SMOOTHING TECHNIQUES FOR COMPUTING NASH EQUILIBRIA OF SEQUENTIAL GAMES

SAMID HODA*, ANDREW GILPIN†, AND JAVIER PEÑA*

Abstract. We develop first-order smoothing techniques for saddle-point problems that arise in the Nash equilibria computation of sequential games. The crux of our work is a construction of suitable prox-functions for a certain class of polytopes that encode the sequential nature of the games. An implementation based on our smoothing techniques computes approximate Nash equilibria for games that are four orders of magnitude larger than what conventional computational approaches can handle.

1. Introduction

The Nash equilibria of two-person, zero-sum sequential games are the solutions to

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

where $\mathcal{X}$ and $\mathcal{Y}$ are polytopes defining the players' strategies and A is the payoff matrix [13, 14]. When the minimizer plays a strategy $\mathbf{x} \in \mathcal{X}$ and the maximizer plays $\mathbf{y} \in \mathcal{Y}$, the expected utility to the maximizer is $\langle \mathbf{y}, A\mathbf{x}\rangle$ and, since the game is zero-sum, the minimizer's expected utility is $\langle \mathbf{y}, -A\mathbf{x}\rangle$. Problem (1) can be expressed as a linear program, but the resulting formulations are prohibitively large for most interesting games. For instance, the payoff matrix A in (1) for limit Texas Hold'em poker has dimension $10^{14} \times 10^{14}$ and contains more than 10^{18} non-zero entries. Problems of this magnitude are far beyond the capabilities of state-of-the-art general-purpose linear programming solvers. Even solving a substantially smaller game with a $10^6 \times 10^6$ payoff matrix containing 50 million non-zeros with conventional linear programming solvers is computationally demanding both in terms of time and memory [3].

We present a novel algorithmic approach for finding approximate solutions to (1). To this end, we define polytopes called *complexes* and concentrate on solving (1) when $\mathcal{X}$ and $\mathcal{Y}$ are polytopes of this type. Complexes generalize simplexes and include as a special case the strategy sets of sequential games. In this paper, we adapt Nesterov's smoothing techniques [11, 12] for approximating (1). In particular, we develop first-order algorithms that take $\mathcal{O}(1/\epsilon)$ iterations to compute $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ such that

$$0 \leq \max_{\mathbf{v}\in\mathcal{Y}} \langle \mathbf{v}, A\mathbf{x}\rangle - \min_{\mathbf{u}\in\mathcal{X}} \langle \mathbf{y}, A\mathbf{u}\rangle \leq \epsilon.$$

Such a pair of strategies is called an ϵ-equilibrium.

An attractive feature of our algorithms is their simplicity and the low computational cost of each iteration. An implementation based on our approach has been successful in obtaining ϵ-equilibria for sequential games where the payoff matrix A is of size $10^8 \times 10^8$

* Tepper School of Business, Carnegie Mellon University.
† Computer Science Department, Carnegie Mellon University.

and contains more than 10^{12} entries [2]. (This is four orders of magnitude larger than what was previously possible.) That implementation is a key component of several successful poker-playing computer programs [4, 5].

The paper is organized as follows. Section 2 summarizes Nesterov's smoothing technique as it applies to problem (1). We highlight that technique's crucial ingredient, a pair of suitable *prox-functions* for the sets $\mathcal{X}$ and $\mathcal{Y}$. Section 3 presents our main idea, a template for constructing suitable prox-functions for complexes. Section 4 considers the special case of *uniform complexes*. For these complexes we provide explicit bounds on the number of iterations needed for finding an ϵ-equilibrium. Section 5 summarizes some computational experience with an implementation based on our approach. Finally, Section 6 summarizes the main conclusions and discusses ideas for future work.

2. Smoothing techniques

Problem (1) can be stated as

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{y}) \tag{2}$$

where

$$f(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}} \langle A\mathbf{y}, \mathbf{x}\rangle \text{ and } \phi(\mathbf{y}) = \min_{\mathbf{x}\in\mathcal{X}} \langle A\mathbf{y}, \mathbf{x}\rangle.$$

The functions f and ϕ are respectively convex and concave non-smooth functions. The left-hand side of (2) is a standard convex minimization problem of the form

$$\bar{h} := \min\{h(\mathbf{x}) : \mathbf{x} \in X\}. \tag{3}$$

First-order methods for solving (3) are algorithms whose search direction at each iteration is obtained using only the first-order information of h, such as its gradient or subgradient. When h is smooth with Lipschitz gradient, there is a first-order algorithm for finding a point $\mathbf{x} \in \mathcal{X}$ such that $h(\mathbf{x}) \leq \bar{h} + \epsilon$ after $\mathcal{O}(1/\sqrt{\epsilon})$ iterations [9]. When h is non-smooth, subgradient algorithms can be applied, but they have a worst-case complexity of $\mathcal{O}(1/\epsilon^2)$ iterations [6]. However, that pessimistic result is based on treating h as a *black-box* where the value and subgradient are accessed via an oracle. For non-smooth functions with a suitable max structure, Nesterov devised first-order algorithms requiring only $\mathcal{O}(1/\epsilon)$ iterations by applying a clever *smoothing technique* [11, 12]. In this paper, we adapt that smoothing technique for solving problem (1).

The key component of Nesterov's smoothing technique is a pair of *prox-functions* for the sets $\mathcal{X}$ and $\mathcal{Y}$. These prox-functions are used to construct smooth approximations $f_\mu \approx f$ and $\phi_\mu \approx \phi$. To obtain approximate solutions to (2), gradient-based algorithms can then be applied to f_μ and ϕ_μ.

Definition 2.1. Assume $Q \subseteq \mathbb{R}^n$ is a convex compact set. A function $d : Q \to \mathbb{R}$ is a *prox-function* if it satisfies the following properties

- d is strongly convex in Q, i.e., there exists $\sigma > 0$ such that for all $\mathbf{x}, \mathbf{y} \in Q$, and $\alpha \in [0, 1]$

$$d(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha d(\mathbf{x}) + (1-\alpha)d(\mathbf{y}) - \frac{1}{2}\sigma\alpha(1-\alpha)\|\mathbf{x}-\mathbf{y}\|^2. \tag{4}$$

  The largest value of the constant σ that satisfies (4) is the *strong convexity modulus* of d.

- $\min\{d(\mathbf{x}) : \mathbf{x} \in Q\} = 0$.

When $d : Q \to \mathbb{R}$ is differentiable, (4) can be equivalently stated in either of the following two forms [10]:

$$d(\mathbf{y}) \geq d(\mathbf{x}) + \langle \nabla d(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}\sigma\|\mathbf{x} - \mathbf{y}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in Q. \tag{5}$$

$$\langle \nabla d(\mathbf{x}) + \nabla d(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \sigma\|\mathbf{x} - \mathbf{y}\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in Q. \tag{6}$$

Assume $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ are prox-functions for the sets $\mathcal{X}$ and $\mathcal{Y}$ respectively. Then for any given $\mu > 0$, the smooth approximations $f_\mu \approx f$ and $\phi_\mu \approx \phi$ are

$$f_\mu(\mathbf{x}) := \max\{\langle \mathbf{x}, A\mathbf{y} \rangle - \mu d_{\mathcal{Y}}(\mathbf{y}) : \mathbf{y} \in \mathcal{Y}\}, \quad \phi_\mu(\mathbf{y}) := \min\{\langle \mathbf{x}, A\mathbf{y} \rangle + \mu d_{\mathcal{X}}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}.$$

The following result of Nesterov provides the theoretical foundation of our first-order algorithms for solving (1). Let $D_{\mathcal{X}} := \max\{d_{\mathcal{X}}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$, and let $\sigma_{\mathcal{X}}$ denote the strong convexity modulus of $d_{\mathcal{X}}$. Let $D_{\mathcal{Y}}$ and $\sigma_{\mathcal{Y}}$ be defined likewise for $\mathcal{Y}$ and $d_{\mathcal{Y}}$.

