Master's Thesis

Convergence analysis and acceleration of the smoothing methods for solving extensive-form games

Guidance

Professor   Nobuo Yamashita
Associate Professor   Ellen Hidemi Fukuda

Keigo Habara

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2023

Abstract

The extensive-form game has been studied considerably in recent years. It can represent games with multiple decision points and incomplete information, and hence it is helpful in formulating games with uncertain inputs, such as poker. We consider an extended-form game with two players and zero-sum, i.e., the sum of their payoffs is always zero. In such games, the problem of finding the optimal strategy can be formulated as a bilinear saddle-point problem. This formulation grows huge depending on the size of the game, since it has variables representing the strategies at all decision points for each player. To solve such large-scale bilinear saddle-point problems, the excessive gap technique (EGT), a smoothing method, has been studied. This method generates a sequence of approximate solutions whose error is guaranteed to converge at $O(1/k)$, where k is the number of iterations. However, it has the disadvantage of having poor theoretical bounds on the error related to the game size. This makes it inapplicable to large games.

Our goal is to improve the smoothing method for solving extensive-form games so that it can be applied to large-scale games. To this end, we make two contributions in this work. First, we slightly modify the strongly convex function used in the smoothing method in order to improve the theoretical bounds related to the game size. Second, we propose a heuristic called centering trick, which allows the smoothing method to be combined with other methods and consequently accelerates the convergence in practice. As a result, we combine EGT with CFR+, a state-of-the-art method for extensive-form games, to achieve good performance in games where conventional smoothing methods do not perform well. The proposed smoothing method is shown to have the potential to solve large games in practice.

Contents

1 Introduction

This study concerns improving the smoothing method for solving extensive-form games with imperfect information. An *extensive-form game* is a description of a game played by multiple players, where the game state corresponds to a node in a rooted tree. The state of the game moves down the tree according to the actions of each player, and each player receives a gain when a leaf of the tree is reached. This game is a good model for games with information gaps between players, such as poker, because it can represent the incomplete information. We only deal with games in which there are two players, and the sum of their payoff is zero. It is known that in such two-person zero-sum games, the problem of finding the optimal strategy can be formulated as a simple minimax problem since the players do not cooperate with each other.

There are two well-known methods for solving extensive-form games: *counterfactual regret minimization* (CFR) [1] and *excessive gap technique* (EGT) [2]. CFR is an application of regret minimization, a framework used in online learning, to extensive-form games. Its variant, CFR+ [3], was developed to solve a variant of a two-player poker game called Texas Hold'em, which was a challenging problem in artificial intelligence [4, 5]. CFR+ is suited for analyzing large games because the solution error is bounded linearly with the game size. However, it is only guaranteed to converge at $O\left(1/\sqrt{k}\right)$ rate, where k is the number of iterations. EGT is a smoothing method for the bilinear saddle-point problem, and its application to extensive-form games has been studied [6]. EGT is theoretically guaranteed to converge with rate $O(1/k)$, which is faster than CFR+, but it is not suitable for solving large games. This is because the term D/σ, which appears in the error bound, depends poorly on the game size. Here σ is the strong convexity parameter of a function d called *prox-function* used in EGT and $D := \max_{\boldsymbol{x}} d(\boldsymbol{x})$. Various prox-functions have been proposed in the literature [6, 7, 8], with D/σ being bounded by a cubic order of the game size.

Our goal is to improve the prox-function so that EGT can be applied to large extensive-form games. To this purpose, we make two contributions. First, we improve the bound of D/σ from $M_Q^3 \ln \max_{I \in \mathcal{I}} |\mathcal{A}(I)|$ to $M_Q^2 \ln |\Sigma|$ by eliminating the first-order term of the prox-function proposed in [8]. Here M_Q is the maximum value of the L_1-norm among the feasible points of the bilinear saddle-point problem, and $|\Sigma|$ is the dimension of the feasible set, both of which are bounded by the game size. Also, $\max_{I \in \mathcal{I}} |\mathcal{A}(I)|$ is the maximum number of legal actions at each decision point. Second, we propose a heuristic in EGT that we call the centering trick. This trick modifies the prox-function in a way that it takes the minimum value in a temporary solution. It is expected to improve the accuracy of the smooth approximation of the bilinear saddle-point problem inside the EGT. This heuristic accelerates convergence in practice and can be combined with other methods by using their solutions. Numerical experiments show that EGT with the heuristic combined with CFR+ performs best among several methods, including CFR+, for games of a scale where conventional EGT does not perform well. This suggests that the proposed smoothing method is effective even for large games.

The structure of this paper is given as follows. Section 2 introduces the basics of convexity analysis. Section 3 presents the bilinear saddle-point problem, the prox-function, and the smoothing method (EGT). Section 4 introduces the extensive-form games and explains how to transform the problem of finding the optimal strategy of the game into a problem covered by

1

EGT. In Section 5, we propose a prox-function, and we will show that it improves theoretical convergence. Section 6 provides the centering heuristic that accelerates the convergence of EGT in practice and confirms its performance through numerical experiments. Section 7 summarizes our contributions.

2   Preliminaries

This section introduces the basic properties of the convex analysis used in this paper. Let us denote $\mathbb{R}^n$ as the n-dimentional Euclidean space and $\mathrm{ri}S$ as the relative interior of $S \subset \mathbb{R}^n$. We will denote $\boldsymbol{x}^\top$ and $\boldsymbol{A}^\top$ as the transpose of the n-dimentional vector $\boldsymbol{x} \in \mathbb{R}^n$ and the $n \times m$ matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$. For a convex compact set $S \subset \mathbb{R}^n$, let us define the conjugate function $f^* \colon \mathbb{R}^n \to \mathbb{R}$ of a function $f \colon S \to \mathbb{R}$ by

$$f^*(\boldsymbol{\xi}) := \max_{\boldsymbol{x} \in S} \left\{ \boldsymbol{\xi}^\top \boldsymbol{x} - f(\boldsymbol{x}) \right\}.$$

If the above maximizer is unique, then by Danskin's theorem, f^* is differentiable at $\boldsymbol{\xi} \in \mathbb{R}^n$, and its derivative is given by

$$\nabla f^*(\boldsymbol{\xi}) = \arg\max_{\boldsymbol{x} \in S} \left\{ \boldsymbol{\xi}^\top \boldsymbol{x} - f(\boldsymbol{x}) \right\}.$$

3   Smoothing methods

In this section, the bilinear saddle-point problem and the prox-function are explained, followed by an overview of EGT.

3.1   Bilinear saddle-point problems

The bilinear saddle-point problems (BSPPs), which EGT covers, are written in the following form:

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

where $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ is a matrix and $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^m$ are convex and compact sets. The *adjoint form* of the problem is given by

$$\max_{\boldsymbol{y} \in \mathcal{Y}} \min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}. \tag{2}$$

By Minimax Theorem, these two problems achieve the same optimal value. In other words, the following equation holds for the optimal solution $\boldsymbol{x}^* \in \mathcal{X}$ and $\boldsymbol{y}^* \in \mathcal{Y}$ of (1) and (2):

$$\max_{\boldsymbol{y} \in \mathcal{Y}} (\boldsymbol{x}^*)^\top \boldsymbol{A} \boldsymbol{y} = \min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}^*.$$

The error of the pair of the solutions $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ can be defined by

$$\varepsilon(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) := \max_{\boldsymbol{y} \in \mathcal{Y}} \bar{\boldsymbol{x}}^\top \boldsymbol{A} \boldsymbol{y} - \min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{x}^\top \boldsymbol{A} \bar{\boldsymbol{y}} \geq 0, \tag{3}$$

and $\varepsilon(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = 0$ if and only if $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ are the optimal solutions of problems (1) and (2).

3.2 Prox-functions

BSPPs (1), the problems covered by EGT, can be regarded as non-smooth minimization problems because $f(x) := \max_{y \in \mathcal{Y}} x^\top A y$ is generally non-smooth. One way to f is to consider a *prox-function* which we define below.

