A dynamical neural network approach for solving stochastic two-player zero-sum games

Dawen Wu
Université Paris Saclay, CNRS, CentraleSupelec
Laboratoire des Signaux et des Systèmes (L2S)

Joint work with Abdel Lisser

ECSO-CMS 2022

July 1st, 2022
Outline of the talk

1. Introduction
2. Stochastic two-player zero-sum games formulation
3. Neurodynamic optimization approach
4. Numerical experiments
5. Conclusion
### Games:

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Games</th>
<th>Main results</th>
</tr>
</thead>
<tbody>
<tr>
<td>von Neumann</td>
<td>1928</td>
<td>two-player zero-sum game</td>
<td>Saddle point existence</td>
</tr>
<tr>
<td>Nash</td>
<td>1950</td>
<td>n-players general-sum game</td>
<td>Nash equilibrium existence</td>
</tr>
<tr>
<td>Charnes</td>
<td>1953</td>
<td>two-player zero-sum game with linear constraints</td>
<td>Saddle point existence</td>
</tr>
<tr>
<td>Singh &amp; Lisser</td>
<td>2019</td>
<td>two-player zero-sum game with chance constraints</td>
<td>Saddle point existence</td>
</tr>
</tbody>
</table>
Neurodynamic optimization approach or Dynamical neural network:

<table>
<thead>
<tr>
<th>Authors</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hopfield et al. (1986)</td>
<td>Linear programming problems (Hopfield network)</td>
</tr>
<tr>
<td>Kennedy et al. (1988)</td>
<td>Nonlinear programming problems based on the penalty method</td>
</tr>
<tr>
<td>Xia et al. (2007)</td>
<td>Nonlinear projection equations</td>
</tr>
<tr>
<td>Xu et al. (2020)</td>
<td>Constrained pseudoconvex programming problems</td>
</tr>
</tbody>
</table>
**Charnes(1953)** studied the two-player zero-sum game problem with linear constrained.

The LP formulation:

\[
\begin{align*}
\text{max } & \quad x^T Ay \\
\text{s.t.} & \quad Bx \leq b \\
& \quad 1^T x = 1 \\
& \quad x \geq 0,
\end{align*}
\]

\[
\begin{align*}
\text{min } & \quad x^T Ay \\
\text{s.t.} & \quad Dy \geq d \\
& \quad 1^T y = 1 \\
& \quad y \geq 0,
\end{align*}
\]

\[x \in \mathbb{R}^n, \ y \in \mathbb{R}^m \quad \ldots \quad \text{decision vector}\]

\[x^* \in \mathbb{R}^n, \ y^* \in \mathbb{R}^m \quad \ldots \quad \text{Saddle point of the above linear programs}\]

\[B \in \mathbb{R}^{p \times n}, \ b \in \mathbb{R}^p \quad \ldots \quad \text{The linear constraint for player 1}\]

\[D \in \mathbb{R}^{q \times m}, \ d \in \mathbb{R}^q \quad \ldots \quad \text{The linear constraint for player 2}\]
Charnes(1953) studied the two-player zero-sum game problem with linear constrained.

The LP formulation:

\[
\begin{align*}
\text{max } & \quad x^T A y \\
\text{s.t.} & \quad Bx \leq b \\
& \quad 1^T x = 1 \\
& \quad x \geq 0,
\end{align*}
\]

\[
\begin{align*}
\text{min } & \quad y^T A x \\
\text{s.t.} & \quad Dy \geq d \\
& \quad 1^T y = 1 \\
& \quad y \geq 0,
\end{align*}
\]

\[x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m \quad \ldots \quad \text{decision vector}\]

\[x^* \in \mathbb{R}^n, \quad y^* \in \mathbb{R}^m \quad \ldots \quad \text{Saddle point of the above linear programs}\]

\[B \in \mathbb{R}^{p \times n}, \quad b \in \mathbb{R}^p \quad \ldots \quad \text{The linear constraint for player 1}\]

\[D \in \mathbb{R}^{q \times m}, \quad d \in \mathbb{R}^q \quad \ldots \quad \text{The linear constraint for player 2}\]
Consider the case of stochastic linear constraints

