



**HAL**  
open science

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

Dawen Wu, Abdel Lisser

► **To cite this version:**

Dawen Wu, Abdel Lisser. A dynamical neural network for solving stochastic two-player zero-sum games. 2021. hal-03345923

**HAL Id: hal-03345923**

**<https://hal.science/hal-03345923>**

Preprint submitted on 16 Sep 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Dawen Wu<sup>a,\*</sup>, Abdel Lisser<sup>a</sup>

<sup>a</sup>*Centralesupelec, Université Paris-Saclay*

---

## Abstract

This paper presents a dynamical neural network approach to solve stochastic two-players zero-sum game problems. The original problem is first transformed into an equivalent convex second-order cone programming problem. We develop a dynamical neural network model to solve the problem, where the model's equilibrium point corresponds to the optimal solution of the game problem. Further, we use a Lyapunov function to show that the equilibrium point of the neural network is globally asymptotically stable. Numerical results are given to show the performance of our approach.

*Keywords:* Stochastic Two-player Zero-sum Game, Saddle point, dynamical neural network

---

## 1. Introduction

In 1928, von Neumann (1928) studies the equilibrium concept in game theory and showed that there always exists a saddle point equilibrium for a finite action two-player zero-sum games. Nash et al. (1950) generalizes this result to n-player games, and shows that there always exists a Nash equilibrium for a finite action  
5 general games with a finite number of players. Charnes (1953) studies a new type of two-player zero-sum game where the mixed strategies of each player are constrained by linear inequalities. He shows that a saddle point equilibrium of such games can be obtained by solving a primal-dual pair of linear programs. Recently, Singh and Lisser (2019) study a stochastic version of two-player zero-sum game, namely chance constrained two-player zero-sum games. They show that a mixed strategy saddle point exists if the random vectors  
10 defining stochastic linear constraints follow elliptically symmetric distributions. As regards the solution of the problem, the saddle point of the two-player zero-sum game can be solved by linear programming. A two-player general sum game can be formulated as a linear complementarity problem (LCP), and Lemke and Howson (1964) gives method to solve it. For n-player general sum cases, Daskalakis et al. (2009) show that the problem of computing nash equilibriums is PPAD-Hard . van der Laan et al. (1987); Govindan and  
15 Wilson (2003); Blum et al. (2006) provide algorithms to solve such problems.

Hopfield and Tank (1985) bridge neural networks and optimization problems. Since then, many neural network models are proposed to solve varied optimization problems, e.g., linear programming by Wang (1993);

---

\*Corresponding author

*Email address:* dawen.wu@centralesupelec.fr, abdel.lisser@12s.centralesupelec.fr (Abdel Lisser)

Xia (1996b), quadratic programming by Xia (1996a); Nazemi (2014), nonlinear programming problems by Kennedy and Chua (1988); Forti et al. (2004); Nazemi and Tahmasbi (2013), minimax problem by Gao et al. (2004); Nazemi (2011), nonlinear complementarity problems by Liao et al. (2001); Dang et al. (2004); Nazemi and Sabeghi (2020); Feizi et al. (2021). The neural network model can be reformulated as a first order ordinary differential equation(ODE).

With regards to the solution of an ODE system with a given initial value, Runge-Kutta and backward differentiation approaches are commonly used, see Curtiss and Hirschfelder (1952); Gottlieb and Shu (1998). Python and Julia provide tools implementing such algorithms, see Virtanen et al. (2020); Rackauckas and Nie (2017). Furthermore, with the rapid development of machine learning, Chen et al. (2019); Raissi et al. (2019) study deep neural network to handle ODE systems.

The remaining of the paper is organized as follows. Section 2 shows the stochastic zero-sum game together with its second-order cone programming problem (SOCP, for short) reformulation. Section 3 gives the KKT conditions related to the SOCP problem, and the neural network reformulation. Numerical experiments are given in Section 4.

The following notations are used in the remainder of the paper.

- $x$  and  $y$  denote the strategies of player1 and player2 respectively.  $m$  and  $n$  denote the sizes of the action set of the player1 and the player2, respectively.
- $\mathcal{J}_1, \mathcal{J}_2$  denote the probabilistic constraints sets for player1 and player2, respectively.  $J_1, J_2$  denote the sets sizes.
- $A, \mu^1, \mu^2, \Sigma^1, \Sigma^2, b, d$  denotes the data for a stochastic two-players zero-sum game.  $A$  denotes the payoff matrix.  $\mu^1, \mu^2, \Sigma^1, \Sigma^2$  are the means and the variances of the probability distributions, respectively.  $\varphi_k^1(t^2)$  and  $\varphi_l^2(t^2)$  are the characteristic functions of the probability distributions, respectively.
- $\alpha^1$  and  $\alpha^2$  are the setting of the confidence level of the players 1 and 2, respectively.
- The  $\Psi_{\xi_l^2}^{-1}(\alpha_l^2), \Psi_{\xi_k^1}^{-1}(\alpha_k^1)$  are quantile functions of 1-dimensional distribution functions induced by characteristic functions  $\varphi_k^1(t^2)$  and  $\varphi_l^2(t^2)$ , respectively.
- $s = (y, v, \delta, \lambda)$  are the decision variables of the optimization problem.  $u$  is the dual variable of the optimization problem.  $r = (s, u) = (y, v, \delta, \lambda, u)$  are the variables of the neural network.
- $nr, ns, nu$  are the number of variables  $r, s, u$ , respectively. Moreover,  $nu$  also denotes the number of constraints of the optimization problem.
- $f(s) = f(y, v, \delta, \lambda)$  and  $g(s) = g(y, v, \delta, \lambda)$  are the objective function and the constraints of the optimization problem.  $f(s), g(s), \nabla f(s), \nabla g(s), \nabla^2 g(s)$  are abbreviated as  $f, g, \nabla f, \nabla g, \nabla^2 g$ .
- $\Phi(r) = \frac{dr}{dt}$  denotes the dynamical neural network.