Definition 3.1. *A function* $d\colon S \to \mathbb{R}$ *is called a prox-function on a convex compact set* $S \subset \mathbb{R}^n$ *when satisfying the following conditions:*

- *d is twice differentiable in* $\mathrm{ri}S$;

- *d is σ-strongly convex in* $\mathrm{ri}S$ *with respect to some norm* $\|\cdot\|_S$ *defined on* $\mathbb{R}^n$, *i.e.,*

$$\xi^\top \nabla^2 d(x)\xi \geq \sigma\|\xi\|_S^2 \quad \forall x \in \mathrm{ri}S, \forall \xi \in \mathbb{R}^n;$$

- $\min_{x \in S} d(x) = 0.$

For some prox-function $d_2\colon \mathcal{Y} \to \mathbb{R}$ and parameter $\mu_2 > 0$, define the *smoothing* of f as

$$f_{\mu_2}(x) := \max_{y \in \mathcal{Y}} \left\{ x^\top A y - \mu_2 d_2(y) \right\} \tag{4}$$
$$= \mu_2 d_2^* \left(A^\top x / \mu_2 \right).$$

Since d_2 is strongly convex, the maximizer of (4) is unique, then d_2^* is differentiable, which in turn shows that f_{μ_2} is also differentiable. Let $D_2 := \max_{y \in \mathcal{Y}} d_2(y)$. By the definition of f_{μ_2}, we have

$$f_{\mu_2}(x) \leq f(x) \leq f_{\mu_2}(x) + \mu_2 D_2 \quad \forall x \in \mathcal{X}, \tag{5}$$

then f_{μ_2} is a good smooth approximation of f for small $\mu_2 > 0$.

3.3 Excessive gap technique

This section provides an overview of EGT. Let $\phi(y) := \min_{x \in \mathcal{X}} x^\top A y$. For some prox-function $d_1\colon \mathcal{X} \to \mathbb{R}$ and $\mu_1 > 0$, define the smooth approximation of ϕ similarly to (4):

$$\phi_{\mu_1}(y) := \min_{x \in \mathcal{X}} \left\{ x^\top A y + \mu_1 d_1(x) \right\},$$

and we also have

$$\phi_{\mu_1}(y) - \mu_1 D_1 \leq \phi(y) \quad \forall y \in \mathcal{Y}, \tag{6}$$

where $D_1 := \max_{x \in \mathcal{X}} d_1(x)$. Here we consider the following condition, called *excessive gap condition*:

$$f_{\mu_2}(x) \leq \phi_{\mu_1}(y). \tag{7}$$

3

If $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ and $\mu_1, \mu_2 > 0$ satisfy the excessive gap condition, the error of $(\boldsymbol{x}, \boldsymbol{y})$, defined in (3), is bounded by

$$
\begin{aligned}
\varepsilon(\boldsymbol{x}, \boldsymbol{y}) &= f(\boldsymbol{x}) - \phi(\boldsymbol{y}) \\
&\leq f_{\mu_2}(\boldsymbol{x}) - \phi_{\mu_1}(\boldsymbol{y}) + \mu_1 D_1 + \mu_2 D_2 \\
&\leq \mu_1 D_1 + \mu_2 D_2,
\end{aligned}
$$

where the first inequality follows from (5) and (6), and the second inequality is given by the excessive gap condition (7). The central concept of EGT is to find $(\boldsymbol{x}, \boldsymbol{y})$ such that the excessive gap condition is satisfied for small $\mu_1, \mu_2 > 0$ in order to reduce $\varepsilon(\boldsymbol{x}, \boldsymbol{y})$. To achieve this, the EGT algorithm consists of two parts: *initialization* and *shrinking*.

- The *initialization* part: Algorithm 1 requires $\mu_1, \mu_2 > 0$ satisfying $\mu_1 \mu_2 \geq \|A\|^2/(\sigma_1 \sigma_2)$, where

$$
\|A\| := \max_{\|\boldsymbol{x}\|_{\mathcal{X}}=1} \max_{\|\boldsymbol{y}\|_{\mathcal{Y}}=1} \boldsymbol{x}^\top A \boldsymbol{y}
$$

  and d_1, d_2 must be σ_1, σ_2-strongly convex with respect to some norm $\|\cdot\|_{\mathcal{X}}, \|\cdot\|_{\mathcal{Y}}$, respectively. Then it generates $(\boldsymbol{x}^0, \boldsymbol{y}^0)$ satisfying the excessive gap condition with its input μ_1, μ_2.

- The *shrinking* part: Algorithm 2 requires $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ and $\mu_1, \mu_2 > 0$ satisfying the excessive gap condition, and the step size $\tau \in (0, 1)$ with $\tau^2/(1 - \tau) \leq \sigma_1 \sigma_2 \mu_1 \mu_2 / \|A\|^2$. Then it generates $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1})$ satisfying the excessive gap condition with $(1 - \tau)\mu_1, \mu_2$, i.e. we can shrink μ_1 by a factor of $1 - \tau$. The shrinking algorithm of the μ_2 version can be obtained similarly.

By using Algorithms 1 and 2 with the parameters and choices between μ_1 and μ_2 to shrink presented in [2], the solution $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ generated by EGT guarantees the following convergence result [2, Theorem 6.3]:

$$
\varepsilon(\boldsymbol{x}^k, \boldsymbol{y}^k) \leq \frac{4\|A\|}{k+1} \sqrt{\frac{D_1 D_2}{\sigma_1 \sigma_2}}. \tag{8}
$$

Therefore, a small $\|A\|$ and large values for $\sigma_1/D_1, \sigma_2/D_2$ which we call *substantial strongly convexity* of d_1, d_2, are required for better convergence, with some norm $\|\cdot\|_{\mathcal{X}}, \|\cdot\|_{\mathcal{Y}}$. In this paper, we only consider the L_1-norm, which is desirable because $\|A\| = \max_{i,j} |A_{ij}|$ does not depend on the dimensions of $\mathcal{X}$ and $\mathcal{Y}$. In practice, we also require that ∇d_1^* and ∇d_2^* are easily computable.

Algorithm 1 Initialization

Input: $\mu_1, \mu_2 > 0$ satisfying $\mu_1 \mu_2 \geq \|A\|^2/(\sigma_1 \sigma_2)$
Output: $\boldsymbol{x}^0, \boldsymbol{y}^0$ satisfying the excessive gap condition (7) with μ_1, μ_2
  1: $\bar{\boldsymbol{x}} \leftarrow \nabla d_1^*(\boldsymbol{0})$
  2: $\boldsymbol{y}^0 \leftarrow \nabla d_2^*(A^\top \bar{\boldsymbol{x}} / \mu_2)$
  3: $\boldsymbol{x}^0 \leftarrow \nabla d_1^*(\nabla d_1(\bar{\boldsymbol{x}}) - A\boldsymbol{y}^0 / \mu_1)$

4

Algorithm 2 Shrinking (μ_1 version)

Input:
  x^k, y^k, μ_1, μ_2 satisfying the excessive gap condition (7)
  $\tau \in (0, 1)$ satisfying $\tau^2/(1 - \tau) \le \sigma_1 \sigma_2 \mu_1 \mu_2 / \|A\|^2$
Output:
  x^{k+1}, y^{k+1} satisfying the excessive gap condition with $(1 - \tau)\mu_1, \mu_2$
1: $\bar{x} \leftarrow \nabla d_1^*(-Ay^k/\mu_1)$
2: $\bar{y} \leftarrow \nabla d_2^*(A^\top((1 - \tau)x^k + \tau\bar{x})/\mu_2)$
3: $y^{k+1} \leftarrow (1 - \tau)y^k + \tau\bar{y}$
4: $x^{k+1} \leftarrow (1 - \tau)x^k + \tau\nabla d_1^*(\nabla d_1(\bar{x}) - \frac{\tau}{(1-\tau)\mu_1}A\bar{y})$

4 Extensive-form games

An extensive-form game is a game played by N players, which can be represented with a rooted tree. A state of the game corresponds to a node of the tree. Moving down the tree can be done either with the player's actions or stochastic events (like dealing cards, etc.) and hereafter, such stochastic events will be considered as *chance player*'s actions. The game ends when a leaf of the tree is reached, and player $i = 1, \ldots, N$ receives a gain $u_i(z)$ corresponding to the terminal node z. This paper only deals with two-person zero-sum games. That is $N = 2$ and $u := -u_1 = u_2$, where u is a loss for player 1 and a gain for player 2.

