et al., 2012, Kroer and Sandholm, 2014, 2016] games more recently.

3 Problem setup

Computing a Nash equilibrium in a two-player zero-sum EFG with perfect recall can be formulated as a Bilinear Saddle Point Problem (BSPP):

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \langle x, Ay\rangle = \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \langle x, Ay\rangle. \tag{1}$$

This is known as the *sequence-form* formulation [Romanovskii, 1962, Koller et al., 1996, von Stengel, 1996]. In this formulation, x and y correspond to the nonnegative strategy vectors for players 1 and 2 and the sets $\mathcal{X}, \mathcal{Y}$ are convex polyhedral reformulations of the sequential strategy space of these players. Here $\mathcal{X}, \mathcal{Y}$ are defined by the constraints $Ex = e, Fy = f$, where each row of E, F encodes part of the sequential nature of the strategy vectors, the right hand-side vectors e, f are $|\mathcal{I}_1|, |\mathcal{I}_2|$-dimensional vectors, and $\mathcal{I}_i$ is the information sets for player i. For a complete treatment of this formulation, see von Stengel [1996].

Our theoretical developments mainly exploit the treeplex domain structure and are independent of other structural assumptions resulting from EFGs. Therefore, we describe our results for general BSPPs. We follow the presentation and notation of Juditsky and Nemirovski [2011a,b] for BSPPs. For notation and presentation of treeplex structure, we follow Kroer et al. [2015].

3.1 Basic notation

We let $\langle x, y\rangle$ denote the standard inner product of vectors x, y. Given a vector $x \in \mathbb{R}^n$, we let $\|x\|_p$ denote its ℓ_p norm given by $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \in [1, \infty)$ and $\|x\|_\infty := \max_{i\in[n]} |x_i|$ for $p = \infty$. Throughout this paper, we use Matlab notation to denote vector and matrices, i.e., $[x; y]$ denotes the concatenation of two column vectors x, y. For a given set Q, we let $\mathrm{ri}\,(Q)$ denote its relative interior. Given $n \in \mathbb{N}$, we denote the simplex $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$.

4 Optimization setup

In its most general form a BSPP is defined as

$$\mathrm{Opt} := \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \phi(x, y), \tag{$\mathcal{S}$}$$

where $\mathcal{X}, \mathcal{Y}$ are nonempty convex compact sets in Euclidean spaces $\mathbf{E}_x, \mathbf{E}_y$ and $\phi(x,y) = \upsilon + \langle a_1, x\rangle + \langle a_2, y\rangle + \langle y, Ax\rangle$. We let $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$; so $\phi(x,y) : \mathcal{Z} \to \mathbb{R}$. In the context of EFG solving, $\phi(x,y)$ is simply the inner product given in (1).

The BSPP $(\mathcal{S})$ gives rise to two convex optimization problems that are dual to each other:

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \min_{x\in\mathcal{X}}[\overline{\phi}(x) := \max_{y\in\mathcal{Y}} \phi(x,y)] & (P) \\ \mathrm{Opt}(D) & = & \max_{y\in\mathcal{Y}}[\underline{\phi}(y) := \min_{x\in\mathcal{X}} \phi(x,y)] & (D) \end{array}$$

with $\mathrm{Opt}(P) = \mathrm{Opt}(D) = \mathrm{Opt}$. It is well known that the solutions to $(\mathcal{S})$ — the saddle points of ϕ on $\mathcal{X} \times \mathcal{Y}$ — are exactly the pairs $z = [x; y]$ comprised of optimal solutions to the problems (P) and (D). We quantify the accuracy of a candidate solution $z = [x; y]$ with the *saddle point residual*

$$\epsilon_{\mathrm{sad}}(z) \;:=\; \overline{\phi}(x) - \underline{\phi}(y) = \underbrace{\big[\overline{\phi}(x) - \mathrm{Opt}(P)\big]}_{\geq 0} + \underbrace{\big[\mathrm{Opt}(D) - \underline{\phi}(y)\big]}_{\geq 0}.$$

In the context of EFG, $\epsilon_{\mathrm{sad}}(z)$ measures the proximity to being an ϵ-Nash equilibrium.

4.1 General framework for FOMs

Most FOMs capable of solving BSPP $(\mathcal{S})$ are quite flexible in terms of adjusting to the geometry of the problem characterized by the domains $\mathcal{X}, \mathcal{Y}$ of the BSPP $(\mathcal{S})$. The following components are standard in forming the setup for such FOMs (we present components for $\mathcal{X}$, analogous components are used for $\mathcal{Y}$):

- *Vector norm*: $\|\cdot\|_{\mathcal{X}}$ on the Euclidean space $\mathbf{E}$ where the domain $\mathcal{X}$ of $(\mathcal{S})$ lives, along with its dual norm $\|\zeta\|_{\mathcal{X}}^* = \max\limits_{\|x\|_{\mathcal{X}}\leq 1} \langle \zeta, x\rangle$.
- *Matrix norm*: $\|A\| = \max_y \{\|Ay\|_{\mathcal{X}}^* : \|y\|_{\mathcal{Y}} = 1\}$ based on the vector norms $\|\cdot\|_{\mathcal{X}}, \|\cdot\|_{\mathcal{Y}}$.
- *Distance-Generating Function* (DGF): A function $\omega_{\mathcal{X}}(x) : \mathcal{X} \to \mathbb{R}$, which is convex and continuous on $\mathcal{X}$, and admits a continuous selection of subgradients $\omega'_{\mathcal{X}}(x)$ on the set $\mathcal{X}^\circ := \{x \in \mathcal{X} : \partial\omega_{\mathcal{X}}(x) \neq \emptyset\}$ (here $\partial\omega_{\mathcal{X}}(x)$ is a subdifferential of $\omega_{\mathcal{X}}$ taken at x), and is strongly convex with modulus $\varphi_{\mathcal{X}}$ w.r.t. the norm $\|\cdot\|_{\mathcal{X}}$:

$$\forall x', x'' \in \mathcal{X}^\circ : \; \langle \omega'_{\mathcal{X}}(x') - \omega'_{\mathcal{X}}(x''), x' - x''\rangle \geq \varphi_{\mathcal{X}} \|x' - x''\|_{\mathcal{X}}^2. \tag{2}$$

- *Bregman distance*: $V(u\|x) := \omega_{\mathcal{X}}(u) - \omega_{\mathcal{X}}(x) - \langle \omega'_{\mathcal{X}}(x), u - x\rangle$ for all $x \in \mathcal{X}^\circ$ and $u \in \mathcal{X}$.
- *Prox-mapping*: Given a *prox center* $x \in \mathcal{X}^\circ$,

$$\mathrm{Prox}_x(\xi) := \operatorname*{argmin}_{u\in\mathcal{X}} \{\langle \xi, u\rangle + V(u\|x)\} : \mathbf{E} \to \mathcal{X}^\circ.$$

  For properly chosen stepsizes, the prox-mapping becomes a contraction. This is critical in the convergence analysis of FOMs. Furthermore, when the DGF is taken as the squared ℓ_2 norm, the prox mapping becomes the usual projection operation of the vector $x - \xi$ onto $\mathcal{X}$.
- *ω-center*: $x_\omega := \operatorname*{argmin}_{x\in\mathcal{X}} \omega_{\mathcal{X}}(x) \in \mathcal{X}^\circ$ of $\mathcal{X}$.
- *Set width*: $\Omega_x := \max\limits_{x\in\mathcal{X}} V(x\|x_\omega) \leq \max\limits_{x\in\mathcal{X}} \omega_{\mathcal{X}}(x) - \min\limits_{x\in\mathcal{X}} \omega_{\mathcal{X}}(x)$.

