Better Regularization for Sequential Decision Spaces: Fast Convergence Rates for Nash, Correlated, and Team Equilibria

GABRIELE FARINA, Carnegie Mellon University, USA
CHRISTIAN KROER, Columbia Univesity, USA
TUOMAS SANDHOLM, Carnegie Mellon University & Strategic Machine, Inc. & Strategy Robot, Inc. & Optimized Markets, Inc., USA

We study the application of iterative first-order methods to the problem of computing equilibria of large-scale two-player extensive-form games. First-order methods must typically be instantiated with a regularizer that serves as a distance-generating function for the decision sets of the players. For the case of two-player zero-sum games, the state-of-the-art theoretical convergence rate for Nash equilibrium is achieved by using the dilated entropy function. In this paper, we introduce a new entropy-based distance-generating function for two-player zero-sum games, and show that this function achieves significantly better strong convexity properties than the dilated entropy, while maintaining the same easily-implemented closed-form proximal mapping. Extensive numerical simulations show that these superior theoretical properties translate into better numerical performance as well.

We then generalize our new entropy distance function, as well as general dilated distance functions, to the scaled extension operator. The scaled extension operator is a way to recursively construct convex sets, which generalizes the decision polytope of extensive-form games, as well as the convex polytopes corresponding to correlated and team equilibria. By instantiating first-order methods with our regularizers, we develop the first accelerated first-order methods for computing correlated equilibra and ex-ante coordinated team equilibria. Our methods have a guaranteed $1/T$ rate of convergence, along with linear-time proximal updates.

1 INTRODUCTION

Large-scale *extensive-form game (EFG)* models have been used in several recent AI milestones, where equilibrium approximation was used as the approach for building AI agents [4, 5, 7, 27]. A crucial component for constructing these agents is a fast method for computing approximate Nash equilibria in large and very-large game models. For the two-player zero-sum setting, an EFG can be solved in polynomial time using a linear program (LP) whose size is linear in the size of the game tree [36]. However, this LP-based approach was not used in any of these AI milestones. Instead, fast iterative methods are preferred [6, 12, 20, 23, 35, 38] as well as sampling-based variants [5, 7, 14, 19, 24, 26, 33]. The reason for this is that constructing the LP, and running simplex or interior-point methods on it, is too expensive for these large-scale models. In contrast, iterative methods only require oracle access to one or two gradient computations, or even estimates thereof, in order to perform an iteration.

From a theoretical standpoint, the fastest iterative methods for solving two-player zero-sum games are *first-order methods (FOMs)* such as the excessive gap technique [30] or mirror prox [29], which converge at a rate of $1/T$, where T is the number of iterations. In order to apply these methods to EFGs, they must be instantiated with a *distance-generating function* (DGF), which yields an appropriate notion of how to measure distances between strategies in the game. In this framework, the convex set of all strategies belonging to a player is referred to as the sequence-form polytope, and alternatively as a *treeplex* [20], which is a tree-like structure of scaled simplexes. Essentially the only sequence-form poltyope DGFs that are known are based on the dilated DGF framework introduced by Hoda et al. [20] (apart from using the standard ℓ_2 distance, which is unsuitable due to projection requirements at each iteration). For example, the dilated entropy distance yields the best current rate of convergence for $1/T$ methods [23]. One drawback of the dilated entropy DGF, as well as other dilated DGFs, is that current analyses incur a dependence of the form $2^{\mathfrak{D}}$, where $\mathfrak{D}$ is the depth of the decision space [20, 22, 23]. In some cases this is reasonable, since the decision space itself may have exponential size in the depth of the game tree. However, in other cases the decision space may have substantial structure, such that this exponential complexity in depth makes the bounds exponential in the size of the game tree.

In this paper, we introduce the first DGF for sequence-form polytopes whose strong convexity is not derived from its structure as a dilated distance function (again, the standard Euclidean distance also satisfies this, but it requires difficult projections). In particular, we show that a weighted version of the negative entropy for the nonnegative unit cube is a superior DGF for sequence-form polytopes. First, we show that this DGF can achieve strong convexity modulus $1/M_Q$ (where M_Q is the maximum value of the ℓ_1 norm on Q), with the largest weights at individual decision points being on the order of $M_Q \log n$ (where $\log n$ is the largest number of actions at any decision point), which improves upon that of the dilated entropy DGF by a factor of $2^{\mathfrak{D}}$. This also translates into an improvement to the theoretical convergence rate of FOMs by a factor $2^{\mathfrak{D}+2}$. A particularly appealing part of this result is that our analysis depends only on the ℓ_1 norm of the sequence-form polytope, and has no exponential dependence on the depth. At the same time, we must also ensure that the new DGF allows fast computation of the associated proximal steps required for, for example, mirror prox or EGT. We show that this is indeed the case: the weights in our new DGF are chosen in a way that allows us to show that this new DGF corresponds to a particular dilated entropy DGF on the sequence-form polytope (while being different outside the sequence-form polytope). This allows us to use existing results on fast proximal-step computation for dilated entropy. We call our new DGF the *dilatable global entropy* (DGE).

After introducing DGE for sequence-form polytopes, we switch our focus to studying DGFs for the more general *scaled extension* operator [17]. The scaled extension operator is a method for

iteratively constructing a convex set as a sequence of convexity-preserving compositions of convex sets. This operator can be used to construct the sequence-form polytope, but more importantly for our purposes it can also be used to construct more general sets such as the polytope of correlation plans needed for computing optimal extensive-form correlated equilibria and ex-ante coordinated team strategies in certain classes of games where it is known that those solution concepts can be computed in polynomial time. First, we show how to extend the class of dilated DGFs to polytopes constructed via scaled extension, thereby generalizing the framework of Hoda et al. [20] beyond sequence-form polytopes, while also giving a simpler proof of strong convexity. This enables DGFs such as the dilated entropy or dilated Euclidean distance to be applied to a much broader class of polytopes. Then, we show that our DGE construction can also be extended to scaled extension. Taken together, we generalize the entire class of known "nice" DGFs for sequence-form polytopes to the set of all polytopes which can be constructed via scaled extension. Applying these results to the problems of computing optimal correlated solution concepts and ex-ante coordinated team strategies yields the *first* method for iteratively solving these problems at a rate of $1/T$, while enjoying fast closed-form solutions at each iteration. In contrast, the only prior result of this form required using the standard Euclidean distance, and thus had to perform expensive projections at every iteration of the algorithm [16].

Extensive experiments validate the efficacy of our new DGFs. We find that these new DGFs lead to much smaller amounts of smoothing, while still ensuring correctness of the algorithms. Intuitively, this means that we can safely take much larger steps at each iteration.

Paper Outline. The paper is structured as follows. Section 3 presents background on first-order methods, which includes the description of the DGFs needed for setting up these methods. That section can be skimmed for notation, if the reader is already familiar with FOMs. Section 2 gives an introduction to extensive-form games. That section can be skimmed for notation, if the reader is already familiar with EFGs. Section 4 introduces the sequence-form polytope, and presents our new DGF for that polytope, along with the convergence rate obtained when combined with a FOM. Section 6 develops DGFs for the scaled extension operator, and shows how this leads to efficient FOMs with a $1/T$ convergence rate for correlated and team equilibria. Section 7 provides an extensive set of computational evaluations of our new DGFs for various games and types of equilibrium.

2 PRELIMINARIES ON EXTENSIVE-FORM GAMES

An extensive form game (EFG) is a game played on a tree. Every node in the tree belongs to some player, whose turn it is to act, and the set of branches at the node correspond to the set of actions available to the player. In general, a strategy for a player may consist of choosing a probability distribution over the actions at each node in the game. Additionally, there may be special nodes called *chance nodes*, which have a fixed distribution over actions associated with them. These nodes model stochastic outcomes, for example, the dealing out of cards in a card game or the valuation signals sent to buyers in a sequential auction. At leaf nodes the game ends, and each leaf node is associated with a vector of payoffs, one payoff per player. The goal of each player in the game is to maximize the expected value of their leaf-node payoffs. Finally, an EFG can model imperfect information: an *information set* is a group of nodes belonging to a player such that the player cannot distinguish among those nodes, and is therefore required to have the same probability distribution over actions at each node in the information set. An example of an information set would be in a poker game, where the information set represents all the cards seen by the player, as well as all bets (which are public). Each node in the information set would correspond to different possible hands held by the other player(s).
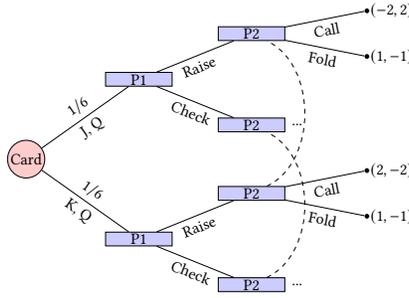
Fig. 1. Example poker game. The red "Card" node is a chance node. Only a subset of the possible cards dealt out by chance are shown ("J,Q" and "K,Q"). Dotted lines denote nodes that belong to the same information set.

Figure 1 illustrates a part of a poker game tree. At each node, either a card is dealt at random to each player (for space reasons, we only show two branches of cards dealt, even though there would be more in a real poker game) or some player acts. The payoffs are at the leaves. They are zero sum in this example.

A *solution concept* provides a definition of rationality. For a given EFG, the application of a solution concept yields a set of equilibria, where each equilibrium has one strategy per player. A strategy describes how a player acts at every one of her information sets. For example, a *Nash equilibrium* is a set of strategies such that each player cannot improve their expected utility by switching to another strategy, when the strategies of other players are held fixed. We will introduce various solution concepts in the sections where we give algorithms for them.

3 PRELIMINARIES ON FIRST-ORDER METHODS

The types of FOMs that we will consider rely on access to a function d which is used to construct a notion of distance between pairs of points in the decision space. The soundness of the algorithm requires such a function to satisfy a number of properties. One of the conditions that we will need is that d be a *Legendre function*, which is defined as follows. (The definition is not entirely standardized; the one we use here is the same one used by Cesa-Bianchi and Lugosi [9].)

Definition 3.1. Let $d : \mathcal{A} \to \mathbb{R}$ be a convex function whose convex domain $\mathcal{A}$ has nonempty interior int($\mathcal{A}$). The function d is said to be a *Legendre function* if

(1) d is differentiable on int($\mathcal{A}$) of $\mathcal{A}$;
(2) for any sequence $x_1, x_2, \dots \in \text{int}(\mathcal{A})$ that converges to a boundary point of $\mathcal{A}$, the norms $\|\nabla d(x_n)\| \to \infty$;
(3) d is strictly convex on int($\mathcal{A}$).

A *proximal setup* for a convex compact set X consists of:

- A norm $\| \cdot \|$ on the Euclidean space E embedding X
- A *distance-generating function (DGF)* $d : \mathcal{A} \to \mathbb{R}$, whose convex domain $\mathcal{A}$ has nonempty interior and is such that $X \subseteq \mathcal{A} \subseteq E$.
  When used as part of a proximal setup, we make the standard requirement that d be a Legendre function that is strongly convex with modulus one with respect to $\| \cdot \|$ on the interior int($\mathcal{A}$) of its domain, specifically,

$$(\nabla d(x) - \nabla d(x'))^\top (x - x') \geq \|x - x'\|^2 \qquad \forall x, x' \in \text{int}(\mathcal{A}).$$

For twice-differentiable d, the strong convexity condition (Condition 3 above) can be equivalently characterized as

$$\boldsymbol{h}^\top \nabla^2 d(\boldsymbol{x}) \boldsymbol{h} \geq \|\boldsymbol{h}\|^2, \ \forall \boldsymbol{x} \in \mathcal{A}, \boldsymbol{h} \in E. \tag{1}$$

Furthermore, we make the common assumption that

$$\min_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}) = 0.$$

This can always be assumed without loss of generality, as d can always be shifted by a constant amount without losing the other properties.

- The *Bregman divergence* $D_d : \mathcal{A} \times \operatorname{int}(\mathcal{A}) \to \mathbb{R}_{\geq 0}$ associated with d yields a notion of distance between points defined as[1]

$$D_d(\boldsymbol{x} \,\|\, \boldsymbol{x}') := d(\boldsymbol{x}) - d(\boldsymbol{x}') - \nabla d(\boldsymbol{x}')^\top (\boldsymbol{x} - \boldsymbol{x}')$$

- The *d-diameter of $\mathcal{X}$* is

$$\Omega_{d,\mathcal{X}} := \max_{\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}} D_d(\boldsymbol{x} \,\|\, \boldsymbol{x}') \leq \max_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}) - \min_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x})$$

- Finally, we denote the largest possible value of the ℓ_1 norm on a $\mathcal{X}$ with the symbol $M_\mathcal{X} := \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_1$.