Let H_i denote the set of nodes where player $i = 1, 2$ acts. Partitions of H_i describe the imperfect information of the game. Such a partition $\mathcal{I}_i$ is called an *information partition*, and each element $I \in \mathcal{I}_i$ (i.e. $I \subset H_i$) is called an *information set*. Player $i = 1, 2$ cannot distinguish between nodes belonging to the same information set. The information partition must satisfy the following natural constraint: all nodes belonging to the same information set must have equal sets of legal actions. Also, in this paper, we only consider games satisfying *perfect recall*, i.e., the information partition is consistent with the assumption that each player can remember their own past actions. See Figures 2 and 5 in Appendix A for an extensive-form game representation of Kuhn poker and Leduc Hold'em, simplified versions of Texas Hold'em.

Assume that each player $i = 1, 2$ can choose their actions probabilistically at each information set. Let $\pi_i(z)$ be the contribution of player $i = 1, 2, c$ to the probability of reaching the terminal node z from the root, where c means the chance player. Let Z be the set of terminal nodes, then the expected value of u is given by

$$\sum_{z \in Z} \pi_1(z)\pi_2(z)\pi_c(z)u(z).$$

Each player $i = 1, 2$ aims to make this expectation smaller and larger by controlling π_1 and π_2, respectively. Note that π_c is constant.

Now consider the feasible region of π_i. Let

$$\Sigma_i := \{\varnothing\} \cup \{(I, a) \mid I \in \mathcal{I}_i, a \in \mathcal{A}(I)\},$$

where $\mathcal{A}(I) \ne \emptyset$ is the set of legal actions at information set I. Furthermore, by the assumption of *perfect recall*, we can define the *parent function* $p_i \colon \mathcal{I}_i \to \Sigma_i$ such that $p_i(I) = (I', a)$ if and

only if $I' \in \mathcal{I}_i$ is the last information set visited before $I \in \mathcal{I}_i$ and the action $a \in \mathcal{A}(I')$ is chosen there, and $p_i(I) = \varnothing$ if and only if there is no player i's information set that comes before $I \in \mathcal{I}_i$. We see that the function p_i forms a tree. That is, for a set of vertices $\Sigma_i \cup \mathcal{I}_i$, the graph with edges from $p_i(I)$ to I for all $I \in \mathcal{I}_i$ and from I to (I, a) for all $I \in \mathcal{I}_i$ and $a \in \mathcal{A}(I)$ is a tree rooted at $\varnothing$. See Figures 3, 4, and 6 for this *information trees* in each game.

Then let us consider the following convex compact set:

$$Q_i := \left\{ \boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|\Sigma_i|} \,\middle|\, x_\varnothing = 1, \ x_{p_i(I)} = \sum_{a \in \mathcal{A}(I)} x_{I,a} \ \forall I \in \mathcal{I}_i \right\}, \tag{9}$$

which we call the *strategy set* for player i. Consider mapping $\boldsymbol{x} \in Q_i$ to a probabilistic strategy that chooses action $a \in \mathcal{A}(I)$ at information set $I \in \mathcal{I}_i$ with the following probability:

$$\begin{cases} x_{I,a}/x_{p_i(I)}, & \text{if } x_{p_i(I)} \neq 0, \\ 1/|\mathcal{A}(I)|, & \text{otherwise}, \end{cases}$$

which is a sufficient representation of the player i's probabilistic strategy. Furthermore, the contribution $\pi_i(z)$ is given by $x_{p_i(z)}$ where $p_i(z)$ is a natural extension of the parent function. That is, $p_i(z) = (I, a)$ if and only if $I \in \mathcal{I}_i$ is the last information set visited before $z \in Z$ and the action $a \in \mathcal{A}(I)$ is chosen there, and $p_i(z) = \varnothing$ if and only if there is no player i's information set before $z \in Z$.

Therefore, extensive-form games can be written as the following BSPP:

$$\min_{\boldsymbol{x} \in Q_1} \max_{\boldsymbol{y} \in Q_2} \sum_{z \in Z} x_{p_1(z)} y_{p_2(z)} \pi_c(z) u(z).$$

As can be seen from this formulation, the matrix A in BSPP (1), which represents an extensive-form game, has only at most $|\Sigma|$ non-zero elements. In other words, A is sparse in most cases.

5 Prox-function over the strategy set

For solving extensive-form games with EGT, we need a prox-function defined on the convex compact set Q_i, defined by (9), the strategy set of each player $i = 1, 2$. To guarantee good convergence, it is necessary to define a prox-function $d_i : Q_i \to \mathbb{R}$ with large *substantial strong convexity* σ_i/D_i.

For simplicity, we omit i denoting the player in this section. Let $M_Q := \max_{\boldsymbol{x} \in Q} \|\boldsymbol{x}\|_1$, which represents the scale of the game. The prox-function proposed in a previous study [8] is shown to be $1/(M_Q^3 \ln \max_{I \in \mathcal{I}} |\mathcal{A}(I)|)$-strongly convex substantially with respect to the L_1-norm. We propose a slightly modified version of this prox-function and show that it is $1/(M_Q^2 \ln |\Sigma|)$-strongly convex substantially with respect to the L_1-norm, which is a better guarantee for most games.

Now we propose the prox-function $d : Q \to \mathbb{R}$ defined by

$$d(\boldsymbol{x}) := x_\varnothing \ln x_\varnothing + \sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} \left(w_I - \sum_{p(I') = (I,a)} w_{I'} \right) x_{I,a} \ln x_{I,a}, \tag{10}$$

6

where $w_I \in \mathbb{R}$ is defined recursively:

$$w_I := 1 + \max_{a \in \mathcal{A}(I)} \sum_{p(I')=(I,a)} w_{I'} \quad \forall I \in \mathcal{I}, \tag{11}$$

and its base case is given by $I \in \mathcal{I}$ with $\{I' \in \mathcal{I} \mid p(I') = (I,a)\} = \emptyset$ for all $a \in \mathcal{A}(I)$. We will denote $\ln x$ as the natural logarithm of $x \geq 0$ and assume $0 \ln 0 = 0$. Note that (10) does not satisfy $\min_{x \in Q} d(x) = 0$, the third condition for prox-function (see Definition 3.1), so $d - \min_{x \in Q} d(x)$ must be used instead. For simplicity, however, we will treat d as a prox-function in the following. This is because the additional constant term is not essential; it does not affect the strong convexity or ∇d^* and only shifts d^* by a constant.

Note that the prox-function proposed in [8] is given by

$$d(x) + \sum_{I \in \mathcal{I}} w_I x_{p(I)} \ln |\mathcal{A}(I)|, \tag{12}$$

which is shown to have a minimum value zero. We have eliminated the first-order term in (12) by neglecting the adjustment of the minimum to zero. As a result, we succeeded in giving a better theoretical guarantee of substantial strong convexity.

Theorem 5.1. *The prox-function* (10) *is* $1/M_Q$-*strongly convex with respect to the L_1-norm.*

Proof. This proof is the same as the proof of [8, Theorem 5]. For $\xi \in \mathbb{R}^{|\Sigma|}$ and $x \in \mathrm{ri}Q$, we have

$$\begin{aligned}
\xi^\top \nabla^2 d(x)\xi &= \frac{(\xi_\emptyset)^2}{x_\emptyset} + \sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} \left(w_I - \sum_{p(I')=(I,a)} w_{I'} \right) \frac{(\xi_{I,a})^2}{x_{I,a}} \\
&\geq \frac{(\xi_\emptyset)^2}{x_\emptyset} + \sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} \frac{(\xi_{I,a})^2}{x_{I,a}} \\
&= \sum_{j \in \Sigma} \frac{(\xi_j)^2}{x_j} \\
&\geq \frac{\left(\sum_{j \in \Sigma} |\xi_j| \right)^2}{\sum_{j \in \Sigma} x_j} \\
&= \frac{\|\xi\|_1^2}{\|x\|_1} \geq \frac{\|\xi\|_1^2}{M_Q},
\end{aligned}$$

where the first equality follows from (10), the second inequality comes from (11), and the fourth inequality is true from the Cauchy-Schwarz inequality:

$$\sum_{j \in \Sigma} \left(\sqrt{x_j} \right)^2 \cdot \sum_{j \in \Sigma} \left(\frac{|\xi_j|}{\sqrt{x_j}} \right)^2 \geq \left(\sum_{j \in \Sigma} \sqrt{x_j} \cdot \frac{|\xi_j|}{\sqrt{x_j}} \right)^2.$$

$\square$

To consider the properties of the conjugate function of the prox-function d, we present the following corollary.