The distance-generating functions $\omega_{\mathcal{X}}, \omega_{\mathcal{Y}}$ can be used to create *smoothed approximations* to $\overline{\phi}, \underline{\phi}$ as follows [Nesterov, 2005b]:

$$\overline{\phi}_{\mu_2}(x) = \max_{y \in \mathcal{Y}} \{\phi(x,y) - \mu_2 \omega_{\mathcal{Y}}(y)\}, \tag{3}$$

$$\underline{\phi}_{\mu_1}(y) = \min_{x \in \mathcal{X}} \{\phi(x,y) + \mu_1 \omega_{\mathcal{X}}(x)\}, \tag{4}$$

where $\mu_1, \mu_2 > 0$ are smoothness parameters denoting the amount of smoothing applied. Let $y_{\mu_2}(x)$ and $x_{\mu_1}(y)$ refer to the y and x values attaining the optima in (3) and (4). These can be thought of as *smoothed best responses*. Nesterov [2005b] shows that the gradients of the functions $\overline{\phi}_{\mu_2}(x)$ and $\underline{\phi}_{\mu_1}(y)$ exist and are Lipschitz continuous. The gradient operators and Lipschitz constants are given as follows

$$\nabla \overline{\phi}_{\mu_2}(x) = a_1 + A y_{\mu_2}(x) \quad \text{and} \quad \nabla \underline{\phi}_{\mu_1}(y) = a_2 + A^\top x_{\mu_1}(y),$$

$$L_1\left(\overline{\phi}_{\mu_2}\right) = \frac{\|A\|^2}{\varphi_{\mathcal{Y}} \mu_2} \quad \text{and} \quad L_2\left(\underline{\phi}_{\mu_1}\right) = \frac{\|A\|^2}{\varphi_{\mathcal{X}} \mu_1}.$$

Based on this setup, we formally state the *Excessive Gap Technique* (EGT) of Nesterov [2005a] in Algorithm 1.

ALGORITHM 1: EGT

input : ω-center z_ω, DGF weights μ_1, μ_2, and $\epsilon > 0$
output: $z^t (= [x^t; y^t])$
$x^0 = \text{Prox}_{x_\omega}\left(\mu_1^{-1} \nabla \overline{\phi}_{\mu_2}(x_\omega)\right)$;
$y^0 = y_{\mu_2}(x_\omega)$;
$t = 0$; $z_1 := z_\omega$;
while $\epsilon_{\text{sad}}(z^t) > \epsilon$ **do**
  $\tau_t = \frac{2}{t+3}$;
  if *t is even* **then**
    $(\mu_1^{t+1}, x^{t+1}, y^{t+1}) =$ *Step*$(\mu_1^t, \mu_2^t, x^t, y^t, \tau)$
  else
    $(\mu_2^{t+1}, y^{t+1}, x^{t+1}) =$ *Step*$(\mu_2^t, \mu_1^t, y^t, x^t, \tau)$
  end
  $t = t + 1$;
end

ALGORITHM 2: Step

input : μ_1, μ_2, x, y, τ
output: μ_1^+, x_+, y_+
$\hat{x} = (1-\tau)\, x + \tau x_{\mu_1}(y)$;
$y_+ = (1-\tau)\, y + \tau y_{\mu_2}(\hat{x})$;
$\tilde{x} =$ $\text{Prox}_{x_{\mu_1}(y)}\left(\frac{\tau}{(1-\tau)\mu_1} \nabla \overline{\phi}_{\mu_2}(\hat{x})\right)$;
$x_+ = (1-\tau)\, x + \tau \tilde{x}$;
$\mu_1^+ = (1-\tau)\, \mu_1$;

The EGT algorithm alternates between taking steps focused on $\mathcal{X}$ and $\mathcal{Y}$. Algorithm 2 shows a single step focused on $\mathcal{X}$. Steps focused on y are completely analogous. Algorithm 1 shows how the alternating steps and stepsizes are computed, as well as how initial points are selected.

Suppose the initial values μ_1, μ_2 in the EGT algorithm satisfy $\mu_1 = \frac{\varphi_{\mathcal{X}}}{L_1(\overline{\phi}_{\mu_2})}$. Then, at every iteration $t \geq 1$ of the EGT algorithm, the corresponding solution $z^t = [x^t; y^t]$ satisfies $x^t \in \mathcal{X}$, $y^t \in \mathcal{Y}$, and

$$\overline{\phi}(x^t) - \underline{\phi}(y^t) = \epsilon_{\text{sad}}(z^t) \leq \frac{4\|A\|}{T+1} \sqrt{\frac{\Omega_{\mathcal{X}} \Omega_{\mathcal{Y}}}{\varphi_{\mathcal{X}} \varphi_{\mathcal{Y}}}}.$$

Consequently, [Nesterov, 2005a] proves that the EGT algorithm has a convergence rate of $O(\frac{1}{\epsilon})$.

5 Treeplexes

Hoda et al. [2010] introduce the *treeplex*, a class of convex polytopes that encompass the sequence-form description of strategy spaces in perfect-recall EFGs.

Definition 1. Treeplexes are defined recursively:

1. *Basic sets*: The standard simplex Δ_m is a treeplex.
2. *Cartesian product*: If $Q_1, \ldots, Q_k$ are treeplexes, then $Q_1 \times \cdots \times Q_k$ is a treeplex.
3. *Branching*: Given a treeplex $P \subseteq [0,1]^p$, a collection of treeplexes $Q = \{Q_1, \ldots, Q_k\}$ where $Q_j \subseteq [0,1]^{n_j}$, and $l = \{l_1, \ldots, l_k\} \subseteq \{1, \ldots, p\}$, the set defined by

$$P \boxed{l} Q \coloneqq \left\{ (u, q_1, \ldots, q_k) \in \mathbb{R}^{p+\sum_j n_j} : \; u \in P, \; q_1 \in u_{l_1} \cdot Q_1, \; \ldots, q_k \in u_{l_k} \cdot Q_k \right\}$$

   is a treeplex. In this setup, we say u_{l_j} is the branching variable for the treeplex Q_j.

A treeplex is a tree of simplexes where children are connected to their parents through the branching operation. In the branching operation, the child simplex domain is scaled by the value of the parent branching variable. Understanding the treeplex structure is crucial because the proofs of our main results rely on induction over these structures. For EFGs, the simplexes correspond to the information sets of a single player and the whole treeplex represents that player's strategy space. The branching operation has a sequential interpretation: The vector u represents the decision variables at certain stages, while the vectors q_j represent the decision variables at the k potential following stages, depending on external outcomes. Here $k \leq p$ since some variables in u may not have subsequent decisions. For treeplexes, von Stengel [1996] has suggested a polyhedral representation of the form $Eu = e$ where the matrix E has its entries from $\{-1, 0, 1\}$ and the vector e has its entries in $\{0, 1\}$.

For a treeplex Q, we denote by S_Q the index set of the set of simplexes contained in Q (in an EFG S_Q is the set of information sets belonging to the player). For each $j \in S_Q$, the treeplex rooted at the j-th simplex Δ^j is referred to as Q_j. Given vector $q \in Q$ and simplex Δ^j, we let $\mathbb{I}_j$ denote the set of indices of q that correspond to the variables in Δ^j and define q^j to be the sub vector of q corresponding to the variables in $\mathbb{I}_j$. For each simplex Δ^j and branch $i \in \mathbb{I}_j$, the set $\mathcal{D}_j^i$ represents the set of indices of simplexes reached immediately after Δ^j by taking branch i (in an EFG $\mathcal{D}_j^i$ is the set of potential next-step information sets for the player). Given a vector $q \in Q$, simplex Δ^j, and index $i \in \mathbb{I}_j$, each child simplex Δ^k for every $k \in \mathcal{D}_j^i$ is scaled by q_i. Conversely, for a given simplex Δ^j, we let p_j denote the index in q of the parent branching variable q_{p_j} that Δ^j is scaled by. We use the convention that $q_{p_j} = 1$ if Q is such that no branching operation precedes Δ^j. For each $j \in S_Q$, d_j is the maximum depth of the treeplex rooted at Δ^j, that is, the maximum number of simplexes reachable through a series of branching operations at Δ^j. Then d_Q gives the depth of Q. We use b_Q^j to identify the number of branching operations preceding the j-th simplex in Q. We will say that a simplex j such that $b_Q^j = 0$ is a *root simplex*.

Figure 1 illustrates an example treeplex Q. Q is constructed from nine two-to-three-dimensional simplexes $\Delta^1, \ldots, \Delta^9$. At level 1, we have two root simplexes, Δ^1, Δ^2, obtained by a Cartesian product (denoted by $\times$). We have maximum depths $d_1 = 2$, $d_2 = 1$ beneath them. Since there are no preceding branching operations, the parent variables for these simplexes Δ^1 and Δ^2 are $q_{p_1} = q_{p_2} = 1$. For Δ^1, the corresponding set of indices in the vector q is $\mathbb{I}_1 = \{1, 2\}$, while for Δ^2 we have $\mathbb{I}_2 = \{3, 4, 5\}$. At level 2, we have the simplexes $\Delta^3, \ldots, \Delta^7$. The parent variable of

Δ^3 is $q_{p_3} = q_1$; therefore, Δ^3 is scaled by the parent variable q_{p_3}. Similarly, each of the simplexes $\Delta^3, \ldots, \Delta^7$ is scaled by their parent variables q_{p_j} that the branching operation was performed on. So on for Δ^8 and Δ^9 as well. The number of branching operations required to reach simplexes Δ^1, Δ^3 and Δ^8 is $b_Q^1 = 0, b_Q^3 = 1$ and $b_Q^8 = 2$, respectively.