3.1 "Nice" distance-generating functions

While not a part of the assumptions on the DGF d, it is typically assumed that d allows one to efficiently compute the following two quantities, which come up at every iteration of most FOMS:

- the *gradient* $\nabla d(\boldsymbol{x})$ of d at any point $\boldsymbol{x} \in \operatorname{int}(\mathcal{X})$;
- the gradient of the convex conjugate d^* of d at any point $\boldsymbol{g} \in E$:

$$\nabla d^*(\boldsymbol{g}) = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \left\{ \boldsymbol{g}^\top \boldsymbol{x} - d(\boldsymbol{x}) \right\}.$$

The gradient of the convex conjugate can be intuitively thought of as a linear maximization problem over $\mathcal{X}$ (i.e., the *support function* of $\mathcal{X}$, which is a non-smooth convex optimization problem), *smoothed* by the regularizer d. For that, in this paper we shall refer to $\nabla d^*(\boldsymbol{g})$ either symbolically, or occasionally as the *smoothed support function*.

Because the above two quantities arise so frequently in optimization methods, it is important that the chosen distance-generating function allow for efficient computation of them. In particular, in this paper we are concerned with "nice" DGFs that enable linear-time (in the dimension of E) exact computation of those two quantities.

Definition 3.2. A distance-generating function d is said to be "nice" if $d(\boldsymbol{x})$, $\nabla d(\boldsymbol{x})$ and $\nabla d^*(\boldsymbol{g})$ can be computed exactly in linear time in the dimension of the domain of d.

Hoda et al. [20] also introduce a notion of a "nice" DGF. Their definition is similar to ours, but only states that $\nabla d^*(\boldsymbol{g})$ should be "easily computable". In contrast, we choose a concrete meaning to that statement: we take it to mean linear time in the dimension of the domain.

Finally, we mention a closely related operation that comes up often in optimization methods: the *proximal operator* (or *prox operator*), defined as

$$\begin{aligned}
\operatorname{prox}_{\tilde{\boldsymbol{x}}}(\boldsymbol{g}) &:= \arg\min_{\boldsymbol{x} \in \mathcal{X}} \left\{ \boldsymbol{g}^\top \boldsymbol{x} - D_d(\boldsymbol{x} \,\|\, \tilde{\boldsymbol{x}}) \right\} \\
&= -\nabla d^*(-\boldsymbol{g} + \nabla d(\tilde{\boldsymbol{x}}))
\end{aligned} \tag{2}$$

---

[1]A Bregman divergence need not be symmetric and thus might not be a metric in the technical sense.

for any $\tilde{x} \in \text{int } A$ and $g \in E$. In light of (2), the prox operator can be implemented efficiently provided that ∇d and ∇d^* can. So, prox operators can be computed exactly in linear time in the dimension of E for "nice" DGFs.

3.2 Bilinear saddle-point problems

We will be interested in solving *bilinear saddle-point problems* (BSPPs), whose general form is

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} x^\top A y, \tag{3}$$

where $A \in \mathbb{R}^{n \times m}$ and $\mathcal{X}, \mathcal{Y}$ are convex and compact sets. We will now present the EGT algorithm for solving BSPPs (in the appendix we also present a similar algorithm called mirror prox). EGT depends on two proximal setups: one for $\mathcal{X}$ and one for $\mathcal{Y}$. We will denote the distance-generating functions chosen for $\mathcal{X}$ and $\mathcal{Y}$ as d_x and d_y, respectively. Let $\|\cdot\|_x$ and $\|\cdot\|_y$ be the norms associated with the strong convexity of d_x and d_y in the given proximal setup. The convergence rate of EGT then depends on the following *operator norm* of the payoff matrix A:

$$\|A\| \coloneqq \max\{x^\top A y : \|x\|_x \le 1, \|y\|_y \le 1\}.$$

We will primarily be concerned with DGFs that are strongly convex with respect to either the ℓ_1 or ℓ_2 norms. The magnitude of $\|A\|$ is the primary way in which the norm matters: if both d_x and d_y are strongly convex with respect to the ℓ_2 norm, then $\|A\|$ can be on the order of $\sqrt{nm}$, whereas if both are with respect to the ℓ_1 norm, then $\|A\|$ is simply equal to its largest entry.

3.3 The Excessive Gap Technique (EGT)

The *excessive gap technique (EGT)* is a first-order method introduced by Nesterov [31], and one of the primary applications is to solve BSPPs such as Equation (3). EGT assumes access to a proximal setup for $\mathcal{X}$ and $\mathcal{Y}$, with one-strongly-convex DGFs d_x, d_y for $\mathcal{X}$ and $\mathcal{Y}$, and constructs smoothed approximations of the optimization problems faced by the x and y players. Based on this setup, we formally state the EGT of [30] in Algorithm 1. EGT alternatingly takes steps focused on decreasing one or the other smoothing parameter. These steps are called SHRINKX and SHRINKY in Algorithm 1.

ALGORITHM 1: Excessive Gap Technique (EGT) algorithm.

---

1 **function** INITIALIZE()
2    $t \leftarrow 0$
3    $\mu_x^0 \leftarrow \|A\|, \ \mu_y^0 \leftarrow \|A\|$
4    $\tilde{x} \leftarrow \arg\min_{\hat{x} \in \mathcal{X}} d_x(\hat{x})$
5    $y^0 \leftarrow \nabla d_y^*(A^\top \tilde{x}/\mu_y^0)$
6    $x^0 \leftarrow \text{prox}_{\tilde{x}}(Ay^0/\mu_x^0)$

7 **function** ITERATE()
8    $t \leftarrow t + 1, \ \tau \leftarrow 2/(t+2)$
9   **if** t *is even* **then** SHRINKX()
10   **else** SHRINKY()

11 **function** SHRINKX()
12    $\bar{x} \leftarrow -\nabla d_x^*(-Ay^{t-1}/\mu_x^{t-1})$
13    $\hat{x} \leftarrow (1-\tau)x^{t-1} + \tau\bar{x}$
14    $\bar{y} \leftarrow \nabla d_y^*(A^\top \hat{x}/\mu_y^{t-1})$
15    $\tilde{x} \leftarrow \text{prox}_{\bar{x}}\left(\frac{\tau}{(1-\tau)\mu_x^{t-1}} A\bar{y}\right)$
16    $x^t \leftarrow (1-\tau)x^{t-1} + \tau\tilde{x}$
17    $y^t \leftarrow (1-\tau)y^{t-1} + \tau\bar{y}$
18    $\mu_x^t \leftarrow (1-\tau)\mu_x^{t-1}$

19 **function** SHRINKY()
20    $\bar{y} \leftarrow \nabla d_y^*(A^\top x^{t-1}/\mu_y^{t-1})$
21    $\hat{y} \leftarrow (1-\tau)y^{t-1} + \tau\bar{y}$
22    $\bar{x} \leftarrow -\nabla d_x^*(-A\hat{y}/\mu_x^{t-1})$
23    $\tilde{y} \leftarrow \text{prox}_{\bar{x}}\left(\frac{-\tau}{(1-\tau)\mu_y^{t-1}} A^\top \bar{x}\right)$
24    $y^t \leftarrow (1-\tau)y^{t-1} + \tau\tilde{y}$
25    $x^t \leftarrow (1-\tau)x^{t-1} + \tau\tilde{x}$
26    $\mu_y^t \leftarrow (1-\tau)\mu_y^{t-1}$

Algorithm 1 shows how initial points are selected and the alternating steps and stepsizes are computed. Nesterov [30] proves that the EGT algorithm converges at a rate of $O(1/T)$:

THEOREM 3.3 (NESTEROV [30] THEOREM 6.3). *At every iteration $t \ge 1$ of the EGT algorithm, the solution (x^t, y^t) satisfies $x^t \in \mathcal{X}, y^t \in \mathcal{Y}$, and*

$$\max_{y \in \mathcal{Y}} (x^t)^\top A y - \min_{x \in \mathcal{X}} x^\top A y^t \le \frac{4\|A\|\sqrt{\Omega_{d_x, \mathcal{X}} \Omega_{d_y, \mathcal{Y}}}}{t+1}.$$

4 THE SEQUENCE-FORM POLYTOPE

4.1 Preliminaries on Sequence-Form Polytopes and Nash Equilibria

We now describe how the set of Nash equilibria of a two-player zero-sum EFG can be represented as a bilinear saddle-point problem. The sequential nature of the decision spaces is represented via the *sequence form*, where each strategy space $\mathcal{X}$ and $\mathcal{Y}$ has the form of a convex polytope.

The sequence-form polytope for a given player is as follows: We assume that we have a set of decision points $\mathcal{J}$, and each decision point $j \in \mathcal{J}$ has a set of actions A_j, with $|A_j| = n_j$ actions in total. If the agent takes a given action $a \in A_j$ at decision point j, then $C_{ja} \subset \mathcal{J}$ denotes the set of next potential decision points that the agent may face (which may be empty if no more decisions can occur after taking action a at j). We assume that the decision points form a tree, meaning that $C_{ja} \cap C_{j'a'} = \emptyset$ for any two pairs ja and $j'a'$ such that $j \neq j'$ or $a \neq a'$. This is equivalent to assuming that the corresponding EFG has *perfect recall*, meaning that no agent ever forgets any past information.

For an EFG, the decision points $\mathcal{J}$ for a given player correspond to the set of information sets in the game belonging to that player, and a pair ja consisting of a decision point j and action a is referred to as a *sequence*. Let Σ be the set of all sequences.

For a two-player zero-sum EFG with perfect recall, the problem of computing a Nash equilibrium can be cast as a BSPP in the form of Equation (3). In this formulation, $\mathcal{X}$ and $\mathcal{Y}$ are the sequence-form polytopes for each player. The *payoff matrix* A is such that for a pair of sequence-form strategies x, y, the objective $x^\top A y$ is equal to the expected value achieved by the second player under those strategies. Thus, the second player wishes to maximize this objective, while the first player wishes to minimize it. Each cell in $A \in \mathbb{R}^{|\Sigma_1| \times |\Sigma_2|}$ corresponds to a pair of sequences, one for each player. The matrix is often sparse: each non-zero entry corresponds to a pair of sequences such that they are the last sequences on the path to some leaf node (and thus we have zeroes for all cells such that the corresponding sequences are never the last pair of sequences before the game ends). The value at a cell is the payoff to the second player at that leaf, times the product of all chance probabilities on the path to the leaf.

4.2 Preliminaries on Dilated Distance-Generating Functions

Dilated distance-generating functions are a general framework for constructing "nice" DGFs (in the sense of Section 3.1) for sequence-form polytopes [20]. Specifically, a dilated DGF for a sequence-form polytope is constructed by taking a weighted sum over suitable *local* regularizers $d_\emptyset$ and d_j ($j \in \mathcal{J}$), and is of the form

$$d : x \mapsto \alpha_\emptyset d_\emptyset(x_\emptyset) + \sum_{j \in \mathcal{J}} \alpha_j x_{p_j} d_j \left(\frac{(x_{ja})_{a \in A_j}}{x_{p_j}} \right) \tag{4}$$

Each local function d_j takes as input $(x_{ja})_{a \in A_j}/x_{p_j} \in \Delta^{n_j}$, and is assumed to be continuous, strongly convex modulus one on the probability simplex Δ^{n_j}, and differentiable in the relative interior of Δ^{n_j}. By dividing $(x_{ja})_{a \in A_j}$ by x_{p_j}, we renormalize $(x_{ja})_{a \in A_j}$ to the simplex, measure the DGF there, and then scale that value by x_{p_j} to make it proportional to the "size" of $x_{p_j} \cdot \Delta^{n_j}$. Finally, the weight α_j is a flexible weight term that can be chosen to ensure good properties. [20] showed that if each local DGF d_j is strongly convex, then the dilated DGF d is also strongly convex (although they do not give an explicit modulus), and they show that the associated smoothed support function can easily be computed, provided that the smoothed support function for each d_j can easily be computed.

The gradient of a dilated DGF and of its convex conjugate can be computed exactly in closed form by combining the gradients of each d_j and their convex conjugates, as shown in Algorithm 2.

ALGORITHM 2: Gradient and smoothed support function implementation for general dilated DGFs.