Lemma 5.2. *The prox-function* (10) *satisfies the following equation for* $x \in \mathrm{ri}Q$:

$$d(x) = \sum_{I \in \mathcal{I}} w_I x_{p(I)} \sum_{a \in \mathcal{A}(I)} \frac{x_{I,a}}{x_{p(I)}} \ln \frac{x_{I,a}}{x_{p(I)}}.$$

Proof. First, note that $x_j \neq 0$ for $j \in \Sigma$ for $x \in \mathrm{ri}Q$. Then for $x \in \mathrm{ri}Q$, we have

$$
\begin{aligned}
d(x) &= \sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} \left(w_I - \sum_{p(I')=(I,a)} w_{I'} \right) x_{I,a} \ln x_{I,a} \\
&= \left(\sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} w_I x_{I,a} \ln x_{I,a} \right) - \sum_{p(I') \neq \varnothing} w_{I'} x_{p(I')} \ln x_{p(I')} \\
&= \sum_{I \in \mathcal{I}} \sum_{a \in \mathcal{A}(I)} w_I x_{I,a} \ln x_{I,a} - \sum_{I \in \mathcal{I}} w_I x_{p(I)} \ln x_{p(I)} \\
&= \sum_{I \in \mathcal{I}} w_I \left\{ \left(\sum_{a \in \mathcal{A}(I)} x_{I,a} \ln x_{I,a} \right) - x_{p(I)} \ln x_{p(I)} \right\} \\
&= \sum_{I \in \mathcal{I}} w_I \left\{ \sum_{a \in \mathcal{A}(I)} x_{I,a} \ln x_{I,a} - \sum_{a \in \mathcal{A}(I)} x_{I,a} \ln x_{p(I)} \right\} \\
&= \sum_{I \in \mathcal{I}} w_I \sum_{a \in \mathcal{A}(I)} x_{I,a} \ln \frac{x_{I,a}}{x_{p(I)}} \\
&= \sum_{I \in \mathcal{I}} w_I x_{p(I)} \sum_{a \in \mathcal{A}(I)} \frac{x_{I,a}}{x_{p(I)}} \ln \frac{x_{I,a}}{x_{p(I)}},
\end{aligned}
$$

where the first and third equalities follow from $x_\varnothing \ln x_\varnothing = 1 \ln 1 = 0$ for $x \in Q$, and the fifth equality comes from $x_{p(I)} = \sum_{a \in \mathcal{A}(I)} x_{I,a}$ for $x \in Q$. $\qquad\square$

From Lemma 5.2 and the fact that (10) is continuous in Q, we have

$$d^*(\xi) = \max_{x \in Q} \left\{ \xi^\top x - d(x) \right\}$$

$$= \sup_{x \in \mathrm{ri}Q} \left\{ \xi^\top x - \sum_{I \in \mathcal{I}} w_I x_{p(I)} \sum_{a \in \mathcal{A}(I)} \frac{x_{I,a}}{x_{p(I)}} \ln \frac{x_{I,a}}{x_{p(I)}} \right\}. \tag{13}$$

Now, choose any $I \in \mathcal{I}$ satisfying $\{I' \in \mathcal{I} \mid p(I') = (I, a)\} = \emptyset$ for all $a \in \mathcal{A}(I)$, then the terms on $(x_{I,a})_{a \in \mathcal{A}(I)}$ of the supreme (13) are given by

$$x_{p(I)} \sup_{z \in \mathrm{ri}\Delta_{|\mathcal{A}(I)|}} \left\{ \sum_{a \in \mathcal{A}(I)} \xi_{I,a} z_a - w_I \sum_{a \in \mathcal{A}(I)} z_a \ln z_a \right\},$$

8

where $z_a := x_{I,a}/x_{p(I)}$ and Δ_n is the n-dimentional simplex. This maximization subproblem can be solved analytically (see Appendix B). That is, the maximizer is given by

$$z_a^* := \frac{\exp(\xi_{I,a}/w_I)}{\sum_{a' \in \mathcal{A}(I)} \exp(\xi_{I,a'}/w_I)},$$

which achieves the following supreme:

$$\text{opt}_I := \sum_{a \in \mathcal{A}(I)} \xi_{I,a} z_a^* - w_I \sum_{a \in \mathcal{A}(I)} z_a^* \ln z_a^*$$

$$= w_I \ln \left\{ \sum_{a \in \mathcal{A}(I)} \exp(\xi_{I,a}/w_I) \right\}.$$

Substituting this result to (13), the terms on $(x_{I,a})_{a \in \mathcal{A}(I)}$ disappear and opt_I is added to $\xi_{p(I)}$, which is the coefficient of $x_{p(I)}$ in (13).

By repeating the above operations in bottom-up order, (13) can be solved, and the total calculation can be performed in $O(|\Sigma|)$. We can also obtain $\nabla d^*(\boldsymbol{\xi}) = \arg\max_{\boldsymbol{x} \in Q} \{\boldsymbol{\xi}^\top \boldsymbol{x} - d(\boldsymbol{x})\}$, however, only the ratio z is obtained in the above operations. Then, after solving (13), we need to calculate $\nabla d^*(\boldsymbol{\xi})$ by multiplying z in top-down order. See Algorithm 3 for details.

Algorithm 3 Calculating $d^*(\boldsymbol{\xi})$ and $\nabla d^*(\boldsymbol{\xi})$

Input: $\boldsymbol{\xi} \in \mathbb{R}^{|\Sigma|}$
Output: $y = d^*(\boldsymbol{\xi})$, $z = \nabla d^*(\boldsymbol{\xi})$
 1: $z \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$
 2: **for** $I \in \mathcal{I}$ in bottom-up order **do**
 3:     **for** $a \in \mathcal{A}(I)$ **do**
 4:         $\xi_{I,a} \leftarrow \exp(\xi_{I,a}/w_I)$
 5:     **end for**
 6:     $\xi_{p(I)} \leftarrow \xi_{p(I)} + w_I \ln \sum_{a \in \mathcal{A}(I)} \xi_{I,a}$
 7:     **for** $a \in \mathcal{A}(I)$ **do**
 8:         $z_{I,a} \leftarrow \xi_{I,a}/\sum_{a \in \mathcal{A}(I)} \xi_{I,a}$
 9:     **end for**
10: **end for**
11: $y \leftarrow \xi_\varnothing$
12: $z_\varnothing \leftarrow 1$
13: **for** $I \in \mathcal{I}$ in top-down order **do**
14:     **for** $a \in \mathcal{A}(I)$ **do**
15:         $z_{I,a} \leftarrow z_{p(I)} z_{I,a}$
16:     **end for**
17: **end for**

Theorem 5.3. *The prox-function* (10) *satisfies*

$$\max_{\boldsymbol{x} \in Q} d(\boldsymbol{x}) - \min_{\boldsymbol{x} \in Q} d(\boldsymbol{x}) \leq M_Q \ln |\Sigma|.$$

Proof. Since $Q \subset [0,1]^{|\Sigma|}$ and $x \ln x \leq 0$ for $x \in [0,1]$, we have $\max_{x \in Q} d(x) \leq 0$. Then it is sufficient to show $-\min_{x \in Q} d(x) = d^*(\mathbf{0}) \leq M_Q \ln |\Sigma|$. From the procedure for computing $d^*(\xi)$ presented above, we see that $d^*(\mathbf{0})$ satisfies the following recursive equation:

$$d^*(\mathbf{0}) = \sum_{p(I)=\varnothing} \mathrm{opt}_I$$

$$\mathrm{opt}_I = w_I \ln \left\{ \sum_{a \in \mathcal{A}(I)} \exp \left(\frac{\sum_{p(I')=(I,a)} \mathrm{opt}_{I'}}{w_I} \right) \right\} \quad \forall I \in \mathcal{I} \tag{14}$$