From linear constrained to linear individual chance constraints:

\[ Bx \leq b \implies P \{ B_k^w x \leq b_k \} \geq \alpha^1_k, \quad \forall k \in J_1 \]

\[ Dy \geq d \implies P \{ D_l^w y \geq d_l \} \geq \alpha^2_l, \quad \forall l \in J_2 \]

- \( J_1 = \{1, \ldots, p\} \) and \( J_2 = \{1, \ldots, q\} \) are the index set of constraints.
- Now, each row vector \( B_k \) and \( D_l \) follows an elliptical distribution i.e. \( B_k^w \sim \text{Ellip}_m (\mu^1_k, \Sigma^1_k, \varphi^1_k) \) and \( D_l^w \sim \text{Ellip}_n (\mu^2_l, \Sigma^2_l, \varphi^2_l) \).
- \( \alpha^1 = (\alpha^1_k)_{k \in J_1} \) and \( \alpha^2 = (\alpha^2_l)_{l \in J_2} \) are confidence levels.
Consider the case of stochastic linear constraints

From **linear constrained** to linear **individual chance constraints**: 

\[ Bx \leq b \implies P \{ B_k^w x \leq b_k \} \geq \alpha_k^1, \quad \forall k \in J_1 \]

\[ Dy \geq d \implies P \{ D_l^w y \geq d_l \} \geq \alpha_l^2, \quad \forall l \in J_2 \]

- \( J_1 = \{1, \ldots, p\}, \ J_2 = \{1, \ldots, q\} \) are the index set of constraints.
- Now, each row vector \( B_k \) and \( D_l \) follows an elliptical distribution i.e. \( B_k^w \sim \text{Ellip}_m (\mu_k^1, \Sigma_k^1, \varphi_k^1) \) and \( D_l^w \sim \text{Ellip}_n (\mu_l^2, \Sigma_l^2, \varphi_l^2) \).
- \( \alpha^1 = (\alpha_k^1)_{k \in J_1} \) and \( \alpha^2 = (\alpha_l^2)_{l \in J_2} \) are confidence levels.
Consider the case of stochastic linear constraints

From **linear constrained** to **linear individual chance constraints:**

\[
Bx \leq b \implies P \{B_k^w x \leq b_k\} \geq \alpha_1, \quad \forall k \in J_1
\]

\[
Dy \geq d \implies P \{D_l^w y \geq d_l\} \geq \alpha_2, \quad \forall l \in J_2
\]

- \(J_1 = \{1, \ldots, p\}, \ J_2 = \{1, \ldots, q\}\) are the index set of constraints.
- Now, each row vector \(B_k\) and \(D_l\) follows an elliptical distribution i.e. \(B_k^w \sim Ellip_m (\mu_k^1, \Sigma_k^1, \varphi_k^1)\) and \(D_l^w \sim Ellip_n (\mu_l^2, \Sigma_l^2, \varphi_l^2)\).
- \(\alpha^1 = (\alpha_k^1)_{k \in J_1}\) and \(\alpha^2 = (\alpha_l^2)_{l \in J_2}\) are confidence levels.
From two-player zero-sum game with linear constraint to the **stochastic two-player zero-sum game problem**.

Here is the stochastic optimization problem we are going to solve, denoted as $G(\alpha)$:

\[
\begin{align*}
\max_x & \quad x^T Ay \\
\text{s.t.} & \quad P \{ B_{k} w x \leq b_k \} \geq \alpha_k^1, \quad \forall k \in \mathcal{J}_1 \\
& \quad 1^T x = 1 \\
& \quad x \geq 0,
\end{align*}
\]

\[
\begin{align*}
\min_y & \quad x^T Ay \\
\text{s.t.} & \quad P \{ D_{l} w y \geq d_l \} \geq \alpha_l^2, \quad \forall l \in \mathcal{J}_2 \\
& \quad 1^T y = 1 \\
& \quad y \geq 0.
\end{align*}
\]
Denote the feasible strategy sets of the two player as $S_1(\alpha^1)$ and $S_2(\alpha^2)$, respectively.