1 **function** GRADIENT($x \in \text{int } Q$)

2     $g \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$

3   **for** $j \in \mathcal{J}$ *in bottom-up order* **do**

4         $(g_{ja})_{a \in A_j} \leftarrow (g_{ja})_{a \in A_j} + \alpha_j \nabla d_j\!\left(\dfrac{(x_{ja})_{a \in A_j}}{x_{p_j}}\right)$

5         $g_{p_j} \leftarrow g_{p_j} + \alpha_j \, d_j\!\left(\dfrac{(x_{ja})_{a \in A_j}}{x_{p_j}}\right)$

6         $g_{p_j} \leftarrow g_{p_j} - \alpha_j \nabla d_j\!\left(\dfrac{(x_{ja})_{a \in A_j}}{x_{p_j}}\right)^{\!\top}\!\!\left(\dfrac{(x_{ja})_{a \in A_j}}{x_{p_j}}\right)$

7     $g_\varnothing \leftarrow g_\varnothing + \alpha_\varnothing \nabla d_\varnothing(x_\varnothing)$

8   **return** g

1 **function** CONJUGATEGRADIENT($g \in \mathbb{R}^{|\Sigma|}$)

2     $z \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$

3     $z_\varnothing \leftarrow 1$

4   **for** $j \in \mathcal{J}$ *in bottom-up order* **do**

5         $(z_{ja})_{a \in A_j} \leftarrow \nabla d_j^*((g_{ja})_{a \in A_j})$

6         $g_{p_j} \leftarrow g_{p_j} - d_j((z_{ja})_{a \in A_j}) + \sum_{a \in A_j} g_{ja} z_{ja}$

7   **for** $j \in \mathcal{J}$ *in top-down order* **do**

8   **for** $a \in A_j$ **do**

9             $z_{ja} \leftarrow z_{p_j} \cdot z_{ja}$

10   **return** z

The local DGFs must be chosen so that they are compatible with the simplex. For a given simplex Δ^k, these are usually chosen either as the *entropy DGF* $d(\boldsymbol{y}) = \log k + \sum_i y_i \log y_i$ or *Euclidean DGF* $d(\boldsymbol{y}) = \frac{1}{2} \sum_i (y_i - 1/k)^2$. These are both 1-strongly convex on Δ^k (for entropy wrt. the ℓ_1 norm and for Euclidean wrt. the ℓ_2 norm), and their associated smoothed support functions can be computed in $O(k)$ and $O(k \log k)$ time, respectively.

One of the most important properties of dilated DGFs is that they lead to a "nice" DGF as long as each local convex conjugate gradient ∇d_j^* can be computed in time linear in $|A_j|$. In particular, this makes the dilated entropy DGF a "nice" DGF. The dilated Euclidean DGF is "almost" "nice", in the sense that ∇d_j^* can be computed in $|A_j| \log |A_j|$ time, and $|A_j|$ is typically very small. In contrast, the standard Euclidean DGF applied to the overall polytope Q is not "nice": It requires $|\Sigma| \log |\Sigma|$ time to resolve its convex conjugate gradient.

For the dilated DGFs, the strongest general result on the strong-convexity modulus comes from Farina et al. [13], where the authors show that if each local DGF d_j is strongly convex modulus one with respect to the ℓ_2 norm, and we set $\alpha_j = 2 + 2 \max_{a \in A_j} \sum_{j' \in C_{ja}} \alpha_{j'}$ for each $j \in \mathcal{J}$, then d is strongly convex modulus one with respect to the ℓ_2 norm on Q.

4.3 Preliminaries on the Dilated Entropy Distance-Generating Function

The dilated *entropy* DGF is the instantiation of the general dilated DGF framework of Section 4.2 with the particular choice of using the (negative) entropy function at each decision node. In particular, it is any regularizer of the form[2]

$$[0,1]^{|\Sigma|} \ni \boldsymbol{x} \mapsto \alpha_\varnothing x_\varnothing \log x_\varnothing + \sum_{j \in \mathcal{J}} \alpha_j \left(x_{p_j} \log |A_j| + \sum_{a \in A_j} x_{ja} \log\!\left(\frac{x_{ja}}{x_{p_j}}\right) \right). \qquad (5)$$

for some choice of the weights $\alpha_\varnothing, \alpha_j > 0$.

We will now briefly review existing results specific to the dilated entropy DGF, for which stronger results are known than for the general class of dilated DGFs. A central result in the present paper is to show that there exist DGFs for sequence-form polytopes which are better than the dilated entropy DGF, but that these DGFs can be partially recast in a dilated form, in order to enable efficient computation of the smoothed support function.

First, as a direct consequence of the more general discussion in Section 4.2 and Algorithm 2, the dilated entropy DGF is a "nice" DGF (in the precise sense of Section 3.1) no matter the choice of weights

---

[2]In this paper, we let $0 \log(0) = 0$. Since d is a Legendre function, it is guaranteed that all iterates and prox-steps will remain the interior of the optimization domain at all times, thus avoiding the non-differentiability issue of the entropy function at the boundary of $[0,1]^{|\Sigma|}$.

α. In particular, in the case of the negative entropy functions $d_j(\boldsymbol{x}) = \log|A_j| + \sum_{a \in A_j} x_{ja} \log x_{ja}$, one has

$$\left(\nabla d_j(\boldsymbol{x})\right)_a = 1 + \log x_a, \quad \left(\nabla d_j^*(\boldsymbol{g})\right)_a = \frac{e^{g_a}}{\sum_{a' \in A_j} e^{g_{a'}}} \qquad \forall a \in A_j, \boldsymbol{x} \in \mathrm{int}(\Delta^{|A_j|}), \boldsymbol{g} \in \mathbb{R}^{|A_j|}.$$

By plugging the above expression in the template of Algorithm 2 we obtain linear-time exact algorithms to compute $\nabla\varphi$ and $\nabla\varphi^*$.

Kroer et al. [23] show that the dilated entropy DGF is Legendre and strongly convex modulus $1/\max_{x \in Q}\|x\|_1$ with respect to the ℓ_1 norm, when the weights α are chosen as in the following definition.

Definition 4.1 (Kroer et al. dilated entropy DGF, ψ). Define the DGF weights β_j recursively as

$$\beta_\varnothing := 2 + 2\sum_{j \in C_\varnothing} \beta_j, \qquad \beta_j := 2 + 2\max_{a \in A_j} \sum_{j' \in C_{ja}} \beta_{j'} \qquad \forall j \in \mathcal{J}.$$

The resulting instantiation of the dilated entropy DGF is called the *Kroer et al. dilated entropy DGF* and denoted ψ:

$$\psi : [0,1]^{|\Sigma|} \ni \boldsymbol{x} \mapsto \beta_\varnothing x_\varnothing \log x_\varnothing + \sum_{j \in \mathcal{J}} \beta_j \left(x_{p_j} \log|A_j| + \sum_{a \in A_j} x_{ja} \log\left(\frac{x_{ja}}{x_{p_j}}\right) \right).$$

Remark 1. *On the surface, the strong convexity modulus of $\frac{1}{M_Q}$ with respect to the ℓ_1 norm might appear less appealing than the modulus 1 obtained by using the ℓ_2 norm. However, recall that the norm that is used to measure strong convexity affects the value of the operator norm of $\boldsymbol{A}$, which is significantly smaller under the $\ell_1 - \ell_\infty$ operator norm (where it is equal to $\max_{ij}|A_{ij}|$) than the $\ell_2 - \ell_2$ operator norm for strong convexity with respect to the ℓ_2 norm.*

One drawback of both the general and entropy-specific dilated DGFs developed in the past is that they have an exponential dependence on the depth of the sequence-form polytope. In particular, note that the factor of 2 in the recursive definition of the weights means that the factor β_j for some root decision point is growing at least on the order of $2^{\mathfrak{D}_Q}$, where $\mathfrak{D}_Q$ is the depth of the sequence-form polytope Q. For many sequence-form polytopes this might be acceptable: if the sequence-form polytope is reasonably balanced, then the number of decision points is also exponential in depth. However, for other sequence-form polytopes this would be unacceptable: the most extreme case would be a single line of decision points, where the number of decision points is linear in $\mathfrak{D}_Q$, but the β_j at the root is exponentially large. This exponential dependence on depth also enters the convergence rate of the FOMs, since it effectively acts as a scalar on the polytope diameter Ω induced by $\mathfrak{D}_Q$. This paper was motivated by the need to soundly resolve that drawback, thus introducing the first "nice" DGF (in the sense of Section 3.1) with guaranteed polynomially-small diameter for any decision point.

5 THE DILATABLE GLOBAL ENTROPY DISTANCE-GENERATING FUNCTION

We now develop our new DGF for sequence-form polytopes. The DGF will be based on a scaled variant of the standard entropy DGF (which we will also refer to as the "global entropy" to distinguish it from the dilated entropy), which is strongly convex modulus one on the hypercube $[0,1]^{|\Sigma|}$. We start by giving a new set of weights for the dilated entropy DGF, and we will then later show that this particular choice of weights can be directly related to the global entropy DGF over the sequence-form polytope, when the global entropy DGF is scaled in a particular way.

First we define the weighting scheme that we will use for the dilated representation of our DGF. Consider the scalars γ_j ($j \in \mathcal{J}$) defined recursively as

$$\gamma_\varnothing = 1 + \sum_{j \in C_\varnothing} \gamma_j, \qquad \gamma_j := 1 + \max_{a \in A_j} \left\{ \sum_{j' \in C_{ja}} \gamma_{j'} \right\} \qquad \forall j \in \mathcal{J}. \qquad (6)$$

These weights are very similar to the ones given for the dilated DGFs in the previous section, except that the whole expression is smaller by a factor of two. Avoiding this factor of two is crucial, because it allows us to avoid the exponential dependence on depth. Here, it is easy to see that γ_j is upper bounded by the number of decision points in the subtree rooted at j, so γ_j is at most polynomial in the size of the sequential decision problem. In fact, it is not hard to show that if j is the sole root decision point, then γ_j is equal to $\max_{x \in Q} \|x\|_1$.

With the above weights, we now instantiate a dilated entropy DGF, which we denote as φ:

$$\varphi : [0,1]^{|\Sigma|} \ni x \mapsto \gamma_\varnothing x_\varnothing \log x_\varnothing + \sum_{j \in \mathcal{J}} \gamma_j \left(x_{p_j} \log |A_j| + \sum_{a \in A_j} x_{ja} \log \left(\frac{x_{ja}}{x_{p_j}} \right) \right).$$

Next, we define the global entropy function that we will be using, which also depends on γ_j.

Definition 5.1 (Dilatable global entropy). The *dilatable global entropy distance generating function* $\tilde{\varphi}$ is the function $\tilde{\varphi} : [0,1]^{|\Sigma|} \to \mathbb{R}_{\geq 0}$ defined as

$$\tilde{\varphi} : x \mapsto w_\varnothing x_\varnothing \log(x_\varnothing) + \sum_{j \in \mathcal{J}} \sum_{a \in A_j} w_{ja} x_{ja} \log x_{ja} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j|,$$

where each w_σ ($\sigma \in \Sigma$) is defined as

$$w_\varnothing := \gamma_\varnothing - \sum_{j \in C_\varnothing} \gamma_j,$$

$$w_{ja} := \gamma_j - \sum_{j' \in C_{ja}} \gamma_{j'} = 1 + \max_{a' \in A_j} \left\{ \sum_{j' \in C_{ja'}} \gamma_{j'} \right\} - \sum_{j' \in C_{ja}} \gamma_{j'} \qquad \forall ja \in \Sigma,$$

and is a scalar lower bounded by 1.

Dilatability. The adjective *dilatable* comes from the key property that the dilatable global entropy is equal to the dilated entropy φ, *on the sequence-from strategy space Q*. That equality does not hold outside the sequence-form polytope, and this means that the gradient of $\tilde{\varphi}(x)$, which is used in the FOMs that we consider, may differ as well.