50 **2. Problem formulation**

In this section, we present the stochastic two-player zero sum game with a chance constraint. A two player zero-sum game involves two persons called player 1 and player 2. These games are described by a matrix  $A$  with  $m$  rows and  $n$  columns. Matrix  $A$  represents the payoffs of player 1 and matrix  $-A$  represents the payoffs of player 2. Let  $I = \{1, 2, \dots, m\}$  be the action set of player 1 and  $J = \{1, 2, \dots, n\}$  be the action sets of player 2. We call pure strategies the actions in sets  $I$  and  $J$  for the players 1 and 2, respectively. We call mixed strategies of a given player, a probability distribution defined over his action set. Let  $X = \{x \in \mathbb{R}^m \mid \sum_i x_i = 1, x_i \geq 0, \forall i \in I\}$  and  $Y = \{y \in \mathbb{R}^n \mid \sum_j y_j = 1, y_j \geq 0, \forall j \in J\}$  the sets of mixed strategies of player 1 and player 2, respectively. The payoffs of player 1 and player 2 are defined by  $x^T A y$  and  $x^T (-A) y$  respectively, for a given strategy pair  $(x, y) \in X \times Y$ . von Neumann (1928) showed that there exists a saddle point equilibrium in mixed strategies in zero-sum games. Dantzig (1951) showed that the saddle point equilibrium is a solution of primal-dual pair of linear programs. Charnes (1953) studied a linear constrained two-player zero-sum game problem. For a given a strategy  $y$  of player 2, the objective of player 1 is to choose a strategy  $x$  which solves the linear programming problem (1). Similarly, the aim of player 2 is to choose a strategy  $y$  that solves problem (2) for given a strategy  $x$  of player 1.

$$\begin{aligned} & \max_x x^T A y \\ & \text{s.t.} \\ & Bx \leq b \\ & \mathbf{1}^T x = 1 \\ & x \geq 0, \end{aligned} \tag{1}$$

$$\begin{aligned} & \min_y x^T A y \\ & \text{s.t.} \\ & Dy \geq d \\ & \mathbf{1}^T y = 1 \\ & y \geq 0, \end{aligned} \tag{2}$$

where  $B \in \mathbb{R}^{J_1 \times m}, D \in \mathbb{R}^{J_2, n}, b \in \mathbb{R}^{J_1}, d \in \mathbb{R}^{J_2}$ . A strategy pair  $(x, y)$  is said to be a saddle point equilibrium for the above constrained zero-sum game if  $x$  is an optimal solution of (1) for a given  $y$ , and  $y$  is an optimal solution of (2) for the given  $x$ .

Singh and Lisser (2019) consider the problem where each row vector  $B_k$  and  $D_l$  of  $B$  and  $D$ , respectively, follows an elliptical distribution i.e.  $B_k^w \sim \text{Ellip}_m(\mu_k^1, \Sigma_k^1, \varphi_k^1)$  and  $D_l^w \sim \text{Ellip}_n(\mu_l^2, \Sigma_l^2, \varphi_l^2)$ . The  $\Psi_{\xi_l^2}^{-1}(\alpha_l^2), \Psi_{\xi_k^1}^{-1}(\alpha_k^1)$  are the quantile functions of 1-dimensional distribution functions induced by characteristic functions  $\varphi_k^1(t^2)$  and  $\varphi_l^2(t^2)$ , respectively. The chance constrained optimization problem can be written

as

$$\begin{aligned} & \max_x x^T A y \\ & \text{s.t.} \\ & P \{B_k^w x \leq b_k\} \geq \alpha_k^1, \quad \forall k \in \mathcal{J}_1 \end{aligned} \quad (3)$$

$$\begin{aligned} & \mathbf{1}^T x = 1 \\ & x \geq 0, \\ & \min_y x^T A y \\ & \text{s.t.} \\ & P \{D_l^w y \geq d_l\} \geq \alpha_l^2, \quad \forall l \in \mathcal{J}_2 \end{aligned} \quad (4)$$

$$\mathbf{1}^T y = 1$$

$$y \geq 0.$$

We use the SOCP reformulation from Henrion (2007) to rewrite the probabilistic constraints in (3) and (4) as follows

$$x^T \mu_k^1 + \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|(\Sigma_k^1)^{\frac{1}{2}} x\| \leq b_k, \quad \forall k \in \mathcal{J}_1, \quad (5)$$

$$-y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \|(\Sigma_l^2)^{\frac{1}{2}} y\| \leq -d_l, \quad \forall l \in \mathcal{J}_2. \quad (6)$$

We denote the stochastic two-players zero-sum game by  $G(\alpha)$  and the feasible strategy sets of the two players by  $S_1(\alpha^1)$  and  $S_2(\alpha^2)$ ,

$$S_1(\alpha^1) = \left\{ x \in \mathbb{R}^m \mid \mathbf{1}^T x = 1, x \geq 0, x^T \mu_k^1 + \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|(\Sigma_k^1)^{\frac{1}{2}} x\| \leq b_k, \quad \forall k \in \mathcal{J}_1 \right\}, \quad (7)$$

$$S_2(\alpha^2) = \left\{ y \in \mathbb{R}^n \mid \mathbf{1}^T y = 1, y \geq 0, -y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \|(\Sigma_l^2)^{\frac{1}{2}} y\| \leq -d_l, \quad \forall l \in \mathcal{J}_2 \right\}. \quad (8)$$

### Assumption 1.

1. The set  $S_1(\alpha^1)$  is strictly feasible, i.e., there exists an  $x \in \mathbb{R}^m$  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  $x \in \mathbb{R}^n$  which is a feasible point of  $S_2(\alpha^2)$  and the inequality constraints of  $S_2(\alpha^2)$  are strictly satisfied by  $y$ .

$(x^*, y^*)$  is called a saddle point equilibrium of  $G(\alpha)$ , if the following inequality holds:

$$x^T A y^* \leq x^{*T} A y^* \leq x^{*T} A y, \quad \forall x \in S_1(\alpha^1), y \in S_2(\alpha^2). \quad (9)$$

The following theorem shows the saddle point existence of the stochastic two-player zero-sum game  
65 problem.

**Theorem 1** (Singh and Lisser (2019), Theorem 3.5). *Consider a constrained zero-sum matrix game where the matrices  $B^w$  and  $D^w$  defining the constraints of both the players, respectively, are random. Let the row vectors  $B_k^w \sim \text{Ellip}_m(\mu_k^1, \Sigma_k^1, \varphi_k^1)$ ,  $k \in \mathcal{J}_1$ , and  $D_l^w \sim \text{Ellip}_n(\mu_l^2, \Sigma_l^2, \varphi_l^2)$ ,  $l \in \mathcal{J}_2$ . For all  $k$  and  $l$ ,  $\Sigma_k^1 \succ 0$  and  $\Sigma_l^2 \succ 0$ . Then, there exists a saddle point equilibrium for the game  $G(\alpha)$  for all  $\alpha \in (0.5, 1]^{\mathcal{J}_1} \times (0.5, 1]^{\mathcal{J}_2}$ .*

70 We refer the reader to Singh and Lisser (2019) for more details about the proof of this theorem and the related results.