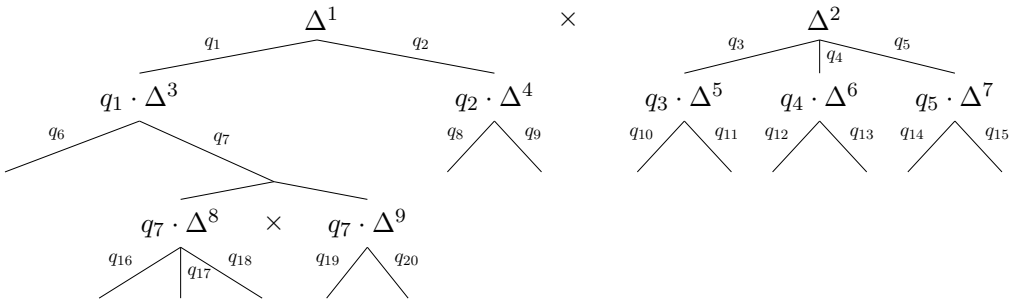

Figure 1: An example treeplex constructed from 9 simplexes. Cartesian product operation is denoted by $\times$.

Note that we allow more than two-way branches; hence our formulation follows that of Kroer et al. [2015] and differs from that of Hoda et al. [2010]. As discussed in Hoda et al. [2010], it is possible to model sequence-form games by treeplexes that use only two-way branches. Yet, this can cause a large increase in the depth of the treeplex, thus leading to significant degradation in the strong convexity parameter. Because we handle multi-way branches directly in our framework, our approach is more effective in taking into account the structure of the sequence-form game and thereby resulting in better bounds on the associated strong convexity parameters and thus overall convergence rates.

Our analysis requires a measure of the size of a treeplex Q. Thus, we define $M_Q \coloneqq \max_{q \in Q} \|q\|_1$.

In the context of EFGs, suppose Q encodes player 1's strategy space; then M_Q is the maximum number of information sets with nonzero probability of being reached when player 1 has to follow a pure strategy while the other player may follow a mixed strategy. We also let

$$M_{Q,r} \coloneqq \max_{q \in Q} \sum_{j \in S_Q : b_Q^j \leq r} \|q^j\|_1. \tag{5}$$

Intuitively, $M_{Q,r}$ gives the maximum value of the ℓ_1 norm of any vector $q \in Q$ after removing the variables corresponding to simplexes that are not within r branching operations of the root of Q.

6 Dilated entropy functions with bounded strong convexity

In this section we introduce DGFs for domains with treeplex structures and establish their strong convexity parameters with respect to a given norm (see (2)).

The basic building block in our construction is the *entropy* DGF given by $\omega_e(z) = -\sum_{i=1}^n z_i \log(z_i)$, for the simplex Δ_n. It is well-known that $\omega_e(\cdot)$ is strongly convex with modulus 1 with respect to the ℓ_1 norm on Δ_n (see Juditsky and Nemirovski [2011a]). We will show that a suitable modification of this function achieves a desirable strong convexity parameter for the treeplex domain.

The treeplex structure is naturally related to the *dilation operation* [Hiriart-Urruty and Lemaréchal, 2001] defined as follows: Given a compact set $K \subseteq \mathbb{R}^d$ and a function $f : K \to \mathbb{R}$, we first define

$$\bar{K} \coloneqq \left\{ (t, z) \in \mathbb{R}^{d+1} : \ t \in [0, 1] \,, \ z \in t \cdot K \right\}.$$

Definition 2. Given a function $f(z)$, the *dilation operation* is the function $\bar{f} : \bar{K} \to \mathbb{R}$ given by

$$\bar{f}(z,t) = \begin{cases} t \cdot f(z/t) & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases}.$$

Based on the dilation operation, we define the dilated entropy function over the simplexes of a treeplex:

Definition 3. Given a treeplex Q and weights $\beta_j > 0$ for each $j \in S_Q$, we define the *dilated entropy function* as

$$\omega(q) = \sum_{j \in S_Q} \beta_j \sum_{i \in \mathbb{I}_j} q_i \log \frac{q_i}{q_{p_j}} \quad \text{for any } q \in Q,$$

where we follow the treeplex notation and p_j is the index of the branching variable preceding Δ^j, with the convention that $q_{p_j} = 1$ if Δ^j has no branching operation preceding it.

Remark 1. *Note that the dilated entropy function $\omega(\cdot)$ defined above is twice differentiable in the relative interior of treeplex Q and admits a continuous gradient selection. Moreover, for large enough weights β_j, we will demonstrate that it is strongly convex w.r.t. the ℓ_1 norm. Thus, the dilated entropy function is compatible with the ℓ_1 norm, as required by the BSPP setup.* ∎

Definition 3 above leads to a subset of the DGFs considered by Hoda et al. [2010]. Our main theoretical result shows that by a careful selection of the weights β_j, we can significantly improve the strong convexity bounds associated with the dilated entropy function. We will consider weights that satisfy the following recurrence:

$$\begin{aligned} &\alpha_j = 1 + \max_{i \in \mathbb{I}_j} \sum_{k \in \mathcal{D}_j^i} \frac{\alpha_k \beta_k}{\beta_k - \alpha_k}, && \forall j \in S_Q, \\ &\beta_j > \alpha_j, && \forall i \in \mathbb{I}_j \text{ and } \forall j \in S_Q \text{ s.t. } b_Q^j > 0, \\ &\beta_j = \alpha_j, && \forall i \in \mathbb{I}_j \text{ and } \forall j \in S_Q \text{ s.t. } b_Q^j = 0. \end{aligned} \tag{6}$$

Intuitively, α_j represents the negative terms that the weight β_j has to cancel out: the constant 1 represents the negative term resulting from the squared norm in the strong convexity requirement; the summation term represents the amount of negative terms accumulated from the induction on simplexes descending from simplex j. The qualifications on β_j ensure that β_j is set such that it at least cancels out the negative terms; the difference $\beta_j - \alpha_j$ controls the amount of negative value the parent simplex has to make up. This is why we set $\beta_j = \alpha_j$ when $b_Q^j = 0$. As part of the proof of Lemma 2 we will see why we require a strict inequality $\beta_j > \alpha_j$ for non-root simplexes.

Based on recurrence (6), our main results establish strong convexity of our dilated entropy DGF w.r.t. the ℓ_2 and ℓ_1 norms:

Theorem 1. *For a treeplex Q, the dilated entropy function with weights satisfying recurrence (6) is strongly convex with modulus 1 with respect to the ℓ_2 norm.*

Theorem 2. *For a treeplex Q, the dilated entropy function with weights satisfying recurrence (6) is strongly convex with modulus $\frac{1}{M_Q}$ with respect to the ℓ_1 norm.*

We give the proofs of Theorems 1 and 2 in Section 6.2. Based on Theorem 2, we get the following corollary:

Corollary 1. *For a treeplex Q, the dilated entropy function with weights $\beta_j = 2+\sum_{r=1}^{d_j} 2^r(M_{Q_j,r}-1)$ for all $j \in S_Q$ is strongly convex with modulus $\frac{1}{M_Q}$ w.r.t. the ℓ_1 norm.*

Corollary 1 follows easily from Theorem 2 and a recursive interpretation of the weights, which is presented as Fact 2 in the next section. In particular, a specific choice of weights in Fact 2 immediately satisfies the recurrence (6) and leads to Corollary 1.

To our knowledge, the best strong convexity bounds for general treeplexes were proved in Kroer et al. [2015]. Corollary 1 improves the prior bounds by exchanging a factor of $|S_Q|$ with a factor of M_Q. Note that $|S_Q|$ is tied to the branching factor associated with branching operations in the treeplex Q whereas M_Q is not. Thus, our result removes the dependence of the strong convexity parameter on the branching factor and hence significantly improves upon Kroer et al. [2015].

In Theorem 3 we use our strong convexity result to establish a polytope diameter that has only a logarithmic dependence on the branching factor. As a consequence, the associated dilated entropy DGF when used in FOMs such as MP and EGT for solving EFGs leads to the same improvement in their convergence rate.

6.1 Preliminary results for the proofs of our main results

We start with some simple facts and a few technical lemmas that are used in our proofs.