THEOREM 5.2. *The dilatable global entropy DGF and the dilated entropy DGF coincide on the polytope of sequence-form strategies Q, that is, $\tilde{\varphi}(x) = \varphi(x)$ for all $x \in Q$.*

PROOF. We start by expanding the definition of $\varphi(x)$:

$$\varphi(x) := \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log \left(\frac{x_{ja}}{x_{p_j}} \right) + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j|$$

$$= \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log x_{ja} - \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log x_{p_j} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j|. \qquad (7)$$

Given the assumption $x \in Q$, it holds that $\sum_{a \in A_j} x_{ja} = x_{p_j}$ for all $j \in \mathcal{J}$ and so we can simplify the middle summation in (7) and obtain

$$
\begin{aligned}
\varphi(x) &= \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log x_{ja} - \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log x_{p_j} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j| \\
&= \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log x_{ja} - \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \sum_{j' \in C_{ja}} \gamma_{j'} x_{ja} \log x_{ja} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j| \\
&= \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \gamma_j x_{ja} \log x_{ja} - \sum_{j \in \mathcal{J}} \sum_{a \in A_j} \left(\sum_{j' \in C_{ja}} \gamma_{j'} \right) x_{ja} \log x_{ja} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j| \\
&= \sum_{j \in \mathcal{J}} \sum_{a \in A_j} w_{ja} x_{ja} \log x_{ja} + \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j| = \tilde{\varphi}(x),
\end{aligned}
$$

as we wanted to show. □

"Nice"ness. We now show that our dilatable global entropy regularizer is "nice" in the sense of Section 3.1, that is, its gradient and the gradient of its convex conjugate can be computed exactly in linear time in $|\Sigma|$.

- The gradient of $\tilde{\varphi}$ can be trivially computed in closed form and linear time in $|\Sigma|$ starting from Definition 5.1 as

$$
(\nabla \tilde{\varphi}(x))_\sigma = (1 + \log x_\sigma) w_\sigma + \sum_{j \in C_\sigma} \gamma_j \log |A_j| \qquad \forall \sigma \in \Sigma, x \in \text{int}(Q).
$$

- Using the dilatability property, we have that the gradient of the convex conjugate satisfies

$$
\nabla \tilde{\varphi}^*(g) = \arg\max_{x \in Q} \{ g^\top x - \tilde{\varphi}(x) \} = \arg\max_{x \in Q} \{ g^\top x - \varphi(x) \} = \nabla \varphi^*(g),
$$

where we used the dilatability property (Theorem 5.2) in the second equality. Therefore, since φ is a dilated DGF and its smoothed support function can be computed in linear time, the smoothed support function of $\tilde{\varphi}$ can be computed in linear time in $|\Sigma|$. Similarly, $\text{prox}_c(g) = -\nabla \varphi^*(-g + \nabla \tilde{\varphi}(c))$, where the internal gradient is with respect to $\tilde{\varphi}$, as opposed to φ as in the general reduction in Equation (2).

Strong convexity. On the other hand, we now show that $\tilde{\varphi}$ has the advantage of a better strong convex modulus on $(0, 1)^{|\Sigma|}$, compared to the class of dilated entropy DGFs.

THEOREM 5.3. *The dilatable global entropy function $\tilde{\varphi} : [0, 1]^{|\Sigma|} \to \mathbb{R}_{\geq 0}$ is a Legendre function, and 1-strongly convex with respect to the ℓ_2 norm on $(0, 1)^{|\Sigma|}$.*

THEOREM 5.4. *The dilatable global entropy function $\tilde{\varphi}$ is strongly convex modulus $1/M_Q$ with respect to the ℓ_1 norm on the sequence-form polytope Q.*

Diameter. The properties above immediately imply that our dilatable global entropy DGF satisfies all the requirements for a prox setup on the polytope of sequence-form strategies Q. Here we complete the analysis by giving bounds on the diameter induced by $\tilde{\varphi}$.

THEOREM 5.5. *The $\tilde{\varphi}$-diameter $\Omega_{\tilde{\varphi}, Q}$ of Q is at most $M_Q^2 \max_{j' \in \mathcal{J}} \log |A_{j'}|$.*

Kroer et al. [23] show that the dilated entropy DGF with weights β leads to a polytope diameter

$$
2^{\mathfrak{D}_Q + 2} M_Q^2 \max_{j' \in \mathcal{J}} \log |A_{j'}|.
$$

Our DGF improves that polytope diameter by a factor of $2^{\mathfrak{D}_Q+2}$. Thus, we are the first to achieve a polytope diameter with no exponential dependence on the depth $\mathfrak{D}_Q$ of the sequence-form polytope.

Summing up our results on the dilatable global entropy, we have shown that it enjoys the same fast smoothed support function computation as the dilated entropy DGF while having a better way to achieve strong convexity modulus $1/M_Q$. In particular, the existing dilated entropy setup requires the weight parameters β to grow exponentially in the depth of the sequence-form polytope, whereas we have only a linear growth in those weights. More concretely, this means that the largest weights $\max_{j \in \mathcal{J}} \beta_j$ in the dilated entropy DGF are larger than the largest weights $\max_{j \in \mathcal{J}} \gamma_j$ in the dilatable global entropy DGF by a factor of more than $2^{\mathfrak{D}_Q}$. This in turn allowed us to achieve a better polytope diameter by a factor of $2^{\mathfrak{D}_Q+2}$ while retaining the same strong convexity modulus.

6 SCALED EXTENSION AND CORRELATED DECISION SPACES

In this section we extend and generalize both the framework of dilated DGFs and the dilatable global entropy DGF to more complex combinatorial domains than sequence-form polytopes. In particular, we show that dilated DGFs and the dilatable global entropy apply to sets that can be constructed through composition of *scaled extension*, a convexity-preserving operation that was recently proposed as a general way of constructing sequential decision spaces in the presence of correlation between the strategies of two or more players [17].

Our generalization begets the first "nice" regularizers (in the sense of Section 3.1) for correlated strategy spaces, which in turn enables us to construct the first FOMs that guarantee convergence to optimal correlated equilibria and optimal ex-ante team coordinated equilibria with rate $1/T$ in certain classes of games where these equilibria can be found in polynomial time.

We start by recalling the definition of scaled extension.

Definition 6.1 (Scaled extension [17]). Let $\mathcal{U}$ and $\mathcal{V}$ be nonempty compact convex sets, and let $h : \mathcal{U} \to \mathbb{R}_{\geq 0}$ be a nonnegative affine real function. The *scaled extension* of $\mathcal{U}$ with $\mathcal{V}$ via h is defined as the convex and compact set $\mathcal{U} \overset{h}{\triangleleft} \mathcal{V} := \{(\boldsymbol{u}, \boldsymbol{v}) : \boldsymbol{u} \in \mathcal{U}, \ \boldsymbol{v} \in h(\boldsymbol{u})\mathcal{V}\}$.

To simplify the proofs, in this paper we only consider scaled extensions that use *linear* functions h, instead of more general affine functions.

6.1 Preliminaries on Correlation and Triangle-Freeness

As reviewed earlier in the paper, Nash equilibria in two-player zero-sum EFGs can be expressed as BSPPs. It turns out that several other solution concepts can also be formulated as BSPPs via more intricate convex-polytope constructions. In this section we briefly describe two important solution concepts that can be expressed as BSPPs: several variants of optimal correlated equilibria and ex-ante team coordinated equilibria.

In correlated equilibria, the rationality assumption of Nash equilibrium is relaxed in order to allow for coordination between the player. It is assumed that a *mediator* will recommend actions to be taken. In *correlated equilibria*, each player sees the recommended action before deciding whether to take it. In *coarse correlated equilibria*, the players must commit to acting according to the recommended strategy before the recommendation is revealed. In all these solution concepts, the recommended strategy is sampled by a mediator, from some correlated distribution which is known to the players.

We will consider three types of correlated equilibria in two-player EFGs.

(1) **Extensive-form correlated equilibrium (EFCE)**: The mediator incrementally recommends individual moves to the players. Every time a player faces a decision point, the mediator privately reveals a recommended move for that decision point to that player. If a player chooses to

disregard a recommendation, then the mediator immediately stops issuing recommendations to that player forever [37].

(2) **Extensive-form coarse correlated equilibrium (EFCCE)**: The mediator incrementally recommends individual moves to the players, but at each decision point, the player must decide whether to follow the recommendation *before* seeing the recommendation [18].

(3) **Normal-form coarse correlated equilibrium (NFCCE)**: each player will be recommended a strategy from the normal-form representation of the EFG, but they must decide whether to commit to playing the recommended strategy before seeing the recommendation [28].

Farina and Sandholm [18] show that EFCE are a subset of EFCCE, and that EFCCE are a subset of NFCCE. They also show that for triangle-free decision problems (or EFGs),[3] the set Ξ of all possible correlated plans between two players can be represented via recursive applications of the scaled extension operator. Thus, we can write BSPPs of the form

$$\arg\min_{\boldsymbol{x}\in\Xi}\max_{\boldsymbol{y}\in\mathcal{Y}}\boldsymbol{x}^{\top}\mathcal{A}\boldsymbol{y}, \tag{8}$$

where the minimization over $\boldsymbol{x}$ represents the choice of a correlated plan for the two players, and the maximization over $\mathcal{Y}$, intuitively, represents different ways of rejecting the mediator's recommendation. By carefully choosing $\mathcal{Y}$, we can enforce different types of correlated equilibrium behavior. Now, as long as we have a "nice" DGF for scaled extensions and a "nice" DGF for the polytope $\mathcal{Y}$, we can apply fast FOMs to the computation of the corresponding type of correlated equilibrium. Appropriate polytopes for $\mathcal{Y}$ exist in the case of EFCE [16], EFCCE [10], and NFCCE [10]. In each case, $\mathcal{Y}$ is itself a polytope that can be constructed via scaled extension (though simpler than Ξ).

Reviewing the details of the scaled-extension-based construction of the polytope of correlation plans Ξ is beyond the scope of this paper. Here, we take the decomposition as a given, and we are concerned with the task of constructing a suitable proximal setup for sets that, like $\mathcal{Y}$ and Ξ, can be expressed through composition of scaled extension operations.

In addition to two-player correlated equilibrium problems discussed above, one can also capture *adversarial team games* with the scaled extension DGFs that we will construct. In the adversarial team game that we consider, two players on a team (meaning that they share the same payoffs) are trying to correlate their strategies so as to maximize utility against an opponent whose utility is exactly the opposite of theirs (i.e., it is a zero-sum game between the team and the opponent). This solution concept is called *team-maxmin equilibrium with coordination (TMECor)* [8]. The set of correlated plans for the two players on the team can again be expressed with Ξ if their two decision spaces are triangle free [11]. Then, the strategies of the opponent are simply a sequence-form polytope Q, so $\mathcal{Y} = Q$. It follows that DGFs for both polytopes can be chosen as the DGE from our paper.

For EFCE, EFCCE, NFCCE, and TMECor, it will follow from our results below that it is possible to construct a "nice" DGF for the polytope of correlation plans when the game is triangle free. By applying this DGF to a method such as EGT or mirror prox, we get the first $1/T$ iterative method for converging to each of these solution concepts, with only a linear cost per iteration. In contrast, prior iterative approaches converge at a rate of $1/\sqrt{T}$, and sometimes still require significantly more expensive projections at every iteration (e.g., Farina et al. [16, 17]).

---

[3]The triangle-free condition is rather technical, and so we omit its exact definition here as it is beyond the scope of the paper. The most natural class of games that it captures is the set of EFGs where all chance moves are public, that is, observed by all players.

6.2 Dilated Distance-Generating Functions for Scaled Extension

In this section we show that the construction of dilated DGFs can be generalized to sets obtained through scaled extension. Let $\mathcal{Z} := \mathcal{U} \overset{h}{\triangleleft} \mathcal{V}$ be a set constructed by scaled extension of $\mathcal{U}$ with $\mathcal{V}$, and assume that "nice" DGFs d_u, d_v for $\mathcal{U}$ and $\mathcal{V}$ respectively have been chosen. In Proposition 1 we show that d_u and d_v can be combined to give a composite "nice" DGF for $\mathcal{Z}$.

PROPOSITION 1. *Let*

- $\mathcal{Z} := \mathcal{U} \overset{h}{\triangleleft} \mathcal{V}$, *where h is the linear function $h : \boldsymbol{u} \mapsto \boldsymbol{a}^\top \boldsymbol{u} \geq 0$;*
- $d_u : \mathcal{A} \to \mathbb{R}$ *and $d_v : \mathcal{B} \to \mathbb{R}$, for some domains $\mathcal{A} \supseteq \mathcal{U}, \mathcal{B} \supseteq \mathcal{V}$ whose interiors are nonempty and such that $\boldsymbol{v}/h(\boldsymbol{u}) \in \mathcal{B}$ for all $\boldsymbol{u} \in \mathcal{A}, \boldsymbol{v} \in \mathcal{B}$, be 1-strongly-convex Legendre functions;*
- $\alpha_v > 0$ *be a positive scalar.*

Then, the composite DGF

$$d_z : \mathcal{A} \times \mathcal{B} \ni (\boldsymbol{u}, \boldsymbol{v}) \mapsto d_u(\boldsymbol{u}) + \alpha_v \, h(\boldsymbol{u}) d_v\left(\frac{\boldsymbol{v}}{h(\boldsymbol{u})}\right)$$

is a Legendre function. Furthermore, if d_u and d_v are "nice" DGFs, so is d_z.

In principle, the construction of Section 4.2 could be repeated inductively on the scaled-extension-based decomposition of the polytope of correlation plans Ξ, giving rise to a "nice" DGF for that polytope. However, before the DGF can be effectively used as part of a prox setup, its strong convexity properties need to be assessed. Unfortunately, just like in sequence-form strategy spaces, dilated DGFs constructed in this way might require weights α that grow exponentially fast with the dimension of the set.

PROPOSITION 2. *Consider the same setup as Proposition 1, and assume that:*

- $\max_{\boldsymbol{v} \in \mathcal{V}} \|\boldsymbol{v}\| \leq 1$;
- $\max_{\boldsymbol{u} \in \mathcal{U}} h(\boldsymbol{u}) \leq 1$;
- d_v *is 1-strongly convex in the interior of $\mathcal{V}$;*
- d_u *satisfies the property*

$$(\nabla d_u(\boldsymbol{u}) - \nabla d_u(\boldsymbol{u}'))^\top (\boldsymbol{u} - \boldsymbol{u}') \geq \frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}')^\top D_u(\boldsymbol{u} - \boldsymbol{u}') \qquad \forall \boldsymbol{u}, \boldsymbol{u}' \in \mathcal{U}$$

for some real matrix D_u.

Then, the composite DGF d_z satisfies

$$(\nabla d_z(\boldsymbol{u}, \boldsymbol{v}) - \nabla d_z(\boldsymbol{u}', \boldsymbol{v}'))^\top \begin{pmatrix} \boldsymbol{u} - \boldsymbol{u}' \\ \boldsymbol{v} - \boldsymbol{v}' \end{pmatrix} \geq \frac{1}{2}\begin{pmatrix} \boldsymbol{u} - \boldsymbol{u}' \\ \boldsymbol{v} - \boldsymbol{v}' \end{pmatrix}^\top \left(\begin{array}{c|c} D_u - 2\alpha_v\|\boldsymbol{a}\|_2^2 I & 0 \\ \hline 0 & \alpha_v I \end{array}\right)\begin{pmatrix} \boldsymbol{u} - \boldsymbol{u}' \\ \boldsymbol{v} - \boldsymbol{v}' \end{pmatrix}$$

REMARK 2. *The minimum eigenvalue of the matrix D_z defined in Proposition 2 determines the strong convexity modulus (with respect to the Euclidean norm) of d_z. Note that we can use Proposition 2 inductively to pick weights α in the dilated DGF that guarantee that it is μ-strongly convex for any μ. Indeed, consider a composition of scaled extensions $X_1 \overset{h_1}{\triangleleft} X_2 \overset{h_2}{\triangleleft} \cdots \overset{h_{k-1}}{\triangleleft} X_k$, where we assume that each h_i is of the form $h_i : \boldsymbol{x} \mapsto \boldsymbol{a}_i^\top \boldsymbol{x}$ for all $i = 1, \ldots, k$. Denote by d_i, for $i = 1, \ldots, k$ the dilated DGF obtained by composing the construction of Proposition 2 for sets $X_1, \ldots, X_i$, and let D_i be the corresponding matrix obtained by applying Proposition 2 recursively. Since from Proposition 2 we have that*

$$D_k = \left(\begin{array}{c|c} D_{k-1} - 2\alpha_k\|\boldsymbol{a}_k\|_2^2 I & 0 \\ \hline 0 & \alpha_k I \end{array}\right),$$

in order to have that the final DGF d_k be 1-strongly-convex, it is necessary that $\alpha_k \geq \mu_k$, or the diagonal elements in the bottom-right matrix block would be lower than the desired μ_k bound. But then d_{k-1} must be $(1 + 2\|\boldsymbol{a}_k\|_2^2)$-strongly convex to compensate for the subtraction in the top-right block of the matrix. Recursively, this implies that $\alpha_{k-1} \geq 2(1 + 2\|\boldsymbol{a}_k\|_2^2)$, and to compensate d_{k-1} must be $(1 + 2(1 + 2\|\boldsymbol{a}_k\|_2^2 + \|\boldsymbol{a}_{k-1}\|_2^2))$-strongly convex, and so on. In summary, the weights required to guarantee strong convexity need to grow exponentially fast, rendering the construction prone to the same criticisms that apply to prior dilated DGFs for the sequence-form polytope. In the next section we show that in some cases of interest this issue can be soundly circumvented, by generalizing some of the ideas we presented in Section 5.

6.3 Dilatable Global Entropy for Scaled Extension

In this section we instantiate the generic framework of dilated DGFs, as defined in the previous section, to the chains of scaled extensions with *simplex domains*, that is, sets of the form $\Delta^{s_1} \overset{h_1}{\triangleleft} \Delta^{s_2} \overset{h_2}{\triangleleft} \Delta^{s_3} \overset{h_3}{\triangleleft} \cdots \overset{h_n}{\triangleleft} \Delta^{s_{n+1}}$, where we assume that $h_k(\boldsymbol{x}) = \boldsymbol{a}_k^\top \boldsymbol{x}$ with $\boldsymbol{a}_k \in [0,1]^{s_1 + \cdots + s_{k-1}}$ for all $k = 1, \ldots, n$. This setup encompasses both sequence-form strategy spaces and the polytope of correlation plans.

The *dilated entropy DGF* for one such set is the dilated DGF obtained by recursively applying the general construction of Proposition 1 with the (negative) entropy function at each Δ^{s_i}. Specifically, if ψ_k is the dilated entropy DGF constructed for the first k scaled extensions, we define ψ_{k+1} as

$$\psi_{k+1} : [0,1]^{s_1 + \cdots s_k} \times [0,1]^{s_{k+1}} \ni (\boldsymbol{u}, \boldsymbol{v}) \mapsto \psi_k(\boldsymbol{u}) + \alpha_{k+1}\left(h_k(\boldsymbol{u}) \log s_{k+1} + \sum_{i=1}^{s_{k+1}} v_i \log\left(\frac{v_i}{h_k(\boldsymbol{u})}\right) \right).$$

By using the same manipulations of the logarithms that we used in Theorem 5.2, it is immediate to show by induction that ψ_{k+1} coincides, on $\Delta^{s_1} \overset{h_1}{\triangleleft} \Delta^{s_2} \overset{h_2}{\triangleleft} \Delta \overset{h_3}{\triangleleft} \cdots \overset{h_k}{\triangleleft} \Delta^{s_{k+1}}$ and for all $k = 1, 2, \ldots$, with the function

$$\tilde{\varphi}_{k+1} : [0,1]^{s_1 + \cdots + s_k} \times [0,1]^{s_{k+1}} \ni (\boldsymbol{u}, \boldsymbol{v}) \mapsto \tilde{\varphi}_k(\boldsymbol{u}) + \alpha_{k+1}\left(h_k(\boldsymbol{u})(1 - \log h_k(\boldsymbol{u})) + \sum_{i=1}^{s_{k+1}} v_i \log v_i \right). \tag{9}$$

For this reason, similarly to what we did for extensive-form strategy spaces, we coin $\tilde{\varphi}_k$ the *dilatable global entropy DGF*. It is immediate to see by induction that $\nabla\psi$ can be computed exactly in linear time. furthermore, because ψ_k is a "nice" DGF by virtue of Proposition 1, and $\psi_k = \tilde{\varphi}_k$ on $\Delta^{s_1} \overset{h_1}{\triangleleft} \Delta^{s_2} \overset{h_2}{\triangleleft} \Delta^{s_3} \overset{h_3}{\triangleleft} \cdots \overset{h_n}{\triangleleft} \Delta^{s_n}$, we immediately obtain that $\tilde{\varphi}$ is a "nice" DGF.

We conclude this section by showing that it does not suffer from the issue we pointed out in Remark 2. We start by providing a refined version of Proposition 2, which follows immediately from taking a Hessian in (9).

LEMMA 6.2. *At any $\boldsymbol{u} \in (0,1)^{s_1 + \cdots + s_k}, \boldsymbol{v} \in (0,1)^{s_{k+1}}$, the Hessian matrix of the dilatable global entropy DGF satisfies*

$$\nabla^2\tilde{\varphi}_{k+1}(\boldsymbol{u}, \boldsymbol{v}) \geq \left(\begin{array}{c|c} \nabla^2\tilde{\varphi}_k(\boldsymbol{u}) - \mathrm{diag}\left(\left\{\frac{\alpha_{k+1}a_k}{u_i}\right\}_{i=1}^m\right) & \boldsymbol{0} \\ \hline \boldsymbol{0} & \mathrm{diag}\left(\left\{\frac{\alpha_{k+1}}{v_i}\right\}_{i=1}^n\right) \end{array} \right).$$

Unlike the bound in Proposition 2, the term subtracted in the top left matrix block is not multiplied by a factor of 2. Hence, the approach described in Remark 2 to set the DGF weights α_i ($i = 1, \ldots, k$) yields a DGF that is guaranteed to be 1-strongly convex that only requires polynomially small (in the dimension of the space)

7 EXPERIMENTS

In this section we study the numerical performance of our DGFs. First we study the performance of the dilatable global entropy for computing Nash equilibria in zero-sum EFGs, and second we study the performance for computing correlated equilibria and team equilibria.

Our experiments will be shown on nine different games, which span a variety of poker games, other recreational games, as well as a pursuit-evasion game played on a graph. All games are standard benchmarks in the computational game theory literature, and a full description of the games is given in the Appendix. In Table 1(a) we summarize some key dimensions of the game instances we use: the number of decision points $|\mathcal{J}_1|, |\mathcal{J}_2|$ for Player 1 and 2, respectively, the number of sequences $|\Sigma_1|, |\Sigma_2|$, and the number of terminal nodes (leaves).

Our experiments will show performance on three algorithms. First, we will plot the performance for both the EGT and mirror prox algorithms, with stepsizes and smoothing chosen according to the theoretical values dictated by Theorems 3.3 and B.1. Second, we will also show results on a tweaked variant of EGT called EGT/AS, which implements several heuristics that typically lead to better performance in practice, as seen in [20, 22, 23]. These heuristic are: *μ balancing*: At each iteration, we take a step on the player i whose smoothing parameter μ_i is larger. *Aggressive Stepsizing*: The original stepsize of EGT at iteration t is $\tau = 2/(3 + t)$, which is typically too conservative in practice. Instead, EGT/AS maintains some current value τ, initially set at $\tau = 0.5$. EGT/AS then repeatedly attempts to take steps with the current τ, and after every step checks whether the invariant condition of EGT still holds. If not, then we undo the step, decrease τ, and repeat the process. *Initial μ fitting*: The initial EGT values for μ_x, μ_y are much too conservative. Instead, At the beginning of the algorithm we perform a search over initial values for $\mu_x = \mu_y$. The search starts at the candidate value $\mu = 10^{-6}$ and stops as soon as the choice of $\mu_x = \mu_y = \mu$ yields an excessive gap value above 0.1. If the current choice does not, μ is incremented by 20% and the fitting continues. For all parameters above, we use the same values as in Kroer et al. [22], even though those values were tuned for the dilated entropy DGF, rather than dilatable global entropy.

In the presentation of the numerical performance, we will generally plot the number of iterations of the FOM on the x axis, rather than plot wall-clock time. Since we hold the algorithmic setup fixed in each plot, apart from the DGF, this gives a fair representation of performance, since they all use the same set of operations (in particular the same number of gradient computations, which is typically the most expensive operation). For EGT/AS, we will instead plot the number of gradient computations on the x axis, since the number of gradient computations can vary for each DGF, depending on the amount of backtracking incurred.

7.1 Nash Equilibrium Computation

We will focus on comparing our new dilatable global entropy for the sequence-form polytope (Definition 5.1) to the prior state-of-the-art dilated entropy DGF (Definition 4.1) from [23].

Before we study the numerical performance, we look at the size of the DGF weights β and γ for each of the games. Table 1 column **(b)** shows the average and maximum size of the dilated entropy, and Table 1 column **(c)** shows the corresponding values for the DGE. We see that the DGE requires vastly less weight, especially in terms of the maximal weights near the root of each decision space.