**Proposition 1.** *The chance constrained optimization problems (3) and (4) can be reformulated as the following SOCP problems ( $\mathcal{P}$ ) and ( $\mathcal{D}$ ).*

$$\left. \begin{array}{l} \min_{y, v^1, (\delta_k^1)_{k \in \mathcal{J}_1}, \lambda^1} v^1 + \sum_{k \in \mathcal{J}_1} \lambda_k^1 b_k \\ \text{s.t.} \\ (i) Ay - \sum_{k \in \mathcal{J}_1} \lambda_k^1 \mu_k^1 - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k^1 \leq v^1 \mathbf{1}_m \\ (ii) -y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \left\| (\Sigma_l^2)^{\frac{1}{2}} y \right\| \leq -d_l, \quad \forall l \in \mathcal{J}_2 \\ (iii) \|\delta_k^1\| \leq \lambda_k^1 \Psi_{\xi_k^1}^{-1}(\alpha_k^1), \quad \forall k \in \mathcal{J}_1 \\ (iv) \mathbf{1}^T y = 1 \\ (v) y \geq 0 \\ (vi) \lambda_k^1 \geq 0, \quad \forall k \in \mathcal{J}_1 \end{array} \right\} (\mathcal{P})$$

$$\left. \begin{array}{l} \max_{x, v^2, (\delta_l^2)_{l \in \mathcal{J}_2}, \lambda^2} v^2 + \sum_{l \in \mathcal{J}_2} \lambda_l^2 d_l \\ \text{s.t.} \\ (i) A^T x - \sum_{l \in \mathcal{J}_2} \lambda_l^2 \mu_l^2 - \sum_{l \in \mathcal{J}_2} (\Sigma_l^2)^{\frac{1}{2}} \delta_l^2 \geq v^2 \mathbf{1}_n \\ (ii) x^T \mu_k^1 + \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \left\| (\Sigma_k^1)^{\frac{1}{2}} x \right\| \leq b_k, \quad \forall k \in \mathcal{J}_1 \\ (iii) \|\delta_l^2\| \leq \lambda_l^2 \Psi_{\xi_l^2}^{-1}(\alpha_l^2), \quad \forall l \in \mathcal{J}_2 \\ (iv) \mathbf{1}^T x = 1 \\ (v) x \geq 0 \\ (vi) \lambda_l^2 \geq 0, \quad \forall l \in \mathcal{J}_2 \end{array} \right\} (\mathcal{D})$$

*Proof.* We show the process that generate ( $\mathcal{P}$ ) from (3).

The chance constrained optimization problem (3) with the second-order cone reformulation (5) is

$$\begin{aligned}
& \max_x x^T Ay \\
& \text{s.t.} \\
& \text{(i)} x^T \mu_k^1 + \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|(\Sigma_k^1)^{\frac{1}{2}} x\| \leq b_k, \quad \forall k \in \mathcal{J}_1 \\
& \text{(iii)} \mathbf{1}^T x = 1 \\
& \text{(v)} x \geq 0.
\end{aligned} \tag{10}$$

Given a strategy  $y$  of player 2, the problem can be written as the following SOCP problem, where  $(t_k^1)_{k \in \mathcal{J}_1}$  are auxiliary variables,

$$\begin{aligned}
& \max_{x, (t_k^1)_{k \in \mathcal{J}_1}} x^T Ay \\
& \text{s.t.} \\
& \text{(i)} -x^T \mu_k^1 - \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|t_k\| + b_k \geq 0, \quad \forall k \in \mathcal{J}_1 \\
& \text{(ii)} t_k^1 - (\Sigma_k^1)^{\frac{1}{2}} x = 0, \quad \forall k \in \mathcal{J}_1 \\
& \text{(iii)} \mathbf{1}^T x = 1 \\
& \text{(iv)} x \geq 0
\end{aligned} \tag{11}$$

The saddle point of the lagrangian of (11) is

$$\min_{v^1, (\delta_k^1)_{k \in \mathcal{J}_1}, \lambda^1 \geq 0} \max_{x, (t_k^1)_{k \in \mathcal{J}_1}} \left[ x^T Ay + v^1 \mathbf{1}^T x + \sum_{k \in \mathcal{J}_1} (\delta_k^1)^T \left( t_k^1 - (\Sigma_k^1)^{\frac{1}{2}} x \right) + \sum_{k \in \mathcal{J}_1} \lambda_k^1 \left( -x^T \mu_k^1 - \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|t_k\| + b_k \right) \right] \tag{12}$$

For the fixed  $v^1, (\delta_k^1)_{k \in \mathcal{J}_1}, \lambda^1$ , we have

$$\begin{aligned}
& \max_{x, (t_k^1)_{k \in \mathcal{J}_1}} \left[ x^T Ay + v^1 \mathbf{1}^T x + \sum_{k \in \mathcal{J}_1} (\delta_k^1)^T \left( t_k^1 - (\Sigma_k^1)^{\frac{1}{2}} x \right) + \sum_{k \in \mathcal{J}_1} \lambda_k^1 \left( -x^T \mu_k^1 - \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|t_k\| + b_k \right) \right] \\
& = \max_x \left[ x^T \left( Ay - \sum_{k \in \mathcal{J}_1} \lambda_k^1 \mu_k^1 - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k^1 - v^1 \mathbf{1} \right) \right] + \max_{(t_k^1)_{k \in \mathcal{J}_1}} \left[ \sum_{k \in \mathcal{J}_1} \left( (\delta_k^1)^T t_k^1 - \lambda_k^1 \Psi_{\xi_k^1}^{-1}(\alpha_k^1) \|t_k\| \right) \right] + v^1 + \sum_{k \in \mathcal{J}_1} \lambda_k^1 b_k
\end{aligned} \tag{13}$$

The first and second max problem are unbound unless the following conditions hold,

$$Ay - \sum_{k \in \mathcal{J}_1} \lambda_k^1 \mu_k^1 - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k^1 \leq v^1 \mathbf{1} \tag{14}$$

$$\|\delta_k^1\| \leq \lambda_k^1 \Psi_{\xi_k^1}^{-1}(\alpha_k^1), \quad \forall k \in \mathcal{J}_1 \tag{15}$$

The lagrangian dual of the inner maximum problem is given by the following SOCP,

$$\begin{aligned}
& \min_{v^1, (\delta_k^1), \lambda^1} v^1 + \sum_{k \in \mathcal{J}_1} \lambda_k^1 b_k \\
& \text{s.t.} \\
& \text{(i)} Ay - \sum_{k \in \mathcal{J}_1} \lambda_k^1 \mu_k^1 - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k^1 \leq v^1 \mathbf{1}_m \\
& \text{(ii)} \|\delta_k^1\| \leq \lambda_k^1 \Psi_{\xi_k^1}^{-1}(\alpha_k^1), \forall k \in \mathcal{J}_1 \\
& \text{(iii)} \lambda_k^1 \geq 0, \forall k \in \mathcal{J}_1
\end{aligned} \tag{16}$$

With the second-order cone constraint (6) for  $y$ , we finally get the first SOCP ( $\mathcal{P}$ ).