Fact 1. *Given a treeplex Q, we have, respectively, for all $i \in \mathbb{I}_j, j \in S_Q$ and all $d = 1, \ldots, d_Q, q \in Q$:*

$$(a) \quad M_{Q_j} \geq 1 + \sum_{l \in \mathcal{D}_j^i} M_{Q_l}, \qquad\qquad (b) \quad M_Q \geq \sum_{j \in S_Q : d_j = d} q_{p_j} M_{Q_j}.$$

Proof. The first inequality was established in Kroer et al. [2015, Lemma 5.7]; the last one follows directly from our treeplex structure and the definition of M_Q. □

Our next observation follows from Fact 1(a) and is advantageous in suggesting a practically useful choice of the weights β_j that can be used for Theorem 2 to arrive at Corollary 1.

Fact 2. *Let Q be a treeplex and $\beta_j = 2 + \sum_{r=1}^{d_j} 2^r(M_{Q_j,r} - 1)$ for all $j \in S_Q$ as in Corollary 1. Then Fact 1(a) implies $\beta_j \geq 2 + \sum_{k \in \mathcal{D}_j^i} 2\beta_k, \forall i \in \mathbb{I}_j$ and $\forall j \in S_Q$.*

Consequently, by selecting $\beta_j = 2\alpha_j$, and $\alpha_j = 1 + \sum_{r=1}^{d_j} 2^{r-1}(M_{Q_j,r} - 1)$ for all $i \in \mathbb{I}_j$ and for all $j \in S_Q$ such that $b_Q^j > 0$, we immediately satisfy the conditions of the recurrence in (6).

Given a twice differentiable function f, we let $\nabla^2 f(z)$ denote its Hessian at z. Our analysis is based on the following sufficient condition for strong convexity of a twice differentiable function:

Fact 3. *A twice-differentiable function f is strongly convex with modulus φ with respect to a norm $\|\cdot\|$ on nonempty convex set $C \subset \mathbb{R}^n$ if $h^\top \nabla^2 f(z) h \geq \varphi \|h\|, \ \forall h \in \mathbb{R}^n, z \in C^\circ$.*

For simplexes Δ^j at depth 1, there is no preceding branching operation; so the variables h_{p_j}, q_{p_j} do not exist. We circumvent this with the convention $h_{p_j} = 0, q_{p_j} = 1$ for such $j \in S_Q$.

In our proofs we will use the following expression for $h^\top \nabla^2 \omega(q) h$.

Lemma 1. *Given a treeplex Q and a dilated entropy function $\omega(\cdot)$ with weights $\beta_j > 0$, we have*

$$h^\top \nabla^2 \omega(q) h = \sum_{j \in S_Q} \beta_j \left[\sum_{i \in \mathbb{I}_j} \left(\frac{h_i^2}{q_i} - \frac{2 h_i h_{p_j}}{q_{p_j}} \right) + \frac{h_{p_j}^2}{q_{p_j}} \right] \quad \forall q \in \operatorname{ri}(Q) \text{ and } \forall h \in \mathbb{R}^n. \tag{7}$$

We provide the proof of Lemma 1 in the appendix. It simply follows from taking the second-order partial derivatives and rearranging terms.

6.2 Proofs of our main theorems

The majority of the work for our strong-convexity results is performed by the following lemma, from which our strong convexity results follow easily.

Lemma 2. *For any treeplex Q, the dilated entropy function with weights satisfying recurrence* (6) *satisfies the following inequality:*

$$h^\top \nabla^2 \omega(q) h \geq \sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \quad \forall q \in \operatorname{ri}(Q) \text{ and } \forall h \in \mathbb{R}^n. \tag{8}$$

Proof. We will first show the following inductive hypothesis over the set of non-root simplexes $\widehat{S}_Q = \left\{ j \in S_Q : \; b_Q^j > 0 \right\}$ for any depth $d \geq 0$:

$$\sum_{j \in \widehat{S}_Q : d_j \leq d} \beta_j \left[\sum_{i \in \mathbb{I}_j} \left(\frac{h_i^2}{q_i} - \frac{2 h_i h_{p_j}}{q_{p_j}} \right) + \frac{h_{p_j}^2}{q_{p_j}} \right] - \sum_{j \in \widehat{S}_Q : d_j \leq d} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \geq - \sum_{j \in \widehat{S}_Q : d_j = d} \frac{\beta_j \alpha_j}{\beta_j - \alpha_j} \frac{h_{p_j}^2}{q_{p_j}}$$

We begin with the inductive step, as the base case will follow from the same logic. Consider a treeplex Q of depth $d > 0$. By applying the inductive hypothesis we have

$$\begin{aligned}
&\sum_{j \in \widehat{S}_Q : d_j \leq d} \beta_j \left[\sum_{i \in \mathbb{I}_j} \left(\frac{h_i^2}{q_i} - \frac{2 h_i h_{p_j}}{q_{p_j}} \right) + \frac{h_{p_j}^2}{q_{p_j}} \right] - \sum_{j \in \widehat{S}_Q : d_j \leq d} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \\
&\geq \sum_{j \in \widehat{S}_Q : d_j = d} \beta_j \left[\sum_{i \in \mathbb{I}_j} \left(\frac{h_i^2}{q_i} - \frac{2 h_i h_{p_j}}{q_{p_j}} \right) + \frac{h_{p_j}^2}{q_{p_j}} \right] - \sum_{j \in \widehat{S}_Q : d_j = d} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} - \sum_{j \in \widehat{S}_Q : d_j = d-1} \frac{\beta_j \alpha_j}{\beta_j - \alpha_j} \frac{h_{p_j}^2}{q_{p_j}}
\end{aligned} \tag{9}$$

Now we can rearrange terms: The sum over $j \in \widehat{S}_Q$ such that $d_j = d-1$ is equivalent to a sum over the immediate descendant information sets $k \in \mathcal{D}_j^i$ inside the square brackets, and we can move the sum over $i \in \mathbb{I}_j$ outside the square brackets by using the fact that $\sum_{i \in \mathbb{I}_j} \frac{q_i}{q_{p_j}} = 1$ and splitting the term $\frac{h_{p_j}^2}{q_{p_j}}$ into separate terms multiplied by $\frac{q_i}{q_{p_j}}$, this gives

$$\begin{aligned}
(9) &= \sum_{j \in \widehat{S}_Q : d_j = d} \sum_{i \in \mathbb{I}_j} \left[\left(\beta_j - 1 - \sum_{k \in \mathcal{D}_j^i} \frac{\beta_k \alpha_k}{\beta_k - \alpha_k} \right) \frac{h_i^2}{q_i} - \left(\frac{2 \beta_j h_i h_{p_j}}{q_{p_j}} \right) + \frac{q_i \beta_j h_{p_j}^2}{q_{p_j}^2} \right] \\
&\geq \sum_{j \in \widehat{S}_Q : d_j = d} \sum_{i \in \mathbb{I}_j} \left[(\beta_j - \alpha_j) \frac{h_i^2}{q_i} - \left(\frac{2 \beta_j h_i h_{p_j}}{q_{p_j}} \right) + \frac{q_i \beta_j h_{p_j}^2}{q_{p_j}^2} \right],
\end{aligned} \tag{10}$$

where the last inequality follows from the definition of α_j.

For indices $j \in S_Q$ such that $b_Q^j > 0$ and $i \in \mathbb{I}_j$, the relations in (6) imply $\beta_j > \alpha_j$, and so the expression inside the square brackets in (10) is a convex function of h_i. Taking its derivative w.r.t. h_i and setting it to zero gives $h_i = \frac{\beta_j}{\beta_j - \alpha_j} \frac{q_i}{q_{p_j}} h_{p_j}$. Thus, we arrive at

$$\begin{aligned}(10) &\geq \sum_{j \in \widehat{S}_Q : d_j = d} \sum_{i \in \mathbb{I}_j} \left[\frac{\beta_j^2}{\beta_j - \alpha_j} \frac{q_i h_{p_j}^2}{q_{p_j}^2} - \frac{\beta_j^2}{\beta_j - \alpha_j} \frac{2 q_i h_{p_j}^2}{q_{p_j}^2} + \frac{q_i \beta_j h_{p_j}^2}{q_{p_j}^2} \right] \\ &= \sum_{j \in \widehat{S}_Q : d_j = d} \frac{h_{p_j}^2}{q_{p_j}} \left[\left(\frac{-\beta_j^2}{\beta_j - \alpha_j} + \beta_j \right) \frac{\sum_{i \in \mathbb{I}_j} q_i}{q_{p_j}} \right] = - \sum_{j \in \widehat{S}_Q : d_j = d} \frac{\beta_j \alpha_j}{\beta_j - \alpha_j} \frac{h_{p_j}^2}{q_{p_j}}.\end{aligned}$$

Hence, the induction step is complete. For the base case $d = 0$ we do not need the inductive assumption: Because $\mathcal{D}_j^i = \emptyset$, $\alpha_j = 1$, and we get (10) by definition; we can then apply the same convexity argument. This proves our inductive hypothesis.