First, we study the theoretically-correct way to use the DGFs. In particular, we instantiate both EGT and mirror prox with the stepsizes and DGFs as specified in Theorems 3.3 and B.1, for the dilatable global entropy and dilated entropy. The results for EGT are shown in Figures 2 and 3. Across both algorithms and all nine games, we see that our new dilatable global entropy DGF performs better, sometimes by over an order of magnitude (e.g. in Liar's dice and pursuit evasion (6 turns)). This is in line with the fact that our new DGF has a better strong convexity modulus,
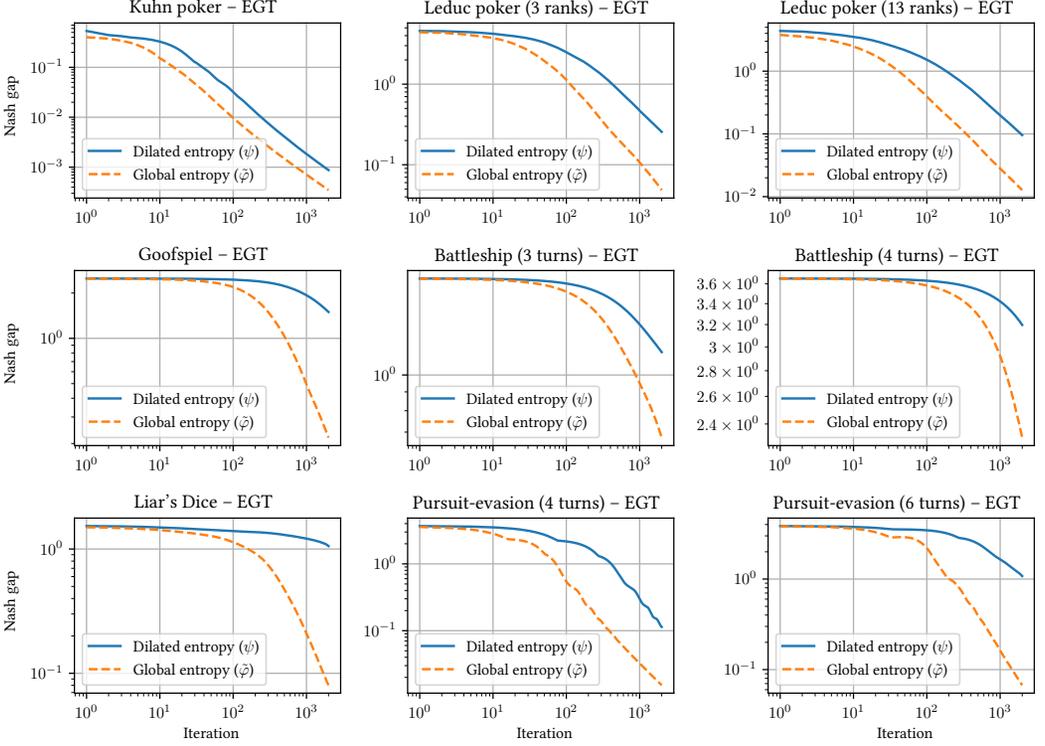
Fig. 2. Performance of the EGT algorithm instantiated with the two entropy DGFs across nine games. The x axis shows the number of EGT iterations, and the y axis shows the distance to Nash equilibrium.

| Game instance | Decision Points | | Sequences | | Leaves | Weights β | | Weights γ | |
	$\|\mathcal{J}_1\|$	$\|\mathcal{J}_2\|$	$\|\Sigma_1\|$	$\|\Sigma_2\|$	$\|Z\|$	Avg	Max	Avg	Max
Kuhn poker	6	6	13	13	30	8.86	38	2.29	7
Leduc poker (3 ranks)	144	144	337	337	1116	11.77	686	2.12	43
Leduc poker (13 ranks)	2574	2574	6007	6007	98 956	12.06	12 326	2.13	703
Goofspiel	17 476	17 476	21 329	21 329	13 824	6.91	23 442	1.70	917
Battleship (3 turns)	18 152	62 875	73 130	253 940	552 132	3.29	2894	1.24	99
Battleship (4 turns)	316 520	734 203	968 234	2 267 924	3 487 428	3.75	27 470	1.33	483
Liar's dice	12 288	12 288	24 571	24 571	147 420	15.56	65 546	2.04	1399
Pursuit-evasion (4 turns)	34	348	52	2029	15 898	8.29	62	1.94	5
Pursuit-evasion (6 turns)	58	11 830	78	68 951	118 514	19.42	254	2.51	7
	(a) — Game instances and sizes					**(b)**		**(c)**	

Table 1. Column **(a)**: various measures of the size of each of the games that we test algorithms on. Columns **(b)** and **(c)**: the magnitude of the dilated entropy DGF and DGE weights.

which allows for a much smaller amount of smoothing, while still guaranteeing correctness. This in turns allows the algorithms to safely take larger steps, thereby progressing faster.

Secondly, we investigate the numerical performance of the two entropy DGFs in the EGT/AS algorithm in Figure 4. Here we see a smaller performance improvement. For most of the games, we get a small factor of improvement for the first 100 or so iterations, but then the performance
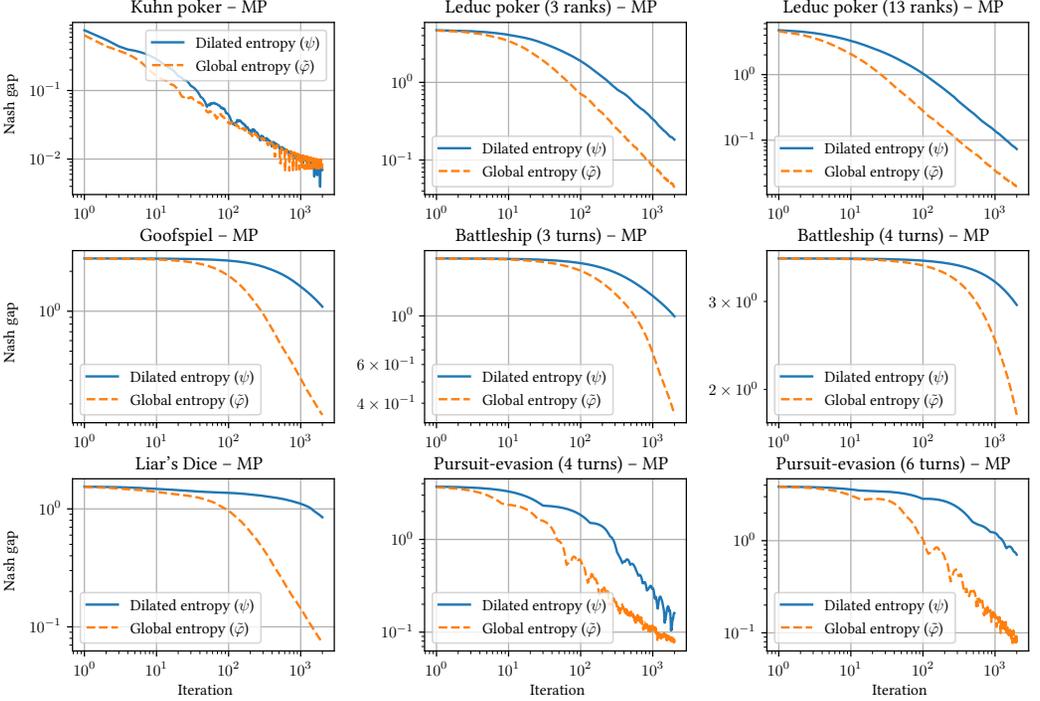
Fig. 3. Performance of the MP algorithm instantiated with the two entropy DGFs across nine games.

is similar thereafter. For Liar's Dice there is a persistent improvement to using dilatable global entropy across all iterations.

7.2 Correlated and Team Equilibrium Computation

Next we investigate the computational performance of our extension of both the dilated entropy DGF and the DGE DGF to the scaled extension operator. In particular, we will consider the problem of computing an NFCCE, which we saw in Section 6.1 can be formulated as a BSPP via the scaled extension operator. Since the constructed polytope is the same for ex-ante team correlated equilibria, extensive form correlated, and extensive form coarse correlated, we restrict our attention to NFCCE and leave the numerical investigation on the other solution concepts for future work. We expect the takeaways to be similar.

Figure 5 shows the results for instantiating the mirror prox algorithm with our two DGFs. We see that, similar to the case of zero-sum Nash equilibrium, the DGE DGF performs much better than the dilated entropy DGF, again likely due to the smaller weights needed in order to make the DGF strongly convex on the correlation-plan polytope.

8 CONCLUSIONS AND FUTURE RESEARCH

We introduced the dilatable global entropy as a distance-generating function for sequential decision-making polytopes such as those encountered in sequential games. We showed that the DGE function leads to better strong-convexity properties than prior DGFs for the sequence-form polytope, and it improves the associated polytope diameter, as well as the convergence rate of FOMs, by a factor of $2^{\mathfrak{D}_Q}$. Experiments confirmed that this leads to a superior notion of distance. We then extended the DGE, as well as the general dilation framework, to the scaled extension operation. We thereby
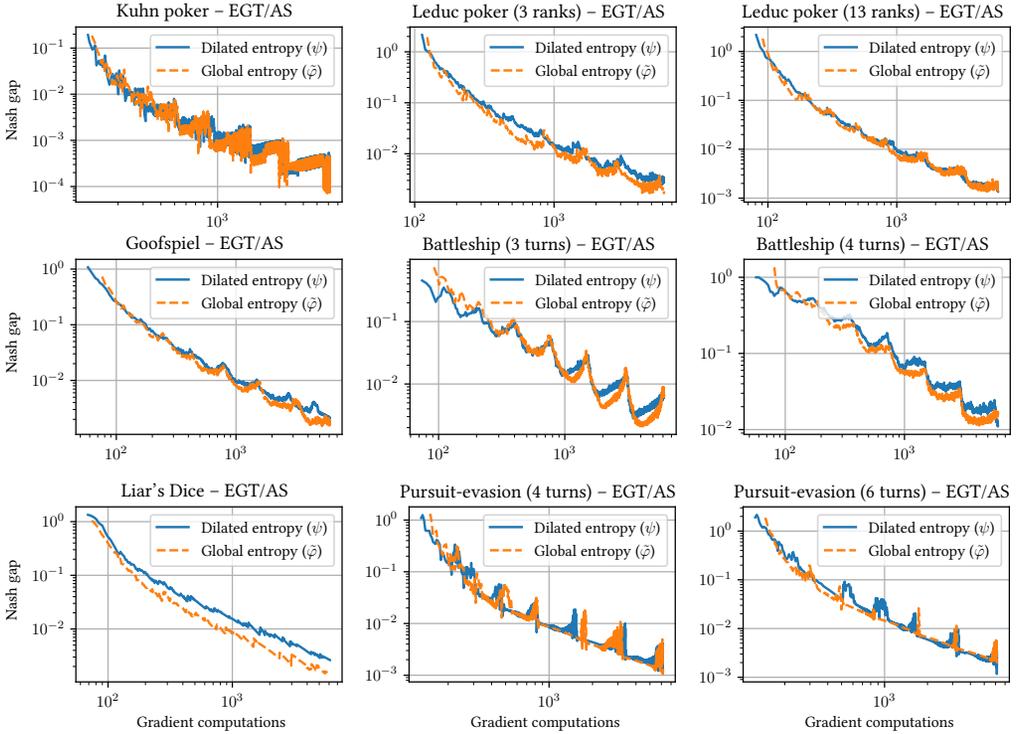
Fig. 4. Performance of the EGT/AS algorithm instantiated with the two entropy DGFs, as well as aggressive stepsizing, μ balancing, and initial μ fitting.
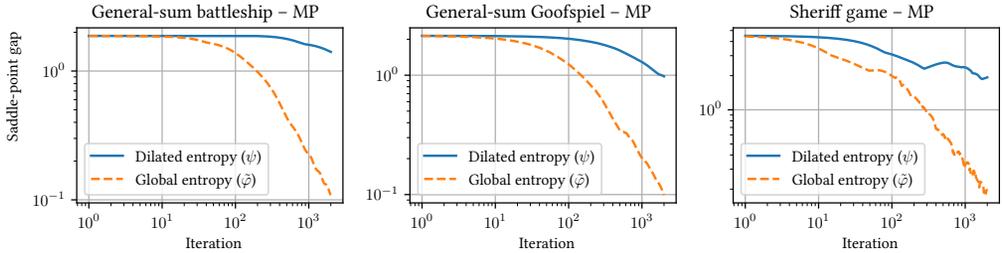

Fig. 5. Performance of the MP algorithm for finding normal-form coarse-correlated equilibria in three general-sum games.

showed how to construct suitable DGFs for the convex polytopes encounted when computing certain correlated equilibria, as well as team equilibria. Based on these extensions, we showed the first algorithm that achieves a $1/T$ convergence rate to the set of of various correlated equilibria and ex-ante team coordinated equilibria, while requiring only linear time (in the polytope size) for each iteration.