Now define $\gamma_I \in \mathbb{R}$ recursively:

$$\gamma_I := |\mathcal{A}(I)| + \sum_{a \in \mathcal{A}(I)} \sum_{p(I')=(I,a)} \gamma_{I'} \quad \forall I \in \mathcal{I}, \tag{15}$$

then let us show

$$\mathrm{opt}_I \leq w_I \ln \gamma_I \quad \forall I \in \mathcal{I} \tag{16}$$

recursively. Assume that (16) holds for I' which is below I in the sense of the rooted tree, discussed in Section 4. Then, we have

$$\sum_{p(I')=(I,a)} \mathrm{opt}_{I'} \leq \sum_{p(I')=(I,a)} w_{I'} \ln \gamma_{I'}$$

$$\leq \left(1 + \sum_{p(I')=(I,a)} w_{I'} \right) \ln \left(1 + \sum_{p(I')=(I,a)} \gamma_{I'} \right)$$

$$\leq w_I \ln \left(1 + \sum_{p(I')=(I,a)} \gamma_{I'} \right),$$

where the first inequality follows from the assumption, and the third inequality comes from (11). By substituting this to (14), we have

$$\mathrm{opt}_I \leq w_I \ln \left\{ \sum_{a \in \mathcal{A}(I)} \left(1 + \sum_{p(I')=(I,a)} \gamma_{I'} \right) \right\}$$

$$= w_I \ln \gamma_I.$$

Therefore (16) is shown for all $I \in \mathcal{I}$. By the definition (15), we have

$$1 + \sum_{p(I)=\varnothing} \gamma_I = 1 + \sum_{I \in \mathcal{I}} |\mathcal{A}(I)| = |\Sigma|.$$

By the definition of w_I, we also have

$$1 + \sum_{p(I)=\varnothing} w_I = \max_{x \in Q} \|x\|_1 = M_Q.$$

10

Finally, we obtained the following inequality that we wanted to show.

$$
\begin{aligned}
d^*(\mathbf{0}) &= \sum_{p(I)=\varnothing} \mathrm{opt}(I) \\
&\leq \sum_{p(I)=\varnothing} w_I \ln \gamma_I \\
&\leq \left(1 + \sum_{p(I)=\varnothing} w_I\right) \ln \left(1 + \sum_{p(I)=\varnothing} \gamma_I\right) \\
&= M_Q \ln |\Sigma|.
\end{aligned}
$$

$\square$

We conclude this section by proving the following theorem.

Theorem 5.4. *The prox-function d defined by* (10) *is* $1/(M_Q^2 \ln |\Sigma|)$*-strongly convex substantially, with respect to the L_1-norm. In other words, assume that d is σ-strongly convex with respect to the L_1-norm, and let $D := \max_{\mathbf{x} \in Q} d(\mathbf{x}) - \min_{\mathbf{x} \in Q} d(\mathbf{x})$, then the following inequality holds:*

$$
\frac{D}{\sigma} \leq M_Q^2 \ln |\Sigma|.
$$

Proof. It follows from Theorem 5.1 and Theorem 5.3. $\square$

6   Numerical experiments

In this section, we report the results of solving extensive-form games by using the prox-function (10) with EGT. As toy instances, we used three games, Kuhn poker, Leduc Hold'em (3 ranks), and Leduc Hold'em (13 ranks), which we explain in detail in Appendix A. All implementations used in the experiments are available on `https://github.com/habara-k/egt-on-efg`.

6.1   Heuristics to accelerate the convergence of EGT

The parameters and choices between μ_1 and μ_2 to shrink proposed in [2] guarantee the convergence of (8) but are very conservative, then heuristic-based parameter selection will perform better in most cases. First, we always use the following heuristics in [8]:

- To start with small μ_1 and μ_2, we call the initializing algorithm 1 with $\mu_1 = \mu_2 = 10^{-6}$. Increase μ_1 and μ_2 by 20% until the output satisfies the excessive gap condition.

- In each step, shrink the larger between μ_1 and μ_2.

- To obtain a large step size, we call the shrinking algorithm 2 with the global τ, which is initially set to 0.5 and is decreased by 50% while the output does not satisfy the excessive gap condition.

See Algorithm 4 for details. In addition to these heuristics, this paper proposes a *centering trick*, which is still not considered in the related literature to the best of our knowledge. For $x' \in \mathrm{ri}Q$, we can define the following prox-function:

$$\tilde{d}(x; x') := d(x) - d(x') - \nabla d(x')^\top (x - x'),$$

which we call x'-*centered* prox-function because $\arg\min_{x \in Q} \tilde{d}(x; x') = x'$ holds. The centering trick uses $\tilde{d}(x; x')$ as a prox-function for EGT, where x' is a solution obtained by some other method. The reason for using this centered prox-function is to improve the accuracy of the smoothing approximation $f_{\mu_2}(\phi_{\mu_1})$ in EGT. In fact, when the prox-function $d_2(d_1)$ takes the minimum value 0 in the optimal solution $y^*(x^*)$ of BSPP, the minimization of f and f_{μ_2} (the maximization of ϕ and ϕ_{μ_1}) are equivalent, for any $\mu_2(\mu_1) > 0$. In addition, we show that

$$\nabla \tilde{d}^*(\xi; x') := \arg\max_{x \in Q} \left\{ \xi^\top x - \tilde{d}(x; x') \right\} = \nabla d^*(\xi + \nabla d(x')),$$

so the cost required for the calculation is also $O(|\Sigma|)$. Numerical experiments evaluate the performance of the centering trick.

6.2 Results of the experiments

Five methods are used: CFR [1], CFR+ [3], EGT, EGT-centering, and EGT-centering with CFR+. EGT uses the prox-function defined in (10). In EGT-centering, the first EGT is performed in 10% of the total steps, and the remaining 90% of the steps are performed in the second EGT using the prox-function centered on the solution obtained from the first EGT. EGT-centering with CFR+ is a variant of EGT-centering, using CFR+ instead of the first EGT.

From Figure 1, note that EGT-centering and EGT-centering with CFR+ both use EGT and CFR+ for the first 10% of iterations, respectively, so only the last 90% of iterations are drawn. First, since Kuhn poker is a very small game, EGT, which is unsuitable for larger games, performs similarly to CFR+. EGT with centering tricks using the solutions of these two methods, namely EGT-centering and EGT-centering with CFR+, converge faster than the other methods. Second, since Leduc Hold'em is a larger game than Kuhn poker, EGT converges worse than CFR+. However, we see that EGT-centering performs as well as CFR+ and that EGT-centering with CFR+ converges better than pure CFR+.