Then using Lemma 1, we now have

$$\begin{aligned}h^\top \nabla^2 \omega(q) h - \sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} &= \sum_{j \in S_Q} \beta_j \left[\sum_{i \in \mathbb{I}_j} \left(\frac{h_i^2}{q_i} - \frac{2 h_i h_{p_j}}{q_{p_j}} \right) + \frac{h_{p_j}^2}{q_{p_j}} \right] - \sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \\ &\geq \sum_{j \in S_Q : b_Q^j = 0} \left[\sum_{i \in \mathbb{I}_j} \beta_j \frac{h_i^2}{q_i} - \sum_{k \in \mathcal{D}_j^i} \frac{\beta_k \alpha_k}{\beta_k - \alpha_k} \frac{h_i^2}{q_i} - \frac{h_i^2}{q_i} \right] \geq 0.\end{aligned}$$

The first inequality follows from the fact that $h_{p_j} = 0$ for all $j \in S_Q$ such that $b_Q^j = 0$, and for all $j \in S_Q$ such that $b_Q^j > 0$, we used our induction. The last inequality follows from (6) and $q_i, h_i^2 \geq 0$. This then proves (8). □

We are now ready to prove our two main theorems, which we restate before proving them.

Theorem 1. *For a treeplex Q, the dilated entropy function with weights satisfying recurrence (6) is strongly convex with modulus 1 with respect to the ℓ_2 norm.*

Proof. Since $q_i \leq 1$, Lemma 2 implies $h^\top \nabla^2 \omega(q) h \geq \sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} h_i^2 = \|h\|_2^2$ for all $q \in \operatorname{ri}(Q)$ and for all $h \in \mathbb{R}^n$. Because the dilated entropy function $\omega(q)$ is twice differentiable on $\operatorname{ri}(Q)$, from Fact 3, we conclude that $\omega(\cdot)$ is strongly convex w.r.t. the ℓ_2 norm on Q with modulus 1.

This analysis is tight: By choosing a vector $q \in \{0,1\}^{|Q|}$ such that $\|q\|_1 = M_Q$, and setting $h_i = \frac{\beta_j}{\beta_j - \alpha_j} \frac{q_i}{q_{p_j}} h_{p_j}$ for all indices i such that $q_i = 1$ and $h_i = 0$ otherwise, every inequality in the proof of Lemma 2 becomes an equality. □

Theorem 2. *For a treeplex Q, the dilated entropy function with weights satisfying recurrence (6) is strongly convex with modulus $\frac{1}{M_Q}$ with respect to the ℓ_1 norm.*

Proof. To show strong convexity with modulus 1 w.r.t. the ℓ_1 norm, we lower bound the right-hand side of (8) in Lemma 2:

$$\sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \geq \frac{1}{M_Q} \left(\sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} q_i \right) \sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{h_i^2}{q_i} \geq \frac{1}{M_Q} \left(\sum_{j \in S_Q} \sum_{i \in \mathbb{I}_j} \frac{|h_i|}{\sqrt{q_i}} \sqrt{q_i} \right)^2 = \frac{1}{M_Q} \|h\|_1^2,$$

where the first inequality follows from the fact that M_Q is an upper bound on $\|q\|_1$ for any $q \in Q$, and the second inequality follows from the Cauchy-Schwarz inequality.

Hence, we deduce $h^\top \nabla^2 \omega(q) h \geq \frac{1}{M_Q}\|h\|_1^2$ holds for all $q \in \mathrm{ri}\,(Q)$ and for all $h \in \mathbb{R}^n$. Because the dilated entropy function $\omega(q)$ is twice differentiable on $\mathrm{ri}\,(Q)$, from Fact 3, we conclude that $\omega(\cdot)$ is strongly convex w.r.t. the ℓ_1 norm on Q with modulus $\varphi = \frac{1}{M_Q}$. □

6.3 Treeplex width

The convergence rates of FOMs such as MP and EGT algorithms depend on the diameter-to-strong convexity parameter ratio $\frac{\Omega}{\varphi}$, as described in Section 4.1. In order to establish full results on the convergence rates of these FOMs, we now bound this ratio using Corollary 1.

Theorem 3. *For a treeplex Q, the dilated entropy function with simplex weights $\beta_j = M_Q(2 + \sum_{r=1}^{d_j} 2^r(M_{Q_j,r} - 1))$ for each $j \in S_Q$ results in $\frac{\Omega}{\varphi} \leq M_Q^2 2^{d_Q+2} \log m$ where m is the dimension of the largest simplex Δ^j for $j \in S_Q$ in the treeplex structure.*

Proof. For our choice of weights β_j, Corollary 1 implies that the resulting dilated entropy function is strongly convex with modulus $\varphi = 1$. Hence, we only need to bound Ω.

Any vector $q \in Q$ satisfying $q_i \in \{0, 1\}$ for all i maximizes $\omega(q)$ and results in $\max_{q \in Q} \omega(q) = 0$. For the minimum value, consider any $q \in \mathrm{ri}\,(Q)$. Applying the well-known lower bound of $\log m$ for the entropy function on an m-dimensional simplex, we have

$$\begin{aligned}
\omega(q) &= \sum_{j \in S_Q} \beta_j q_{p_j} \sum_{i \in \mathbb{I}_j} \frac{q_i}{q_{p_j}} \log \frac{q_i}{q_{p_j}} \geq -\sum_{j \in S_Q} \beta_j q_{p_j} \log m = -\sum_{d=0}^{d_Q} \sum_{j \in S_Q : d_j = d} \beta_j q_{p_j} \log m \\
&= -\sum_{d=1}^{d_Q} \sum_{j \in S_Q : d_j = d} \beta_j q_{p_j} \log m - \sum_{j \in S_Q : d_j = 0} \beta_j q_{p_j} \log m \\
&= -M_Q \log m \sum_{d=1}^{d_Q} \sum_{j \in S_Q : d_j = d} q_{p_j} \left(2 + \sum_{r=1}^{d} 2^r (M_{Q_j,r} - 1)\right) - M_Q \sum_{j \in S_Q : d_j = 0} 2 q_{p_j} \log m \\
&\geq -M_Q \log m \sum_{d=1}^{d_Q} \sum_{j \in S_Q : d_j = d} q_{p_j} M_{Q_j} \sum_{r=1}^{d} 2^r - 2 M_Q \log m \sum_{j \in S_Q : d_j = 0} q_{p_j}, \qquad (11)
\end{aligned}$$

where the last inequality follows because for each $j \in S_Q$ with $d_j = 0$, the definition of M_Q implies $\sum_{j \in S_Q : d_j = 0} q_{p_j} \leq M_Q$, and for each $j \in S_Q$ with $d_j = d \geq 1$, we have $2 + \sum_{r=1}^{d} 2^r (M_{Q_j,r} - 1) \leq \sum_{r=1}^{d} 2^r M_{Q_j,r} \leq \sum_{r=1}^{d} 2^r M_{Q_j}$ since $M_{Q_j,r} \leq M_{Q_j}$. Also, from Fact 1(b), we have $\sum_{j \in S_Q : d_j = d} q_{p_j} M_{Q_j} \leq M_Q$. Then we arrive at

$$\begin{aligned}
(11) &\geq -M_Q^2 \log m \left(2 + \sum_{d=1}^{d_Q} \sum_{r=1}^{d} 2^r\right) = -M_Q^2 \log m \left(2 + \sum_{d=1}^{d_Q} (2^{d+1} - 2)\right) \\
&= -M_Q^2 \log m \left(2 + \sum_{d=1}^{d_Q} 2^{d+1} - 2 d_Q\right) \geq -M_Q^2 (\log m) 2^{d_Q+2},
\end{aligned}$$

where the last inequality follows because for $d_Q = 0$ we have $2^{d_Q+2} = 4 > 2$ and for $d_Q \geq 1$ we have $2 d_Q \geq 2$.