In future research, it would be interesting to investigate whether our new DGFs can be used to achieve numerical performance comparable to that of the currently practically-fastest algorithms, that is, new CFR variants [6, 15, 35], which have worse theoretical convergence rate. In particular, we think that stochastic methods could be a promising line of research for this, because it is harder to tune the stepsize in those methods, in order to account for the weights previously used in the dilated entropy DGF.

REFERENCES

[1] Ahron Ben-Tal and Arkadi Nemirovski. 2001. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications.* Vol. 2. Siam.

[2] Branislav Bošanský and Jiří Čermák. 2015. Sequence-form algorithm for computing Stackelberg equilibria in extensive-form games. In *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

[3] Branislav Bošanský, Christopher Kiekintveld, V Lisý, and Michal Pěchouček. 2014. An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information. *Journal of Artificial Intelligence Research* (2014), 829–866.

[4] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up Limit Hold'em Poker is Solved. *Science* 347, 6218 (Jan. 2015).

[5] Noam Brown and Tuomas Sandholm. 2017. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* (Dec. 2017), eaao1733.

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

[7] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.

[8] Andrea Celli and Nicola Gatti. 2018. Computational results for extensive-form adversarial team games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[9] Nicolo Cesa-Bianchi and Gabor Lugosi. 2006. *Prediction, learning, and games.* Cambridge University Press.

[10] Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. 2020. Coarse correlation in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1934–1941.

[11] Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. 2020. Faster Algorithms for Optimal Ex-Ante Coordinated Collusive Strategies in Extensive-Form Zero-Sum Games. *arXiv preprint arXiv:2009.10061* (2020).

[12] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Online Convex Optimization for Sequential Decision Processes and Extensive-Form Games. In *AAAI Conference on Artificial Intelligence.*

[13] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Optimistic Regret Minimization for Extensive-Form Games via Dilated Distance-Generating Functions. In *Advances in Neural Information Processing Systems, NeurIPS 2019,.* 5222–5232.

[14] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2020. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning.* PMLR, 3018–3028.

[15] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2021. Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[16] Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. 2019. Correlation in Extensive-Form Games: Saddle-Point Formulation and Benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS).*

[17] Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. 2019. Efficient Regret Minimization Algorithm for Extensive-Form Correlated Equilibrium. In *Advances in Neural Information Processing Systems, NeurIPS 2019.* 5187–5197.

[18] Gabriele Farina and Tuomas Sandholm. 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.*

[19] Richard Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. 2012. Generalized Sampling and Variance in Counterfactual Regret Minimization. In *AAAI Conference on Artificial Intelligence (AAAI).*

[20] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. 2010. Smoothing Techniques for Computing Nash Equilibria of Sequential Games. *Mathematics of Operations Research* 35, 2 (2010).

[21] Christian Kroer, Gabriele Farina, and Tuomas Sandholm. 2018. Robust Stackelberg Equilibria in Extensive-Form Games and Extension to Limited Lookahead. In *AAAI Conference on Artificial Intelligence (AAAI).*

[22] Christian Kroer, Gabriele Farina, and Tuomas Sandholm. 2018. Solving Large Sequential Games with the Excessive Gap Technique. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS).*

[23] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. 2020. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming* (2020).

[24] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. 2015. Faster First-Order Methods for Extensive-Form Game Solving. In *Proceedings of the ACM Conference on Economics and Computation (EC).*

[25] H. W. Kuhn. 1950. A Simplified Two-Person Poker. In *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker (Eds.). Annals of Mathematics Studies, 24, Vol. 1. Princeton University Press, Princeton, New Jersey, 97–103.

[26] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. Monte Carlo Sampling for Regret Minimization in Extensive Games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS).*

[27] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* (May 2017).

[28] H. Moulin and J.-P. Vial. 1978. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory* 7, 3-4 (1978), 201–221.

[29] Arkadi Nemirovski. 2004. 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).

[30] Yurii Nesterov. 2005. Excessive Gap Technique in Nonsmooth Convex Minimization. *SIAM Journal of Optimization* 16, 1 (2005).

[31] Yurii Nesterov. 2005. Smooth Minimization of Non-Smooth Functions. *Mathematical Programming* 103 (2005).

[32] Sheldon M Ross. 1971. Goofspiel—the game of pure strategy. *Journal of Applied Probability* 8, 3 (1971), 621–625.

[33] Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. 2019. Variance reduction in monte carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2157–2164.

[34] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. 2005. Bayes' Bluff: Opponent Modelling in Poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*.

[35] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. 2015. Solving Heads-up Limit Texas Hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.

[36] Bernhard von Stengel. 1996. Efficient Computation of Behavior Strategies. *Games and Economic Behavior* 14, 2 (1996), 220–246.

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

[38] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

A PROOFS MISSING FROM THE MAIN BODY

A.1 Proof of Theorem 5.3

PROOF. The interior of domain is $(0, 1)^{|\Sigma|}$, a nonempty set, and $\tilde{\varphi}$ is clearly twice-differentiable on it. Furthermore, $\tilde{\varphi}$ is easily seen to be 1-strongly convex, as the Hessian is

$$\nabla^2 \tilde{\varphi}(x) = \text{diag}\left(\left\{\frac{w_{ja}}{x_{ja}}\right\}_{ja \in \Sigma}\right) \succeq I,$$

since $0 \le x_{ja} \le 1$ and $w_{ja} \ge 1$. As strong convexity implies strict convexity, $\tilde{\varphi}$ satisfies all the properties listed in the statement. □

A.2 Proof of Theorem 5.4

PROOF. Using the second-order definition of strong convexity, we wish to show that the inequality $h^\top \nabla^2 \tilde{\varphi}(x) h \ge \frac{1}{M_Q} \|h\|_1^2$ holds for any $h \in \mathbb{R}^{|\Sigma|}$. Expanding the Hessian matrix gives

$$h^\top \nabla^2 \tilde{\varphi}(x) h = h^\top \text{diag}\left(\left\{\frac{w_{ja}}{x_{ja}}\right\}_{ja \in \Sigma}\right) h \ge \sum_{ja \in \Sigma} \frac{h_{ja}^2}{x_{ja}}. \tag{10}$$

On the other hand, by expanding the definition of $\|h\|_1^2$ and applying the Cauchy-Schwarz inequality, we have

$$\|h\|_1^2 = \left(\sum_{ja} |h_{ja}|\right)^2 = \left(\sum_{ja} \frac{|h_{ja}|}{\sqrt{x_{ja}}} \sqrt{x_{ja}}\right)^2 \le \left(\sum_{ja} \frac{h_{ja}^2}{x_{ja}}\right)\left(\sum_{ja} x_{ja}\right) \le \left(\sum_{ja} \frac{h_{ja}^2}{x_{ja}}\right) M_Q.$$

Substituting (10) into the last inequality yields a proof of the desired strong convexity modulus $1/M_Q$. □

A.3 Proof of Theorem 5.5

PROOF. By the definition of the polytope diameter and the fact that we chose our DGFs such that $\min_{x \in \mathcal{A}} \tilde{\varphi}(x) = 0$, we have

$$\Omega_{\tilde{\varphi}, Q} \le \max_{x \in Q} \tilde{\varphi}(x) \le \sum_{j \in \mathcal{J}} \gamma_j x_{p_j} \log |A_j| \le \max_{j' \in \mathcal{J}} \log |A_{j'}| \sum_{j \in \mathcal{J}} \gamma_j x_{p_j}$$

$$\le M_Q \max_{j' \in \mathcal{J}} \log |A_{j'}| \sum_{j \in \mathcal{J}} x_{p_j} \le M_Q^2 \max_{j' \in \mathcal{J}} \log |A_{j'}|.$$

The first inequality is by noting that $\log x_{ja} \le 0$ since $x_{ja} \le 1$ for all $j, a \in \Sigma$. The second inequality is by taking the maximum. The third inequality is by noting that γ_j is largest at root decision points, where it is at most M_Q. The fourth inequality upper bounds $\sum_{j \in \mathcal{J}} x_{p_j}$ by M_Q. □

A.4 Proof of Proposition 1

PROOF. Since $\text{int}(\mathcal{A} \times \mathcal{B}) = \text{int}(\mathcal{A}) \times \text{int}(\mathcal{B})$, the domain of d_z has nonempty interior. Showing differentiability in the interior is easy, and in particular at any $(u, v) \in \text{int}(\mathcal{A}) \times \text{int}(\mathcal{B})$ it holds that

$$\nabla d_z(u, v) = \begin{pmatrix} \nabla d_u(u) + \alpha_v \left(d_v\left(\frac{v}{h(u)}\right) - \nabla d_v\left(\frac{v}{h(u)}\right)^\top \frac{v}{h(u)}\right) a \\ \alpha_v \nabla d_v\left(\frac{v}{h(u)}\right) \end{pmatrix}.$$

Hence, the only task missing to complete the verification that d_z is Legendre is to show that d_z is strictly convex on $\text{int}(\mathcal{A}) \times \text{int}(\mathcal{B})$. The proof is standard and we omit it.

We now turn to the second part of the statement, and assume that d_u and d_v are "nice" DGFs. It is clear that the gradient (A.4) can be computed in linear time: we only need to take inner products which takes time linear in the dimension of $\mathcal{V}$, compute the value of $h(\boldsymbol{u})$ which is linear in the dimension of $\mathcal{U}$, and compute corresponding values of d_u and d_v, which takes linear time by assumption. As for the gradient of the convex conjugate, we start by noting that

$$\max_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}\left\{\boldsymbol{g}_u^\top\boldsymbol{u}+h(\boldsymbol{u})\boldsymbol{g}_v^\top\boldsymbol{v}-d_u(\boldsymbol{u})-\alpha_v h(\boldsymbol{u})d_v(\boldsymbol{v})\right\}$$

$$=\max_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}\left\{\boldsymbol{g}_u^\top\boldsymbol{u}-d_u(\boldsymbol{u})+\alpha_v\,h(\boldsymbol{u})\left[\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)^\top\boldsymbol{v}-d_v(\boldsymbol{v})\right]\right\}$$

$$=\max_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}\left\{\boldsymbol{g}_u^\top\boldsymbol{u}-d_u(\boldsymbol{u})+\alpha_v\,h(\boldsymbol{u})\left[\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)^\top\boldsymbol{v}-d_v(\boldsymbol{v})\right]\right\}$$

$$=\max_{\boldsymbol{u}\in\mathcal{U}}\left\{\boldsymbol{g}_u^\top\boldsymbol{u}-d_u(\boldsymbol{u})+\alpha_v\,h(\boldsymbol{u})\max_{\boldsymbol{v}\in\mathcal{V}}\left\{\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)^\top\boldsymbol{v}-d_v(\boldsymbol{v})\right\}\right\}$$

$$=\max_{\boldsymbol{u}\in\mathcal{U}}\left\{\left(\boldsymbol{g}_u+\alpha_v\,d_v^*\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)\boldsymbol{a}\right)^\top\boldsymbol{u}-d_u(\boldsymbol{u})\right\},$$

where the third equality follows since $\alpha_v>0$ and $h(\boldsymbol{u})\geq 0$ by hypothesis, and the fourth equality follows from the definition of $h(\boldsymbol{u})=\boldsymbol{a}^\top\boldsymbol{u}$ for all $\boldsymbol{u}$. From that observation, it follows that

$$\nabla d_z^*(\boldsymbol{g}_u,\boldsymbol{g}_v)=\begin{pmatrix}\nabla d_u^*\left(\boldsymbol{g}_u+\alpha_v\,d_v^*\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)\boldsymbol{a}\right)\\\nabla d_v^*\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right).\end{pmatrix}\tag{11}$$

Since

$$d_v^*\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)=\max_{\boldsymbol{v}\in\mathcal{V}}\left\{\left(\frac{\boldsymbol{g}_v}{\alpha_v}\right)^\top\boldsymbol{v}-d_v(\boldsymbol{v})\right\}$$

can be computed in linear time starting from $\nabla d_v^*(\boldsymbol{g}_v/\alpha_v)$, the gradient of the conjugate of d_z, given in (11), can be evaluated in linear time provided that ∇d_u^* and ∇d_v^* can, which is assumed by hypothesis since d_u and d_v are "nice".   □