These results show that EGT combined with CFR+ by the centering trick converges faster than pure CFR+ in large games where EGT alone performs worse than CFR+. In addition, although we could not experiment in this paper, the results suggest that EGT with the centering trick has the potential to further exploit the performance of CFR+ in very large games such as those used in [4, 5]. Note that the implementations of all five methods are optimized in the same way, resulting in EGT (including the centering trick) taking at most 2 to 3 times longer per iteration than CFR (CFR+). Thus, although the horizontal axis in Figure 1 represents the number of iterations, changing this to the running time (computational cost) yields almost the same result.
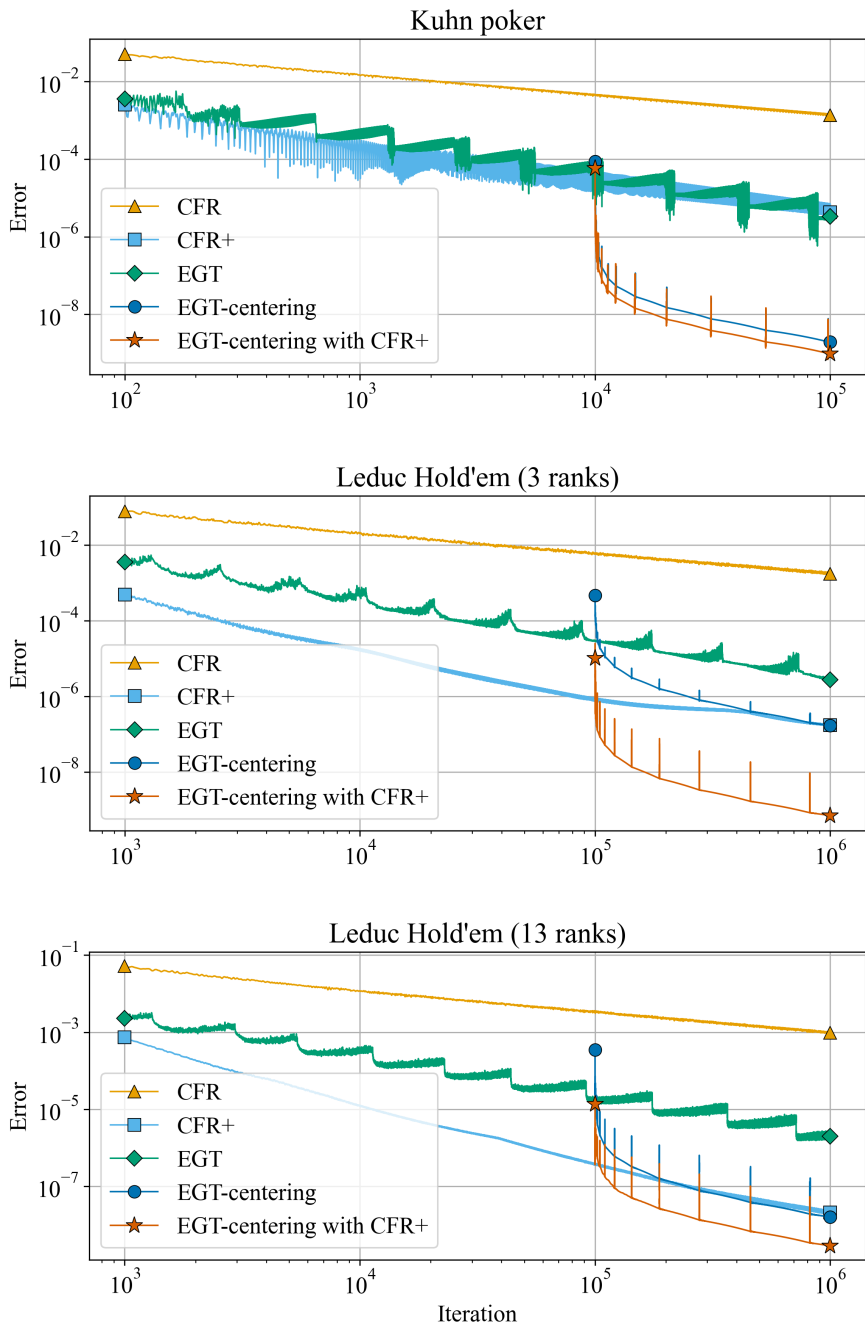
Figure 1: Performance of the five methods for solving three games.

Algorithm 4 Excessive gap technique (with heuristics in [8])

1: **function** INIT()
2:       $\mu \leftarrow 10^{-6}$
3:       **while** true **do**
4:             $x, y \leftarrow$ Call algorithm 1 with $\mu_1 = \mu_2 = \mu$.
5:             **if** $f_{\mu_2}(x) \leq \phi_{\mu_1}(y)$ **then**
6:                   **return** x, y, μ, μ
7:             **end if**
8:             $\mu \leftarrow 1.2\mu$
9:       **end while**
10: **end function**

11: **function** DECREASE$\mu_1(x, y, \mu_1, \mu_2, \tau)$
12:       **while** true **do**
13:             $\bar{x}, \bar{y} \leftarrow$ Call algorithm 2 with x, y, μ_1, μ_2, and τ.
14:             **if** $f_{\mu_2}(\bar{x}) \leq \phi_{(1-\tau)\mu_1}(\bar{y})$ **then**
15:                   **return** $\bar{x}, \bar{y}, (1 - \tau)\mu_1, \tau$
16:             **end if**
17:             $\tau \leftarrow 0.5\tau$
18:       **end while**
19: **end function**

Output: $x \in Q_1, y \in Q_2$: solutions of EGT
20: $x, y, \mu_1, \mu_2 \leftarrow$ INIT()
21: $\tau \leftarrow 0.5$
22: **for** $k = 1, \ldots, T - 1$ **do**
23:       **if** $\mu_1 > \mu_2$ **then**
24:             $x, y, \mu_1, \tau \leftarrow$ DECREASE$\mu_1(x, y, \mu_1, \mu_2, \tau)$
25:       **else**
26:             decrease μ_2 similarly.
27:       **end if**
28: **end for**

---

7 Conclusions

To make the smoothing method for solving extensive-form games applicable to large games, we proposed a modified version of the prox-function used internally in the smoothing method to improve its theoretical convergence guarantees. We also proposed a heuristic to update the prox-function with temporary solutions, which is expected to improve the accuracy of the internal approximation of the smoothing method. Numerical experiments confirm that the smoothing method applying the heuristic with the solution of CFR+, the state-of-the-art method for large games, converges faster than pure CFR+. Some future works include confirming the performance of the smoothing method when applied to larger-scale extensive-form games and proving theoretically that the proposed heuristic improves convergence.

Acknowledgments

References

[1] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20:1729–1736, 2007.

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

[3] Oskari Tammelin. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*, 2014.

[4] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[5] Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[6] Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

[7] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, 179(1-2):385–417, 2020.

[8] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Better regularization for sequential decision spaces: Fast convergence rates for Nash, correlated, and team equilibria. *arXiv preprint arXiv:2105.12954*, 2021.

[9] Harold W Kuhn. A simplified two-person poker. *Contributions to the Theory of Games*, 1:97–103, 1950.

[10] Finnegan Southey, Michael P Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*, 2012.

[11] Todd W Neller and Marc Lanctot. An introduction to counterfactual regret minimization. In *Proceedings of Model AI Assignments, The Fourth Symposium on Educational Advances in Artificial Intelligence (EAAI-2013)*, volume 11, 2013.

A Toy instances

In this section, we present some well-known extensive-form games. These games are used in the numerical experiments in Section 6.

A.1 Kuhn poker

Kuhn poker is a simple variant of Texas Hold'em proposed in [9], played with three cards, J, Q, and K. Each player is dealt a card privately, and the game begins with each player betting 1 chip. Player 1 is the first to act. Player 1 can either *check* (do nothing and pass the turn to Player 2) or *raise* (bet an additional 1 chip).

(a) If player 1 raises, player 2 chooses to *call* (bet an additional 1 chip) or *fold* (surrender and lose the 1 chip already bet); if player 2 calls, the player with the higher-ranked card wins all chips (showdown).

(b) If player 1 checks, player 2 chooses to check (and showdown) or raise; if player 2 raises, player 1 chooses to call or fold as in (a).

The game tree of Kuhn poker is shown in Figure 2, and its information tree of each player is shown in Figures 3 and 4.

A.2 Leduc Hold'em

Leduc Hold'em is a simple variant of Texas Hold'em proposed in [10], played with pairs of cards J, Q, and K, totaling six cards. The general rules are the same as in Kuhn poker, but some differences exist.

- Each player can raise against the opponent's raise, which is called *re-raise*, but cannot raise against a re-raise.
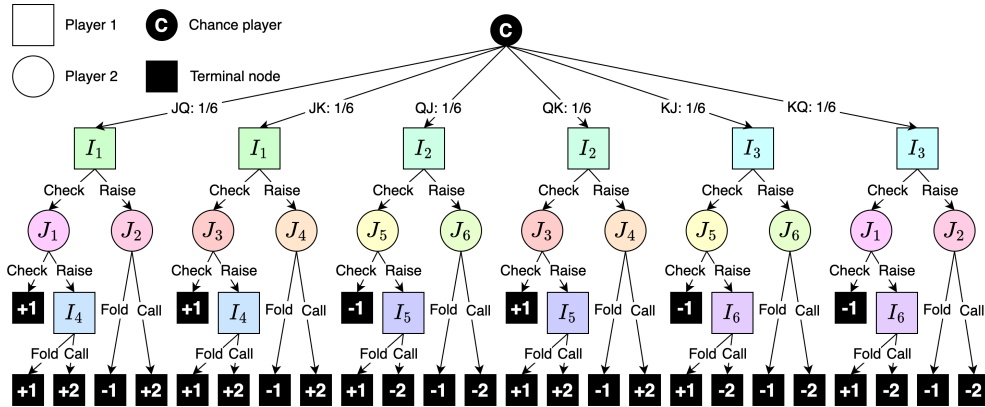
Figure 2: Kuhn poker game tree. The nodes marked with "C" on a black background are played by the chance player, and the square and circle nodes are played by player 1 and player 2. The terminal nodes are represented by a black square with corresponding $u(z)$. The arrows correspond to actions, and the state moves down to the pointed node. Chance player action probabilities are given and are written on the arrows. The nodes played by player 1 and player 2 are partitioned according to the information partition $\mathcal{I}_1 = \{I_1, \ldots, I_6\}$ and $\mathcal{I}_2 = \{J_1, \ldots, J_6\}$. Each player does not know which card is dealt to their opponent, so they cannot distinguish between nodes belonging to the same information partition.
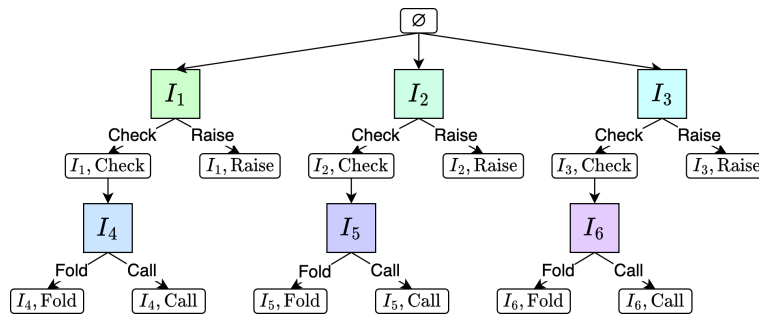