Theorem 2.2 (Nesterov [11, 12])**.** *There is a procedure based on the above smoothing technique that after N iterations generates a pair of points $(\mathbf{x}^N, \mathbf{y}^N) \in \mathcal{X} \times \mathcal{Y}$ such that*

$$0 \leq f(\mathbf{x}^N) - \phi(\mathbf{y}^N) \leq \frac{4\,\|A\|}{N+1}\sqrt{\frac{D_{\mathcal{X}} D_{\mathcal{Y}}}{\sigma_{\mathcal{X}}\sigma_{\mathcal{Y}}}}. \tag{7}$$

Furthermore, each iteration of the procedure performs some elementary operations, three matrix-vector multiplications by A, and requires the exact solution of three subproblems of the form

$$\max_{\mathbf{x} \in \mathcal{X}} \{\langle \mathbf{g}, \mathbf{x} \rangle - d_{\mathcal{X}}(\mathbf{x})\} \quad \textit{or} \quad \max_{\mathbf{y} \in \mathcal{Y}} \{\langle \mathbf{g}, \mathbf{y} \rangle - d_{\mathcal{Y}}(\mathbf{y})\}. \tag{8}$$

In Section 5, we will present an algorithm based on Theorem 2.2. Before that, we first provide a method for solving the subproblems in (8) as these are critical steps in the algorithm. These subproblems can be phrased in terms of the conjugate of the functions $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$. The conjugate of $d : Q \to \mathbb{R}$ is the function $d^* : \mathbb{R}^n \to \mathbb{R}$ defined by

$$d^*(\mathbf{s}) := \max\{\langle \mathbf{s}, \mathbf{x} \rangle - d(\mathbf{x}) : \mathbf{x} \in Q\}.$$

If d is strongly convex and Q is compact, then the conjugate d^* is differentiable everywhere and

$$\nabla d^*(\mathbf{s}) = \operatorname{argmax}\{\langle \mathbf{s}, \mathbf{x} \rangle - d(\mathbf{x}) : \mathbf{x} \in Q\}.$$

For an algorithm based on Theorem 2.2 to be practical, the subproblems (8) must be solvable quickly since they are solved three times at each iteration of the algorithm. In other words, the conjugates $d^*_{\mathcal{X}}$ and $d^*_{\mathcal{Y}}$ and their gradients $\nabla d^*_{\mathcal{X}}$ and $\nabla d^*_{\mathcal{Y}}$ should be easily computable. This motivates the following definition.

Definition 2.3. Assume $Q \subseteq \mathbb{R}^n$ is a compact convex set. We say that $d : Q \to \mathbb{R}$ is a *nice prox-function* for Q if it satisfies the following three conditions:

(i) d is continuous and strongly convex in Q, and differentiable in the relative interior of Q.
(ii) The conjugate d^* satisfies $d^*(\mathbf{0}) = 0$.
(iii) The conjugate function d^* and its gradient ∇d^* are easily computable.

Example 1. For the k-dimensional simplex Δ_k, the entropy function $d(x) = \ln k + \sum_{i=1}^k x_i \ln x_i$, and the Euclidean distance function $d(x) = \frac{1}{2}\sum_{i=1}^k (x_i - 1/k)^2$ are nice prox-functions. Indeed, for the entropy prox-function, the gradient of the conjugate $\nabla d^*(\mathbf{s})$ is given by the closed-form expression

$$\nabla_i d^*(\mathbf{s}) = \frac{e^{s_i}}{\sum_{j=1}^k e^{s_j}},\ i = 1, \ldots, k.$$

For the Euclidean prox-function, the gradient of the conjugate $\nabla d^*(\mathbf{s})$ is given by the expression

$$\nabla_i d^*(\mathbf{s}) = (s_i - \lambda)^+,\ i = 1, \ldots, k,$$

where $\lambda \in \mathbb{R}$ is such that $\sum_{j=1}^k (s_j - \lambda)^+ = 1$. This value of λ can be found in $\mathcal{O}(k \ln k)$ steps via a binary search in the sorted components of $\mathbf{s}$.

3. Complexes

This section presents the essential elements of our approach. We define the class of *complex* polytopes and provide a generic technique for constructing nice prox-functions for complexes, using as building blocks any family of nice prox-functions for simplexes. This allows us to create practical first-order algorithms based on Theorem 2.2 for solving the saddle-point problem (1) over complexes $\mathcal{X}$ and $\mathcal{Y}$.

A complex can be seen as a tree whose nodes are simplexes. The tree structure endows the complex with a certain kind of sequential feature. In particular, complexes include the types of polytopes that arise in the computation of Nash equilibria of sequential games. The latter is an immediate consequence of the *sequence form* formulation of Nash equilibria for sequential games, as detailed in [13, 14].

Definition 3.1. The class of complexes is recursively defined as follows:

- *Basic sets:* Every standard simplex $\Delta_m := \left\{\mathbf{x} \in [0,1]^m : \sum_{j=1}^m x_j = 1\right\}$ is a complex.
- *Cartesian product:* If $Q_1, \ldots, Q_k$ are complexes then $Q_1 \times \cdots \times Q_k$ is a complex.
- *Branching:* If $P \subseteq [0,1]^p$ and $Q \subseteq [0,1]^q$ are complexes and $i \in \{1, \ldots, p\}$ then

$$P \boxed{i} Q := \left\{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p+q} : \mathbf{x} \in P,\ \mathbf{y} \in x_i \cdot Q\right\}$$

  is a complex.

The Branching operation in Definition 3.1 has the following sequential interpretation: the vector $\mathbf{x}$ is the set of "current stage" decision variables, and the vector $\mathbf{y}$ is the set of "next stage" decision variables following the i-th current decision variable x_i. Notice that a complex can be written in the form $\{\mathbf{x} \geq 0 : E\mathbf{x} = \mathbf{e}\}$ for some matrix E with entries in $\{-1, 0, 1\}$ and vector $\mathbf{e}$ with entries in $\{0, 1\}$.

We now present our general procedure for constructing nice prox-functions for complexes. The construction relies on the following *dilation* operation from convex analysis [7]. Given a compact set $K \subseteq \mathbb{R}^d$ and a function $\Phi \colon K \to \mathbb{R}$, define the set $\bar{K} \subseteq \mathbb{R}^{d+1}$ as

$$\bar{K} := \left\{(x, \mathbf{y}) \in \mathbb{R}^{d+1} : x \in [0,1],\ \mathbf{y} \in x \cdot K\right\},$$

and define the function $\bar{\Phi}\colon \bar{K} \to \mathbb{R}$ as

$$\bar{\Phi}(x, \mathbf{y}) = \begin{cases} x \cdot \Phi\left(\frac{\mathbf{y}}{x}\right) & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

Proposition 3.2. *If K is compact and Φ is continuous in K, then $\bar{\Phi}$ is continuous in $\bar{K}$. Also if $(x, \mathbf{y}) \in \bar{K}$ is such that $x > 0$ and $\nabla\Phi(\mathbf{y}/x)$ exists, then $\nabla\bar{\Phi}(x, \mathbf{y})$ exists and*

$$\begin{aligned} \nabla_x \bar{\Phi}(x, \mathbf{y}) &= \Phi\left(\tfrac{\mathbf{y}}{x}\right) - \left\langle \nabla\Phi\left(\tfrac{\mathbf{y}}{x}\right), \tfrac{\mathbf{y}}{x} \right\rangle, \\ \nabla_{\mathbf{y}} \bar{\Phi}(x, \mathbf{y}) &= \nabla\Phi\left(\tfrac{\mathbf{y}}{x}\right). \end{aligned} \tag{9}$$