This lower bound on the minimum value, i.e., $\min_{q \in Q} \omega(q) \geq -M_Q^2 (\log m) 2^{d_Q+2}$, coupled with $\max_{q \in Q} \omega(q) \leq 0$, establishes the theorem. □

7 EGT for extensive-form game solving

We now describe how to instantiate EGT for solving two-player zero-sum EFGs of the form (1) with treeplex domains. Below we state the customization of all the definitions from Section 4 for our problem.

Let m be the size of the largest simplex in either of the treeplexes $\mathcal{X}, \mathcal{Y}$. Because $\mathcal{X}$ and $\mathcal{Y}$ are treeplexes, it is immediately apparent that they are closed, convex, and bounded. We use the ℓ_1 norm on both of the embedding spaces $\mathbf{E}_x, \mathbf{E}_y$. As our DGFs for $\mathcal{X}, \mathcal{Y}$ are compatible with the ℓ_1 norm, we use the dilated entropy DGF scaled with weights given in Theorem 3. Then Theorem 3 gives our bound on $\frac{\Omega_\mathcal{X}}{\varphi_\mathcal{X}}$ and $\frac{\Omega_\mathcal{Y}}{\varphi_\mathcal{Y}}$. Because the dual norm of the ℓ_1 norm is the ℓ_∞ norm, the matrix norm is given by: $\|A\| = \max_{y \in \mathcal{Y}} \{\|Ay\|_1^* : \|y\|_1 = 1\} = \max_{i,j} |A_{i,j}|$.

Remark 2. *Note that $\|A\|$ is not at the scale of the maximum payoff difference in the original game. The values in A are scaled by the probability of the observed nature outcomes on the path of each sequence. Thus, $\|A\|$ is exponentially smaller (in the number of observed nature steps on the path to the maximizing sequence) than the maximum payoff difference in the original EFG.*

Theorem 3 immediately leads to the following convergence rate result for FOMs equipped with dilated entropy DGFs to solve EFGs (and more generally BSPPs over treeplex domains).

Theorem 4. *Consider a BSPP over treeplex domains $\mathcal{X}, \mathcal{Y}$. Then EGT algorithm equipped with the dilated entropy DGF with weights $\beta_j = 2 + \sum_{r=1}^{d_j} 2^r (M_{\mathcal{X}_j,r} - 1)$ for all $j \in S_\mathcal{X}$ and the corresponding setup for $\mathcal{Y}$ will return an ϵ-accurate solution to the BSPP in at most the following number of iterations:*

$$\frac{\max_{i,j} |A_{i,j}| \sqrt{M_\mathcal{X}^2 2^{d_\mathcal{X}+2} M_\mathcal{Y}^2 2^{d_\mathcal{Y}+2}} \log m}{\epsilon}.$$

This rate in Theorem 4, to our knowledge, establishes the state-of-the-art for FOMs with $O(\frac{1}{\epsilon})$ convergence rate for EFGs.

7.1 Improvements in extensive-form game convergence rate

The ratio $\frac{\Omega}{\varphi}$ of set diameter over the strong convexity parameter is important for FOMs that rely on a prox function, such as EGT and MP. Compared to the rate obtained by [Kroer et al., 2015], we get the following improvement: for simplicity, assume that the number of actions available at each information set is on average a, then our bound improves the convergence rate of [Kroer et al., 2015] by a factor of $\Omega(d_\mathcal{X} \cdot a^{d_\mathcal{X}} + d_\mathcal{Y} \cdot a^{d_\mathcal{Y}})$.

As mentioned previously, Hoda et al. [2010] proved only explicit bounds for the special case of uniform treeplexes that are constructed as follows: 1) A base treeplex Q_b along with a subset of b indices from it for branching operations is chosen. 2) At each depth d, a Cartesian product operation of size k is applied. 3) Each element in a Cartesian product is an instance of the base treeplex with a size b branching operation leading to depth $d-1$ uniform treeplexes constructed in the same way. Given bounds Ω_b, φ_b for the base treeplex, the bound of Hoda et al. [2010] for a uniform treeplex with d uniform treeplex levels (note that the total depth of the constructed treeplex is $d \cdot d_{Q_b}$, where d_{Q_b} is the depth of the base treeplex Q_b) is

$$\frac{\Omega}{\varphi} \leq O\left(b^{2d-2} k^{2d+2} d^2 M_{Q_b}^2 \frac{\Omega_b}{\varphi_b}\right),$$

Then when the base treeplex is a simplex of dimension m, their bound for the dilated entropy on a uniform treeplex Q becomes

$$\frac{\Omega}{\varphi} \leq O\left(|S_Q|^2 d_Q^2 \log m\right).$$

Even for the special case of a uniform treeplex with a base simplex, comparing Theorem 3 to their bound, we see that our general bound improves the associated constants by exchanging $O(|S_Q|^2 d_Q^2)$ with $O(M_Q^2 2^{d_Q})$. Since M_Q does not depend on the branching operation in the treeplex, whereas $|S_Q|$ does, these are also the first bounds to remove any dependence on the branching operation. Note also that there exist games where $M_Q = \sqrt{|S_Q|}$, and in general M_Q is much smaller than $|S_Q|$. Consequently, our results establish the best known convergence results for all FOMs based on dilated entropy DGF such as EGT, MP, and stochastic variants of BSPP algorithms.

CFR, CFR+, and EGT all need to keep track of a constant number of current and/or average iterates, so the memory usage of all three algorithms is of the same order; when gradients are computed using an iterative approach as opposed to storing matrices or matrix decompositions, each algorithm requires a constant times the number of sequences in the sequence-form representation. Therefore, we compare mainly the number of iterations required by each algorithm. Since the theoretical properties of CFR and CFR+ are comparable, we compare to CFR, with all statements being valid for CFR+ as well.

CFR has a $O(\frac{1}{\epsilon^2})$ convergence rate; but its dependence on the number of information sets is only linear (and sometimes sublinear [Lanctot et al., 2009]). Since our results have a quadratic dependence on M_Q^2, CFR sometimes has a better dependence on game constants and can be more attractive for obtaining low-quality solutions quickly for games with many information sets. MCCFR and CFR+ have a similar convergence rate [Lanctot et al., 2009], though MCCFR has cheaper iterations.

Gilpin et al. [2012] give an equilibrium-finding algorithm presented as $O(\ln(\frac{1}{\epsilon}))$; but this form of their bound has a dependence on a certain condition number of the A matrix. Specifically, their iteration bound for sequential games is $O(\frac{\|A\|_{2,2} \cdot \ln(\|A\|_{2,2}/\epsilon) \cdot \sqrt{D}}{\delta(A)})$, where $\delta(A)$ is the condition number of A, $\|A\|_{2,2} = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$ is the Euclidean matrix norm, and $D = \max_{x,\bar{x} \in \mathcal{X}, y,\bar{y} \in \mathcal{Y}} \|(x,y) - (\bar{x},\bar{y})\|_2^2$. Unfortunately, the condition number $\delta(A)$ is only shown to be finite for these games. Without any such unknown quantities based on condition numbers, Gilpin et al. [2012] establish a convergence rate of $O(\frac{\|A\|_{2,2} \cdot D}{\epsilon})$. This algorithm, despite having the same dependence on ϵ as ours in its convergence rate, i.e., $O(\frac{1}{\epsilon})$, suffers from worse constants. In particular, there exist matrices such that $\|A\|_{2,2} = \sqrt{\|A\|_{1,\infty}\|A\|_{\infty,1}}$, where $\|A\|_{1,\infty}$ and $\|A\|_{\infty,1}$ correspond to the maximum absolute column and row sums, respectively. Then together with the value of D, this leads to a cubic dependence on the dimension of Q. For games where the players have roughly equal-size strategy spaces, this is equivalent to a constant of $O(M_Q^4)$ as opposed to our constant of $O(M_Q^2)$.

8 Numerical experiments

We carry out numerical experiments to investigate the practical performance of EGT on EFGs when instantiated with our DGF.

First, we investigate the impact of applying the weights used in recurrence (6), as compared to the previous scheme introduced in Kroer et al. [2015]. To instantiate recurrence (6) we have to choose a way to set β_j relative to α_j. Experimentally, we found that the best way to instantiate the recurrence is to use $\beta_j = \alpha_j$ for all j, in spite of the strict inequality required for our proof. This scheme will henceforth be referred to as new weights. We compare these new weights to the

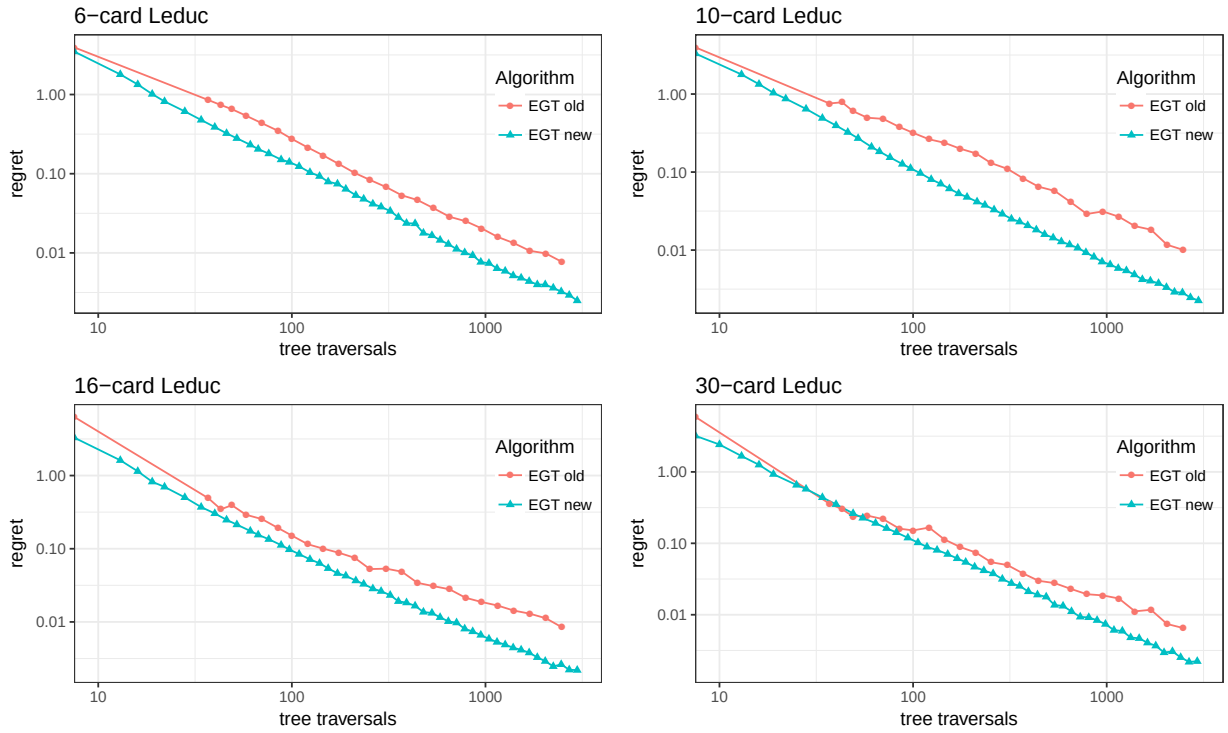

Figure 2: Regret as a function of the number of iterations for EGT with our weighting scheme (EGT new) and with the weighting scheme from Kroer et al. [2015] (EGT old). Both axes are on a log scale.

weights used in Kroer et al. [2015] (henceforth referred to as old weights). Figure 2 shows the result of running EGT with the old and the new weights. For both the old and the new weights, we found that the scalars M_Q and $|S_Q|$ applied to each DGF in order to achieve strong convexity modulus 1 according to Corollary 1 and Theorem 5.4 of Kroer et al. [2015], respectively, are too conservative. Instead, we show the results after tuning these parameters for the corresponding algorithms to yield the best results for each weight scheme. Anecdotally, we found that the old weights are more sensitive and more difficult to tune. The performance also seems more jittery; this is evident even in the strongest parameter we found (especially noticeable on 10, 16, and 30-card Leduc in Figure 2).

We compare the performance of EGT to that of CFR and CFR+ algorithms on a scaled up variant of the poker game Leduc hold'em [Southey et al., 2005], a benchmark problem in the imperfect-information game-solving community. In our version, the deck consists of k pairs of cards $1 \dots k$, for a total deck size of $2k$. Setting $k = 3$ yields the standard Leduc game. Each player initially pays one chip to the pot, and is dealt a single private card. After a round of betting, a community card is dealt face up. After a subsequent round of betting, if neither player has folded, both players reveal their private cards. If either player pairs their card with the community card, they win the pot. Otherwise, the player with the highest private card wins. In the event both players have the same private card, they draw and split the pot.

The results are shown in Figure 3. Each graph is a loglog plot that shows the results for a particular instance of Leduc with $6, 10, 16$ and 30 card decks, respectively. For each graph, we show the performance of all three algorithms, with the x-axis showing the number of tree traversals, and the y-axis showing the sum of regrets over the two players. We find that EGT instantiated with our DGF significantly outperforms both CFR and CFR+ across all four variants of Leduc. This

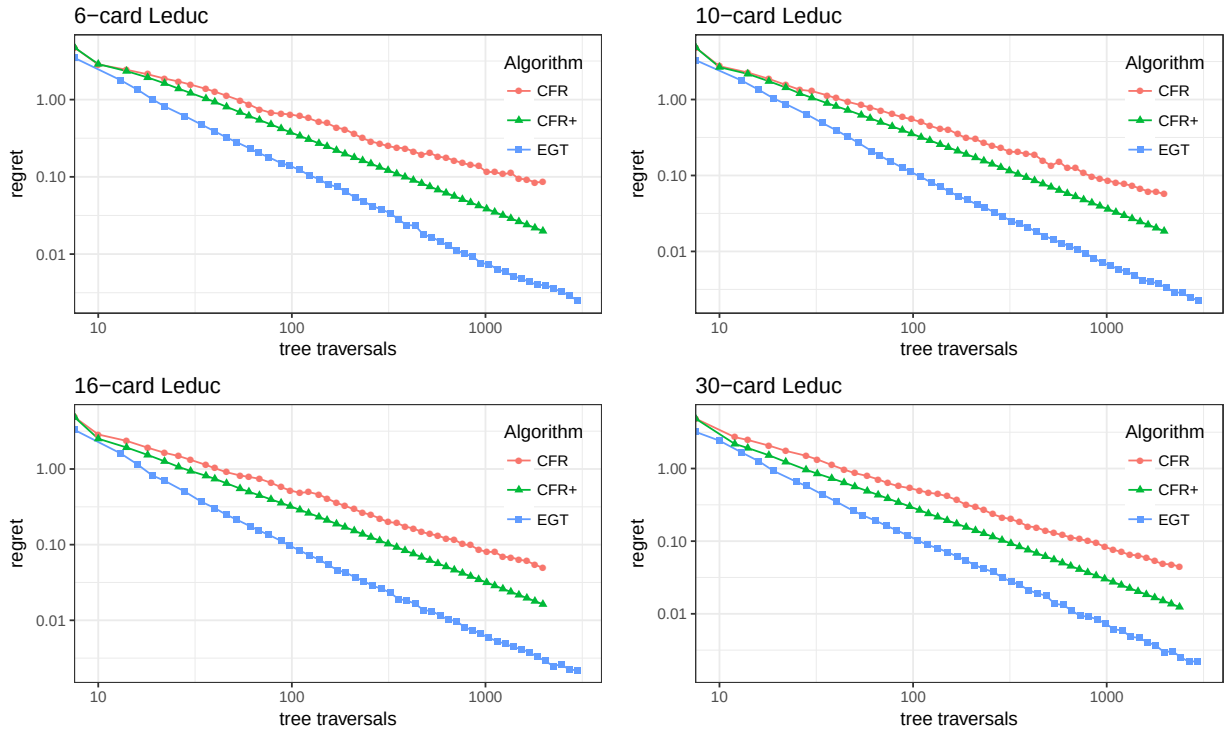

Figure 3: Regret as a function of the number of tree traversals in four different variants of Leduc hold'em for the CFR, CFR+, and EGT algorithms. Both axes are shown on a log scale.

is the case across all iterations; EGT finds a stronger initial point in x^0, y^0 (see Algorithm 1), and maintains a stronger convergence rate across all iterations.

The performance we get from EGT relative to CFR and CFR+ is surprising due to what the conventional wisdom in the field has been. In Kroer et al. [2015] it was found that, while EGT has better convergence rate, CFR (which performs worse than CFR+) had better initial performance, and it was only after a certain number of iterations that EGT took over. Furthermore, the switch point where EGT is preferable was found to shift outward on the x-axis as the Leduc game size was increased. This sentiment has been mirrored by Brown and Sandholm [2016]. In contrast to this, we find that our DGF along with proper initialization leads to EGT performing better than not only CFR, but also CFR+, at every point on the x-axis. Furthermore, scaling up the game size does not seem to adversely affect this relationship.