**Assumption 1**

1. The set $S_1(\alpha^1)$ is strictly feasible, i.e., there exists an $x \in \mathbb{R}^n$ which is a feasible point of $S_1(\alpha^1)$ and the inequality constraints of $S_1(\alpha^1)$ are strictly satisfied by $x$.

2. The set $S_2(\alpha^2)$ is strictly feasible, i.e., there exists an $y \in \mathbb{R}^m$ which is a feasible point of $S_2(\alpha^2)$ and the inequality constraints of $S_2(\alpha^2)$ are strictly satisfied by $y$. 

Dawen Wu
The saddle point existence theorem of $G(\alpha)$:

**Theorem 1 (Singh & Lisser (2019))**

Consider a stochastic two-player zero-sum game $G(\alpha)$. Let the row vectors $B^w_k \sim \text{Ellip}_m (\mu^1_k, \Sigma^1_k, \varphi^1_k), k \in J_1$, and $D^w_l \sim \text{Ellip}_n (\mu^2_l, \Sigma^2_l, \varphi^2_l), l \in J_2$. For all $k$ and $l$, $\Sigma^1_k \succ 0$ and $\Sigma^2_l \succ 0$. Then, there exists a saddle point equilibrium for the game $G(\alpha)$ for all $\alpha \in (0.5, 1]^{J_1} \times (0.5, 1]^{J_2}$. 

Dawen Wu
Obtaining the optimal mixed strategy $y^*$ of Player 2 can be reformulated as the following SOCP:

\[
\begin{align*}
\min_{y, v^1, (\delta_k^1)_{k \in J_1}, \lambda^1} & \quad v^1 + \sum_{k \in J_1} \lambda^1_k b_k \\
\text{s.t.} & \\
(1) & \quad Ay - \sum_{k \in J_1} \lambda^1_k \mu^1_k - \sum_{k \in J_1} (\Sigma^1_k)^{\frac{1}{2}} \delta^1_k \leq v^1 \mathbf{1}_m \\
(2) & \quad -y^T \mu^2_l + \psi^{-1} (\xi^2_l) \left\| (\Sigma^2_l)^{\frac{1}{2}} y \right\| \leq -d_l, \quad \forall l \in J_2 \\
(3) & \quad \left\| \delta^1_k \right\| \leq \lambda^1_k \psi^{-1} (\xi^1_k) \left( \alpha^1_k \right), \quad \forall k \in J_1 \\
(4) & \quad \mathbf{1}^T y = 1 \\
(5) & \quad y \geq 0 \\
(6) & \quad \lambda^1_k \geq 0, \quad \forall k \in J_1
\end{align*}
\]
Obtaining the optimal mixed strategy $x^*$ of Player 1 can be reformulated as the following SOCP:

$$
\begin{align*}
(\mathcal{D}) \quad \max_{x, v^2, (\delta^2_l)_{l \in J_2}, \lambda^2} & \quad v^2 + \sum_{l \in J_2} \lambda^2_l d_l \\
\text{s.t.} & \quad (1) A^T x - \sum_{l \in J_2} \lambda^2_l \mu^2_l - \sum_{l \in J_2} \left( \Sigma^2_l \right)^{1/2} \delta^2_l \geq v^2 \mathbf{1}_n \\
& \quad (2) x^T \mu^1_k + \Psi^{-1}_{\xi^1_k} \left( \alpha^1_k \right) \left\| \left( \Sigma^1_k \right)^{1/2} x \right\| \leq b_k, \quad \forall k \in J_1 \\
& \quad (3) \left\| \delta^2_l \right\| \leq \lambda^2_l \Psi^{-1}_{\xi^2_l} \left( \alpha^2_l \right), \quad \forall l \in J_2 \\
& \quad (4) \mathbf{1}^T x = 1 \\
& \quad (5) x \geq 0 \\
& \quad (6) \lambda^2_l \geq 0, \quad \forall l \in J_2
\end{align*}
$$
Theorem 2 (Singh & Lisser (2019))

