



# An algorithm for finding approximate Nash equilibria in bimatrix games

Lunshan Gao<sup>1,2</sup>

Published online: 10 August 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

This paper describes an algorithm for calculating approximately mixed Nash equilibria (NE) in bimatrix games. The algorithm fuzzifies the strategies with normalized possibility distributions. The fuzzification takes advantage of the piecewise linearity of possibility distributions and transforms the NE problem of bimatrix games into a reduced form. The algorithm is guaranteed to find approximate NE in bimatrix games and ensures that the approximate NE is the saddle point of expected payoff functions in the reduced form. The algorithm provides a method of determining how close an approximate NE is to a solution during computation. Numerical results show that the new algorithm is approximately seven-time faster than the Lemke–Howson (LH) algorithm when the game size is 96, and the value of approximation deviation can be as small as 0.1.

**Keywords** Nash equilibria (NE) · Approximate NE · Fuzzification · Triangular fuzzy number · Possibility distribution

## 1 Introduction

Calculating Nash equilibria (NE) in two-player games is polynomial parity argument directed version (PPAD) complete (Chen et al. 2009; Daskalakis et al. 2009). Chen et al. conjectured that there is a  $\mathcal{O}(n^{k+\epsilon^{-c}})$ -time algorithm for finding an  $\epsilon$ -NE in a two-player game for some constants  $c$  and  $k$ , where  $n$  is the number of strategies (Chen et al. 2009). Ortiz and Irfan proposed a fully polynomial time approximation scheme algorithm that is able to calculate an  $\epsilon$ -NE in graphical multi-hypermatrix games (Ortiz and Irfan 2017). They used a discretization scheme to discretize the probability space and the payoff space, and formed a gradually increased probability distribution for each player. The discretization scheme depends on the number of strategies. Rubinstein proved that computing  $\epsilon$ -NE in 2 player  $n \times n$  games requires  $n^{\log^{1-\mathcal{O}(1)}n}$  time (Rubinstein 2016). Fearnley et al. (2012) indicated that a well-supported  $\epsilon$ -NE in bimatrix games can be computed within less than two-third

approximate ratio. Kontogiannis and Spirakis used linear programming techniques and proved that finding  $\epsilon$ -NE in win-lose bimatrix games can achieve  $\epsilon = 0.5$ , and finding  $\epsilon$ -NE in arbitrary  $[0, 1]$ -bimatrix games can achieve  $\epsilon = 0.658$  approximate ratio (Kontogiannis and Spirakis 2007). Lipton et al. used  $k$ -uniformed mixed strategies to approximate the randomized strategies in  $[0, 1]$  and they indicated that the approximate ratio must be bigger than  $(12 \ln(n)/k)^{1/2}$  where  $n$  was the number of strategies and  $k \leq n$  was the number of nonzero values (Lipton et al. 2003). Bosse et al. proved that calculating an approximate NE in a bimatrix game can achieve  $\epsilon = 0.36392$  approximate ratio (Bosse et al. 2010).

A well-known efficient algorithm for solving bimatrix games is the Lemke–Howson algorithm (LH). However, there is no way of determining how close one is to a solution during computation due to the lack of an objective function in the LH algorithm (McKelvey and McLennan 1996). They also used sets of 100 random games to measure average computational time for the LH algorithm (McKelvey and McLennan 1996). The classical LH algorithm finds one exact NE in exponential time (Rubinstein 2016).

Fuzzy sets theory has been applied to matrix game theory (Bector and Chandra 2005; Campos et al. 1992; Nishizaki and Sakawa 2001; Vijay et al. 2005; Kacher and Larbanib 2006; Larbani 2009; Li and Zhang 2011; Gao 2012; Li 2016). The fuzzy sets approach to game theory is based on the fact that players are not able to exactly estimate payoffs in

Communicated by V. Loia.

✉ Lunshan Gao  
sgao@conestogac.on.ca; lgao@wlu.ca

<sup>1</sup> Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON, Canada

<sup>2</sup> School of Applied Computer Science and IT, Conestoga College, Kitchener, ON, Canada

noncooperative games due to inadequate information on the environment (Lar bani 2009; Li 2016). The imprecision and uncertainty are appropriately modeled with fuzzy sets (Li and Zhang 2011; Li 2016; Nishizaki and Sakawa 2001). Note that the gains and losses for the players should be identical for a zero-sum matrix game. However, the aforementioned approach cannot always guarantee that the gain-floor and loss-ceiling for the players in zero-sum matrix games are identical (Li 2016).

This paper describes a novel algorithm that achieves the following: (1) the transformation of the NE problem in bimatrix games into a reduced form using fuzzification; (2) the solutions of the algorithm are the saddle points of the expected payoff functions; (3) the performance of the new algorithm is superior to the LH algorithm; (4) the determination of the approximation deviation during computation. This paper is organized as follows: Sect. 2 gives concepts that are related to game theory and fuzzy sets theory. Section 3 describes the algorithm in detail. Section 4 analyzes the algorithm and determines the approximation deviations. Section 5 gives an example to demonstrate the procedure of the algorithm and numerical results. Section 6 gives a conclusion.

## 2 Preliminaries

**Definition 1 (Fuzzy number)** A fuzzy number is a fuzzy set that is defined in domain  $R$ . The membership function  $\lambda(t)$  of a fuzzy number is a continuous function and its range is  $[0, 1]$  such that  $0 \leq \lambda(t) \leq 1$  (Dijkman et al. 1983).

**Definition 2 (Triangular fuzzy number (TFN))** A triangular fuzzy number is a special case of fuzzy number, whose membership function forms a triangle. A TFN  $\lambda$  is usually denoted with three values such that  $\lambda = (d^l, d^m, d^r)$ , where  $d^m$  is called the mean value of the TFN  $\lambda$ , such that  $\lambda(d^m) = 1$ ,  $d^l$  and  $d^r$  are the lower and upper limits of the TFN, such that  $\lambda(d^l) = \lambda(d^r) = 0$  (Kaufmann and Gupta 1998).

**Definition 3 (Fuzzification)** Fuzzification is the process of assigning the numerical inputs of a system to fuzzy sets with some degree of membership function.

The concept of fuzzification is commonly used in fuzzy sets theory and fuzzy control theory (Gao 1999; Zimmermann 2011).

**Definition 4 (Possibility distribution)** Suppose  $x \in [0, 1]^n \subseteq R^n$ , where  $R^n$  is a  $n$ -dimensional Euclidean space. Each  $x_i \in x$  is mapped to a fuzzy number  $\lambda_i (i = 1, 2, \dots, n)$ . A possibility distribution associated with the variable  $x$  is defined as  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ , where the support of  $\lambda_i$  is in  $[0, 1]$  such that  $supp \lambda_i \in [0, 1] (i = 1, 2, \dots, n)$ . When

$\sum_{j=1}^n \lambda_j(t) > 0 (t \in [0, 1])$ , we say that  $\lambda$  is a suitable possibility distribution.

As Zadeh described in his initial paper, the sum of a possibility distribution is not necessarily 1 (Zadeh 1978). However, practical problems sometimes require the normalization of the possibility distributions. Researchers studied the normalization of possibility distributions (Dubois and Prade 2015; Zimmermann 2011). The definition of normalized possibility distribution is given as follows.

**Definition 5 (The normalization of possibility distribution)**

When fuzzy numbers  $\lambda_i (i = 1, 2, \dots, n)$  in the possibility distribution  $\lambda$  are TFNs and  $\lambda$  is a suitable possibility distribution, the normalization of a possibility distribution  $\mu(t)$  is defined as follows.

$$\mu(t) = \left( \frac{\lambda_1(t)}{\sum_{j=1}^n \lambda_j(t)}, \frac{\lambda_2(t)}{\sum_{j=1}^n \lambda_j(t)}, \dots, \frac{\lambda_n(t)}{\sum_{j=1}^n \lambda_j(t)} \right), \quad t \in [0, 1]. \tag{2.1}$$

where  $\mu_i(t) = \frac{\lambda_i(t)}{\sum_{j=1}^n \lambda_j(t)} (i = 1, 2, \dots, n), t \in [0, 1]$ .