While the experiments in Figure 3 are very interesting from the perspective of which algorithm to use for large-scale EFG-solving in practice going forward, there are some caveats to keep in mind. First, we only considered number of tree traversals in our performance calculations. However, CFR algorithms have the ability to avoid parts of the tree traversal. For games where accelerated best-response calculation [Johanson et al., 2011] can be applied, e.g., poker-like games, this is unlikely to have a big effect. But, for some other games, this aspect can be important, though note that Brown et al. [2017] showed experimentally that pruning can be used in EGT as well. Second, to get superior performance from EGT, we had to hand-tune initialization parameters relating to our DGF, whereas CFR+ requires no tuning. Development of an algorithmic scheme for choosing this tuning parameter in EGT can make it significantly easier to apply the tuned variant of EGT in practice. Third, on another practical aspect, CFR+ is a conceptually very simple algorithm, and thus also easy to implement. In contrast to this, EGT and our DGF requires a safe-guarded

numerical implementation because the prox operator associated with our DGF requires taking exponentials.

9 Conclusions

We have investigated FOMs for computing Nash equilibria in two-player zero-sum perfect-recall EFGs. On the theoretical side, we analyzed the strong convexity properties of the dilated entropy DGF over treeplexes. By introducing specific weights that are tied to the structure of the treeplex, we improved prior results on treeplex diameter from $O(|S_Q|^2 d_Z^2)$ to $O(M_Q^2 2^{d_Z})$, thereby removing all dependence on branching from the Cartesian product operator in the treeplex definition. Our results generalize to any treeplex, whereas the prior results were for only uniform treeplexes, a significant restriction. These results lead to significant improvements in the convergence rates of many FOMs that can be equipped with dilated entropy DGFs and used for EFG solving including but not limited to EGT, MP, and Stochastic MP.

We numerically investigated the performance of EGT and compared it to the practical state-of-the-art algorithms CFR and CFR+. Our experiments showed that EGT with the dilated entropy DGF, when tuned with a proper scaling, has better practical, as well as theoretical, convergence rate than CFR+, the current state-of-the-art algorithm in practice. While our scaling parameter for the DGF did not require extensive tuning, we believe a more principled way of setting it is worthy of further future investigation.

Theorems 1 and 2 establish bounds for a general class of weights β_j satisfying the recurrence (6). Then in Corollary 1, we have selected a particular weighting scheme for β_j satisfying (6) and performed our numerical tests. There may be other interesting choices of β_j satisfying the recurrence (6). Thus, finding a way to optimally choose among the set of weights satisfying (6) to minimize the polytope diameter for specific games is appealing.

On a separate note, in practice CFR is often paired with an abstraction technique [Sandholm, 2010] such as those mentioned in Section 2. This is despite the lack of any theoretical justification. Effective ways to pair FOMs such as MP and EGT with practical abstraction techniques [Brown et al., 2015] or abstraction techniques that achieve solution-quality guarantees [Lanctot et al., 2012, Kroer and Sandholm, 2014, 2016] are also worth further consideration.

References

B Bosansky, Christopher Kiekintveld, V Lisy, and Michal Pechoucek. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research*, 2014.

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218), January 2015.

Noam Brown and Tuomas Sandholm. Regret transfer and parameter optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

Noam Brown and Tuomas Sandholm. Strategy-based warm starting for regret minimization in games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

Noam Brown, Sam Ganzfried, and Tuomas Sandholm. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit Texas Hold'em

agent. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015.

Noam Brown, Christian Kroer, and Tuomas Sandholm. Dynamic thresholding and pruning for regret minimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1), 2009.

Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. *Games and Economic Behavior*, 92, 2015.

Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5), 2007.

Andrew Gilpin, Javier Peña, and Tuomas Sandholm. First-order algorithm with $\mathcal{O}(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games. *Mathematical Programming*, 133(1–2), 2012.

John Hawkin, Robert Holte, and Duane Szafron. Automated action abstraction of imperfect information extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

John Hawkin, Robert Holte, and Duane Szafron. Using sliding windows to generate action abstractions in extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. 2001.

Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2), 2010.

Albert Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 2011.

Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

Anatoli Juditsky and Arkadi Nemirovski. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. *Optimization for Machine Learning*, 2011a.

Anatoli Juditsky and Arkadi Nemirovski. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, 2011b.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2), 1996.

Christian Kroer and Tuomas Sandholm. Extensive-form game abstraction with bounds. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2014.

Christian Kroer and Tuomas Sandholm. Imperfect-recall abstractions with bounds in games. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster first-order methods for extensive-form game solving. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2015.

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

Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003. ACM.

Michael Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in no-limit poker. *arXiv preprint arXiv:1701.01724*, 2017.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1), 2004.

Yurii Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal of Optimization*, 16(1), 2005a.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103, 2005b.

I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962.

Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, Winter 2010. Special issue on Algorithmic Game Theory.

Tuomas Sandholm. Abstraction for solving large incomplete-information games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. Senior Member Track.

Tuomas Sandholm and Satinder Singh. Lossy stochastic game abstraction with bounds. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 2012.

Jiefu Shi and Michael Littman. Abstraction methods for game theoretic poker. In *CG '00: Revised Papers from the Second International Conference on Computers and Games*, London, UK, 2002. Springer-Verlag.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2), 1996.

Kevin Waugh and Drew Bagnell. A unified view of large-scale zero-sum equilibrium computation. In *Computer Poker and Imperfect Information Workshop at the AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

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

A Omitted proofs

A.1 Proof of Lemma 1

Proof. Consider $q \in \mathrm{ri}\,(Q)$ and any $h \in \mathbb{R}^n$. For each $j \in S_Q$ and $i \in \mathbb{I}_j$, the second-order partial derivates of $\omega(\cdot)$ w.r.t. q_i are:

$$\nabla^2_{q_i^2}\omega(q) = \frac{\beta_j}{q_i} + \sum_{k\in\mathcal{D}_j^i}\sum_{l\in\mathbb{I}_k}\frac{\beta_k q_l}{q_i^2} = \frac{\beta_j}{q_i} + \sum_{k\in\mathcal{D}_j^i}\frac{\beta_k}{q_i}, \tag{12}$$

where the last equality holds because $k \in \mathcal{D}_j^i$ and thus $\sum_{l\in\mathbb{I}_k} q_l = \|q^k\|_1 = q_{p_k} = q_i$. Also, for each $j \in S_Q, i \in \mathbb{I}_j, k \in \mathcal{D}_j^i$, and $l \in \mathbb{I}_k$, the second-order partial derivates w.r.t. q_i, q_l are given by:

$$\nabla^2_{q_i,q_l}\omega(q) = \nabla^2_{q_l,q_i}\omega(q) = -\frac{\beta_k}{q_i}. \tag{13}$$

Then equations (12) and (13) together imply

$$h^\top\nabla^2\omega(q)h = \sum_{j\in S_Q}\sum_{i\in\mathbb{I}_j}\left[h_i^2\left(\frac{\beta_j}{q_i} + \sum_{k\in\mathcal{D}_j^i}\frac{\beta_k}{q_i}\right) - \sum_{k\in\mathcal{D}_j^i}\sum_{l\in\mathbb{I}_k} h_i h_l\frac{2\beta_k}{q_i}\right]. \tag{14}$$

Given $j \in S_Q$ and $i \in \mathbb{I}_j$, we have $p_k = i$ for each $k \in \mathcal{D}_j^i$ and for any $k \in \mathcal{D}_j^i$, there exists some other $j' \in S_Q$ corresponding to k in the outermost summation. Then we can rearrange the following terms:

$$\sum_{j\in S_Q}\sum_{i\in\mathbb{I}_j} h_i^2\sum_{k\in\mathcal{D}_j^i}\frac{\beta_k}{q_i} = \sum_{j\in S_Q}\beta_j\frac{h_{p_j}^2}{q_{p_j}} \quad\text{and}\quad \sum_{j\in S_Q}\sum_{i\in\mathbb{I}_j}\sum_{k\in\mathcal{D}_j^i}\sum_{l\in\mathbb{I}_k} h_i h_l\frac{2\beta_k}{q_i} = \sum_{j\in S_Q}\sum_{i\in\mathbb{I}_j}\beta_j\frac{2h_i h_{p_j}}{q_{p_j}}.$$

Using these two equalities in (14) leads to (7) and proves the lemma. □