Consider a stochastic two-player zero-sum game $G(\alpha)$. Let the row vector $B^w \sim \text{Ellip}_m \left( \mu^1_k, \Sigma^1_k, \varphi^1_k \right), k \in J_1$, where $\Sigma^1_k \succ 0$, and the row vector $D^w_l \sim \text{Ellip}_n \left( \mu^2_l, \Sigma^2_l, \varphi^2_l \right), l \in J_2$ where $\Sigma^2_l \succ 0$. Let Assumption 1 holds. Then, for a given $\alpha \in (0.5, 1]^p \times (0.5, 1]^q$, $(x^*, y^*)$ is a saddle point equilibrium of the game $G(\alpha)$ if and only if

$\left( y^*, v^1, (\delta^1_k)_{k \in J_1}, \lambda^1 \right)$ and $\left( x^*, v^2, (\delta^2_l)_{l \in J_2}, \lambda^2 \right)$ are optimal solutions of primal-dual pair of SOCPs $(\mathcal{P})$ and $(\mathcal{D})$, respectively.

- The saddle point $(x^*, y^*)$ of the game $G(\alpha)$ is contained in the optimal solution of the SOCPs $(\mathcal{P})$ and $(\mathcal{D})$.
- $(\mathcal{P})$ and $(\mathcal{D})$ are a primal-dual pair of SOCPs.
The SOCP ($\mathcal{P}$) can be written as:

$$
\begin{align*}
\min_{s} & \quad f(s) \\
\text{s.t.} & \quad g(s) \leq 0,
\end{align*}
$$

(1)

The related KKT conditions are:

$$
\nabla f(s) + \nabla g(s)^T u = 0 \\
g(s) \leq 0, \quad u^T \geq 0, \quad u^T g(s) = 0
$$

(2)

- $s = \left( y, v^1, (\delta^1_k)_{k \in J_1}, \lambda^1 \right)$ is the primal variable.
- $u = \left( x, v^2, (\delta^2_l)_{l \in J_2}, \lambda^2 \right)$ is the dual variable.
- The saddle point $(x^*, y^*)$ can be obtained by solving the KKT conditions (2).
Here, we introduce the time variable $t$, and $s(t)$ and $u(t)$ become time dependent functions.

Let $r(t)$ contain $s(t), u(t)$, i.e.,

$$r(t) = (s(t), u(t)) = (y(t), v(t), \delta(t), \lambda(t), u(t))^T.$$  

We use this following ODE system, $\frac{dr}{dt} = \Phi(r)$, to solve the primal-dual pair of SOCPs, $(\mathcal{P})$ and $(\mathcal{D})$.

$$
\frac{dr}{dt} = 
\begin{bmatrix}
\frac{dy}{dt} \\
\frac{dv}{dt} \\
\frac{d\delta}{dt} \\
\frac{d\lambda}{dt} \\
\frac{du}{dt}
\end{bmatrix}
= 
\begin{bmatrix}
- (\nabla f_y + \nabla g_y^T(u + g)^+) \\
- (\nabla f_v + \nabla g_v^T(u + g)^+) \\
- (\nabla f_{\delta} + \nabla g_{\delta}^T(u + g)^+) \\
- (\nabla f_{\lambda} + \nabla g_{\lambda}^T(u + g)^+) \\
(u + g)^+ - u
\end{bmatrix},
$$ (3)
Neurodynamic optimization approach

Theorems

Theorem 3

The point \( r^* = (y^*, v^*, \delta^*, \lambda^*, u^*)^T \) is the equilibrium point of the ODE system (3) if and only if it is also the KKT point of the SOCP problem.

Lemma 4

The equilibrium point of the proposed ODE system (3) is unique.

Theorem 5

The equilibrium point \( r^* = (y^*, v^*, \delta^*, \lambda^*, u^*) \) of the proposed ODE system (3) is globally asymptotically stable.