## 3 The algorithm

### 3.1 The problem

**Definition 6 (Bimatrix games)** A bimatrix game is described as follows.

$$G = (2, S, u), S = (S_1, S_2), u = (u_1, u_2), \tag{3.1}$$

where  $S_1 = (s_1^1, s_2^1, \dots, s_k^1) (S_2 = (s_1^2, s_2^2, \dots, s_l^2))$  is a set of strategies for player 1 (2);  $u_i$  is the expected payoff function of player  $i (i = 1, 2)$  that is defined as follows.

$$\begin{cases} u_1(x, y) = x^T A y \\ u_2(x, y) = x^T B y \end{cases} \tag{3.2}$$

where  $x^T = (x_1, x_2, \dots, x_k)$  and  $x \in \Delta^k (\Delta^k$  is a  $k$ -dimensional standard simplex);  $y^T = (y_1, y_2, \dots, y_l)$  and  $y \in \Delta^l$ ;  $A, B$  is a  $k \times l$  matrix for player 1, player 2, respectively.

**Definition 7 (Nash equilibrium (NE))** The pair of strategies  $(x^0, y^0)$  is called a NE, if and only if

$$\begin{cases} u_1(x, y^0) \leq u_1(x^0, y^0), \forall x \in S_1 \\ u_2(x^0, y) \leq u_2(x^0, y^0), \forall y \in S_2 \end{cases} \tag{3.3}$$

**Definition 8 ( $\epsilon$ -NE) (Lipton et al. 2003)** For any  $\epsilon > 0$ , a pair of strategies  $(x^*, y^*)$  is an  $\epsilon$ -NE for the bimatrix game (3.2) if

- (i)  $u_1(x, y^*) \leq u_1(x^*, y^*) + \epsilon, \forall x \in S_1.$
- (ii)  $u_2(x^*, y) \leq u_2(x^*, y^*) + \epsilon, \forall y \in S_2.$

Suppose  $t \in R$  and  $s \in R$  are two free independent variables. If one can find two transformations such that,

$$x = \mu(t) \in \Delta^k, t \in R, \tag{3.4}$$

$$y = \eta(s) \in \Delta^l, s \in R. \tag{3.5}$$

Then, according to the parameterized methodology (Downey and Fellows 2013), (3.2) can be described as follows.

$$\begin{cases} u_1(\mu(t), \eta(s)) = (\mu(t))^T A \eta(s), (t \in R; s \in R), \\ u_2(\mu(t), \eta(s)) = (\mu(t))^T B \eta(s), (t \in R; s \in R). \end{cases} \tag{3.6}$$

The key question is how we can construct the two transformations (3.4) and (3.5). This paper proposes a method to construct these two transformations  $\mu(t)$  and  $\eta(s)$  by using the fuzzification of mixed strategies  $(x, y)$ .

### 3.2 The fuzzification

The expected payoff function of player 1 is  $u_1(x, y)$ .  $u_1(x, y)$  has two variables such that  $x \in \Delta^k$  and  $y \in \Delta^l$ , where  $\Delta^k(\Delta^l)$  is  $k(l)$ -dimensional standard simplex. The fuzzification is to assign a fuzzy number to each  $x_i$ . As a result, a set of fuzzy numbers is assigned to variable  $x \in R^k$ .

Note that a  $x = (x_1, x_2, \dots, x_k) \in R^k$  and  $x \in \Delta^k$ . That is,  $x_i \in [0, 1]$  can be fuzzified with a TFN  $\lambda_i(t)$  because of  $\lambda_i(t) \in [0, 1]$ . Therefore,  $x \in R^k$  can be represented with a set of TFNs.

However, the above fuzzification has a problem that  $\lambda \in \Delta^k$  is not guaranteed because  $x \in \Delta^k$ . In order to guarantee that the set of TFNs is over the standard simplex and to make the fuzzification reasonable, instead of assigning a set of TFNs  $\lambda$  directly to  $x$ , the fuzzification in this paper assigns the normalization of a possibility distribution to  $x$  such that,

$$x = \mu(t) = \frac{\lambda(t)}{\sum_{j=1}^k \lambda_j(t)}, (t \in D^1 = [0, 1]). \tag{3.7}$$

where  $\lambda_i = (d_i^l, d_i^m, d_i^r)$  is a TFN, and  $\lambda(t) = (\lambda_1(t), \dots, \lambda_k(t))$  is a set of TFNs. When a set of TFNs  $\lambda(t)$  is given in domain  $D^1 = [0, 1]$  ( $D^1$  means for player 1), the mean value  $d_i^m$  of TFN  $\lambda_i$  will distribute in domain  $D^1 = [0, 1]$ . Without loss of generality, we suppose that

$$0 \leq d_1^m \leq d_2^m \leq \dots \leq d_k^m \leq 1.$$

These mean values decompose the domain  $D^1 = [0, 1]$  into  $M$  intervals such that,

$$D^1 = \bigcup_{i=1}^M D_i^1, \tag{3.8}$$

where  $1 \leq M \leq k$  and  $D_1^1 = [d_1^l, d_1^m], D_M^1 = [d_{k-1}^m, d_k^r], D_i^1 = [d_{i-1}^m, d_i^m](i = 2, \dots, M - 1)$ . It is obvious that we have the following proposition. The proof of the proposition can be found in Gao (2017).

**Proposition 1** Suppose that  $\lambda_i(t)$  is a membership function of a TFN  $\lambda_i = (d_i^l, d_i^m, d_i^r)$ , then,  $\lambda_i(t)$  is a monotonic, continuous, and linear function in domain  $D_j^1(j = 1, 2, \dots, M)$ .

Based on proposition 1,  $\lambda_i(t)$  can be denoted as follows.

$$\begin{aligned} \lambda_i(t) &= p_i t + q_i, p_i \in R, q_i \in R (i = 1, 2, \dots, k), \\ t &\in D_j^1 (j = 1, 2, \dots, M) \end{aligned}$$

Equation (3.8) shows that domain  $D^1 = [0, 1]$  is decomposed by the mean values  $d_i^m$  into  $M$  sub-domains, where  $1 \leq M \leq k$ . The fuzzification can be denoted as follows.

$$x = \mu(t), x \in \Delta^k, \mu(t) \in \Delta^k, t \in D_i^1 (i = 1, 2, \dots, M). \tag{3.9}$$

where  $\mu(t)$  is defined in Eq. (3.7).

Similarly, the variable  $y \in R^l$  can be fuzzified with another normalization of a possibility distribution  $\eta(s)$  such that,

$$y = \eta(s) = \frac{\theta(s)}{\sum_{j=1}^l \theta_j(s)}, (s \in D^2 = [0, 1]). \tag{3.10}$$

where  $\theta_j = (g_j^l, g_j^m, g_j^r)$  is a TFN, and  $\eta(s) = (\eta_1(s), \dots, \eta_l(s))$  is a set of TFNs. When a set of TFNs  $\theta(s)$  is give in domain  $D^2 = [0, 1]$  ( $D^2$  means for player 2), the mean value  $g_j^m$  of  $\theta_j(s)$  will distribute in domain  $D^2 = [0, 1]$ . Without loss of generality, we suppose that

$$0 \leq g_1^m \leq g_2^m \leq \dots \leq g_l^m \leq 1.$$

These mean values of TFNs  $\theta_j$  decompose the domain  $D^2 = [0, 1]$  ( $D^2$  means for player 2) into  $L$  intervals such that,

$$D^2 = \bigcup_{j=1}^L D_j^2, \tag{3.11}$$

where  $1 \leq L \leq l$  and  $D_1^2 = [g_1^l, g_1^m], D_L^2 = [g_l^m, g_l^r], D_j^2 = [g_{j-1}^m, g_j^m](j = 2, \dots, L - 1)$ . Similarly, TFN  $\theta_j(s)$  can be defined as follows.

$$\begin{aligned} \theta_j(s) &= e_j s + h_j, e_j \in R, h_j \in R (j = 1, 2, \dots, l), \\ s &\in D_i^2 (i = 1, 2, \dots, L - 1). \end{aligned}$$

Equation (3.11) shows that domain  $D^2 = [0, 1]$  is decomposed by the mean values  $g_j^m$  into  $L$  sub-domains, where  $1 \leq L \leq l$ . The fuzzification can be denoted as follows.

$$y = \eta(s), y \in \Delta^l, \eta(s) \in \Delta^l, s \in D_j^2 (j = 1, 2, \dots, L). \tag{3.12}$$

where  $\eta(s)$  is defined in Eq. (3.10).