$$\left. \begin{aligned}
& \min_{y, v^1, (\delta_k^1)_{k \in \mathcal{J}_1}, \lambda^1} v^1 + \sum_{k \in \mathcal{J}_1} \lambda_k^1 b_k \\
& \text{s.t.} \\
& \text{(i)} Ay - \sum_{k \in \mathcal{J}_1} \lambda_k^1 \mu_k^1 - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k^1 \leq v^1 \mathbf{1}_m \\
& \text{(ii)} -y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \left\| (\Sigma_l^2)^{\frac{1}{2}} y \right\| \leq -d_l, \quad \forall l \in \mathcal{J}_2 \\
& \text{(iii)} \|\delta_k^1\| \leq \lambda_k^1 \Psi_{\xi_k^1}^{-1}(\alpha_k^1), \quad \forall k \in \mathcal{J}_1 \\
& \text{(iv)} \mathbf{1}^T y = 1 \\
& \text{(v)} y \geq 0 \\
& \text{(vi)} \lambda_k^1 \geq 0, \quad \forall k \in \mathcal{J}_1
\end{aligned} \right\} (\mathcal{P})$$

The dual problem ( $\mathcal{D}$ ) can be generated similarly. □

75 The following theorem shows the existence of the saddle point for the chance constrained zero-sum game  $G(\alpha)$ .

**Theorem 2** (Singh and Lisser (2019), Theorem 3.7). *Consider a constrained zero-sum game where the matrices  $B^w$  and  $D^w$  defining the constraints of player 1 and player 2, respectively, are random. Let the row vector  $B^w \sim \text{Ellip}_m(\mu_k^1, \Sigma_k^1, \varphi_k^1)$ ,  $k \in \mathcal{J}_1$ , where  $\Sigma_k^1 \succ 0$ , and the row vector  $D_l^w \sim \text{Ellip}_n(\mu_l^2, \Sigma_l^2, \varphi_l^2)$ ,  $l \in \mathcal{J}_2$  where  $\Sigma_l^2 \succ 0$ . 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 there exist  $(v^{1*}, (\delta_k^{1*})_{k \in \mathcal{J}_1}, \lambda^{1*})$  and  $(v^{2*}, (\delta_l^{2*})_{l \in \mathcal{J}_2}, \lambda^{2*})$  such that  $(y^*, v^{1*}, (\delta_k^{1*})_{k \in \mathcal{J}_1}, \lambda^{1*})$  and  $(x^*, v^{2*}, (\delta_l^{2*})_{l \in \mathcal{J}_2}, \lambda^{2*})$  are optimal solutions of primal-dual pair of SOCPs ( $\mathcal{P}$ ) and ( $\mathcal{D}$ ), respectively.*

We refer the reader to Singh and Lisser (2019) for more details about the proof of this theorem and the related results. 85



### 3. Methodology

In this section, we study a neural network approach to solve the second-order cone programming problem given in section 2. We provide the necessary and sufficient KKT conditions of problem  $(\mathcal{P})$ . We use a neural network to solve the KKT conditions that the equilibrium point of the neural network is the satisfied KKT point. Then, we study the stability of the equilibrium point by analyzing a Lyapunov function.

We transform the equality constraint  $\mathbf{1}^T y = 1$  in  $(\mathcal{P})$  into inequality  $\mathbf{1}^T y - 1 \leq 0, 1 - \mathbf{1}^T y \leq 0$ . For sake of simplicity, we consider only the primal problem. Denote  $s = (y, v, \delta, \lambda) = (y, v^1, (\delta_k^1)_{k \in \mathcal{J}_1}, \lambda^1)$ , where  $\delta = (\delta_k^1)_{k \in \mathcal{J}_1} = [\delta_1^1{}^T, \dots, \delta_{J_1}^1{}^T]^T$  and  $\lambda = \lambda^1 = [\lambda_1^1, \dots, \lambda_{J_1}^1]^T$ . Such that, the optimization problem  $(\mathcal{P})$  can be simplified as

$$\begin{aligned} & \min_s f(s) \\ & \text{s.t.} \\ & g(s) \leq 0, \end{aligned} \tag{17}$$

where the objective function  $f : \mathbb{R}^{ns} \rightarrow \mathbb{R}$ , and the constraints  $g : \mathbb{R}^{ns} \rightarrow \mathbb{R}^{nu}$ .

#### 3.1. KKT conditions

Since the SOCP constraints of  $g(s)$  are not differentiable, we introduce the following perturbation  $\epsilon = 10^{-6}$ , i.e.,  $\sqrt{\|s\|^2 + \epsilon^2}$ . Thanks to this smoothness technique, the KKT conditions of the SOCP are necessary and sufficient optimality conditions.

The KKT conditions of the SOCP problem  $(\mathcal{P})$  are

$$\begin{aligned} & \nabla f(s) + \nabla g(s)^T u = 0 \\ & g(s) \leq 0, \quad u^T \geq 0, \quad u^T g(s) = 0 \end{aligned} \tag{18}$$

where the  $\nabla f, u, g, \nabla g$  are as follows

$$\nabla f(s) = \begin{bmatrix} \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial v} \\ \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ b \end{bmatrix} \tag{19}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_{41} \\ u_{42} \\ u_5 \\ u_6 \end{bmatrix} \tag{20}$$

$$g(s) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_{41} \\ g_{42} \\ g_5 \\ g_6 \end{bmatrix} = \begin{bmatrix} Ay - v\mathbf{1} - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k - \sum_{k \in \mathcal{J}_1} \lambda_k \mu_k^1 \\ (-y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \left\| (\Sigma_l^2)^{\frac{1}{2}} y \right\| + d_l)_{l \in \mathcal{J}_2} \\ (\|\delta_k\| - \lambda_k \Psi_{\xi_k^1}^{-1}(\alpha_k^1))_{k \in \mathcal{J}_1} \\ \mathbf{1}^T y - 1 \\ -\mathbf{1}^T y + 1 \\ -y \\ -\lambda \end{bmatrix} \quad (21)$$

$$\nabla g(s) = \begin{bmatrix} \frac{\partial g_1}{\partial y} & \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial \delta} & \frac{\partial g_1}{\partial \lambda} \\ \frac{\partial g_2}{\partial y} & \frac{\partial g_2}{\partial v} & \frac{\partial g_2}{\partial \delta} & \frac{\partial g_2}{\partial \lambda} \\ \frac{\partial g_3}{\partial y} & \frac{\partial g_3}{\partial v} & \frac{\partial g_3}{\partial \delta} & \frac{\partial g_3}{\partial \lambda} \\ \frac{\partial g_{41}}{\partial y} & \frac{\partial g_{41}}{\partial v} & \frac{\partial g_{41}}{\partial \delta} & \frac{\partial g_{41}}{\partial \lambda} \\ \frac{\partial g_{42}}{\partial y} & \frac{\partial g_{42}}{\partial v} & \frac{\partial g_{42}}{\partial \delta} & \frac{\partial g_{42}}{\partial \lambda} \\ \frac{\partial g_5}{\partial y} & \frac{\partial g_5}{\partial v} & \frac{\partial g_5}{\partial \delta} & \frac{\partial g_5}{\partial \lambda} \\ \frac{\partial g_6}{\partial y} & \frac{\partial g_6}{\partial v} & \frac{\partial g_6}{\partial \delta} & \frac{\partial g_6}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} A & -\mathbf{1} & -(\Sigma_k^1)^{\frac{1}{2}}_{k \in \mathcal{J}_1} & -(\mu_k^1)_{k \in \mathcal{J}_1} \\ (-\mu_l^2)^T + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \frac{\Sigma_l^{2 \frac{1}{2} T} \Sigma_l^{2 \frac{1}{2}} y}{\|\Sigma_l^{2 \frac{1}{2}} y\|} & 0 & 0 & 0 \\ 0 & 0 & (\frac{\delta_k}{\|\delta_k\|})_{k \in \mathcal{J}_1} & (-\Psi_{\xi_k^1}^{-1}(\alpha_k^1))_{k \in \mathcal{J}_1} \\ \mathbf{1}^T & 0 & 0 & 0 \\ -\mathbf{1}^T & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} \quad (22)$$