A.5 Proof of Proposition 2

PROOF. By using algebraic manipulations, it is easy to show that, for all $(\boldsymbol{u},\boldsymbol{v})\in\mathcal{Z}$,

$$(\nabla d_z(\boldsymbol{u},\boldsymbol{v})-\nabla d_z(\boldsymbol{u},\boldsymbol{v}))^\top\begin{pmatrix}\boldsymbol{u}-\boldsymbol{u}'\\\boldsymbol{v}-\boldsymbol{v}'\end{pmatrix}\geq\frac{1}{2}(\boldsymbol{u}-\boldsymbol{u}')^\top D_u(\boldsymbol{u}-\boldsymbol{u}')+\alpha_v h\left(\frac{\boldsymbol{u}+\boldsymbol{u}'}{2}\right)\left\|\frac{\boldsymbol{v}}{h(\boldsymbol{u})}-\frac{\boldsymbol{v}'}{h(\boldsymbol{u})}\right\|^2.\tag{12}$$

Now, we note that

$$\|\boldsymbol{v}-\boldsymbol{v}'\|=\left\|h(\boldsymbol{u})\frac{\boldsymbol{v}}{h(\boldsymbol{u})}-h(\boldsymbol{u}')\frac{\boldsymbol{v}'}{h(\boldsymbol{u}')}\right\|$$

$$=\left\|h\left(\frac{\boldsymbol{u}+\boldsymbol{u}'}{2}\right)\left(\frac{\boldsymbol{v}}{h(\boldsymbol{u})}-\frac{\boldsymbol{v}'}{h(\boldsymbol{u}')}\right)+(h(\boldsymbol{u})-h(\boldsymbol{u}'))\left[\frac{1}{2}\left(\frac{\boldsymbol{v}}{h(\boldsymbol{u})}+\frac{\boldsymbol{v}'}{h(\boldsymbol{u}')}\right)\right]\right\|$$

$$\leq h\left(\frac{\boldsymbol{u}+\boldsymbol{u}'}{2}\right)\left\|\frac{\boldsymbol{v}}{h(\boldsymbol{u})}-\frac{\boldsymbol{v}'}{h(\boldsymbol{u}')}\right\|+(h(\boldsymbol{u})-h(\boldsymbol{u}')),$$

where we used the first assumption to bound the norm of $\frac{1}{2}\left(\frac{v}{h(u)} + \frac{v'}{h(u')}\right)$, which belongs to $\mathcal{V}$ by definition of $\mathcal{Z}$. Hence, by rearranging and taking squares we have

$$
h\left(\frac{u + u'}{2}\right)\left\|\frac{v}{h(u)} - \frac{v'}{h(u')}\right\| \geq h\left(\frac{u + u'}{2}\right)^2\left\|\frac{v}{h(u)} - \frac{v'}{h(u')}\right\|^2
$$
$$
\geq \frac{1}{2}\|v - v'\|^2 - (h(u) - h(u'))^2
$$
$$
= \frac{1}{2}\|v - v'\|^2 - (a^\top(u - u))^2.
$$

Plugging the last expression into (12) and using the fact that $aa^\top \preceq \|a\|_2^2 I$ we obtain the statement. $\qquad\square$

B DETAILED DESCRIPTION OF GAME INSTANCES USED IN NUMERICAL EXPERIMENTS

Here we describe each of the games that we consider in the experimental section of the paper.

Kuhn poker is a standard benchmark in the EFG-solving community [25]. In Kuhn poker, each player puts an ante worth 1 into the pot. Each player is then privately dealt one card from a deck that contains 3 unique cards (Jack, Queen, King). Then, a single round of betting then occurs, with the following dynamics. First, Player 1 decides to either check or bet 1. Then,
- If Player 1 checks Player 2 can check or raise 1.
  - If Player 2 checks a showdown occurs; if Player 2 raises Player 1 can fold or call.
    * If Player 1 folds Player 2 takes the pot; if Player 1 calls a showdown occurs.
- If Player 1 raises Player 2 can fold or call.
  - If Player 2 folds Player 1 takes the pot; if Player 2 calls a showdown occurs.

When a showdown occurs, the player with the higher card wins the pot and the game immediately ends.

Leduc poker is another standard benchmark in the EFG-solving community [34]. The game is played with a deck of R unique cards, each of which appears exactly twice in the deck. The game is composed of two rounds. In the first round, each player places an ante of 1 in the pot and is dealt a single private card. A round of betting then takes place, with Player 1 acting first. At most two bets are allowed per player. Then, a card is is revealed face up and another round of betting takes place, with the same dynamics described above. After the two betting round, if one of the players has a pair with the public card, that player wins the pot. Otherwise, the player with the higher card wins the pot. All bets in the first round are worth 1, while all bets in the second round are 2.

Goofspiel is another popular benchmark game, originally proposed by Ross [32]. It is a two-player card game, employing three identical decks of k cards each whose values range from 1 to k. At the beginning of the game, each player gets dealt a full deck as their hand, and the third deck (the "prize" deck) is shuffled and put face down on the board. In each turn, the topmost card from the prize deck is revealed. Then, each player privately picks a card from their hand. This card acts as a bid to win the card that was just revealed from the prize deck. The selected cards are simultaneously revealed, and the highest one wins the prize card.
In the zero-sum version of the game, if the players' played cards are equal, the prize card is split. In the general-sum version of the game, denoted "General-sum Goofspiel" and used in the experiments on NFCCE, the prize card is thrown out on tie. Either way, the players' scores are computed as the sum of the values of the prize cards they have won.

Pursuit-evasion is a security-inspired pursuit-evasion game played on the graph shown in Figure 6. It is a zero-sum variant of the one used by Kroer et al. [21], and a similar search game has been considered by Bošanskỳ et al. [3] and Bošanskỳ and Čermák [2].
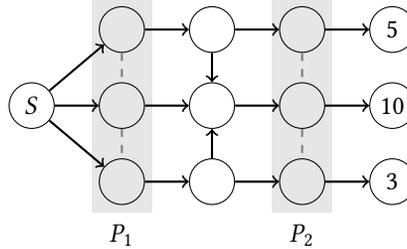

Fig. 6. The graph on which the search game is played.

In each turn, the attacker and the defender act simultaneously. The defender controls two patrols, one per each respective patrol areas labeled P_1 and P_2. Each patrol can move by one step along the grey dashed lines, or stay in place. The attacker starts from the leftmost node (labeled S) and at each turn can move to any node adjacent to its current position by following the black directed edges. The attacker can also choose to wait in place for a time step in order to hide all their traces. If a patrol visits a node that was previously visited by the attacker, and the attacker did not wait to clean up their traces, they will see that the attacker was there. The goal of the attacker is to reach any of the rightmost nodes, whose corresponding payoffs are 5, 10, or 3, respectively, as indicated in Figure 6. If at any time the attacker and any patrol meet at the same node, the attacker is loses the game, which leads to a payoff of −1 for the attacker and of 1 for the defender. The game times out after m simultaneous moves, in which case both players defender receive payoffs 0.

Battleship is a parametric version of a classic board game, where two competing fleets take turns shooting at each other [16]. At the beginning of the game, the players take turns at secretly placing a set of ships on separate grids (one for each player) of size 3×2. Each ship has size 2 (measured in terms of contiguous grid cells) and a value of 4, and must be placed so that all the cells that make up the ship are fully contained within each player's grids and do not overlap with any other ship that the player has already positioned on the grid. After all ships have been placed. the players take turns at firing at their opponent. Ships that have been hit at all their cells are considered sunk. The game continues until either one player has sunk all of the opponent's ships, or each player has completed R shots. At the end of the game, each player's payoff is calculated as the sum of the values of the opponent's ships that were sunk, minus the sum of the values of ships which that player has lost.

In the general-sum variant we consider in the NFCCE experiments, we set $R = 3$, and furthermore we set each player's payoff is calculated as the sum of the values of the opponent's ships that were sunk, minus the sum of the values of ships which that player has lost *times two*. This modification makes the game general-sum, and makes the players more risk-averse. Because of that, it was observed by Farina et al. [16] that the introduction of a mediator in the game (through the correlated solution concept) enables to players to reach equilibrium states with significantly larger social welfare.

Liar's dice is another standard benchmark in the EFG-solving community [?]. In our instantiation, each of the two players initially privately rolls an unbiased 6-face die. The first player begins bidding, announcing any face value up to 6 and the minimum number of dice that the player

believes are showing that value among the dice of both players. Then, each player has two choices during their turn: to make a higher bid, or to challenge the previous bid by declaring the previous bidder a "liar". A bid is higher than the previous one if either the face value is higher, or the number of dice is higher. If the current player challenges the previous bid, all dice are revealed. If the bid is valid, the last bidder wins and obtains a reward of +1 while the challenger obtains a negative payoff of −1. Otherwise, the challenger wins and gets reward +1, and the last bidder obtains reward of −1.

Sheriff The Sheriff game is inspired by the Sheriff of Nottingham board game and was introduced by Farina et al. [16] as a benchmark game for correlated solution concepts in extensive-form game. Player 1 (the "smuggler") selects the number of illegal items to be placed in the cargo (in our case, between 0 and 3). The selected number is unknown to Player 2 (the "sheriff"). Then, the game proceeds for 3 bargaining rounds. In each round, the following happens:

- The smuggler selects an integer bribe amount, in the range 0 to 3 (inclusive). The selected amount is public information. However, the smuggler does not actually give money to the sheriff, unless this is the final round.
- Then, the sheriff tells the smuggler whether he is planning to inspect the cargo. However, no cargo is actually inspected other than in the final round. The sheriff can change his mind in later rounds, except for the final round.

B.1 Mirror Prox (MP)

Next we consider the *Mirror Prox (MP)* algorithm [29]. As for EGT, we will assume that we have proximal setups for both $\mathcal{X}$ and $\mathcal{Y}$, with one-strongly-convex DGFs d_x, d_y. Rather than construct smoothed approximations, mirror prox directly uses the DGFs to take first-order steps. Algorithm 3 shows the sequence of steps. Compared to EGT, mirror prox has a somewhat simpler structure: it simply takes repeated extrapolated proximal steps. First, a proximal step in the descent direction is taken for both x and y. Then, the gradient at those new points is used to take a proximal step starting from the previous iterate (this is the extrapolation part: a step is taken starting from the previous iterate, but with the extrapolated gradient). Finally, the *average* strategy is output.

ALGORITHM 3: Mirror Prox (MP) algorithm.

11 **function** INITIALIZE()
12      $t \leftarrow 0$
13      $z_x^0 \leftarrow \arg\min_{\hat{x} \in \mathcal{X}} d_x(\hat{x})$
14      $z_y^0 \leftarrow \arg\min_{\hat{y} \in \mathcal{Y}} d_y(\hat{y})$

15 **function** ITERATE()
16      $t \leftarrow t + 1$
17      $w_x^t \leftarrow \text{prox}_{z_x^t}\left(\eta^t A z_y^{t-1}\right)$
18      $w_y^t \leftarrow \text{prox}_{z_y^t}\left(-\eta^t A^\top z_x^{t-1}\right)$
19      $z_x^{t+1} \leftarrow \text{prox}_{z_x^t}\left(\eta^t A w_y^t\right)$
20      $z_y^{t+1} \leftarrow \text{prox}_{z_y^t}\left(-\eta^t A^\top w_x^t\right)$
21      $x^t \leftarrow [\sum_{\tau=1}^t \eta^\tau]^{-1} \sum_{\tau=1}^t \eta^\tau w_x^\tau$
22      $y^t \leftarrow [\sum_{\tau=1}^t \eta^\tau]^{-1} \sum_{\tau=1}^t \eta^\tau w_y^\tau$

Note: $\{\eta^t\}$ is a sequence of step-size parameters. A well-known and theoretically-sound choice for η^t is $\eta^t := \frac{1}{\|A\|}$ for all $t = 0, 1, \ldots$ (see also Theorem B.1).

The mirror prox algorithm also converges at rate $O(1/T)$:

THEOREM B.1 (BEN-TAL AND NEMIROVSKI [1] THEOREM 5.5.1). *Suppose the stepsize in Algorithm 3 is set as $\eta_t = 1/\|A\|$. Then we have*

$$\max_{y \in \mathcal{Y}} (x^t)^\top A y - \min_{x \in \mathcal{X}} x^\top A y^t \le \frac{\|A\|(\Omega_{d_x, \mathcal{X}} + \Omega_{d_y, \mathcal{Y}})}{2t}.$$