Proof. The continuity follows via a straightforward limiting argument: Assume $(x^i, \mathbf{y}^i), (x, \mathbf{y}) \in \bar{K}$ and $(x^i, \mathbf{y}^i) \to (x, \mathbf{y})$. If $x > 0$ then $\mathbf{y}^i/x_i, \mathbf{y}/x \in K$ and $\mathbf{y}^i/x_i \to \mathbf{y}/x$. Hence, since Φ is continuous, we get

$$\bar{\Phi}(x^i, \mathbf{y}^i) = \Phi(\mathbf{y}^i/x^i) \to \Phi(\mathbf{y}/x) = \bar{\Phi}(x, \mathbf{y}).$$

On the other hand, if $x = 0$ then $x^i \to 0$. It then follows that

$$|\bar{\Phi}(x^i, \mathbf{y}^i)| = |x^i \Phi(\mathbf{y}^i/x^i)| \leq x^i \max\{\Phi(\mathbf{z}) : \mathbf{z} \in K\} \to 0 = \bar{\Phi}(x, \mathbf{y}).$$

Finally, the identities in (9) follow by applying the chain rule. □

Assume we are given nice prox-functions d_m for Δ_m, $m \in \mathbb{Z}^+$. Using this family, we recursively construct functions for complexes as follows:

- *Basic sets:* For $Q = \Delta_m$, let $d_Q := d_m$.
- *Cartesian product:* If $Q_1, \ldots, Q_k$ are complexes and $Q = Q_1 \times \cdots \times Q_k$, let

$$d_Q(\mathbf{x}^1, \ldots, \mathbf{x}^k) := \sum_{i=1}^{k} d_{Q_i}(\mathbf{x}^i)$$

  where $d_{Q_1}, \ldots, d_{Q_k}$ are nice prox-functions for their respective complexes.
- *Branching:* If $P \subseteq [0,1]^p$ and $R \subseteq [0,1]^r$ are complexes, $i \in \{1, \ldots, p\}$, and $Q = P \boxed{i} R$, let

$$d_Q(\mathbf{x}, \mathbf{y}) := d_P(\mathbf{x}) + \bar{d}_R(x_i, \mathbf{y}) \tag{10}$$

  where d_P and d_R are nice prox-functions for P and R.

Theorem 3.3. *The functions d_Q defined above are nice prox-functions for each complex Q.*

To prove Theorem 3.3, it suffices to show that the properties of nice prox-functions are preserved for the Cartesian product and Branching steps. Since the Cartesian product step is straightforward, we concentrate on the Branching step as stated in the following proposition.

Proposition 3.4. *Assume $P \subseteq [0,1]^p$ and $R \subseteq [0,1]^r$ are complexes, $i \in \{1, \ldots, p\}$, and $Q = P \boxed{i} R$. Furthermore, assume d_P and d_R are nice prox-functions for P and R respectively and*

$$d_Q(\mathbf{x}, \mathbf{y}) := d_P(\mathbf{x}) + \bar{d}_R(x_i, \mathbf{y}).$$

Then

(i) *d_Q is continuous and strongly convex in Q and differentiable in the relative interior of Q.*

(ii) *d_Q^* and ∇d_Q^* are computable via the following expressions*

$$d_Q^*(\mathbf{u},\mathbf{v}) = d_P^*(\tilde{\mathbf{u}}) \tag{11}$$

$$\nabla d_Q^*(\mathbf{u},\mathbf{v}) = (\nabla d_P^*(\tilde{\mathbf{u}}), \nabla_i d_P^*(\tilde{\mathbf{u}}) \cdot \nabla d_R^*(\mathbf{v})) \tag{12}$$

where

$$\tilde{u}_j = \begin{cases} u_j & \text{if } j \neq i, \\ u_i + d_R^*(\mathbf{v}) & \text{if } j = i. \end{cases}$$

Proof.

(i) The continuity of d_Q in Q and the differentiability in the relative interior of Q follow from (10) and Proposition 3.2. Since d_Q is continuous in Q, to prove its strong convexity, from (6) it suffices to show that there exists $\sigma > 0$ such that

$$\langle \nabla d_Q(\mathbf{x},\mathbf{y}) - \nabla d_Q(\tilde{\mathbf{x}},\tilde{\mathbf{y}}), (\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}}) \rangle \geq \sigma \|(\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}})\|^2 \tag{13}$$

for all $(\mathbf{x},\mathbf{y})$ and $(\tilde{\mathbf{x}},\tilde{\mathbf{y}})$ in the relative interior of Q.

Assume $(\mathbf{x},\mathbf{y})$ and $(\tilde{\mathbf{x}},\tilde{\mathbf{y}})$ are in the relative interior of Q and set $\mathbf{z} := \mathbf{y}/x_i$ and $\tilde{\mathbf{z}} := \tilde{\mathbf{y}}/\tilde{x}_i$. From (10), Proposition 3.2, and some elementary calculations we get

$$\begin{aligned} \langle \nabla d_Q(\mathbf{x},\mathbf{y}) - \nabla d_Q(\tilde{\mathbf{x}},\tilde{\mathbf{y}}), (\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}}) \rangle &= \langle \nabla d_P(\mathbf{x}) - \nabla d_P(\tilde{\mathbf{x}}), \mathbf{x} - \tilde{\mathbf{x}} \rangle \\ &+ x_i \cdot (d_R(\mathbf{z}) - d_R(\tilde{\mathbf{z}}) + \langle \nabla d_R(\tilde{\mathbf{z}}), \tilde{\mathbf{z}} - \mathbf{z} \rangle) \\ &+ \tilde{x}_i \cdot (d_R(\tilde{\mathbf{z}}) - d_R(\mathbf{z}) + \langle \nabla d_R(\mathbf{z}), \mathbf{z} - \tilde{\mathbf{z}} \rangle). \end{aligned}$$

Therefore, since d_P and d_R are strongly convex, (5) yields

$$\begin{aligned} \langle \nabla d_Q(\mathbf{x},\mathbf{y}) - \nabla d_Q(\tilde{\mathbf{x}},\tilde{\mathbf{y}}), (\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}}) \rangle &\geq \sigma_P \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \tfrac{1}{2}\sigma_R x_i \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 + \tfrac{1}{2}\sigma_R \tilde{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 \\ &= \sigma_P \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_R \hat{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\|^2, \end{aligned} \tag{14}$$

where $\hat{x}_i = \frac{x_i + \tilde{x}_i}{2}$ and $\sigma_P, \sigma_R > 0$ are the strong convexity moduli of d_P and d_R respectively.