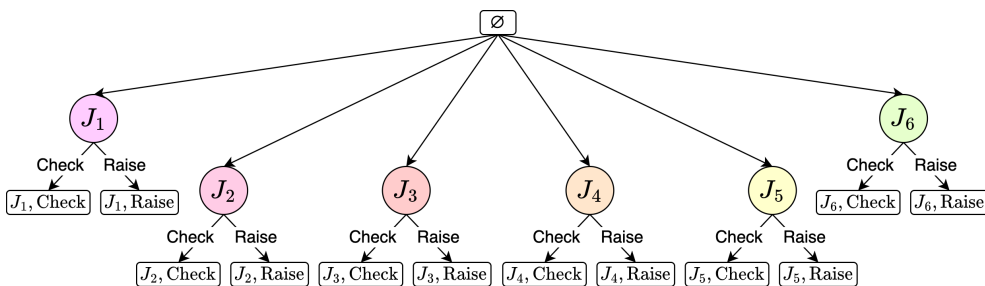Figure 3: The *information tree* of player 1 in Kuhn poker

Figure 4: The *information tree* of player 2 in Kuhn poker
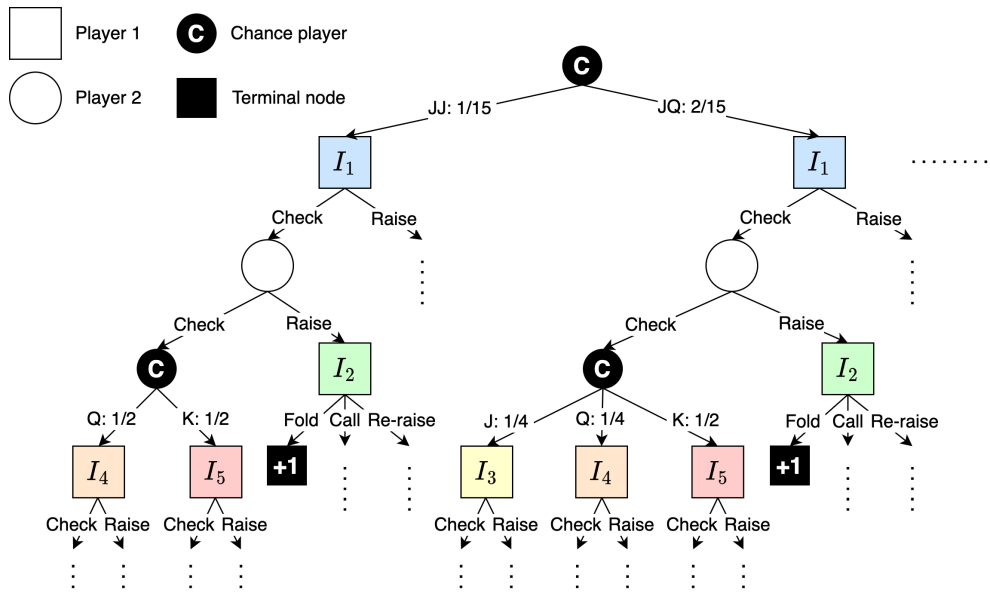
17

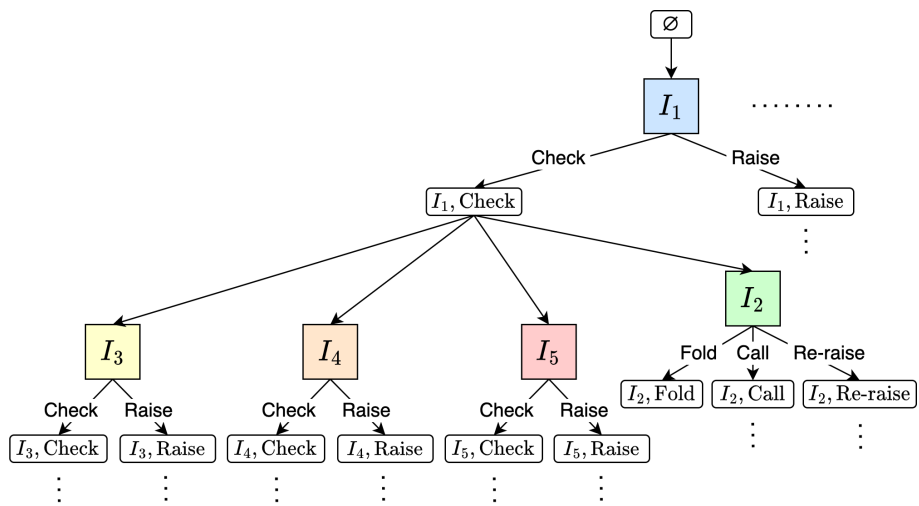Figure 5: Leduc Hold'em game tree

Figure 6: The *information tree* of player 1 in Leduc Hold'em

- If both players check or one player calls, instead of an immediate showdown, a *community card* is revealed randomly from the remaining cards, and the game is resumed only once more.

- The winner of the showdown is the player with the same card as the community card. If there is no such player, the player with the higher-ranked card wins. If both players have the same cards, the game is tied (chips are divided equally).

- The betting amount of raise is 2 and 4 chips in each phase. This means that the maximum move of chips is 1+2+2+4+4=13.

The game tree of Leduc hold'em is shown in Figure 5, and the information tree of player 1 is shown in Figure 6. Experiments are also performed for Leduc Hold'em (13 ranks), a game in which the number of card ranks is changed from 3 to 13.

B Subproblem in the conjugate function

Here, we will compute the analytical solution of the problem below, which is associated to our proposed prox-function. Let us consider the following maximization problem:

$$\max_{z \, \in \, \mathbb{R}^n_{>0}} \quad \sum_{i=1}^{n} \xi_i z_i - w \sum_{i=1}^{n} z_i \ln z_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} z_i = 1, \tag{17}$$

where $w > 0$. Lagrange function $L \colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is defined by

$$L(z, \mu) := \sum_{i=1}^{n} \xi_i z_i - w \sum_{i=1}^{n} z_i \ln z_i + \mu \left(1 - \sum_{i=1}^{n} z_i \right),$$

and the optimal solution z^* of (17) satisfies the following equation with some $\mu^* \in \mathbb{R}$:

$$\frac{\partial L}{\partial z_i}(z^*, \mu^*) = \xi_i - w(1 + \ln z_i^*) - \mu^* = 0, \quad i = 1, \dots, n$$

$$\frac{\partial L}{\partial \mu}(z^*, \mu^*) = 1 - \sum_{j=1}^{n} z_i^* = 0.$$

Eliminate μ^* from the above equation to obtain

$$z_i^* = \frac{\exp(\xi_i/w)}{\sum_{j=1}^{n} \exp(\xi_j/w)},$$

and the optimal value is given by

$$\sum_{i=1}^{n} \xi_i z_i^* - w \sum_{i=1}^{n} z_i^* \ln z_i^* = \sum_{i=1}^{n} \xi_i z_i^* - w \sum_{i=1}^{n} z_i^* \left\{ (\xi_i/w) - \ln \sum_{j=1}^{n} \exp(\xi_j/w) \right\}$$

$$= w \ln \sum_{j=1}^{n} \exp(\xi_j/w).$$

C Implementation of CFR and CFR+

Most implementations of CFR and CFR+ employ the depth-first search of the game tree [3, 11]. This method has the advantage of greatly reducing computation time in huge games, at the expense of accuracy, by using sampling. On the other hand, simply exploring all nodes may be computationally inefficient. For example, in a game with multiple terminal nodes $z \in Z$ such that the pair $(p_1(z), p_2(z))$ is equal (see Section 4 for the definition of p_1, p_2), all terminal nodes are searched at each iteration in the depth-first search type implementation, but in the BSPP type unnecessary.