The stationarity, primal feasibility, dual feasibility, and complementary slackness can be written as follows

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ b \end{bmatrix} + \begin{bmatrix} A & -\mathbf{1} & -(\Sigma_k^1)^{\frac{1}{2}}_{k \in \mathcal{J}_1} & -(\mu_k^1)_{k \in \mathcal{J}_1} \\ (-\mu_l^2)^T + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \frac{\Sigma_l^{2 \frac{1}{2} T} \Sigma_l^{2 \frac{1}{2}} y}{\|\Sigma_l^{2 \frac{1}{2}} y\|} & 0 & 0 & 0 \\ 0 & 0 & (\frac{\delta_k}{\|\delta_k\|})_{k \in \mathcal{J}_1} & (-\Psi_{\xi_k^1}^{-1}(\alpha_k^1))_{k \in \mathcal{J}_1} \\ \mathbf{1}^T & 0 & 0 & 0 \\ -\mathbf{1}^T & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}^T \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_{41} \\ u_{42} \\ u_5 \\ u_6 \end{bmatrix} = 0 \quad (23)$$

$$\begin{bmatrix} Ay - v\mathbf{1} - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k - \sum_{k \in \mathcal{J}_1} \lambda_k \mu_k^1 \\ (-y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \left\| (\Sigma_l^2)^{\frac{1}{2}} y \right\| + d_l)_{l \in \mathcal{J}_2} \\ (\|\delta_k\| - \lambda_k \Psi_{\xi_k^1}^{-1}(\alpha_k^1))_{k \in \mathcal{J}_1} \\ \mathbf{1}^T y - 1 \\ -\mathbf{1}^T y + 1 \\ -y \\ -\lambda \end{bmatrix} \leq 0 \quad (24)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_{41} \\ u_{42} \\ u_5 \\ u_6 \end{bmatrix} \geq 0 \quad (25)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_{41} \\ u_{42} \\ u_5 \\ u_6 \end{bmatrix}^T \begin{bmatrix} Ay - v\mathbf{1} - \sum_{k \in \mathcal{J}_1} (\Sigma_k^1)^{\frac{1}{2}} \delta_k - \sum_{k \in \mathcal{J}_1} \lambda_k \mu_k^1 \\ (-y^T \mu_l^2 + \Psi_{\xi_l^2}^{-1}(\alpha_l^2) \left\| (\Sigma_l^2)^{\frac{1}{2}} y \right\| + d_l)_{l \in \mathcal{J}_2} \\ (\|\delta_k\| - \lambda_k \Psi_{\xi_k^1}^{-1}(\alpha_k^1))_{k \in \mathcal{J}_1} \\ \mathbf{1}^T y - 1 \\ -\mathbf{1}^T y + 1 \\ -y \\ -\lambda \end{bmatrix} = 0 \quad (26)$$

The four decision variables of problem  $(\mathcal{P})$ , namely  $y, v, \delta, \lambda$ , have  $n, 1, J_1 * n, J_1$  components, respectively. The function  $g$  is composed of  $g_1, g_2, g_3, g_{41}, g_{42}, g_5$ , and  $g_6$ , with  $m, J_2, J_1, 1, 1, n, J_1$  components, respectively. The gradient  $\nabla f$  is a  $(J_1 + 1) * (n + 1)$ -vector. The jacobian  $\nabla g$  is  $(2 + m + n + 2 * J_1 + J_2) \times (J_1 + 1) * (n + 1)$ -matrix.

### 3.2. Neural network model

We now propose a neural network model with a given initial value. Let  $r = (y, v, \delta, \lambda, u)^T$  be the variables of the neural network.

$$\frac{dr}{dt} = \begin{bmatrix} \frac{dy}{dt} \\ \frac{dv}{dt} \\ \frac{d\delta}{dt} \\ \frac{d\lambda}{dt} \\ \frac{du}{dt} \end{bmatrix} = \begin{bmatrix} -(\nabla f_y + \nabla g_y^T(u + g)^+) \\ -(\nabla f_v + \nabla g_v^T(u + g)^+) \\ -(\nabla f_\delta + \nabla g_\delta^T(u + g)^+) \\ -(\nabla f_\lambda + \nabla g_{\lambda^1}^T(u + g)^+) \\ (u + g)^+ - u \end{bmatrix} \quad (27)$$

$$r(t_0) = r_0 \quad (28)$$

The complexity for solving the neural network (27) is highly dependent on the number variable. The number of the decision variables  $y, v, \delta, \lambda$  is  $ns = (J_1 + 1) * (n + 1)$ , and the number of the dual variables  $\mu$  is  $nu = 2 + m + n + 2 * J_1 + J_2$ , leading to a total number of variables  $nr = 3 + m + 2N + (3 + n) * J_1 + J_2$  for the neural network.

**Theorem 3.** *The point  $r^* = (y^*, v^*, \delta^*, \lambda^*, u^*)^T$  is the equilibrium point of the neural network (27) if and only if it is also the KKT point of the SOCP problem.*

*Proof.* Let  $r^* = (y^*, v^*, \delta^*, \lambda^*, u^*)^T$  be the equilibrium of the neural network (27). It follows that  $\frac{dr^*}{dt} = 0$ ,

$$\begin{aligned}
-\left(\nabla f_y^* + \nabla g_y^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_v^* + \nabla g_v^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_\delta^* + \nabla g_\delta^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_\lambda^* + \nabla g_\lambda^{T^*}(u^* + g^*)^+\right) &= 0 \\
(u^* + g^*)^+ - u^* &= 0
\end{aligned} \tag{29}$$

Substituting the first four lines by the last line  $u^* = (u^* + g^*)^+$ , we have

$$\begin{aligned}
-\left(\nabla f_y^* + \nabla g_y^{T^*}u^*\right) &= 0 \\
-\left(\nabla f_v^* + \nabla g_v^{T^*}u^*\right) &= 0 \\
-\left(\nabla f_\delta^* + \nabla g_\delta^{T^*}u^*\right) &= 0 \\
-\left(\nabla f_\lambda^* + \nabla g_\lambda^{T^*}u^*\right) &= 0,
\end{aligned} \tag{30}$$

where the KKT conditions stationarity holds. Moreover,  $u^* = (u^* + g^*)^+$  result in

$$g^* \leq 0, \quad u^* \geq 0, \quad u^{*T}g^* = 0, \tag{31}$$

110 where the primal feasibility, the dual feasibility and the complementary slackness hold.

Conversely, let  $r^* = (y^*, v^*, \delta^*, \lambda^*, u^*)$  be the kkt point of the problem  $(\mathcal{P})$ , then we have

$$\begin{aligned}
\nabla f_y^* + \nabla g_y^{T^*}u^* &= 0 \\
\nabla f_v^* + \nabla g_v^{T^*}u^* &= 0 \\
\nabla f_{\delta^*} + \nabla g_\delta^{T^*}u^* &= 0 \\
\nabla f_\lambda^* + \nabla g_\lambda^{T^*}u^* &= 0,
\end{aligned} \tag{32}$$

$$g^* \leq 0, \quad u^* \geq 0, \quad u^{*T}g^* = 0. \tag{33}$$

Conditions (33) lead to  $u^* = (u^* + g^*)^+$ . By substituting this into (32), we obtain

$$\begin{aligned}
-\left(\nabla f_y^* + \nabla g_y^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_v^* + \nabla g_v^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_\delta^* + \nabla g_\delta^{T^*}(u^* + g^*)^+\right) &= 0 \\
-\left(\nabla f_\lambda^* + \nabla g_\lambda^{T^*}(u^* + g^*)^+\right) &= 0 \\
(u^* + g^*)^+ - u^* &= 0
\end{aligned} \tag{34}$$

which is the equilibrium point of the neural network. □

### 3.3. Stability analysis

In this subsection, we study the uniqueness and the stability of the equilibrium point.

115 **Lemma 4.** *The equilibrium point of the proposed neural network (27) is unique.*

*Proof.* Since the problem  $(\mathcal{P})$  has the unique optimal solution  $(y^*, v^*, \delta^*, \lambda^*)$ , the necessary and sufficient KKT conditions (18) have the corresponding unique solution. From Theorem 3, we see that the equilibrium point of the proposed neural network is a necessary and sufficient condition for being a KKT point (18). Thus the equilibrium point of the neural network is unique. □

120 **Lemma 5.** *For an initial value problem (27) and (28), there exists a unique continuous solution  $r(t)$ .*

*Proof.* Since  $g, \nabla f$  and  $\nabla g$  are locally Lipschitz continuous, and the operations  $+, \cdot, (\cdot)^+$  would not change the locally Lipschitz property, such that  $\nabla f + \nabla g^T(u+g)^+$  and  $(u+g)^+ - u$  are locally Lipschitz continuous. According to the Cauchy-Lipschitz theorem, the neural network (27) with an initial point  $r(t_0) = r_0$  has a unique solution  $r(t), t \in [t_0, T)$ , for some  $T > t_0$ . Additionally, if  $r(t)$  is globally bound, the interval  $[t_0, T]$  expand to  $[t_0, +\infty)$ . □

**Lemma 6.** *The Jacobian matrix  $\nabla\Phi(r), \forall r \in \mathbb{R}^{nr}$  is a negative semidefinite matrix.*

*Proof.* We separate the situations into three cases, depending on the different status of  $(u+g)^+ \in \mathbb{R}_+^{nu}$ , and show under all three situations  $\nabla\Phi(r)$  is negative semidefinite matrix.

For the case where  $(u+g)^+$  has zero and non-zero components, such that  $0 < p < nu$

$$(u+g)^+ = \underbrace{(u_1 + g_1, \dots, u_p + g_p)}_p, \underbrace{(0, \dots, 0)}_{nu-p}, \quad (35)$$

the jacobian matrix  $\nabla\Phi(r), \forall r \in \mathbb{R}^{nr}$  is

$$\nabla\Phi(r) = \begin{bmatrix} -(\nabla^2 f + \sum_{k=1}^p ((u_k + g_k) \nabla^2 g_k^p) + \nabla g^{pT} \nabla g^p) & -\nabla g^{pT} \\ \nabla g^p & S \end{bmatrix}, \quad (36)$$

where

$$S = \begin{bmatrix} 0_{p \times p} & 0_{p \times (nu-p)} \\ 0_{(nu-p) \times p} & -I_{(nu-p) \times (nu-p)} \end{bmatrix}. \quad (37)$$

130  $\nabla g$  and  $\nabla^2 g_k$  denote the Jacobian matrix of function  $g$  and the Hessian matrix of function  $g_k$ .  $\nabla g^p$  and  $\nabla^2 g_k^p$  are the same as  $\nabla g$  and  $\nabla^2 g_k$  for first  $p$  row but the remaining  $nu - p$  row are all zero.

Matrix  $\nabla g^{pT} \nabla g^p$  is positive semidefinite. Since the functions  $f$  and  $g$  are assumed to be convex and twice differentiable, the Hessian matrices  $\nabla^2 f$  and  $\nabla^2 g_k, k = 1, 2, \dots, p$ , are positive semidefinite matrices. Furthermore, the positive semidefiniteness of  $\nabla^2 g_k$  implies that  $\nabla^2 g_k^p$  is positive semidefinite matrix. Moreover, it is clear that matrix  $S$  is negative semidefinite matrix. Putting those all together, the Jacobian matrix  $\nabla \Phi$  is a negative semidefinite matrix.

For the case where  $(u + g)^+$  has all non-zero components, such that  $p = nu$

$$(u + g)^+ = (u_1 + g_1, \dots, u_{nu} + g_{nu}), \quad (38)$$

the jacobian matrix  $\nabla \Phi(r)$  is

$$\nabla \Phi(r) = \begin{bmatrix} -(\nabla^2 f + \sum_{k=1}^m ((u_k + g_k) \nabla^2 g_k) + \nabla g^T \nabla g) & -\nabla g^T \\ \nabla g & 0_{nu \times nu} \end{bmatrix}, \quad (39)$$

Similar to the previous case, it is to see that  $\nabla \Phi(r)$  is a  $(ns + nu) \times (ns + nu)$  negative semidefinite matrix.