Next, we bound the right-hand side of (13). Applying the triangle inequality we get

$$\begin{aligned} \|(\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}})\| &\leq \|\mathbf{x} - \tilde{\mathbf{x}}\| + \|x_i \mathbf{z} - \tilde{x}_i \tilde{\mathbf{z}}\| \\ &= \|\mathbf{x} - \tilde{\mathbf{x}}\| + \|\tfrac{1}{2}(x_i + \tilde{x}_i)(\mathbf{z} - \tilde{\mathbf{z}}) + \tfrac{1}{2}(x_i - \tilde{x}_i)(\mathbf{z} + \tilde{\mathbf{z}})\| \\ &\leq \|\mathbf{x} - \tilde{\mathbf{x}}\| + \hat{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\| + \tfrac{1}{2}|x_i - \tilde{x}_i| \|\mathbf{z} + \tilde{\mathbf{z}}\|. \end{aligned} \tag{15}$$

Since R is compact, $M := \max\{\|\mathbf{z}\| : \mathbf{z} \in R\} < \infty$. Therefore from (15) we get

$$\|(\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}})\| \leq (1 + M)\|\mathbf{x} - \tilde{\mathbf{x}}\| + \hat{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\|.$$

Now, by the Cauchy-Schwarz inequality,

$$\|(\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}})\|^2 \leq \left((1 + M)^2 \frac{1}{\sigma_P} + \frac{\hat{x}_i}{\sigma_R} \right) \left(\sigma_P \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_R \hat{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 \right).$$

But $\hat{x}_i \in [0,1]$ because $\mathbf{x}, \tilde{\mathbf{x}} \in P \subseteq [0,1]^p$, so

$$\|(\mathbf{x},\mathbf{y}) - (\tilde{\mathbf{x}},\tilde{\mathbf{y}})\|^2 \leq \left((1 + M)^2 \frac{1}{\sigma_P} + \frac{1}{\sigma_R} \right) \left(\sigma_P \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_R \hat{x}_i \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 \right). \tag{16}$$

From (14) and (16) it follows that (13) holds for

$$\sigma = \frac{1}{\frac{(1+M)^2}{\sigma_P} + \frac{1}{\sigma_R}} > 0.$$

(ii) For a given vector $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{p+r}$ we have

$$\begin{aligned} d_Q^*(\mathbf{u},\mathbf{v}) &= \sup\{\langle(\mathbf{u},\mathbf{v}),(\mathbf{x},\mathbf{y})\rangle - d_Q(\mathbf{x},\mathbf{y}) : (\mathbf{x},\mathbf{y}) \in Q\} \\ &= \sup\{\langle\mathbf{u},\mathbf{x}\rangle + \langle\mathbf{v},\mathbf{y}\rangle - d_P(\mathbf{x}) - \bar{d}_R(x_i,\mathbf{y}) : \mathbf{x} \in P,\ \mathbf{y} \in x_i \cdot R\} \\ &= \sup\{\langle\mathbf{u},\mathbf{x}\rangle - d_P(\mathbf{x}) + x_i \cdot (\langle\mathbf{v},\mathbf{z}\rangle - d_R(\mathbf{z})) : \mathbf{x} \in P,\ \mathbf{z} \in R,\ x_i > 0\} \\ &= \sup\{\langle\mathbf{u},\mathbf{x}\rangle - d_P(\mathbf{x}) + x_i \cdot d_R^*(\mathbf{z}) : \mathbf{x} \in P\} \\ &= \sup\{\langle\tilde{\mathbf{u}},\mathbf{x}\rangle - d_P(\mathbf{x}) : \mathbf{x} \in P\} \\ &= d_P^*(\tilde{\mathbf{u}}). \end{aligned} \tag{17}$$

The third and fourth steps above hold by the continuity of $\bar{d}_R$ and d_P. Hence (11) is proven. To prove (12), observe that the maximizer in the second to last step in (17) is $\bar{\mathbf{x}} = \nabla d_P^*(\tilde{\mathbf{u}})$. Next, consider two cases depending on the value of $\bar{x}_i$. If $\bar{x}_i > 0$ then the maximizer in the third step in (17) is $\bar{\mathbf{z}} = \nabla d_R^*(\tilde{\mathbf{v}})$, and consequently the maximizer in the first step in (17) is $(\bar{\mathbf{x}}, \bar{x}_i \cdot \bar{\mathbf{z}})$. If $\bar{x}_i = 0$ then the maximizer in the first step in (17) is $(\bar{\mathbf{x}}, \mathbf{0})$. In either case the maximizer in the first step in (17) is $\nabla d_Q^*(\mathbf{u},\mathbf{v}) = (\bar{\mathbf{x}}, \bar{x}_i \cdot \bar{\mathbf{z}}) = (\nabla d_P^*(\tilde{\mathbf{u}}), \nabla_i d_P^*(\tilde{\mathbf{u}}) \cdot \nabla d_R^*(\mathbf{v}))$. □

Remark 3.5. We can generalize the above construction and results to weighted versions of the prox-functions. More precisely, in the Branching step, we could define $d_Q(\mathbf{x},\mathbf{y}) := w_P d_P(\mathbf{x}) + w_R \bar{d}_R(x_i,\mathbf{y})$ for some constants $w_P, w_R > 0$. We will elaborate on this idea to obtain prox-functions yielding better complexity guarantees for uniform complexes.

4. Uniform complexes

In this section we derive complexity results for first-order smoothing algorithms for the problem (1) in the special case when $\mathcal{X}$ and $\mathcal{Y}$ are *uniform* complexes. This special case of (1) covers the formulation of Nash equilibrium for instances of many interesting games. Indeed, as will be discussed in Section 5, uniform complexes naturally arise in multi-round sequential games such as poker.

Definition 4.1. Assume that a complex $Q \subseteq [0,1]^q$, an index set $I = \{i_1, \ldots, i_b\} \subseteq \{1, \ldots, q\}$, and a positive integer k are given. Define $Q_r,\ r = 1, 2, \ldots,$ as follows

- $Q_1 := Q \times \cdots \times Q$ (k times).
- $Q_{r+1} := \hat{Q}_r \times \cdots \times \hat{Q}_r$ (k times), where

$$\hat{Q}_r := Q \boxed{I} Q_r := \{(\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^b) : \mathbf{x} \in Q,\ \mathbf{y}^j \in x_{i_j} \cdot Q_r,\ j = 1, \ldots, b\}.$$

We will refer to Q_r as the r-th uniform complex generated by Q, I, k and will sometimes write it as $\mathcal{Q}(Q, I, k, r)$.

Remark 4.2. Notice that the operation $\boxed{I}$ is the same as the operation $\boxed{i}$ applied b times. More precisely,

$$Q \boxed{I} Q_r = Q \boxed{i_1} Q_r \boxed{i_2} \cdots \boxed{i_b} Q_r.$$

Given a nice prox-function d_Q for Q and constants $w_r > 0,\ r = 1, 2, \ldots,$ consider the following weighted version of our previous construction of prox-functions for complexes.

- For $Q_1 = Q \times \cdots \times Q$ (k times) let

$$d_{Q_1}(\mathbf{x}^1, \ldots, \mathbf{x}^k) := \sum_{j=1}^{k} d_Q(\mathbf{x}^j)$$

- For $Q_{r+1} = \hat{Q}_r \times \cdots \times \hat{Q}_r$ (k times), let

$$d_{Q_{r+1}}(\mathbf{u}^1, \ldots, \mathbf{u}^k) := \sum_{j=1}^{k} d_{\hat{Q}_r}(\mathbf{u}^j),$$

where $d_{\hat{Q}_r}$ is defined as follows

$$d_{\hat{Q}_r}(\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^b) := w_r \cdot d_Q(\mathbf{x}) + \sum_{j=1}^{b} \bar{d}_{Q_r}(x_{i_j}, \mathbf{y}).$$

We now present an explicit iteration complexity bound for a first-order smoothing algorithm for the saddle-point problem (1), when $\mathcal{X}$ and $\mathcal{Y}$ are uniform complexes.

Theorem 4.3. *Suppose A, $\mathcal{X}$, $\mathcal{Y}$, $d_{\mathcal{X}}$, and $d_{\mathcal{Y}}$ satisfy the following conditions:*

(i) $\mathcal{X} = \mathcal{Q}(Q, I, k, r) \subseteq \mathbb{R}^m$ *and* $\mathcal{Y} = \mathcal{Q}(\tilde{Q}, \tilde{I}, \tilde{k}, \tilde{r}) \subseteq \mathbb{R}^n$.