This paper presents CFR and CFR+ implementations using the BSPP representation of the extensive-form game introduced in Section 4. These implementations are also available on `https://github.com/habara-k/egt-on-efg`.

Algorithm 5 CFR and CFR+ (in BSPP form)

1: **function** PROD($z \in \mathbb{R}^{|\Sigma|}_{\geq 0}$)
2:     $x \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$
3:     $x_\varnothing \leftarrow 1$
4:     **for** $I \in \mathcal{I}$ in top-down order **do**
5:         **for** $a \in \mathcal{A}(I)$ **do**
6:             $x_{I,a} \leftarrow x_{p(I)} z_{I,a}$
7:         **end for**
8:     **end for**
9:     **return** x
10: **end function**

11: **function** NORMALIZE($r \in \mathbb{R}^{|\Sigma|}_{\geq 0}$)
12:     $z \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$
13:     **for** $I \in \mathcal{I}$ **do**
14:         **if** $\exists a \in \mathcal{A}(I)$ s.t. $r_{I,a} > 0$ **then**
15:             **for** $a \in \mathcal{A}(I)$ **do**
16:                 $z_{I,a} \leftarrow r_{I,a} / \sum_{a \in \mathcal{A}(I)} r_{I,a}$
17:             **end for**
18:         **else**
19:             **for** $a \in \mathcal{A}(I)$ **do**
20:                 $z_{I,a} \leftarrow 1/|\mathcal{A}(I)|$
21:             **end for**
22:         **end if**
23:     **end for**
24:     **return** z
25: **end function**

26: **function** REGRET($z \in \mathbb{R}^{|\Sigma|}, u \in \mathbb{R}^{|\Sigma|}$)
27:     $r \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma|}$
28:     **for** $I \in \mathcal{I}$ in bottom-up order **do**
29:         **for** $a \in \mathcal{A}(I)$ **do**
30:             $r_{I,a} \leftarrow u_{I,a} - \sum_{a \in \mathcal{A}(I)} u_{I,a} z_{I,a}$
31:         **end for**
32:         $u_{p(I)} \leftarrow u_{p(I)} + \sum_{a \in \mathcal{A}(I)} u_{I,a} z_{I,a}$
33:     **end for**
34:     **return** r
35: **end function**

Output: $x \in Q_1, y \in Q_2$: solutions of CFR
36: $r_\mathcal{X}, r_\mathcal{Y} \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma_1|}, \mathbf{0} \in \mathbb{R}^{|\Sigma_2|}$
37: $z_\mathcal{X}, z_\mathcal{Y} \leftarrow \text{NORMALIZE}(r_\mathcal{X}), \text{NORMALIZE}(r_\mathcal{Y})$
38: $x^1, y^1 \leftarrow \text{PROD}(z_\mathcal{X}), \text{PROD}(z_\mathcal{Y})$
39: **for** $k = 1, \ldots, T - 1$ **do**
40: $\quad r_\mathcal{X} \leftarrow r_\mathcal{X} + \text{REGRET}(z_\mathcal{X}, -Ay^k)$
41: $\quad r_\mathcal{Y} \leftarrow r_\mathcal{Y} + \text{REGRET}(z_\mathcal{Y}, A^\top x^k)$
42: $\quad z_\mathcal{X} \leftarrow \text{NORMALIZE}(\max(r_\mathcal{X}, \mathbf{0}))$
43: $\quad z_\mathcal{Y} \leftarrow \text{NORMALIZE}(\max(r_\mathcal{Y}, \mathbf{0}))$
44: $\quad x^{k+1} \leftarrow \text{PROD}(z_\mathcal{X})$
45: $\quad y^{k+1} \leftarrow \text{PROD}(z_\mathcal{Y})$
46: **end for**
47: $x, y \leftarrow \frac{1}{T} \sum_{k=1}^{T} k x^k, \frac{1}{T} \sum_{k=1}^{T} k y^k$

Output: $x \in Q_1, y \in Q_2$: solutions of CFR+
48: $r_\mathcal{X}, r_\mathcal{Y} \leftarrow \mathbf{0} \in \mathbb{R}^{|\Sigma_1|}, \mathbf{0} \in \mathbb{R}^{|\Sigma_2|}$
49: $z_\mathcal{X}, z_\mathcal{Y} \leftarrow \text{NORMALIZE}(r_\mathcal{X}), \text{NORMALIZE}(r_\mathcal{Y})$
50: $x^1, y^1 \leftarrow \text{PROD}(z_\mathcal{X}), \text{PROD}(z_\mathcal{Y})$
51: **for** $k = 1, \ldots, T - 1$ **do**
52: $\quad r_\mathcal{X} \leftarrow \max(r_\mathcal{X} + \text{REGRET}(z_\mathcal{X}, -Ay^k), \mathbf{0})$
53: $\quad z_\mathcal{X} \leftarrow \text{NORMALIZE}(r_\mathcal{X})$
54: $\quad x^{k+1} \leftarrow \text{PROD}(z_\mathcal{X})$
55: $\quad r_\mathcal{Y} \leftarrow \max(r_\mathcal{Y} + \text{REGRET}(z_\mathcal{Y}, A^\top x^{k+1}), \mathbf{0})$
56: $\quad z_\mathcal{Y} \leftarrow \text{NORMALIZE}(r_\mathcal{Y})$
57: $\quad y^{k+1} \leftarrow \text{PROD}(z_\mathcal{Y})$
58: **end for**
59: $x, y \leftarrow \frac{2}{T+T^2} \sum_{k=1}^{T} k x^k, \frac{2}{T+T^2} \sum_{k=1}^{T} k y^k$

Master's Thesis

Convergence analysis and acceleration of the smoothing methods for solving extensive-form games

Guidance

Professor   Nobuo Yamashita
Associate Professor   Ellen Hidemi Fukuda

Keigo Habara

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2023

Convergence analysis and acceleration of the smoothing methods
for solving extensive-form games

Keigo Habara

February 2023

Convergence analysis and acceleration of the smoothing methods for solving extensive-form games

Keigo Habara

Abstract

The extensive-form game has been studied considerably in recent years. It can represent games with multiple decision points and incomplete information, and hence it is helpful in formulating games with uncertain inputs, such as poker. We consider an extended-form game with two players and zero-sum, i.e., the sum of their payoffs is always zero. In such games, the problem of finding the optimal strategy can be formulated as a bilinear saddle-point problem. This formulation grows huge depending on the size of the game, since it has variables representing the strategies at all decision points for each player. To solve such large-scale bilinear saddle-point problems, the excessive gap technique (EGT), a smoothing method, has been studied. This method generates a sequence of approximate solutions whose error is guaranteed to converge at $O(1/k)$, where k is the number of iterations. However, it has the disadvantage of having poor theoretical bounds on the error related to the game size. This makes it inapplicable to large games.

Our goal is to improve the smoothing method for solving extensive-form games so that it can be applied to large-scale games. To this end, we make two contributions in this work. First, we slightly modify the strongly convex function used in the smoothing method in order to improve the theoretical bounds related to the game size. Second, we propose a heuristic called centering trick, which allows the smoothing method to be combined with other methods and consequently accelerates the convergence in practice. As a result, we combine EGT with CFR+, a state-of-the-art method for extensive-form games, to achieve good performance in games where conventional smoothing methods do not perform well. The proposed smoothing method is shown to have the potential to solve large games in practice.