For the case where  $(u + g)^+$  has all zero components, such that  $p = 0$

$$(u + g)^+ = (0, \dots, 0), \quad (40)$$

the jacobian matrix  $\nabla \Phi(r)$  is

$$\nabla \Phi(r) = \begin{bmatrix} -\nabla^2 f & 0_{ns \times nu} \\ 0_{nu \times ns} & -I_{nu \times nu} \end{bmatrix}. \quad (41)$$

In this case also, it is easy to see that  $\nabla \Phi(r)$  is a negative semidefinite matrix. This completes the proof.  $\square$

**Definition 1.** A mapping  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is said to be monotonic if:

$$(x - y)^T (F(x) - F(y)) \geq 0, \quad \forall x, y \in \mathbb{R}^n \quad (42)$$

**Lemma 7** (Ortega and Rheinboldt (2000)). A differentiable mapping  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is monotonic, if and only if the jacobian matrix  $\nabla F(x), \forall x \in \mathbb{R}^n$  is positive semidefinite.

**Theorem 8.** The equilibrium point  $r^* = (y^*, v^*, \delta^*, \lambda^*, u^*)$  of the proposed neural network (27) is globally asymptotically stable.

*Proof.* Consider the following Lyapunov function

$$E(r) = \|\Phi(r)\|^2 + \frac{1}{2} \|r - r^*\|^2. \quad (43)$$

$E(r)$  is a positive definite function because  $E(r^*) = 0$  and  $E(r) > 0, \quad \forall r \neq r^*$ .

$$\frac{d\Phi}{dt} = \frac{\partial\Phi}{\partial r} \frac{dr}{dt} = \nabla\Phi(r)\Phi(r) \quad (44)$$

$$\begin{aligned} \dot{E}(r(t)) &= \left(\frac{d\Phi}{dt}\right)^T \Phi + \Phi^T \left(\frac{d\Phi}{dt}\right) + (r - r^*)^T \frac{dr(t)}{dt} \\ &= \Phi^T (\nabla\Phi(r)^T + \nabla\Phi(r)) \Phi + (r - r^*)^T \Phi(r) \end{aligned} \quad (45)$$

By Lemma 6, we have

$$\Phi^T(r) (\nabla\Phi(r)^T + \nabla\Phi(r)) \Phi(r) \leq 0, \quad \forall r \neq r^*. \quad (46)$$

By lemma 7, we have

$$(r - r^*)^T (\Phi(r) - \Phi(r^*)) = (r - r^*)^T \Phi(r) \leq 0, \quad \forall r \neq r^*. \quad (47)$$

This means that  $\dot{E}(r(t)) \leq 0$ .

145 According to Lyapunov globally stable theorem, the equilibrium  $r^*$  of the neural network (27) is globally stable. Moreover, it follows from (27), (28), (46) and (47), that  $\Phi(r) = 0 \Leftrightarrow \dot{E}(r) = 0$ , which means  $\dot{E}(r) = 0$  is true only for the equilibrium point, such that  $\dot{E}(r)$  is a negative definite function. Therefore, the equilibrium point of the neural network is globally asymptotically stable. □

## 150 4. Numerical Experiments

In this section, we provide numerical results for using the neural network to solve the stochastic two-player zero-sum game problem. We use Python3.8, Scipy1.6, Numpy1.20 softwares run on i7-10610U, 1.8GHz processor with 16 GB RAM to solve our different game problems. We use the following method to study the quality of the KKT point.

**Definition 2.** *The point  $(s, u)$  is an approximate KKT point with  $\epsilon$  error if it satisfies*

$$\begin{aligned} \left\| \nabla f(s) + \nabla g(s)^T \mu \right\| &\leq \epsilon \\ |\min \{-g_i(s), u_i\}| &\leq \epsilon \quad \forall i = 1, \dots, nu \\ \|g(s)_+\| &\leq \epsilon \end{aligned} \quad (48)$$

155 We uniformly generate the data of the game, namely,  $A, b, d, \mu^1, \mu^2$ , i.e.,  $A \sim U(0, 10), b \sim U(7, 10), d \sim U(3, 6), \mu^1 \sim U(0, 10), \mu^2 \sim U(0, 10)$ . For the sake of simplicity, we only consider uniformly distributed

diagonal matrices  $\Sigma^1$  and  $\Sigma^2$ , i.e.,  $\Sigma^1, \Sigma^2 \sim U(0, 3)$ . By way of illustration, the following is a randomly generated game instance with  $4 \times 4$  payoff matrix.

$$\begin{aligned}
 A &= \begin{bmatrix} 5.86 & 6.30 & 5.38 & 1.05 \\ 3.06 & 6.06 & 2.48 & 2.03 \\ 6.36 & 1.46 & 9.32 & 3.79 \\ 3.83 & 0.97 & 4.09 & 2.16 \end{bmatrix}, b = \begin{bmatrix} 8.27 \\ 9.55 \\ 7.13 \end{bmatrix}, d = \begin{bmatrix} 3.02 \\ 5.54 \\ 4.89 \end{bmatrix}, \\
 \mu^1 &= \begin{bmatrix} 0.57 & 0.49 & 4.40 & 7.56 \\ 8.89 & 0.40 & 2.54 & 6.25 \\ 7.97 & 4.70 & 7.48 & 0.76 \end{bmatrix}, \mu^2 = \begin{bmatrix} 5.18 & 2.12 & 8.93 & 4.55 \\ 5.53 & 4.94 & 9.94 & 5.04 \\ 5.94 & 6.66 & 1.62 & 9.49 \end{bmatrix}, \\
 \Sigma_1^1 &= \begin{bmatrix} 1.05 & 0 & 0 & 0 \\ 0 & 0.86 & 0 & 0 \\ 0 & 0 & 2.80 & 0 \\ 0 & 0 & 0 & 1.59 \end{bmatrix}, \Sigma_2^1 = \begin{bmatrix} 2.11 & 0 & 0 & 0 \\ 0 & 1.66 & 0 & 0 \\ 0 & 0 & 0.10 & 0 \\ 0 & 0 & 0 & 2.59 \end{bmatrix}, \Sigma_3^1 = \begin{bmatrix} 0.24 & 0 & 0 & 0 \\ 0 & 0.55 & 0 & 0 \\ 0 & 0 & 2.32 & 0 \\ 0 & 0 & 0 & 2.38 \end{bmatrix}, \\
 \Sigma_1^2 &= \begin{bmatrix} 0.02 & 0 & 0 & 0 \\ 0 & 2.10 & 0 & 0 \\ 0 & 0 & 2.85 & 0 \\ 0 & 0 & 0 & 1.02 \end{bmatrix}, \Sigma_2^2 = \begin{bmatrix} 0.15 & 0 & 0 & 0 \\ 0 & 1.06 & 0 & 0 \\ 0 & 0 & 2.68 & 0 \\ 0 & 0 & 0 & 0.61 \end{bmatrix}, \Sigma_3^2 = \begin{bmatrix} 0.60 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 1.86 & 0 \\ 0 & 0 & 0 & 0.83 \end{bmatrix},
 \end{aligned}$$

Table 1 shows the experiment results for different game sizes, confidence level  $\alpha$ , and probability distributions. Each game size represents the action set size for both players, and each case has linear probability constraints. The data of the games are generated by the above method. We consider the Normal and Laplace distributions, and the confidence levels  $\alpha$  are 0.8 or 0.9. The number of iterations of the neural network algorithm is 100.