(ii) *The prox-functions $d_{\mathcal{X}}, d_{\mathcal{Y}}$ are constructed as above with weights $w_j = (kM)^2(bk)^j, j = 1, \ldots, r-1$ and $\tilde{w}_j = (\tilde{k}\tilde{M})^2(\tilde{b}\tilde{k})^j$, $j = 1, \ldots, \tilde{r} - 1$ respectively, where $b = |I|$, $\tilde{b} = |\tilde{I}|$, $M := \max\{\|\mathbf{u}\| : \mathbf{u} \in Q\}$, $\tilde{M} := \max\{\|\mathbf{u}\| : \mathbf{u} \in \tilde{Q}\}$.*

Then after N iterations the procedure from Theorem 2.2 yields $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ such that

$$0 \leq f(\mathbf{x}) - \phi(\mathbf{y}) = \max_{\mathbf{v} \in \mathcal{Y}} \langle \mathbf{v}, A\mathbf{x} \rangle - \min_{\mathbf{u} \in \mathcal{X}} \langle \mathbf{y}, A\mathbf{u} \rangle \leq \frac{4\|A\|G}{N+1} \sqrt{\frac{D_Q D_{\tilde{Q}}}{\sigma_Q \sigma_{\tilde{Q}}}}, \tag{18}$$

where $G = mn(kMr)(\tilde{k}\tilde{M}\tilde{r})$.

The crux of the proof of Theorem 4.3 is Lemma 4.4 below, which bounds the ratio of the maximum value to the strong convexity modulus for the prox-functions for uniform complexes. This ratio can be seen as a measure of the prox-function's quality. Lemma 4.4 provides an estimate of this ratio for the prox-functions d_{Q_r} constructed above, provided the weights w_r are chosen judiciously.

Lemma 4.4. *Assume Q and Q_r, $r = 1, 2, \ldots$, are as in Definition 4.1. Let σ, σ_r, D, D_r, and M be defined as follows*

$$\sigma := \text{strong convexity modulus of } d_Q, \ \sigma_r := \text{strong convexity modulus of } d_{Q_r},$$

$$D := \max\{d_Q(\mathbf{z}) : \mathbf{z} \in Q\}, \ D_r := \max\{d_{Q_r}(\mathbf{z}) : \mathbf{z} \in Q_r\},$$

$$M := \max\{\|\mathbf{z}\| : \mathbf{z} \in Q\}.$$

If $w_r = (kM)^2(bk)^r$, $r = 1, 2, \ldots$ then

$$\frac{D_r}{\sigma_r} \leq b^{2r-2}k^{2r+2}r^2M^2\frac{D}{\sigma}. \tag{19}$$

Proof of Theorem 4.3. Since $\mathcal{X} = \mathcal{Q}(Q, I, k, r) \subseteq \mathbb{R}^m$, Lemma 4.4 yields

$$\frac{D_{\mathcal{X}}}{\sigma_{\mathcal{X}}} \leq b^{2r-2}k^{2r+2}r^2M^2\frac{D_Q}{\sigma_Q}.$$

In addition, a simple induction argument shows the dimension m of $\mathcal{X} = \mathcal{Q}(Q, I, k, r)$ satisfies $m = kq \cdot \frac{(bk)^r-1}{bk-1}$. Therefore

$$\frac{D_{\mathcal{X}}}{\sigma_{\mathcal{X}}} \leq m^2k^2r^2M^2\frac{D_Q}{\sigma_Q}. \tag{20}$$

Similarly,

$$\frac{D_{\mathcal{Y}}}{\sigma_{\mathcal{Y}}} \leq n^2\tilde{k}^2\tilde{r}^2\tilde{M}^2\frac{D_{\tilde{Q}}}{\sigma_{\tilde{Q}}}. \tag{21}$$

The iteration bound (18) now follows from (7), (20), and (21). □

The proof of Lemma 4.4 in turn relies on the following technical lemma.

Lemma 4.5. *Let* $M_r := \max\{\|\mathbf{z}\| : \mathbf{z} \in Q_r\}$. *Then the strong convexity moduli* σ_r *of* d_{Q_r}, $r = 1, 2, \ldots$ *satisfy*

$$\sigma_{r+1} \geq \frac{1}{\frac{k(1+M_r)^2}{w_r\sigma} + \frac{bk}{\sigma_r}}. \tag{22}$$

Proof. Let $\hat{\sigma}_r$ be the strong convexity modulus of $d_{\hat{Q}_r}$. From the construction of d_{Q_r}, it follows that $\sigma_{r+1} \geq \hat{\sigma}_r/k$. Hence it suffices to bound $\hat{\sigma}_r$. Proceeding as in the proof of Proposition 3.4(i), it follows that for all $\mathbf{w} = (\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^b)$ and $\tilde{\mathbf{w}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}^1, \ldots, \tilde{\mathbf{y}}^b)$ in the relative interior of $\hat{Q}_r$ we have

$$\langle \nabla d_{\hat{Q}_r}(\mathbf{w}) - \nabla d_{\hat{Q}_r}(\tilde{\mathbf{w}}), \mathbf{w} - \tilde{\mathbf{w}} \rangle \geq w_r\sigma\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_r \sum_{j=1}^{b} \hat{x}_{i_j}\|\mathbf{z}^j - \tilde{\mathbf{z}}^j\|^2 \tag{23}$$

and

$$\|\mathbf{w} - \tilde{\mathbf{w}}\| \leq (1 + M_r)\|\mathbf{x} - \tilde{\mathbf{x}}\| + \sum_{j=1}^{b} \hat{x}_{i_j}\|\mathbf{z}^j - \tilde{\mathbf{z}}^j\|, \tag{24}$$

where $\mathbf{z}^j = \mathbf{y}^j/x_{i_j}$, and $\tilde{\mathbf{z}}^j = \tilde{\mathbf{y}}^j/\tilde{x}_{i_j}$ for $j = 1, \ldots, b$. Applying the Cauchy-Schwarz inequality to (24) we get

(25)

$$\begin{aligned}
\|\mathbf{w} - \tilde{\mathbf{w}}\|^2 &\leq \left(\frac{(1 + M_r)^2}{w_r\sigma} + \frac{\sum_{j=1}^{b} \hat{x}_{i_j}}{\sigma_r}\right)\left(w_r\sigma\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_r \sum_{j=1}^{b} \hat{x}_{i_j}\|\mathbf{z}^j - \tilde{\mathbf{z}}^j\|^2\right) \\
&\leq \left(\frac{(1 + M_r)^2}{w_r\sigma} + \frac{b}{\sigma_r}\right)\left(w_r\sigma\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sigma_r \sum_{j=1}^{b} \hat{x}_{i_j}\|\mathbf{z}^j - \tilde{\mathbf{z}}^j\|^2\right).
\end{aligned}$$

From (23), (25), and the continuity of $d_{\hat{Q}_r}$ we obtain

$$\hat{\sigma}_r \geq \frac{1}{\frac{(1+M_r)^2}{w_r\sigma} + \frac{b}{\sigma_r}},$$

which yields (22) since $\sigma_{r+1} \geq \hat{\sigma}_r/k$. □

Proof of Lemma 4.4. Let $M_r := \max\{\|\mathbf{z}\| : \mathbf{z} \in Q_r\}$. We have $M_1 = kM$ and $M_{r+1} \leq k(M + bM_r)$, so

$$1 + M_r \leq kM(bk)^r,\ r = 1, 2, \ldots.$$

Hence $w_r \geq \frac{(1+M_r)^2}{(bk)^r}$, and consequently (22) yields

$$\frac{1}{(bk)^{r+1}\sigma_{r+1}} \leq \frac{1}{b\sigma} + \frac{1}{(bk)^r\sigma_r}.$$

Therefore, since $\sigma_1 \geq \sigma/k$, it follows that

$$\frac{1}{(bk)^r\sigma_r} \leq \frac{r}{b\sigma},\ r = 1, 2, \ldots. \tag{26}$$

On the other hand, from the construction of Q_r and d_{Q_r} we have

$$D_1 \leq kD,\ D_{r+1} \leq k(w_r D + bD_r),\ r = 1, 2, \ldots$$

so,

$$D_r \leq kD\left((bk)^{r-1} + \sum_{j=1}^{r-1} w_j (bk)^{r-1-j}\right).$$

Thus

$$\begin{array}{rcl} D_r & \leq & kD\left((bk)^{r-1} + \sum_{j=1}^{r-1} w_j (bk)^{r-1-j}\right) \\ & = & kD\left((bk)^{r-1} + (kM)^2 \sum_{j=1}^{r-1} (bk)^j (bk)^{r-1-j}\right) \\ & = & kD(1 + (kM)^2(r-1))(bk)^{r-1} \\ & \leq & krD(kM)^2(bk)^{r-1}. \end{array} \tag{27}$$

Finally (19) follows by putting together (26) and (27). □

For the special case when the norm in $\mathbb{R}^q$ and each $\mathbb{R}^{q_r}$ is the Euclidean norm, we can sharpen the bound in Lemma 4.4, and thus also the bound in Theorem 4.3.

Lemma 4.6. *Assume b, M, D, D_r, σ, and σ_r, are as in Lemma 4.4, and the norm in $\mathbb{R}^q$ and each $\mathbb{R}^{q_r}$ is the Euclidean norm. If $w_r = kM^2k^r,\ r = 1, 2, \ldots$, then*

$$\frac{D_r}{\sigma_r} \leq b^{2r-2}k^{r+1}r^2M^2\frac{D}{\sigma}. \tag{28}$$

Proof. For the Euclidean norm we have $\sigma_{r+1} = \hat{\sigma}_r$, where $\hat{\sigma}_r$ is the strong convexity modulus of $d_{\hat{Q}_r}$. Next, we proceed to bound $\hat{\sigma}_r$ as in the proof of Lemma 4.5. For all $\mathbf{w} = (\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^b)$ and $\tilde{\mathbf{w}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}^1, \ldots, \tilde{\mathbf{y}}^b)$ in the relative interior of $\hat{Q}_r$ the inequality (23) holds. Next, instead of (24) we can use

$$\begin{array}{rcl} \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 & = & \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sum_{j=1}^{b} \|x_{i_j}\mathbf{z}^j - \tilde{x}_{i_j}\tilde{\mathbf{z}}^j\|^2 \\ & \leq & \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 + \sum_{j=1}^{b} \left(|x_{i_j} - \tilde{x}_{i_j}|M_r + \hat{x}_{i_j}\|\mathbf{z}^j - \tilde{\mathbf{z}}^j\|\right)^2. \end{array}$$

Hence, by the Cauchy-Schwarz inequality, we get

(29)
$$\begin{aligned}\|\mathbf{w}-\tilde{\mathbf{w}}\|^2 &\leq \|\mathbf{x}-\tilde{\mathbf{x}}\|^2+\left(\frac{M_r^2}{w_r\sigma}+\frac{b}{\sigma_r}\right)\left(w_r\sigma\|\mathbf{x}-\tilde{\mathbf{x}}\|^2+\sigma_r\sum_{j=1}^{b}\hat{x}_{i_j}\|\mathbf{z}^j-\tilde{\mathbf{z}}^j\|^2\right)\\ &\leq \left(\frac{(1+M_r^2)}{w_r\sigma}+\frac{b}{\sigma_r}\right)\left(w_r\sigma\|\mathbf{x}-\tilde{\mathbf{x}}\|^2+\sigma_r\sum_{j=1}^{b}\hat{x}_{i_j}\|\mathbf{z}^j-\tilde{\mathbf{z}}^j\|^2\right).\end{aligned}$$

Thus, the bound in Lemma 4.5 can be sharpened to

$$\sigma_{r+1}=\hat{\sigma}_r\geq\frac{1}{\frac{1+M_r^2}{w_r\sigma}+\frac{b}{\sigma_r}}. \tag{30}$$

Furthermore, in this case $M_1^2=kM^2$ and $M_{r+1}^2\leq k(M^2+bM_r^2)$ which implies

$$1+M_r^2\leq kM^2(bk)^r.$$

Hence $w_r\geq\frac{1+M_r^2}{b^r}$, and consequently (30) yields

$$\frac{1}{b^{r+1}\sigma_{r+1}}\leq\frac{1}{b\sigma}+\frac{1}{b^r\sigma_r}.$$

Therefore, since $\sigma_1=\sigma$, it follows that

$$\frac{1}{b^r\sigma_r}\leq\frac{r}{b\sigma},\ r=1,2,\ldots. \tag{31}$$

On the other hand, since $D_1=kD$ and $D_{r+1}\leq k(w_rD+bD_r)$, it follows that

$$\begin{aligned}D_r &\leq kD\left((bk)^{r-1}+\textstyle\sum_{j=1}^{r-1}w_j(bk)^{r-1-j}\right)\\ &= kD\left((bk)^{r-1}+kM^2\textstyle\sum_{j=1}^{r-1}k^j(bk)^{r-1-j}\right)\\ &\leq kD(1+kM^2(r-1))(bk)^{r-1}\\ &\leq k^2rDM^2(bk)^{r-1}.\end{aligned} \tag{32}$$

Finally (28) follows by putting together (31) and (32). □

We conclude this section with an estimate on the number of extreme points of the uniform complexes $\mathcal{Q}(Q,I,k,r)$. This highlights the intricate geometry of these polytopes.

Proposition 4.7. *Assume Q,I,b,k are as in Definition 4.1. For $r=1,2,\ldots,$ the number of extreme points of Q_r is at least v^{k^r}, where v is the number of extreme points of Q. When Q is a simplex, the number of extreme points of Q_r is $b^{\frac{k^r-k}{k-1}}v^{k^r}$.*

Proof. Let v_r and $\hat{v}_r$ denote the number of extreme points of Q_r and $\hat{Q}_r$ respectively. Observe that

$$v_1=v^k,\ v_{r+1}=\hat{v}_r^k\geq v_r^k,\ r=1,2,\ldots.$$

Thus $v_r\geq v^{k^r}$, $r=1,\ldots$.

When Q is a simplex, we have

$$v_1=v^k,\ v_{r+1}=\hat{v}_r^k=(bv_r)^k,\ r=1,2,\ldots.$$

Thus $v_r=b^{\frac{k^r-k}{k-1}}v^{k^r}$. □

5. Implementation and application

In this section we discuss our computational experience with our implementation of Nesterov's *excessive gap technique* [11]. Section 5.1 presents a description of this algorithm specialized for the problem (1). The prox-functions used in our implementation are the same as the ones designed in this paper. In Section 5.2 we briefly summarize some computational experiments, and in Section 5.3 we discuss how the prox-functions designed in this paper have enabled the application of Nesterov's excessive gap technique to solving real-world instances of poker.