By substituting  $x, y$  in Eq. (3.2) with  $\mu(t), \eta(s)$ , respectively, the expected payoff function  $u_1(x, y)(u_2(x, y))$  in (3.2) can be transformed as follows in each domain  $D_i^1 \times D_j^2$ .

$$\begin{cases} u_1(\mu(t), \eta(s)) = (\mu(t))^T A \eta(s), t \in D_i^1, s \in D_j^2. \\ u_2(\mu(t), \eta(s)) = (\mu(t))^T B \eta(s), t \in D_i^1, s \in D_j^2. \end{cases} \tag{3.13}$$

It is clear that Eqs. (3.6) and (3.13) are similar. The only difference between the two formulas is that the domains are different. The domain in (3.6) is  $R \times R$ , and the domain in (3.13) is  $D_i^1 \times D_j^2 (i = 1, 2, \dots, M; j = 1, 2, \dots, L)$ .

In this paper, we use a technique that adds the two payoff functions  $u_1(x, y)$  and  $u_2(x, y)$  together to constitute a single payoff function  $u(x, y)$  such that,

$$\begin{aligned} u(\mu(t), \eta(s)) &= u_1(\mu(t), \eta(s)) + u_2(\mu(t), \eta(s)) \\ &= (\mu(t))^T (A + B) \eta(s), t \in D_i^1, s \in D_j^2. \end{aligned} \tag{3.14}$$

NE points are critical points that satisfy the following equations in domain  $D_i^1 \times D_j^2 (i = 1, 2, \dots, M; j = 1, 2, \dots, L)$ .

$$\begin{cases} \frac{\partial u(\mu(t), \eta(s))}{\partial t} = \frac{\partial u_1(\mu(t), \eta(s))}{\partial t} + \frac{\partial u_2(\mu(t), \eta(s))}{\partial t} = 0, t \in D_i^1, s \in D_j^2 \\ \frac{\partial u(\mu(t), \eta(s))}{\partial s} = \frac{\partial u_1(\mu(t), \eta(s))}{\partial s} + \frac{\partial u_2(\mu(t), \eta(s))}{\partial s} = 0, t \in D_i^1, s \in D_j^2. \end{cases} \tag{3.15}$$

Based on Eqs. (3.7) and (3.10), one can simplify Eq. (3.15) as follows.

$$\begin{cases} \frac{\partial u(\mu(t), \eta(s))}{\partial t} = K_1(t, s) V_1^T (A + B) \theta(s) = 0, t \in D_i^1, s \in D_j^2 \\ \frac{\partial u(\mu(t), \eta(s))}{\partial s} = K_2(t, s) V_2^T (A^T + B^T) \lambda(t) = 0, t \in D_i^1, s \in D_j^2 \end{cases} \tag{3.16}$$

where  $K_i(t, s) (i = 1, 2)$  and  $V_i (i = 1, 2)$  are denoted as follows.

$$K_1(t, s) = \frac{1}{(\sum_{i=1}^k \lambda_i(t))^2 \sum_{j=1}^l \theta_j(s)}, t \in D_i^1; s \in D_j^2 \tag{3.17}$$

$$K_2(t, s) = \frac{1}{(\sum_{i=1}^l \theta_i(s))^2 \sum_{j=1}^k \lambda_j(t)}, t \in D_i^1; s \in D_j^2 \tag{3.18}$$

$$V_1^T = \begin{pmatrix} p_1 \sum_{j=1}^k q_j - q_1 \sum_{j=1}^k p_j, \dots, p_k \sum_{j=1}^k q_j \\ -q_k \sum_{j=1}^k p_j \end{pmatrix} \tag{3.19}$$

$$V_2^T = \begin{pmatrix} e_1 \sum_{j=1}^l h_j - h_1 \sum_{j=1}^l e_j, \dots, e_l \sum_{j=1}^l h_j \\ -h_l \sum_{j=1}^l e_j \end{pmatrix}. \tag{3.20}$$

Based on the definition of membership functions of TFNs, when  $\forall t \in R, \lambda_i(t) \geq 0$  and  $\forall s \in R, \theta_j(s) \geq 0$ . If the fuzzifications for variables  $x$  and  $y$  are reasonable, then the sum of all membership functions of TFNs will be larger than zero such that  $\sum_{i=1}^k \lambda_i(t) > 0$  and  $\sum_{j=1}^l \theta_j(s) > 0$ . If  $K_i(t, s) > 0 (i = 1, 2)$ , then the right hand of Eq. (3.16) can be simplified as follows.

$$\begin{cases} V_1^T (A + B) \theta(s) = 0, t \in D_i^1, s \in D_j^2, \\ V_2^T (A^T + B^T) \lambda(t) = 0, t \in D_i^1, s \in D_j^2. \end{cases} \tag{3.21}$$

When possibility distributions  $\lambda(t) = (\lambda_1(t), \dots, \lambda_k(t)) \in R^k$  and  $\theta(s) = (\theta_1(s), \dots, \theta_l(s)) \in R^l$  are given, where  $\lambda_i(t) = p_i t + q_i (i = 1, 2, \dots, k)$  and  $\theta_j(s) = e_j s + h_j (j = 1, 2, \dots, l)$ , the coefficients  $p_i, q_i (i = 1, 2, \dots, k)$  and  $e_j, h_j (j = 1, 2, \dots, l)$  are known. The elements of  $V_i (i = 1, 2)$  are constant. The elements of  $\lambda(t)$  are linear functions of  $t$ . The elements of  $\theta(s)$  are linear functions of  $s$ . Therefore, the solution of Eq. (3.21) can be explicitly denoted as follows.

$$t^* = \frac{V_2^T (A^T + B^T) Q}{V_2^T (A^T + B^T) P}, \quad s^* = \frac{V_1^T (A + B) H}{V_1^T (A + B) E}. \tag{3.22}$$

where  $E, H, P$ , and  $Q$  are as follows.

$$E^T = (e_1, e_2, \dots, e_l), \quad H^T = (-h_1, -h_2, \dots, -h_l), \tag{3.23}$$

$$P^T = (p_1, p_2, \dots, p_k), \quad Q^T = (-q_1, -q_2, \dots, -q_k). \tag{3.24}$$

The sub-domain  $D_i^1(D_j^2)$  is divided by the mean value  $d_i^m(g_j^m)$  of TFN  $\lambda_i(\eta_j)$ . The necessary and sufficient conditions of Eq. (3.22) being a solution of Eq. (3.21) is as follows.

$$\begin{cases} t^* \in D_i^1, \text{ such that } d_{i-1}^m \leq t^* \leq d_i^m, \\ s^* \in D_j^2, \text{ such that } g_{j-1}^m \leq s^* \leq g_j^m. \end{cases} \tag{3.25}$$

That is,  $(t^*, s^*) \in D_i^1 \times D_j^2 (i = 1, 2, \dots, M; j = 1, 2, \dots, L)$ . If either  $t^*$  or  $s^*$  does not satisfy the inequality

(3.25), then, Eq. (3.21) does not have a solution in  $D_i^1 \times D_j^2$  and the algorithm will try the next sub-domain (Fig. 2). If  $(t^*, s^*)$  satisfies the inequalities in (3.25), then  $(t^*, s^*)$  is a solution of Eq. (3.16).

The diagram of the algorithm is depicted in Fig. 1. First, the mixed strategies  $(x, y)$  are fuzzified with the normalized possibility distributions defined in Eqs. (3.7) and (3.10). Secondly, the algorithm calculates the potential critical point  $(t^*, s^*)$  with Eq. (3.22). Finally, the algorithm verifies if  $(t^*, s^*)$  satisfies the inequality (3.25) to determine if  $(t^*, s^*)$  is a solution of Eq. (3.16). In case if there is no solution is found in all the sub-domains  $D_i^1 \times D_j^2 (i = 1, 2, \dots, M; j = 1, 2, \dots, L)$ , the algorithm is able to adjust the fuzzification and try the procedure again until an approximate NE is found.