From table 1, we can see that the CPU time depends on the game size. When the game size increases, the number of variables in the neural network increases, which determines the CPU time. Further, the epsilon errors in our experiment are all below 0.06 which shows the high performances of our approach in terms of the quality of the neural network solutions.

## 5. Conclusion

In this paper, we studied a neural network model to solve the two-player zero-sum game with stochastic linear constraints problem. We reformulated our problem as an SOCP problem. We show that the equilibrium point of the neural network model is the optimal solution for the original problem. By using the Lyapunov stability theory, we prove the globally asymptotic stability and the uniqueness of the equilibrium point of the proposed neural network. Our numerical experiments show the performance of the neural network to solve



Game size	$\alpha$		Probability distribution	Neural network model			
	$\alpha_1$	$\alpha_2$		Neural network size	CPU time	Value	epsilon
4	0.8	0.8	Normal	39	1.19	3.80	0.01
	0.9	0.9			0.87	4.00	0.01
	0.8	0.8	Laplace		1.14	3.82	0.01
	0.9	0.9			0.86	4.11	0.00
10	0.8	0.8	Normal	75	2.08	4.36	0.06
	0.9	0.9			2.36	4.46	0.03
	0.8	0.8	Laplace		2.11	4.36	0.05
	0.9	0.9			1.95	4.55	0.01
50	0.8	0.8	Normal	315	35.69	5.24	0.05
	0.9	0.9			36.67	5.29	0.04
	0.8	0.8	Laplace		37.14	5.24	0.06
	0.9	0.9			44.29	5.34	0.02
100	0.8	0.8	Normal	615	249.40	5.06	0.02
	0.9	0.9			345.70	5.00	0.05
	0.8	0.8	Laplace		356.80	5.04	0.05
	0.9	0.9			352.70	5.04	0.04
200	0.8	0.8	Normal	1215	8182.92	4.98	0.03
	0.9	0.9			9202.68	5.03	0.03
	0.8	0.8	Laplace		10836.52	4.99	0.03
	0.9	0.9			7726.93	5.04	0.02

Table 1: Solving stochastic zero sum games by the neural network model

large size two-player zero-sum games with up to  $200 \times 200$  payoff matrix within reasonable CPU time.

## 175 Bibliography

Blum B, Shelton CR, Koller D. A continuation method for nash equilibria in structured games. *Journal of Artificial Intelligence Research* 2006;25:457–502.

Charnes A. Constrained games and linear programming. *Proceedings of the National Academy of Sciences of the United States of America* 1953;39(7):639.

180 Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud D. Neural ordinary differential equations. 2019. [arXiv:1806.07366](https://arxiv.org/abs/1806.07366).

Curtiss CF, Hirschfelder JO. Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America* 1952;38(3):235.

Dang C, Leung Y, Gao Xb, Chen Kz. Neural networks for nonlinear and mixed complementarity problems and their applications. *Neural Networks* 2004;17(2):271–83.

Dantzig G. A proof of the equivalence of the programming problem and the game problem. *Activity analysis of production and allocation* 1951;;330–5.

Daskalakis C, Goldberg PW, Papadimitriou CH. The complexity of computing a nash equilibrium. *SIAM Journal on Computing* 2009;39(1):195–259.

- 190 Feizi A, Nazemi A, Rabiei MR. Solving the stochastic support vector regression with probabilistic constraints by a high-performance neural network model. *Engineering with Computers* 2021;;1–16.
- Forti M, Nistri P, Quincampoix M. Generalized neural network for nonsmooth nonlinear programming problems. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2004;51(9):1741–54.
- Gao XB, Liao LZ, Xue W. A neural network for a class of convex quadratic minimax problems with con-  
195 straints. *IEEE transactions on neural networks* 2004;15(3):622–8.
- Gottlieb S, Shu CW. Total variation diminishing runge-kutta schemes. *Mathematics of computation* 1998;67(221):73–85.
- Govindan S, Wilson R. A global newton method to compute nash equilibria. *Journal of Economic Theory* 2003;110(1):65–86.
- 200 Henrion R. Structural properties of linear probabilistic constraints. *Optimization* 2007;56(4):425–40.
- Hopfield JJ, Tank DW. “neural” computation of decisions in optimization problems. *Biological cybernetics* 1985;52(3):141–52.
- Kennedy MP, Chua LO. Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems* 1988;35(5):554–62.
- 205 van der Laan G, Talman A, Van der Heyden L. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Mathematics of operations research* 1987;12(3):377–97.
- Lemke CE, Howson Jr JT. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics* 1964;12(2):413–23.
- 210 Liao LZ, Qi H, Qi L. Solving nonlinear complementarity problems with neural networks: a reformulation method approach. *Journal of Computational and Applied Mathematics* 2001;131(1-2):343–59.
- Nash JF, et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 1950;36(1):48–9.
- Nazemi A. A dynamical model for solving degenerate quadratic minimax problems with constraints. *Journal*  
215 *of Computational and Applied Mathematics* 2011;236(6):1282–95.
- Nazemi A. A neural network model for solving convex quadratic programming problems with some applica-  
tions. *Engineering Applications of Artificial Intelligence* 2014;32:54–62.
- Nazemi A, Sabeghi A. A new neural network framework for solving convex second-order cone constrained  
220 variational inequality problems with an application in multi-finger robot hands. *Journal of Experimental & Theoretical Artificial Intelligence* 2020;32(2):181–203.

- Nazemi A, Tahmasbi N. A high performance neural network model for solving chance constrained optimization problems. *Neurocomputing* 2013;121:540–50.
- von Neumann J. Zur theorie der gesellschaftsspiele. *Mathematische annalen* 1928;100(1):295–320.
- Ortega JM, Rheinboldt WC. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.
- 225 Rackauckas C, Nie Q. *Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia*. *Journal of Open Research Software* 2017;5(1).
- Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 2019;378:686–707.
- 230 Singh VV, Lisser A. A second-order cone programming formulation for two player zero-sum games with chance constraints. *European Journal of Operational Research* 2019;275(3):839–45.
- Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, et al. *Scipy 1.0: fundamental algorithms for scientific computing in python*. *Nature methods* 2020;17(3):261–72.
- 235 Wang J. Analysis and design of a recurrent neural network for linear programming. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 1993;40(9):613–8.
- Xia Y. A new neural network for solving linear and quadratic programming problems. *IEEE Transactions on Neural Networks* 1996a;7(6):1544–8. doi:10.1109/72.548188.
- Xia Y. A new neural network for solving linear programming problems and its application. *IEEE Transactions*  
240 *on Neural Networks* 1996b;7(2):525–9.