5.1. **Nesterov's Excessive Gap Technique.** Assume $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ are nice prox functions for $\mathcal{X}$ and $\mathcal{Y}$ respectively. For $\mu_{\mathcal{X}}$, $\mu_{\mathcal{Y}} > 0$ consider the pair of problems:

$$f_{\mu_{\mathcal{Y}}}(\mathbf{x}) := \max\{\langle \mathbf{x}, A\mathbf{y}\rangle - \mu_{\mathcal{Y}} d_{\mathcal{Y}}(\mathbf{y}) : \mathbf{y} \in \mathcal{Y}\}, \quad \phi_{\mu_{\mathcal{X}}}(\mathbf{y}) := \min\{\langle \mathbf{x}, A\mathbf{y}\rangle + \mu_{\mathcal{X}} d_Q(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}.$$

The algorithm below generates iterates $(\mathbf{x}^k, \mathbf{y}^k, \mu_{\mathcal{X}}^k, \mu_{\mathcal{Y}}^k)$ with $\mu_{\mathcal{X}}^k$, $\mu_{\mathcal{Y}}^k$ decreasing to zero and such that the following *excessive gap condition* is satisfied at each iteration:

$$f_{\mu_{\mathcal{Y}}^k}(\mathbf{x}^k) \leq \phi_{\mu_{\mathcal{X}}^k}(\mathbf{y}^k). \tag{33}$$

Intuitively, for small values of $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$, $f_{\mu_{\mathcal{Y}}} \approx f$ and $\phi_{\mu_{\mathcal{X}}} \approx \phi$. Since $f(\mathbf{x}) \geq \phi(\mathbf{y})$ for all $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$ and the excessive gap condition (33) is maintained for all iterates $(\mathbf{x}^k, \mathbf{y}^k)$ it follows that $f(\mathbf{x}^k) \approx \phi(\mathbf{y}^k)$ when $\mu_{\mathcal{X}}^k$ and $\mu_{\mathcal{Y}}^k$ are small.

The detailed EGT (Excessive Gap Technique) algorithm is as follows.

EGT algorithm

Input: Nice prox functions $d_{\mathcal{X}}, d_{\mathcal{Y}}$ for $\mathcal{X} \subseteq \mathbb{R}^m, \mathcal{Y} \subseteq \mathbb{R}^n$ respectively, $A \in \mathbb{R}^{m \times n}$, and a positive integer N
Output: $\mathbf{x}^N \in \mathcal{X}, \mathbf{y}^N \in \mathcal{Y}$ such that

$$0 \leq f(\mathbf{x}^N) - \phi(\mathbf{y}^N) = \max_{\mathbf{v} \in \mathcal{Y}} \langle \mathbf{v}, A\mathbf{x}^N \rangle - \min_{\mathbf{u} \in \mathcal{X}} \langle \mathbf{y}^N, A\mathbf{u} \rangle \leq \frac{4\|A\|}{N+1} \sqrt{\frac{D_{\mathcal{X}} D_{\mathcal{Y}}}{\sigma_{\mathcal{X}} \sigma_{\mathcal{Y}}}}$$

(1) $\mu_{\mathcal{X}}^0 = \mu_{\mathcal{Y}}^0 = \frac{\|A\|}{\sqrt{\sigma_{\mathcal{X}} \sigma_{\mathcal{Y}}}}$
(2) $\hat{\mathbf{x}} = \nabla d_{\mathcal{X}}^*(\mathbf{0})$
(3) $\mathbf{y}^0 = \nabla d_{\mathcal{Y}}^* \left(\frac{1}{\mu_{\mathcal{Y}}^0} A\hat{\mathbf{x}} \right)$
(4) $\mathbf{x}^0 = \nabla d_{\mathcal{X}}^* \left(\nabla d_{\mathcal{X}} (\hat{\mathbf{x}}) + \frac{1}{\mu_{\mathcal{X}}^0} A^{\mathrm{T}} \mathbf{y}^0 \right)$
(5) `For` $k = 0, 1, \ldots, N$:
  (a) $\tau = \frac{2}{k+3}$
  (b) `If` k `is even:  /* Shrink` $\mu_{\mathcal{X}}$ `*/`
    (i) $\check{\mathbf{x}} = \nabla d_{\mathcal{X}}^* \left(-\frac{1}{\mu_{\mathcal{X}}^k} A^{\mathrm{T}} \mathbf{y}^k \right)$
    (ii) $\hat{\mathbf{x}} = (1-\tau)\mathbf{x}^k + \tau \check{\mathbf{x}}$
    (iii) $\hat{\mathbf{y}} = \nabla d_{\mathcal{Y}}^* \left(\frac{1}{\mu_{\mathcal{Y}}^k} A\hat{\mathbf{x}} \right)$
    (iv) $\tilde{\mathbf{x}} = \nabla d_{\mathcal{X}}^* \left(\nabla d_{\mathcal{X}} (\check{\mathbf{x}}) - \frac{\tau}{(1-\tau)\mu_{\mathcal{X}}^k} A^{\mathrm{T}} \hat{\mathbf{y}} \right)$
    (v) $\mathbf{y}^{k+1} = (1-\tau)\mathbf{y}^k + \tau \hat{\mathbf{y}}$
    (vi) $\mathbf{x}^{k+1} = (1-\tau)\mathbf{x}^k + \tau \tilde{\mathbf{x}}$

(vii) $\mu_{\mathcal{X}}^{k+1} = (1-\tau)\mu_{\mathcal{X}}^{k}$

(c) `If` k `is odd:  /* Shrink` $\mu_{\mathcal{Y}}$ `*/`

(i) $\check{\mathbf{y}} = \nabla d_{\mathcal{Y}}^{*}\left(\frac{1}{\mu_{\mathcal{Y}}^{k}} A\mathbf{x}^{k}\right)$

(ii) $\hat{\mathbf{y}} = (1-\tau)\mathbf{y}^{k} + \tau\check{\mathbf{y}}$

(iii) $\hat{\mathbf{x}} = \nabla d_{\mathcal{X}}^{*}\left(-\frac{1}{\mu_{\mathcal{X}}^{k}} A^{\mathrm{T}}\hat{\mathbf{y}}\right)$

(iv) $\tilde{\mathbf{y}} = \nabla d_{\mathcal{Y}}^{*}\left(\nabla d_{\mathcal{Y}}(\check{\mathbf{y}}) + \frac{\tau}{(1-\tau)\mu_{\mathcal{Y}}^{k}} A\hat{\mathbf{x}}\right)$

(v) $\mathbf{x}^{k+1} = (1-\tau)\mathbf{x}^{k} + \tau\hat{\mathbf{x}}$

(vi) $\mathbf{y}^{k+1} = (1-\tau)\mathbf{y}^{k} + \tau\tilde{\mathbf{y}}$

(vii) $\mu_{\mathcal{Y}}^{k+1} = (1-\tau)\mu_{\mathcal{Y}}^{k}$

5.2. **Computational experiments.** Here we summarize our computational experiments which were published in a separate paper [2]. We first conducted experiments comparing the performance of the EGT algorithm for the prox-functions induced by the entropy and Euclidean prox-functions on simplexes. Based on our experience with a variety of sequential games, we concluded that the use of the entropy prox-function consistently enabled the EGT algorithm to reach a given ϵ about an order of magnitude faster than was possible with the Euclidean prox-function.

We also developed two heuristics for speeding up the convergence of the algorithm in practice. (Although we were not able to improve on the worst-case theoretical complexity bounds given in this paper, we were able to substantially speed up the algorithm in practice, while maintaining the same theoretical worst-case guarantees.) The first heuristic attempts to drive the smoothing parameters $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ to zero at a rate faster (i.e., a larger τ) than suggested by the EGT algorithm, while still maintaining the excessive gap condition. The second heuristic periodically adjusts the $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ parameters to keep them approximately equal to each other, while also aggressively pushing them towards zero in tandem. Our experimental results, conducted on a range of sequential games, indicate that these heuristics provide a substantial speed-up of the basic algorithm, particularly when the game instance is large.

5.3. **Application to Texas Hold'em poker.** Poker is a game involving elements of chance, imperfect information, and counter-speculation. Game-theoretic optimal strategies are far from straightforward, often necessitating such tactics as bluffing and slow-playing. For these reasons, and others, poker has been identified as an important challenge problem for the field of Artificial Intelligence (AI) [1]. Just as the development of a computer program capable of beating the world's best human chess player was once seen as an important milestone, the development of an expert-level poker-playing program is now seen as an equally important milestone for AI.

The prox-function construction described in Section 3 has been instrumental in the development of some recent programs for playing Texas Hold'em poker. One parameter of Texas Hold'em is the *betting structure*. Two common betting structures are *limit*, in which players may bet a fixed amount, and *no-limit*, in which players may bet any number of their chips. Our equilibrium-finding algorithm computed the strategies for both *GS3* [4] and *Tartanian* [5], which play limit and no-limit Texas Hold'em, respectively.

In 2007 the Association for the Advancement of Artificial Intelligence (AAAI) held the second Computer Poker Competition. *GS3* placed third (out of 15) in the Limit Equilibrium competition and third (out of 17) in the Limit Online competition. *Tartanian* placed second (out of 10) in the no-limit competition. This is particularly impressive given the small amount of poker-specific knowledge that was incorporated into those programs. They instead depend on an equilibrium analysis (which in turn relied on our prox-function construction) for determining their strategies. As the developers of *GS3* and *Tartanian* point out, it is currently not feasible to solve their models using off-the-shelf linear programming solvers.

The approach used for constructing the above players is based on creating *lossy* abstractions of the original game. These lossy abstractions are smaller sequential games whose equilibria can be used to construct approximate equilibria for the original games. The larger the abstraction, the better the quality of the approximate equilibrium. For the limit competition, our implementation of the EGT algorithm solved an abstracted game whose payoff matrix was $10^8 \times 10^8$. For the no-limit competition, our algorithm solved a game with payoff matrix of size $10^7 \times 10^7$. The uniform complexes introduced in Section 4 provide a perfect framework for modeling limit Texas Hold'em poker. For this game, the complex $Q_{\mathcal{X}}$ for the first player is a uniform complex. The "basic" complex $Q \subseteq [0,1]^{14}$ has the linear description $Q = \{\mathbf{x} \in [0,1]^{14} : E\mathbf{x} = \mathbf{e}\}$ where

$$E := \begin{bmatrix} 1 & 1 & 1 & & & & & & & & & & & \\ & -1 & & 1 & 1 & 1 & & & & & & & & \\ & & & & & -1 & 1 & 1 & 1 & & & & & \\ & & -1 & & & & & & & 1 & 1 & 1 & & \\ & & & & & & & & & & & -1 & 1 & 1 \end{bmatrix}, \quad \mathbf{e} := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The fourteen columns of E represent the possible sequence of actions that the first player can take during each betting round of the game. Each row in E encodes a simplex over three actions: *fold*, *call*, and *raise*. (The last row only allows fold and call.) The set $I = \{2, 3, 5, 6, 8, 9, 11, 12, 14\}$ indexes the sequences that do not end with a fold. Texas Hold'em is played in four rounds so $r = 4$. Finally, the value of k depends on the quality of the abstraction. The abstractions in [4] range from $k = 6$ to $k = 40$ (the k is actually different in each round). The complex $Q_{\mathcal{Y}}$ for the second player is also a uniform complex with similar characteristics.

One particularly attractive feature of the EGT algorithm is the fact that the only operation performed on the matrix A is a matrix-vector product. This allows for an *implicit* representation of A. In the paper describing our implementation [2], we present an implicit representation for poker that is based on constructing A using Kronecker products of smaller matrices. Recall the Kronecker product applied to two matrices $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{p \times q}$, is

$$B \otimes C = \begin{bmatrix} b_{11}C & \cdots & b_{1n}C \\ \vdots & \ddots & \vdots \\ b_{m1}C & \cdots & b_{mn}C \end{bmatrix} \in \mathbb{R}^{mp \times nq}.$$

For limit Texas Hold'em, we can write the payoff matrix as

$$A := \begin{bmatrix} F_1 \otimes B_1 & & & \\ & F_2 \otimes B_2 & & \\ & & F_3 \otimes B_3 & \\ & & & F_4 \otimes B_4 + S \otimes W \end{bmatrix}.$$

The component matrices F_i, B_i, S, and W are small enough to be explictly represented whereas it is infeasible to explicitly represent A. For the games our algorithm is currently solving, that representation results in memory savings of more than three orders of magnitude, and enables the solution of game instances that would be impossible even to represent explicitly in memory.

6. Conclusions and future research

We developed first-order algorithms to approximate Nash equilibria of two-person zero-sum sequential games. We applied Nesterov's smoothing technique to the saddle-point formulation (1) of the Nash equilibrium problem. The heart of our approach is a construction of nice prox-functions for the complex polytopes in the saddle-point formulation. An implementation based on our approach has been successful is obtaining approximate equilibria for sequential games that are four orders of magnitude larger than what conventional computational approaches can handle.

In contrast to a direct first-order approach to solve the linear programming formulation of (1) such as that proposed in [8], our approach automatically yields algorithms that generate feasible strategies $\mathbf{x} \in \mathcal{X},\ \mathbf{y} \in \mathcal{Y}$ throughout execution. This is of crucial importance because points that violate the constraints defining the complexes $\mathcal{X}, \mathcal{Y}$ even so slightly are typically meaningless strategies. Furthermore, the linear programming formulation of (1) increases the dimension of the problem substantially since it requires a new variable for each constraint in the description of the complexes $\mathcal{X}, \mathcal{Y}$.

In addition to our first-order smoothing approach to the problem (1), it is conceivable that specialized versions of other algorithmic approaches may also lead to effective algorithms for solving the saddle-point problem (1). For example, a specialized interior-point algorithm could use an appropriately designed iterative method to solve the system of equations at each main iteration. No such approach has been successfully developed so far. These interesting alternative algorithmic approaches will be the subject of future research.

References

[1] D. Billings, L. Peña, J. Schaeffer, and D. Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, January 2002. Special Issue on Games, Computers and Artificial Intelligence.

[2] A. Gilpin, S. Hoda, J. Peña, and T. Sandholm. Gradient-based algorithms for finding Nash equilibria in extensive form games. In *Workshop on Internet and Network Economics (WINE-07)*, 2007.

[3] A Gilpin and T. Sandholm. Lossless abstraction of imperfect information games. *JACM: Journal of the ACM*, 54, 2007.

[4] A. Gilpin, T. Sandholm, and T. Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Vancouver, Canada, 2007.

[5] A. Gilpin, T. Sandholm, and T. Sørensen. A heads-up no-limit texas hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Estoril, Portugal, 2008.

[6] J.-L. Goffin. On the convergence rate of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.

[7] J. Hirriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag, Berlin, 2001.

[8] G. Lan, Z. Lu, and R. Monteiro. Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming, December 2006. Manuscript.

[9] Y. Nesterov. A method for unconstrained convex minimization problem with rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983. Translated to English as *Soviet Math. Docl.*

[10] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Kluwer Academic Publishers, 2004.

[11] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.

[12] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[13] B. von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14:220–246, 1996.

[14] B. von Stengel. Equilibrium computation for games in strategic and extensive form. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.