The details of the algorithm are described in Algorithm 1.

**Algorithm 1**

Step 1. Fuzzify mixed strategies  $x$  and  $y$  with the normalized possibility distributions  $\mu(t) \in \Delta^k$  and  $\eta(s) \in \Delta^l$ . Decide the numbers  $M, L$ , and domain  $D_i^1 (i = 1, 2, \dots, M)$  and domain  $D_j^2 (j = 1, 2, \dots, L)$ .  
**while**  $M > 0$  **do**  
    **while**  $L > 0$  **do**  
        Step 2. Verify  $K_1(t, s) > 0$  and  $K_2(t, s) > 0$  based on (3.17) and (3.18). If either  $K_1(t, s) \leq 0$  or  $K_2(t, s) \leq 0$ , then go to step 1.  
        Step 3. Calculate  $V_1$  in  $D_i^1$  with equation (3.19),  $V_2$  in  $D_j^2$  with equation (3.20), and  $\lambda(t)$  in  $D_i^1$  and  $\theta(s)$  in  $D_j^2$ .  
        Step 4. Calculate vectors  $E$  and  $H$  with equation (3.23), and vectors  $P$  and  $Q$  with equation (3.24).  
        Step 5. Based on equation (3.22), calculate solution  $(t^*, s^*)$ . If the inequality (3.25) is not satisfied, then go to Step 2.  
        Step 6. Verify inequalities (4.3) and (4.4) are satisfied. If not, go to step 2.  
        Step 7. Calculate  $(x^*, y^*)$  with equations (3.7) and (3.10).  
        Step 8. Calculate the approximation parameter  $\epsilon_i (i = 1, 2)$  in  $D_i^1 \times D_j^2$  with equation (4.10) and (4.11).  
    **end while**  
**end while**  
Step 9. Calculate  $u_i(x^*, y^*) (i = 1, 2)$ .  
**return**  $(x^*, y^*)$  and  $u_i(x^*, y^*) (i = 1, 2)$

**3.3 The analysis of the fuzzification**

From a conceptual point of view, the proposed algorithm might be considered as a type of divide-and-conquer algorithms. However, it differs from the divide-and-conquer algorithms because it requires neither a base case for sub-problems nor recursive operations.

The algorithm decomposes domain  $D^1 \times D^2 = [0, 1] \times [0, 1]$  into  $M \times L$  sub-domains, and then finds a critical point in each sub-domain  $D_i^1 \times D_j^2$  (Fig. 2). The purpose of the decomposition is to take advantage of the piecewise linearity of possibility distributions. The algorithm calculates  $(t^*, s^*)$  with Eq. (3.22) in each sub-domain. Then, it verifies if  $(t^*, s^*)$  satisfies the inequalities in (3.25) and the inequalities (4.3)

and (4.4) in Sect. 4. If the verification passes, then the algorithm finds a saddle point of the expected payoff functions in the reduced forms in sub-domain  $D_i^1 \times D_j^2 (i = 1, \dots, k; j = 1, \dots, l)$ .

The advantages of the fuzzification are the simplification of the NE problem of bimatrix games and the encapsulation of the standard simplex of strategies  $x$  and  $y$  in possibility distributions.

**4 The analysis of the algorithm**

First, we prove that if  $(t^*, s^*)$  is a solution of Eq. (3.16), then  $(\mu(t^*), \eta(s^*))$  can be a saddle point of function  $u(\mu(t), \eta(s))$  in domain  $D_i^1 \times D_j^2$ . Secondly, we prove that a saddle point of function  $u(\mu(t), \eta(s))$  can also be a saddle point of function  $u_1(\mu(t), \eta(s))$  and function  $u_2(\mu(t), \eta(s))$ . Finally, we discuss how to determine the approximation parameter  $\epsilon$ .

**Theorem 1** *If  $(t^*, s^*)$  is a solution of Eq. (3.16) in domain  $D_i^1 \times D_j^2$  and  $\frac{\partial^2 u}{\partial t \partial s} \Big|_{t=t^*, s=s^*} \neq 0$ , then  $(\mu(t^*), \eta(s^*))$  is a saddle point of function  $u(\mu(t), \eta(s))$  that is denoted in Eq. (3.14).*

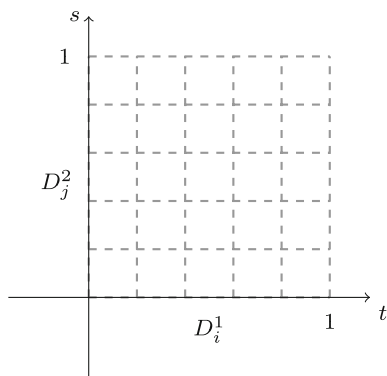
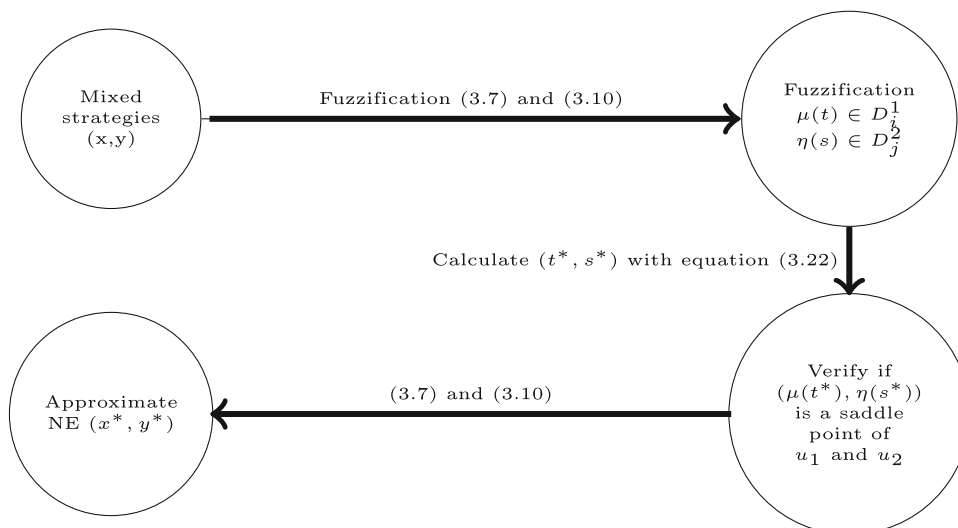
**Proof** Since  $(t^*, s^*)$  is a solution of Eq. (3.16), we have the following.

$$\frac{\partial u(\mu(t), \eta(s))}{\partial t} \Big|_{t=t^*, s=s^*} = 0 \text{ and } \frac{\partial u(\mu(t), \eta(s))}{\partial s} \Big|_{t=t^*, s=s^*} = 0.$$

The Taylor expansion of  $u(\mu(t), \eta(s))$  at  $(t^*, s^*)$  in domain  $D_i^1 \times D_j^2$  is as follows.

$$\begin{aligned} u(\mu(t), \eta(s)) &= u(\mu(t^*), \eta(s^*)) + \frac{\partial u}{\partial t} \Big|_{t=t^*, s=s^*} (t - t^*) \\ &\quad + \frac{\partial u}{\partial s} \Big|_{t=t^*, s=s^*} (s - s^*) \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial t^2} \Big|_{t=t^*, s=s^*} (t - t^*)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial s^2} \Big|_{t=t^*, s=s^*} (s - s^*)^2 \\ &\quad + \frac{\partial^2 u}{\partial t \partial s} \Big|_{t=t^*, s=s^*} (t - t^*)(s - s^*) \\ &\quad + H.O.T., t \in D_i^1, s \in D_j^2. \\ &= u(\mu(t^*), \eta(s^*)) + \frac{1}{2} \frac{\partial^2 u}{\partial t^2} \Big|_{t=t^*, s=s^*} (t - t^*)^2 \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial s^2} \Big|_{t=t^*, s=s^*} (s - s^*)^2 \\ &\quad + \frac{\partial^2 u}{\partial t \partial s} \Big|_{t=t^*, s=s^*} (t - t^*)(s - s^*), t \in D_i^1, s \in D_j^2 \\ &= u(\mu(t^*), \eta(s^*)) \\ &\quad + \frac{1}{2} (t - t^*, s - s^*) \begin{pmatrix} \frac{\partial^2 u}{\partial t^2} & \frac{\partial^2 u}{\partial t \partial s} \\ \frac{\partial^2 u}{\partial t \partial s} & \frac{\partial^2 u}{\partial s^2} \end{pmatrix} \Big|_{t=t^*, s=s^*} \begin{pmatrix} t - t^* \\ s - s^* \end{pmatrix}, \\ &\quad t \in D_i^1, s \in D_j^2. \end{aligned} \tag{4.1}$$

**Fig. 1** The diagram of the algorithm



**Fig. 2** An example of sub-domain  $D_i^1 \times D_j^2$

If we calculate the second-order partial derivative of the first equation in (3.16), we have the following.

$$\frac{\partial^2 u(\mu(t), \eta(s))}{\partial t^2} \Big|_{\substack{t=t^* \\ s=s^*}} = \frac{\partial K_1(t, s)}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} V_1^T (A + B)\theta(s^*) = 0. \tag{4.2}$$

The determinant of the Hessian matrix is as follows.

$$\begin{aligned} \text{DetH} &= \frac{\partial^2 u}{\partial t^2} \Big|_{\substack{t=t^* \\ s=s^*}} \frac{\partial^2 u}{\partial s^2} \Big|_{\substack{t=t^* \\ s=s^*}} - \left( \frac{\partial^2 u}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 \\ &= - \left( \frac{\partial^2 u}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2. \end{aligned}$$

If  $\frac{\partial^2 u(\mu(t), \eta(s))}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \neq 0$ , then  $\text{DetH} < 0$ . That is,  $(\mu(t^*), \eta(s^*))$  is a saddle point of  $u(\mu(t), \eta(s))$ .  $\square$

According to Theorem 1, if  $(t^*, s^*)$  is a solution of Eq. (3.16) in  $D_i^1 \times D_j^2$  and the mixed second-order partial

derivative at  $(t^*, s^*)$  is not zero, then  $(\mu(t^*), \theta(s^*))$  is a saddle point of function  $u(\mu(t), \eta(s))$ . It is interesting to know under what conditions the solution  $(t^*, s^*)$  will be a critical point of the payoff function  $u_1(\mu(t), \eta(s))$  and the payoff function  $u_2(\mu(t), \eta(s))$ . We have the following proposition.

**Proposition 2** If  $(t^*, s^*)$  is a solution of Eq. (3.16) in domain  $D_i^1 \times D_j^2$ , and the following inequalities are satisfied,

$$\frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \frac{\partial u_2(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \geq 0 \tag{4.3}$$

$$\frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \frac{\partial u_2(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \geq 0. \tag{4.4}$$

Then,  $(t^*, s^*)$  is a critical point of  $u_1(\mu(t), \eta(s))$  and  $u_2(\mu(t), \eta(s))$  such that

$$\begin{cases} \frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = 0 \\ \frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = 0. \end{cases} \quad \begin{cases} \frac{\partial u_2(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = 0 \\ \frac{\partial u_2(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = 0. \end{cases}$$

**Proof** Since  $(t^*, s^*)$  is a solution of Eq. (3.16) in domain  $D_i^1 \times D_j^2$ , based on Eq. (3.15), we have the following.

$$\begin{cases} \left( \frac{\partial u}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 = \left( \frac{\partial u_1}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 + \left( \frac{\partial u_2}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 + 2 \frac{\partial u_1}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} \frac{\partial u_2}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = 0. \\ \left( \frac{\partial u}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 = \left( \frac{\partial u_1}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 + \left( \frac{\partial u_2}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right)^2 + 2 \frac{\partial u_1}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \frac{\partial u_2}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = 0. \end{cases} \tag{4.5}$$

According to inequalities (4.3) and (4.4), if and only if the following is satisfied,

$$\begin{cases} \frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = 0 \\ \frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = 0, \end{cases} \quad \begin{cases} \frac{\partial u_2(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = 0 \\ \frac{\partial u_2(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = 0, \end{cases} ,$$

then Eq. (4.5) is satisfied. That is, the solution  $(t^*, s^*)$  is a critical point of function  $u_1(\mu(t), \eta(s))$  and function  $u_2(\mu(t), \eta(s))$ .  $\square$

The following theorem proves that  $(\mu(t^*), \eta(s^*))$  is a saddle point of functions  $u_1(\mu(t), \eta(s))$  and  $u_2(\mu(t), \eta(s))$  if  $(\mu(t^*), \eta(s^*))$  is a saddle point of function  $u(\mu(t), \eta(s))$  and the inequalities (4.3) and (4.4) are satisfied.

**Theorem 2** Suppose that  $(t^*, s^*)$  is a solution of Eq. (3.16) in domain  $D_i^1 \times D_j^2$  and inequalities (4.3) and (4.4) are satisfied.

If  $\frac{\partial^2 u_i}{\partial t \partial s} \Big|_{t=t^*, s=s^*} \neq 0 (i = 1, 2)$ , then,  $(\mu(t^*), \eta(s^*))$  is a saddle point of function  $u_1(\mu(t), \eta(s))$  and  $u_2(\mu(t), \eta(s))$

**Proof** First, we prove that  $(\mu(t^*), \eta(s^*))$  is a saddle point of  $u_1(\mu(t), \eta(s))$ . The Taylor expansion of  $u_1(\mu(t), \eta(s))$  at  $(t^*, s^*)$  in domain  $D_i^1 \times D_j^2$  is as follows.

$$\begin{aligned}
 u_1(\mu(t), \eta(s)) &= u_1(\mu(t^*), \eta(s^*)) \\
 &+ \frac{\partial u_1}{\partial t} \Big|_{t=t^*, s=s^*} (t - t^*) + \frac{\partial u_1}{\partial s} \Big|_{t=t^*, s=s^*} (s - s^*) \\
 &+ \frac{1}{2} \frac{\partial^2 u_1}{\partial t^2} \Big|_{t=t^*, s=s^*} (t - t^*)^2 \\
 &+ \frac{1}{2} \frac{\partial^2 u_1}{\partial s^2} \Big|_{t=t^*, s=s^*} (s - s^*)^2 \\
 &+ \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{t=t^*, s=s^*} (t - t^*)(s - s^*) \\
 &+ H.O.T., t \in D_i^1, s \in D_j^2. \tag{4.6}
 \end{aligned}$$

Since inequalities (4.3) and (4.4) are satisfied and according to Proposition 2,  $(t^*, s^*)$  is a critical point of  $u_1(\mu(t), \eta(s))$  such that,

$$\begin{cases} \frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{t=t^*, s=s^*} = K_1(t, s) V_1^T A \theta(s) \Big|_{t=t^*, s=s^*} = 0 \\ \frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{t=t^*, s=s^*} = K_2(t, s) V_2^T A \lambda(t) \Big|_{t=t^*, s=s^*} = 0. \end{cases} \tag{4.7}$$

If we apply Eqs. (4.7)–(4.6), then Eq. (4.6) is simplified as follows.

$$\begin{aligned}
 u_1(\mu(t), \eta(s)) &= u_1(\mu(t^*), \eta(s^*)) + \frac{1}{2} \frac{\partial^2 u_1}{\partial t^2} \Big|_{t=t^*, s=s^*} (t - t^*)^2 \\
 &+ \frac{1}{2} \frac{\partial^2 u_1}{\partial s^2} \Big|_{t=t^*, s=s^*} (s - s^*)^2 + \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{t=t^*, s=s^*} (t - t^*)(s - s^*), \\
 &= u_1(\mu(t^*), \eta(s^*)) \\
 &+ \frac{1}{2} (t - t^*, s - s^*) \begin{pmatrix} \frac{\partial^2 u_1}{\partial t^2} & \frac{\partial^2 u_1}{\partial t \partial s} \\ \frac{\partial^2 u_1}{\partial t \partial s} & \frac{\partial^2 u_1}{\partial s^2} \end{pmatrix} \Big|_{t=t^*, s=s^*} \begin{pmatrix} t - t^* \\ s - s^* \end{pmatrix}, \\
 t \in D_i^1, s \in D_j^2. \tag{4.8}
 \end{aligned}$$

If we calculate the second-order partial derivative of Eq. (4.7), we get the following.

$$\begin{cases} \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial t^2} \Big|_{t=t^*, s=s^*} = \frac{\partial K_1(t, s)}{\partial t} V_1^T A \theta(s) \Big|_{t=t^*, s=s^*} = 0 \\ \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial s^2} \Big|_{t=t^*, s=s^*} = \frac{\partial K_2(t, s)}{\partial s} V_2^T A \lambda(t) \Big|_{t=t^*, s=s^*} = 0. \end{cases} \tag{4.9}$$

If we apply Eq. (4.9) to Hessian matrix in Eq. (4.8), then the determinant of the Hessian matrix is as follows.

$$\begin{aligned}
 \text{DetH} &= \frac{\partial^2 u_1}{\partial t^2} \Big|_{t=t^*, s=s^*} \frac{\partial^2 u_1}{\partial s^2} \Big|_{t=t^*, s=s^*} - \left( \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{t=t^*, s=s^*} \right)^2 \\
 &= - \left( \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{t=t^*, s=s^*} \right)^2.
 \end{aligned}$$

If  $\frac{\partial^2 u_1}{\partial t \partial s} \Big|_{t=t^*, s=s^*} \neq 0$ , then  $\text{DetH} < 0$ . That is,  $(\mu(t^*), \eta(s^*))$  is a saddle point of  $u_1(\mu(t), \eta(s))$ .

Similarly, we can prove that  $(\mu(t^*), \eta(s^*))$  is a saddle point of function  $u_2(\mu(t), \eta(s))$  if the conditions in Theorem 2 are satisfied.  $\square$

### 4.1 The analysis of approximation ratio

Without loss of generality, suppose that domain  $D^1 = [0, 1] (D^2 = [0, 1])$  is equally decomposed with possibility distributions  $\lambda(t)(\theta(s))$  such that  $|t - t^*| \leq 1/M$  for  $t, t^* \in D_i^1 (|s - s^*| \leq 1/L$  for  $s, s^* \in D_j^2)$ . We have the following theorem.

**Theorem 3** If  $(t^*, s^*)$  is a solution of Eq. (3.16) in domain  $D_i^1 \times D_j^2$ , and inequalities (4.3) and (4.4) are satisfied, then there exist  $\epsilon_1 > 0$  and  $\epsilon_2 > 0$  such that,

$$\epsilon_1 = \max \left\{ \frac{1}{ML} K_1(t^*, s^*) | V_1^T A E |, \frac{1}{ML} K_2(t^*, s^*) | V_2^T A^T P | \right\} \tag{4.10}$$

$$\epsilon_2 = \max \left\{ \frac{1}{ML} K_1(t^*, s^*) | V_1^T B E |, \frac{1}{ML} K_2(t^*, s^*) | V_2^T B^T P | \right\} \tag{4.11}$$

and

$$|u_1(\mu(t), \eta(s)) - u_1(\mu(t^*), \eta(s^*))| \leq \epsilon_1, \forall t \in D_i^1; \forall s \in D_j^2. \tag{4.12}$$

$$|u_2(\mu(t), \eta(s)) - u_2(\mu(t^*), \eta(s^*))| \leq \epsilon_2, \forall t \in D_i^1; \forall s \in D_j^2. \tag{4.13}$$

where  $K_1(t, s)(K_2(t, s))$  is defined in Eqs. (3.17), (3.18),  $V_1(V_2)$  is defined in Eq. (3.19), (3.20), and  $E, P$  is denoted in Eqs. (3.23), (3.24), respectively.

**Proof** First, we prove that there exists  $\epsilon_1$  in (4.10) and  $u_1(\mu(t), \eta(s))$  satisfies (4.12). Since  $(t^*, s^*)$  is a solution of

Eq. (3.16) in domain  $D_i^1 \times D_j^2$  and inequalities (4.3) and (4.4) are satisfied, based on Proposition 2, we have the following.

$$\begin{cases} \frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = K_1(t, s) V_1^T A \theta(s) \Big|_{\substack{t=t^* \\ s=s^*}} = 0, t \in D_i^1, s \in D_j^2, \\ \frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = K_2(t, s) V_2^T A^T \lambda(t) \Big|_{\substack{t=t^* \\ s=s^*}} = 0, t \in D_i^1, s \in D_j^2 \end{cases} \quad (4.14)$$

The Taylor expansion of  $u_1(\mu(t), \eta(s))$  at  $(t^*, s^*)$  in domain  $D_i^1 \times D_j^2$  is as follows.

$$\begin{aligned} u_1(\mu(t), \eta(s)) &= u_1(\mu(t^*), \eta(s^*)) + \frac{1}{2} \frac{\partial^2 u_1}{\partial t^2} \Big|_{\substack{t=t^* \\ s=s^*}} (t - t^*)^2 \\ &+ \frac{1}{2} \frac{\partial^2 u_1}{\partial s^2} \Big|_{\substack{t=t^* \\ s=s^*}} (s - s^*)^2 + \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} (t - t^*)(s - s^*) \\ &+ H.O.T., t \in D_i^1, s \in D_j^2. \end{aligned} \quad (4.15)$$

If we calculate the second-order partial derivative of Eq. (4.14), then we have the following.

$$\begin{cases} \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial t^2} \Big|_{\substack{t=t^* \\ s=s^*}} = \frac{\partial K_1(t, s)}{\partial t} V_1^T A \theta(s) \Big|_{\substack{t=t^* \\ s=s^*}} = 0, t \in D_i^1, j \in D_j^2. \\ \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial s^2} \Big|_{\substack{t=t^* \\ s=s^*}} = \frac{\partial K_2(t, s)}{\partial s} V_2^T A^T \lambda(t) \Big|_{\substack{t=t^* \\ s=s^*}} = 0, t \in D_i^1, j \in D_j^2. \end{cases}$$

Therefore, Eq. (4.15) is simplified as follows.

$$\begin{aligned} u_1(\mu(t), \eta(s)) &= u_1(\mu(t^*), \eta(s^*)) \\ &+ \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} (t - t^*)(s - s^*), t \in D_i^1, s \in D_j^2. \end{aligned} \quad (4.16)$$

If we calculate the mixed second-order partial derivative of the first equation in (4.14), then we have the following.

$$\begin{aligned} \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} &= \frac{\partial K_1(t, s)}{\partial s} V_1^T A \theta(s) \Big|_{\substack{t=t^* \\ s=s^*}} \\ &+ K_1(t, s) V_1^T A \frac{d\theta(s)}{ds} \Big|_{\substack{t=t^* \\ s=s^*}} \\ &= K_1(t, s) V_1^T A \frac{d\theta(s)}{ds} \Big|_{\substack{t=t^* \\ s=s^*}} \\ &= K_1(t^*, s^*) V_1^T A E. \end{aligned} \quad (4.17)$$

If we calculate the mixed second-order partial derivative of the second equation in (4.14), then we have the following.

$$\begin{aligned} \frac{\partial^2 u_1(\mu(t), \eta(s))}{\partial s \partial t} \Big|_{\substack{t=t^* \\ s=s^*}} &= \frac{\partial K_2(t, s)}{\partial s} V_2^T A^T \lambda(t) \Big|_{\substack{t=t^* \\ s=s^*}} \\ &+ K_2(t, s) V_2^T A^T \frac{d\lambda(t)}{dt} \Big|_{\substack{t=t^* \\ s=s^*}} \\ &= K_2(t, s) V_2^T A^T \frac{d\lambda(t)}{dt} \Big|_{\substack{t=t^* \\ s=s^*}} \\ &= K_2(t^*, s^*) V_2^T A^T P. \end{aligned} \quad (4.18)$$

Since  $|t - t^*| \leq 1/M$  and  $|s - s^*| \leq 1/L$ , based on Eq. (4.16), the following inequality is derived.

$$\begin{aligned} &|u_1(\mu(t), \eta(s)) - u_1(\mu(t^*), \eta(s^*))| \\ &\leq \frac{1}{ML} \left| \frac{\partial^2 u_1}{\partial t \partial s} \Big|_{\substack{t=t^* \\ s=s^*}} \right|, t \in D_i^1, s \in D_j^2. \end{aligned} \quad (4.19)$$

Generally speaking, the right hands of Eqs. (4.17) and (4.18) are identical. However, because they are approximations of the second-order partial derivatives, they may be not equal to each other. Therefore, we choose  $\epsilon_1$  as the maximum value of them such that,

$$\epsilon_1 = \max \left\{ \frac{1}{ML} K_1(t^*, s^*) |V_1^T A E|, \frac{1}{ML} K_2(t^*, s^*) |V_2^T A^T P| \right\}.$$

We have the following inequality.

$$|u_1(\mu(t), \eta(s)) - u_1(\mu(t^*), \eta(s^*))| \leq \epsilon_1, t \in D_i^1, s \in D_j^2.$$

Similarly, we can prove that there exist  $\epsilon_2$  in (4.11) and  $u_2(\mu(t), \eta(s))$  satisfies inequality (4.13).  $\square$

Based on the definition of  $\epsilon$ -NE (Lipton et al. 2003), Theorem 3 indicates that  $u_i(\mu(t^*), \eta(s^*)) (i = 1, 2)$  is an  $\epsilon$ -NE. For a given fuzzification, the value of  $\epsilon_i (i = 1, 2)$  is determined. Theoretically,  $\epsilon_i (i = 1, 2)$  can be any value in interval  $[0, +\infty)$ . Theorem 3 provides the measurement of approximation deviation. Numerical results in Sect. 5.1 show that the value of  $\epsilon_i (i = 1, 2)$  can be as small as 0.1.

### 5 Example and numerical results

First, an example of  $5 \times 5$  bimatrix game is given to demonstrate the procedure of the novel algorithm. Secondly, we compare the new algorithm with the LH algorithm by using numerical experiments.

**Example 1** Find  $\epsilon$ -NE in the following bimatrix game.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Step 1. Fuzzify strategies  $x$  and  $y$ . A possibility distribution for player 1 is given as follows.

$$\begin{aligned} \lambda_1 &= (0, 0, 1), \lambda_2 = (0, 1, 1), \lambda_3 = (0, 1, 1), \\ \lambda_4 &= (0, 0, 1), \lambda_5 = (0, 1, 1). \end{aligned}$$



The membership functions  $\lambda_i(t)$  of TFNs  $\lambda_i (i = 1, \dots, 5)$  are as follows.

$$\lambda_1(t) = \begin{cases} 1-t & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases}, \lambda_2(t) = \begin{cases} t & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases},$$

$$\lambda_3(t) = \begin{cases} t & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases},$$

$$\lambda_4(t) = \begin{cases} 1-t & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases}, \lambda_5(t) = \begin{cases} t & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases}.$$

A possibility distribution for player 2 is given as follows.

$$\theta_1 = (0, 1, 1), \theta_2 = (0, 1, 1), \theta_3 = (0, 0, 1),$$

$$\theta_4 = (0, 1, 1), \theta_5 = (0, 1, 1).$$

The membership functions  $\eta_j(s)$  of TFNs  $\theta_j (j = 1, \dots, 5)$  are as follows.

$$\theta_1(s) = \begin{cases} s & s \in [0, 1] \\ 0 & \text{otherwise} \end{cases}, \theta_2(s) = \begin{cases} s & s \in [0, 1] \\ 0 & \text{otherwise} \end{cases},$$

$$\theta_3(s) = \begin{cases} 1-s & s \in [0, 1] \\ 0 & \text{otherwise} \end{cases},$$

$$\theta_4(s) = \begin{cases} s & s \in [0, 1] \\ 0 & \text{otherwise} \end{cases}, \theta_5(s) = \begin{cases} s & s \in [0, 1] \\ 0 & \text{otherwise} \end{cases}.$$

Step 2. Verify  $K_i(t, s) > 0 (i = 1, 2)$ .

$$K_1(t, s) = \frac{1}{(2+t)^2 \times (1+3s)} > 0, t \in [0, 1], s \in [0, 1]$$

$$K_2(t, s) = \frac{1}{(1+3s)^2 \times (2+t)} > 0, t \in [0, 1], s \in [0, 1].$$

Step 3. Calculate  $V_1$  and  $V_2$ . Based on equations (3.19) and (3.20), we can calculate  $V_i (i = 1, 2)$  as follows.

$$V_1^T = (-3, 2, 2, -3, 2), \quad V_2^T = (1, 1, -4, 1, 1).$$

Step 4. Calculate  $E, H, P$  and  $Q$  with Eq. (3.23) and (3.24).

$$P^T = (-1, 1, 1, -1, 1), Q = (-1, 0, 0, -1, 0)$$

$$E^T = (1, 1, -1, 1, 1), H^T = (0, 0, -1, 0, 0).$$

Step 5. Calculate  $(t^*, s^*)$  based on Eq. (3.22).  $(t^*, s^*) = (0.75, 0.4)$ . It is clear that inequalities in (3.25) are satisfied such that  $0.75 \in [0, 1]$  and  $0.4 \in [0, 1]$ .

Step 6. Verify that inequalities (4.3) and (4.4) are satisfied.

$$\frac{\partial u_1(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = K_1(t^*, s^*) V_1^T A \theta(s^*) = 0$$

$$\frac{\partial u_2(\mu(t), \eta(s))}{\partial t} \Big|_{\substack{t=t^* \\ s=s^*}} = K_1(t^*, s^*) V_1^T B \theta(s^*) = 0$$

$$\frac{\partial u_1(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = K_2(t^*, s^*) V_2^T A^T \lambda(t^*) \approx -0.0939$$

$$\frac{\partial u_2(\mu(t), \eta(s))}{\partial s} \Big|_{\substack{t=t^* \\ s=s^*}} = K_2(t^*, s^*) V_2^T B^T \lambda(t^*) \approx -0.0939.$$

Step 7. Calculate  $(x^*, y^*)$ . Based on Eqs. (3.7) and (3.10), we get the following.

$$x^* = (0.0909, 0.2727, 0.2727, 0.0909, 0.2727)$$

$$y^* = (0.1818, 0.1818, 0.2727, 0.1818, 0.1818).$$

Step 8. Calculate the parameter  $\epsilon_i (i = 1, 2)$ . Based on Eqs. (4.10) and (4.11), we can get  $\epsilon_1 \approx 0.3005$  and  $\epsilon_2 \approx 0.3005$ .

Step 9. Calculate  $u_i(\mu(t^*), \eta(s^*)) (i = 1, 2)$ .  $u_1(\mu(t^*), \eta(s^*)) \approx 0.2727$  and  $u_2(\mu(t^*), \eta(s^*)) \approx 0.2727$ .

### 5.1 Numerical results

The algorithm is implemented in C++ with Microsoft Visual Studio 2015 and run in a computer with one Intel Core i5-7200U CPU (2.50GHz) and 8 GB RAM.

First, we compare the new algorithm with the LH algorithm with one instance for each size 12, 16, 24, 32, 48, 64, and 96 of bimatrix games. The instance is generated with C++ generator of random numbers uniformly distributed in the interval [0,1]. Matrices  $A = (a_{ij})$  and  $B = (b_{ij})$  are generated independently for each size of bimatrix games. The data of the LH algorithm in Table 1 are obtained with the Windows version gambit-lcp.exe of GamBit (McKelvey et al. 2014). The data of the new algorithm in Table 1 are obtained when  $\epsilon_i = 0.1 (i = 1, 2)$ .

Table 1 shows that the new algorithm is superior to the LH algorithm (gambit-lcp.exe) for each size game from the perspective of computational time. It may not be prudent to conclude that one algorithm is better than another by using only one instance because the computational times differ from one instance to another.

To have unbiased comparison, we use sets of random bimatrix games to measure the average of the computational times (McKelvey and McLennan 1996). We generate 100 sets of random games by using C++ generator of random numbers uniformly distributed in the interval [0,1]. In all cases, the computational time is measured when one approximate NE is found. The average computational times for each game are shown in Table 2. The data are visualized in Fig. 3.

When the game size is smaller than or equal to 24, the computational time of the new algorithm is similar to that of the classical LH algorithm. However, when game sizes increase, the computational time of the classical LH algorithm surges sharply, while the computational time of the new algorithm

**Table 1** The comparison of the new algorithm with the LH algorithm

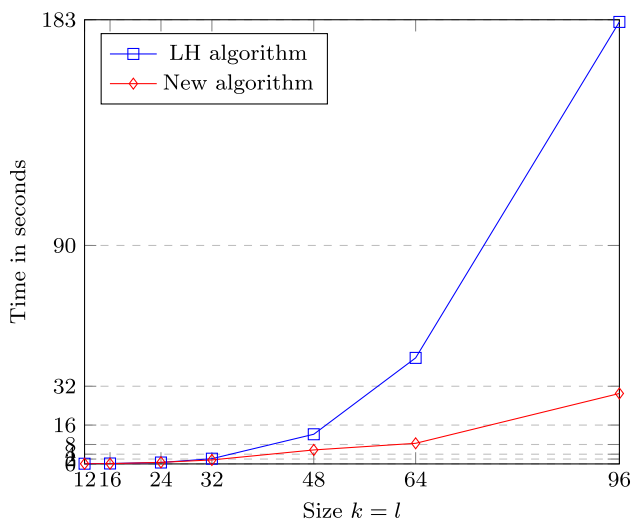
Size ( $k = l$ )	LH algorithm (gambit-lcp.exe)		New algorithm( $\epsilon_1 = \epsilon_2 = 0.1$ )	
	Number of NE	Time (S)	Number of NE	Time (S)
12	9	1.16	4	0.70
16	61	6.91	24	2.06
24	195	216.91	73	9.11
32	3205	8075.7	173	32.34
48	3411	2131.04	726	168.04
64	3411	3312.63	2339	508.85
96	745(1)	11163.66	3876	2543.19

(1) gambit-lcp.exe was terminated by an error bad-alloc when size is 96

**Table 2** Average times on 100 random games in seconds

Size ( $k = l$ )	LH algorithm	New algorithm
12	0.084	0.062
16	0.1737	0.103
24	0.5772	0.653
32	2.153	1.64
48	12.22	5.76
64	43.68	8.49
96	182.1	29.02

The data of the LH algorithm are cited from Table 2 in paper McKelvey and McLennan (1996)

**Fig. 3** Computational time comparison with the LH algorithm

keeps its growth rate within three. For example, the computational time of the classical LH is more than sixfold of that of the new algorithm when the game size is 96.

## 6 Conclusion

This paper proposed a new algorithm for computing approximately mixed NE in bimatrix games. Advantages of the algorithm are that: the encapsulation of the standard simplex into the normalized possibility distributions by using the fuzzifications, the transformation of the NE problem in bimatrix games into a reduced form by taking advantage of the piecewise linearity of possibility distributions, the solutions of the new algorithm are the saddle points of payoff functions, and the new algorithm provides a method of determining how close an approximate NE is to a solution and the numerical results show that the value of approximation deviation can be as small as 0.1.

The example in Sect. 5 demonstrates that the algorithm is efficient and easy to use when proper possibility distributions are defined. The numerical results show that the new algorithm is superior to the LH algorithm from the perspectives of computational time. The essential and vital step of the algorithm is the technique of fuzzifying mixed strategies  $(x, y)$  appropriately. Appropriate fuzzifications result in both better performance and accuracy for calculating mixed NE.

This work leaves many potentials for future research, for instance, comprehensive comparisons with other algorithms such as semidefinite programming and heuristic algorithms, and the theoretical analysis of the new algorithm from the perspective of computational complexity theory. For example, it is necessary to categorize the complexity class of the new algorithm.

**Acknowledgements** The author thanks anonymous reviewers and Gao J and appreciates their constructive suggestions and comments.

## Compliance with ethical standards

**Conflict of interest** The authors have no financial disclosures of conflicts to report.

## References

- Bector CR, Chandra S (2005) Fuzzy mathematical programming and fuzzy matrix games. Springer, Berlin
- Bosse H, Byrka J, Markakis E (2010) New algorithms for approximate Nash equilibria in bimatrix games. *Theor Comput Sci* 411(1):164–173
- Campos L, Gonzalez A, Vila MA (1992) On the use of the ranking function approach to solve fuzzy matrix games in direct way. *Fuzzy Set Syst* 49:193–203
- Chen X, Deng X, Teng SH (2009) Setting the complexity of computing two-player Nash equilibria. *J ACM* 56(3):1–57
- Daskalakis C, Mehta A, Papadimitriou C (2009) A note on approximate Nash equilibria. *Theor Comput Sci* 410:1581–1588
- Dijkman JG, van Haeringen H, DE Lange SJ (1983) Fuzzy numbers. *J Math Anal Appl* 92:301–341
- Downey R, Fellows MR (2013) Fundamentals of parameterized complexity. Springer, Berlin
- Dubois D, Prade H (2015) Practical methods for constructing possibility distributions. *Int J Intell Syst* 31(3):215–239
- Fearnley J, Goldberg PW, Savani R, Sorensen TB (2012) Approximate well-supported Nash equilibria below two-thirds. In: International symposium on algorithmic game theory, SAGT: algorithmic game theory, pp 108–119
- Gao L (2017) The application of possibility distribution for solving standard quadratic optimization problems. *Comput Inf Sci*. <https://doi.org/10.5539/cis.v10n3p60>
- Gao L (2012) The discussion of applications of the fuzzy average to matrix game theory. In: The proceeding of CCECE2012
- Gao L (1999) The fuzzy arithmetic mean. *Fuzzy Sets Syst* 107:335–348
- Kacher F, Larbanib M (2006) Solution concept for a non-cooperative game with fuzzy parameters. *Int Game Theory Rev* 8:489–498
- Kaufmann A, Gupta MM (1998) Fuzzy mathematical models in engineering and management science. Elsevier, Amsterdam
- Kontogiannis SC, Spirakis PG (2007) Efficient algorithms for constant well supported approximate equilibria in bimatrix games. In: Proceeding of ICALP, pp 595–606
- Larban M (2009) Non cooperative fuzzy games in normal form: a survey. *Fuzzy Sets Syst* 160:3184–3210
- Li C, Zhang Q (2011) Nash equilibrium strategy for fuzzy non-cooperative games. *Fuzzy Sets Syst* 176:46–55
- Li DF (2016) Linear programming models and methods of matrix games with payoffs of triangular fuzzy numbers, studies in fuzziness and soft computing 328. Springer, Berlin
- Lipton RJ, Markakis E, Mehta A (2003) Playing large games using simple strategies. In: Proceeding of the fourth ACM conference on electronic commerce. New York, USA, pp 36–41
- McKelvey RD, McLennan A (1996) Chapter 2, computation of equilibria in finite games. *Handb Comput Econ* 1:87–142
- McKelvey RD, McLennan A, Turocy TL (2014) Gambit: software tools for game theory. Version 16:1. <http://www.gambit-project.org/>
- Nishizaki I, Sakawa M (2001) Fuzzy and multiobjective games for conflict resolution. Springer, Berlin
- Ortiz LE, Irfan MT (2017) Tractable algorithms for approximate Nash equilibria in generalized graphical Games with tree structure. In: The Proceeding of the thirty-first conference on artificial intelligence
- Rubinstein A (2016) Settling the complexity of computing approximate two-player Nash equilibria. In: Proceeding of FOCS, pp 258–265
- Vijay V, Chandra S, Bector CR (2005) Matrix games with fuzzy goals and fuzzy payoffs. *Omega Int J Manag Sci* 33:425–429
- Zadeh LA (1978) Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst* 1:3–28
- Zimmermann H-J (2011) Fuzzy set theory and its application, 4th edn. Kluwer Academic Publishers, Berlin

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.