- Theorem 3: The equilibrium point of the ODE system coincides with the optimal solution of the SOCPs.
- Theorem 5: Starting with any initial point \((t_0, r_0)\), the state solution \( r(t) \) converges to the optimal solution \( r^* \) as the time go to infinity, i.e., \( \lim_{t \to \infty} r(t) = r^* \).
Neurodynamic optimization approach

Workflow

The flowchart summarizes how we obtain the saddle point \((x^*, y^*)\) of a stochastic two-player zero-sum game using neurodynamic optimization approach.

1. A stochastic two-player zero-sum game
2. The SOCP problem
3. The KKT system
4. The DNN model

Formation to DNN models

Initial point

\(r_0\)

Interval

\([0, T]\)

The DNN model

Numerical methods

Solution \(r(t)\)

Nash equilibrium contained in \(r(T)\)

Solution of DNN models
We compare our neurodynamic optimization approach with the conventional approach, the SCS method.

**Left:** The SCS method. The objective value with respect to iterations.

**Right:** The neurodynamic approach. The objective value with respect to time $t$ of the ODE system.

The SCS method solve the problem in a **Discrete** manner. The neurodynamic approach solve the problem in a **Time-continuous** manner.
The point \((s, u)\) is an approximate KKT point with \(\epsilon\)-error if it satisfies

\[
\left\| \nabla f (s) + \nabla g (s)^T u \right\| \leq \epsilon,
\]
\[
\| u \circ g(s) \| \leq \epsilon,
\]
\[
\| g (s)_+ \| \leq \epsilon,
\]
\[
\| u_- \| \leq \epsilon,
\]

\(\epsilon\)-error represents how well the solution \((s, u)\) satisfy the KKT conditions.
Numerical results

The $\epsilon$-error comparison between the SCS method and our neurodynamic optimization approach:

![Graph showing the comparison between SCS and DNN methods.]

**Figure:** The x-axis represents the number of iterations (for the SCS method) or the time $t$ (for the Neurodynamic approach). The y-axis represents the $\epsilon$ error.

The SCS method may stop or become slow after some iterations. The neurodynamic approach keep minimizing the $\epsilon$-error thanks to the global convergence theorem.
## Numerical results

Comparasion between Neurodynamic, SCS and CVXOPT

<table>
<thead>
<tr>
<th>Game size</th>
<th>Probability distribution</th>
<th>α</th>
<th>Neurodynamic</th>
<th>SCS</th>
<th>CVXOPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4, 4)</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>1.79</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.59</td>
<td>0.12</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>1.79</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.30</td>
<td>0.05</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>1.66</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.40</td>
<td>0.11</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(10, 10)</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>2.39</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.51</td>
<td>0.01</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>2.67</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.78</td>
<td>0.09</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>2.61</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.53</td>
<td>0.03</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(50, 50)</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>56.33</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.26</td>
<td>0.18</td>
<td>0.063</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>68.78</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.17</td>
<td>0.09</td>
<td>0.062</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>58.37</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.25</td>
<td>0.15</td>
<td>0.055</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>47.07</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.16</td>
<td>0.08</td>
<td>0.046</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(100, 100)</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>381.33</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.02</td>
<td>0.02</td>
<td>0.111</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>369.88</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.99</td>
<td>0.04</td>
<td>0.105</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>319.04</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td>0.02</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>359.95</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.04</td>
<td>0.04</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>8984.51</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.97</td>
<td>0.03</td>
<td>0.283</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(200, 200)</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.8</td>
<td>0.8</td>
<td>8862.24</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.99</td>
<td>0.04</td>
<td>0.281</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>8381.08</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.98</td>
<td>0.04</td>
<td>0.252</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>11218.82</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td>0.05</td>
<td>0.265</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ε-error</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Pros:** 1. Achieve a lower $\epsilon$-error 2. Solve the game of large size.
- **Cons:** Time consuming.
In this paper, we studied a neurodynamic approach to solve a two-player zero-sum game with stochastic linear constraints.

We show that the equilibrium point of the ODE system is the saddle point for the game.

We show that the ODE system can converge to the saddle point of the game.

We use this neurodynamic approach to solve the game of size up to $(200, 200)$. 
Thank you for your attention

Reference:


