Paper Analysis Report

Rank-1 Bi-matrix Games: A Homeomorphism and a Polynomial Time Algorithm [1]

Jianke Zhang 1120200784 2022.6

Abstract: This report is written to introduce a new method of solving the Nash equilibrium in *Rank-1 Bi-matrix Games* mentioned in paper [1] and to arrange and summarize the main ideas and its proofs. I firstly organised the structure of this paper, trying to understand the main deduction to sort out author's train of thought. Secondly, I organised the notation and extracted out prime lemmas and proofs, which is beneficial for comprehending the author's theorem. What's more, as an introduction, I analysed the prominent and weaknesses of this paper by refering another recent work [3]. The methods used to solve this *Rank-1 Bi-matrix Games* can be migrated to other non-cooperative strategic, and are helpful for designing other effective Nash equilibrium algorithms.

Key word: Rank-1 games, Homeomorphism, Nash equilibrium, Polynomial Time Algorithm

Figure 0-1 Chess: an example of Bi-matrix Games

Contents

I. Introduction

1.1 Why I choose this?

Originally, I didn't want to choose Nash equilibrium as my topic because to be honest, it's really hard for us student to totally understand the complicated mathematical proof which exist extensively in solving NE problem. But however, comparing to mechanism design, the complicated deduction of math and design a new algorithm according to it can be really valuable especially for a student majoring in computer science.

Once there's a class introducing LH algorithm (Lemke-Howson), and this algorithm proposed a new method using path following on a polytope and this surprised me that solving Nash equilibrium uxexpectedly is a question of looking for path in a directed graph (later I knew that's what PPAD intends to). So here I choose this paper [1] solving Rank1 Bi-matrix Games to learn more novel approaches of NE problem, meanwhile to practice my math skills. Hope what I wrote make some sense.

1.2 Brief review of the paper

Work in this paper [1] is about a new method to solve the Nash equilibrium problem in polynomial time for a particular class of Bi-matrix games. After the discussion and analysis of the previous studies on this topic, the author proposes a linear game space which is homeomorphism to Nash equilibrium solution space, mainly based on the thought of LH algorithm. With this method, the author realizes a further transformation of the problem. On the basis of these theoretical derivations, a polynomial time algorithm for solving rank 1 games is given, and a new method for proving several classical double matrix games is also proposed.

Strictly speaking, this paper only solves a small class of double matrix game problems. Its value lies in that, on the one hand, many practical game problems met are stricter than the condition of rank-1 games. For example, a large number of classic zero-sum game problems are rank 0 problems, which is much simplier than rank1 games. On the other hand, this paper provides a method to construct homeomorphic linear space so as to simplify the problem. This innovative idea has a great enlightening effect on other problems. This is where I think the real value of this article lies.

1.3 Structure of the paper

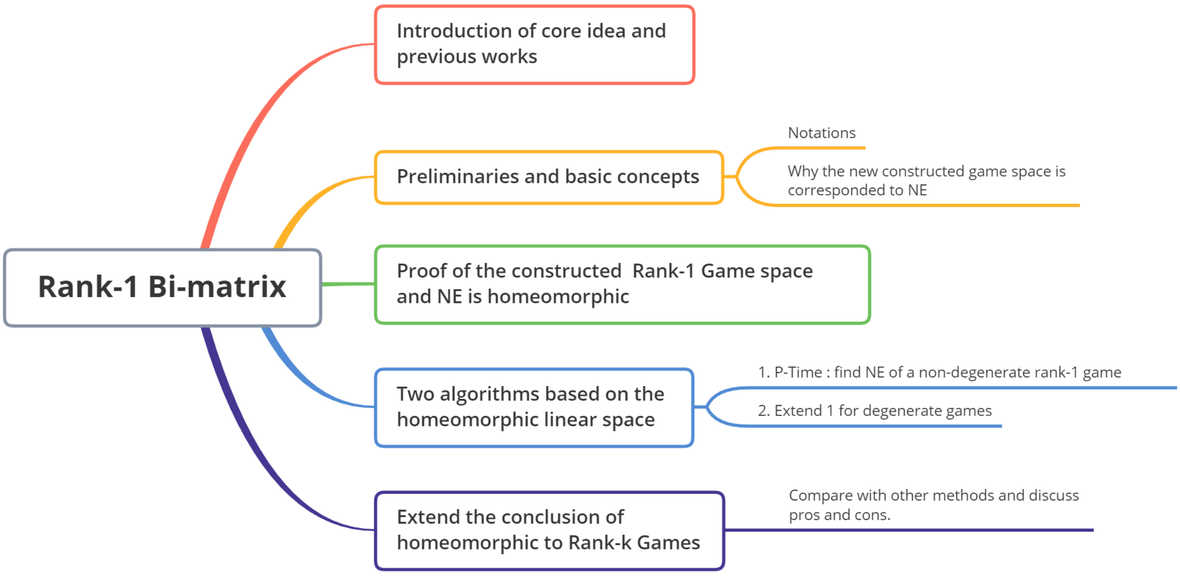

Figure 1-1 Structure of the paper

The whole structure was showed in figure 1-1. Here a more detailed summarize is given to facilitate the understanding of the content and ideas in the paper.

(1) Definition of Nash Equilibrium and Non-cooperative Game Theory of Finite Game (A model to understand strategic interaction of selfish agents in a given organization) .

(2) Introducing bimatrix Games with two agents payoff matrix (A, B) and organized some previous useful conclusions:

- The Classical Lemke-Howson algorithm [3] found a non-polynomial time method to find NE in two-matrix games.
- Subsequent studies show that NE calculation of double matrix games is PPAD-complete, and the NE of $O(\frac{1}{n})$ approximate exact solution is also PPAD, indicating that polynomial time algorithm is unlikely to solve NE of double matrix games.
- For the constant rank-k games:
  - Lipton gives a polynomial time algorithm, but these algorithms are not applicable to the widespread Zero-sum Games (*low expressive power* in the original).
  - Kannan and Theobald gave an approximate accurate NE for calculating the case with fixed $R(A+B) = k$.

(3) Introducing the *polytope strategy*: Rank-0 games are the same as zero-sum games and

Nash equilibria of a zero-sum game is a connected polyhedral set that may be computed in polynomial time by solving a linear program. So solving NE is really looking for a set of equivalent linear programming problems.

(4) The value of studying it: rank-1 Game is an extension of the smallest zero-sum games in the hierarchy. The expressive power of rank-1 games is larger than the zero-sum games, and NE is precisely polynomially solvable in a special *multiplication game* (mentioned in work [4]).

(5) The core of this paper: establish a homeomorphism between bimatrix games and Nash equilibrium. For more details, see in chapter 4.3.

(6) The author's work, here are the general contributions:

   (a) Construct the linear - subspace of rank 1 game, and the exact NE of this kind of game is calculated by proving that the space and NE are homeomorphic.

   (b) Present a new algorithm for rank 1 game.

   (c) Discuss how to extend the method of (a) to other double matrix games.

   (d) Using the idea of (a) to give several new proof methods of classical conclusions, including:

      - Existence of NE
      - Singularity of NE
      - Exponential theorem of two matrix games

   (e) Extend the homeomorphism idea of (a) to the fixed point formula of the rank-k game, which has linear structure and is much easier compared with the classical structure.

II. Preliminaries

In this part, I organized the essential notations used for proof and extracted out the important definition which will be extensively used in later proof of homeomorphism. To be more concise, I only displayed the notations I used in this report for better explaining the origin paper's train of thought.

2.1 Notations

This is a notation table in case you forgot what a sign means in the lemma or deduction. Some complicated definition in the table can be refered in section 2.2, and the possible relation between different notations can be found in section 3.

Table 2-1 Essential notations

Notation	Meaning
R^{2mn}	$R^{mn} \times R^{mn}$
$\Delta = (x_1, x_2, \cdots, x_n)$	Domain of strategy ;$x_1 + x_2 + \cdots + x_n = 1$
$(A, B) \in R^{2mn}$	Bi-Matrix for the two agents
$(x, y) \in \Delta \times \Delta$	Mixed strategy profile of the game
P, Q	Polytope to describe best response strategies
$\Gamma = \{(A, C + \alpha \cdot \beta^T) \vert \alpha \in R^m\}$	Game space
$E^\Gamma = \{(\alpha, x, y) \vert (x, y)\ is\ NESP\ of\ G(\alpha)\}$	Nash equilibrium correspondence of Γ
Q'	a set of Q to encompass $Q(\alpha)$
$L(u)$	The label of point v in the polytope
$\mathcal{N} = \{(v, w) \in P \times Q' \vert L(u) \cup L(w) = [m + n]\}$	The set of fully-labeled pairs in $P \times Q'$
$\Psi(\alpha, x, y)$	A map from $E_\Gamma \to \mathcal{N}$
$\mathcal{E}_v, \mathcal{E}_w$	Polytope vertices relate to a fully-labeled pair.
$LP(\delta)$	An linear program according to δ.
$OPT(\delta)$	the set of optimal points of $LP(\delta)$.

2.2 Some essential definitions and concepts

Def 1 - Homeomorphism : Make (X, T_X) and (Y, T_Y) two topological spaces, and define $f : (X, T_X) \to (Y, T_Y)$ is homeomorphism between these two spaces if and only if

(1) f is a bijection. (2) f is continuous. (3) f^{-1} is continuous.

Def 2 - NESP : A strategy profile is said to be a Nash equilibrium strategy profile (NESP) if no player achieves a better payoff by a unilateral deviation

$$\mathbf{x}^T R\mathbf{y} \geq \mathbf{x}'^T Ry, \forall \mathbf{x}' \in \Delta_m$$

$$\mathbf{x}^T C\mathbf{y} \geq \mathbf{x}^T C\mathbf{y}', \forall \mathbf{y}' \in \Delta_n$$

Def 3 - P/Q : Use a pair of polytope to describe the best response strategies of the row or column player.

$$P = \{(y, \pi_1) \in R^{n+1}) \mid A_i y - \pi_1 \leq 0,\ \forall\, i \in [m];\ y_j \geq 0,\ \forall\, j \in [n];\ \sum_{j=1}^{n} y_j = 1\}$$

$$Q = \{(x, \pi_2) \in R^{m+1}) \mid x^T B_i - \pi_2 \leq 0,\ \forall\, j \in [n];\ x_i \geq 0,\ \forall\, i \in [m];\ \sum_{i=1}^{m} x_i = 1\}$$

where x and y are vector variables, and π_1 and π_2 are scalar variables.

Def 4 - Fully-labeled : Label $L(v)$ of a point in the polytope (P or Q) is the set of indices of the tight inequalities at v. A pair $(v, w) \in P \times Q$ is called fully-labeled pair if $L(v) \cup L(w) = \{1, 2, \cdots, m+n\}$. Let $\mathcal{N} = \{(v, w) \in P \times Q' | L(u) \cup L(w) = [m+n]\}$ is the set of fully-labeled pairs. For convenience, define some useful sign of special vertices set .

$$\mathcal{E}_v = \{w' \in Q' \mid (v, w') \in \mathcal{N}\} \; ; \; \mathcal{E}_w = \{v' \in Q' \mid (v', w) \in \mathcal{N}\}$$

$$\mathcal{N}^P = \{v \in P | \, \mathcal{E}_v \neq \varnothing\} \; ; \; \mathcal{N}^{Q'} = \{w \in Q' | \, \mathcal{E}_w \neq \varnothing\}$$

Def 5 - Expression of Game Spaces :

(1) Bi-matrix game space : $\Omega = \{(A, B) \in R^{2mn}\}$

(2) Now make some changes in Ω in order to present the property of $Rank - 1$:

Any rank-1 matrix can be written as $\alpha \cdot \beta^T$, where $\alpha \in R^m$, $\beta \in R^n$.

So let $C = B - \alpha \cdot \beta^T$, then we denote this form of game space is Γ.

$$\Gamma = \{(A, C + \alpha \cdot \beta^T) | \alpha \in R^m\} \tag{2.1}$$

If the rank of game is 1, we have $C = -A$, namely $\Gamma = \{(A, -A + \alpha \cdot \beta^T)\}$.

(3) Game's NE space: $E_\Gamma = E_\Omega = \{(A, B, x, y) \in R^{2mn} \times \Delta_m \times \Delta_n\}$, where (x, y) is NESP of Γ / Ω.

(4) New polytope Q': a set of $Q(\alpha)$, $Q(\alpha) = Q' \cap \{(x, \lambda, \pi_2) | \sum_{i=1}^m a_i x_i = \lambda\}$

$$Q' = \{(x, \lambda, \pi_2) \in R^{m+2} | \, x^T C_j + \beta_j \lambda - \pi_2 \leq 0, \, \forall \, j \in [n]; \, x_i \geq 0, \, \forall \, i \in [m]; \, \sum_{i=1}^m x_i = 1\}$$

Def 6 - Map Ψ : Let $\Psi \; : \; E_\Gamma \rightarrow \mathcal{N}$ be such that:

$$\Psi(\alpha, x, y) = ((y, \pi_1), (x, \lambda, \pi_2)), \; where \; \pi_1 = x^T A y, \; \lambda = \sum_{i=1}^m \alpha_i x_i, \; \pi_2 = x^T \cdot (C + \alpha \cdot \beta^T) y$$

Def 7 - $LP(\delta)$: Let LP a linear program:

$$LP(\delta) : \qquad max \; \delta(\beta^T y) - \pi_1 - \pi_2$$

$$s.t. (y, \pi_1) \in P; \; (x, \lambda, \pi_2) \in Q'; \; \lambda = \delta$$

III. Important lemmas

To avoid too much complicated mathematical deduction in chapter 4 which will make understanding the main train of thought even harder, I took all the important lemmas out and explained them here.

Some of them are basic conclusions in NE problem, some others are classic conclusion of previous research and the others are the author's unique. Limited by space, I' ll only choose some important lemmas to prove. The lemmas here can well explain why the author's theorem is right.

Lemma 1 - $Fully-labeled \Rightarrow NESP$**: A strategy profile** (x, y) **is a NESP of the game** (A, B) **iff** $((y, \pi_1),\ (x, \pi_2)) \in P \times Q$ **is a fully-labeled pair.**

Explanation : This lemma comes from work in paper [3], the famous LH algorithm. It associated the NE problem with topological polytope. And this lemma is the basis of later conclusions in this paper.

Lemmas after 1 are trying to sort out the structure of the space $\mathcal{N}$.

Lemma 2 - The map $\mathcal{N}$ **is surjective.**

Proof: Let $(v, w) \in \mathcal{N}$ be a fully-labeled pair with $v = (y, \pi_1)$ and $w = (x, \lambda, \pi_2)$. Let $\alpha \in R^m$ be such that $\sum_{i=1}^{m} \alpha_i x_i = \lambda = 0$, then $(v, (x, \pi_2)) \in P(\alpha) \times Q(\alpha)$ is a fully-labeled pair. Therefore, $(\alpha, x, y) \in E_\Gamma$ and $(v, w) = \Psi(\alpha, x, y)$. So that $\mathcal{N}$ is surjective.**Q.E.D.**

Lemma 3 - E_Γ **is connected iff** $\mathcal{N}$ **is a single connected component**.

Explanation : This can be easily proved with Lemma 2 by proving ($\Rightarrow$) and ($\Leftarrow$).

It can been seen in lemma 2 and 3 that E_Γ and $\mathcal{N}$ are closely related.

Next, let's make some deduction.

Let the 'support-pair' of a vertex $(y, \pi_2) \in P$ be (I, J) where $I = \{i \in [m] | A_i y - \pi_2 = 0\}$ and $J = \{j \in [n] | y_j > 0\}$.Let $\beta_{j_s} = \min_j \beta_j$, $i_s = arg\max_i a_i j_s$, $\beta_{j_e} = \max_j \beta_e$, and $i_e = arg\max_i a_i j_e$.

So that the indices of j_s and j_e correspond to the minimum and maximum entries in β respectively, and the indices i_s and i_e correspond to the maximum entry in A_{j_s} and A_{j_e} respectively.

Lemma 4 - Stucture of $\mathcal{N}$**: There exists two vertices** v_s **and** v_e **in P with support-pairs (**i_s**,** j_s**) and (**i_e**,** j_e**) respectively, and edge** $(v, \mathcal{E}_v) \in \mathcal{N}$ **has exactly one bounding vertex if v is**

either v_s or v_e, otherwise it has two bounding vertices.

Explanation : This lemma shows that $\mathcal{N}$ has a simply structure that all other edges have two bounding vertices.

Lemmas after 4 are trying to find out property of $\mathcal{N}$ when $rank(A+B)=1$.

Lemma 5 - Property when rank is 1

$\forall\ (v,w) = ((y,\pi_1),(x,\lambda,\pi_2)) \in P \times Q'), \lambda(\beta^T y) - \pi_1 - \pi_2 \leq 0$, it also holds when $(v,w) \in \mathcal{N}$.

Explanation : This can be easily proved with when substitute $C = -A$ into the definition of P and Q.

Lemma 6 - OPT property

$\forall\ a \in R, OPT(a) = \{((y,\pi_1),(x,\lambda,\pi_2)) \in \mathcal{N} |\ \lambda = a\}$ and $OPT(a) \neq \varnothing$.

Lemma 7 - Monotonicity in $\mathcal{N}$

$\beta^T y$ and also λ monotonically increases on the directed path $\mathcal{N}$.

Explanation : This can be proved based on proposition 1 and by using the property of OPT, it's a little bit complicated and the detailed proof can be found in page 6 of work [1]. The later algorithm uses this lemma for rapid search on $\mathcal{N}$.

IV. Core ideas

In this chapter, I'll use the notations, definitions and lemmas to assemble the important propositions mentioned in the paper step by step. I'll try best to explain the conclusion more specifically using concrete desciption instead of abstract mathematical deduction.

4.1 Can we depict the structure of $\mathcal{N}$ easily?

Surely, the answer is yes! Let's think about a vertex $w = (x,\lambda,\pi_2) \in \mathcal{N}^{Q'}$ (click here to recall its definition 2.2), and then we know there exists an $\mathcal{E}_w \in \mathcal{N}^P$ that is in plain $A_i y = \pi_1$. However, the π_1 is bounded between 0 and 1 and A_i is fixed, so this $\mathcal{E}_w$ in its origin P is surely bounded as well. But there are two exceptions here: $(v_s, \mathcal{E}_{v_s})$, $(v_s, \mathcal{E}_{v_s})$ (refer the lemma 3), which are the only two unbounded edges in $\mathcal{N}$.

With this, let's imagine what $\mathcal{N}$ really looks like. Two unbounded edges with exactly 1 adjacent edges and many other bounded vertices? Surely, this shape is just as simple as a stick with infinite length and various circles surrounding it or just besides it, because only circles can ensure the edge is bounded and has two adjacent neighbor on both side.

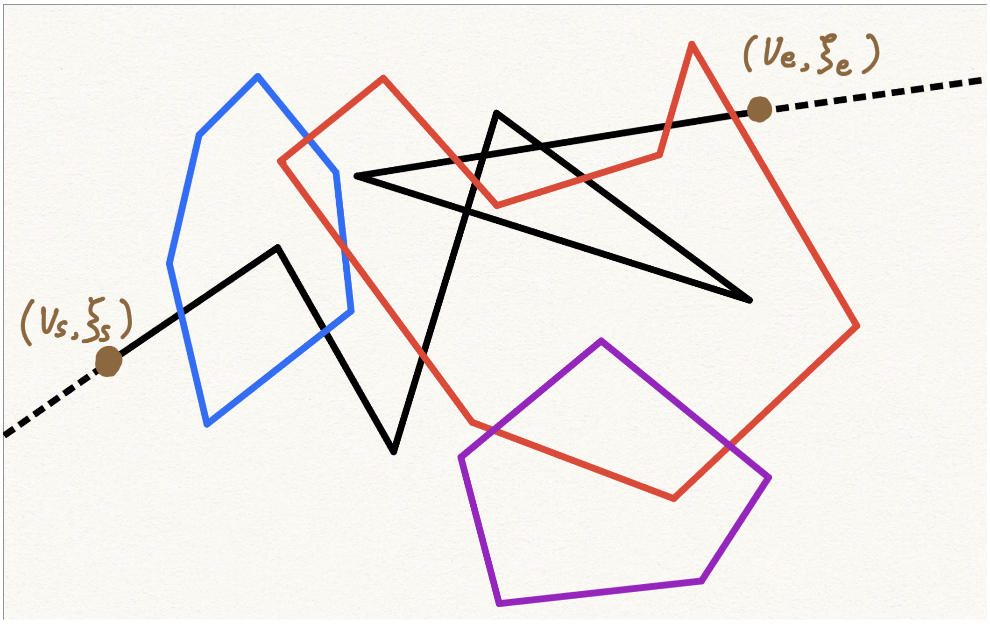

Figure 4-1 What the $\mathcal{N}$ might looks like?

Actually, with this conclusion, we can even infer the result of LH Algorithm mentioned in work [3].

Then, back the paper, we get the proposition 1:

Proposition 1 - Stucture of $\mathcal{N}$

The set of fully-labeled points $\mathcal{N}$ admits the following decomposition into mutually disjoint connected components: $\mathcal{N} = \mathcal{P} \cup \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_k, k{\geq}0$, where $\mathcal{P}$ and $\mathcal{C}_i$ respectively form a path and cycles on 1-skeleton of $P \times Q'$.

4.2 Why rank-1 is special?

Begin with the game space for rank-1 games: $\Gamma = \{(A, -A + \alpha \cdot \beta^T) | \alpha \in R^m\}$ (recall it in definition 2.2). As Def 7, the paper defined a linear program $LP(\delta)$ (to be honest I didn't quite figure out how to come into this).

From proposition 1, it is clear that there is always a path P in $\mathcal{N}$. And for every $a \in R$, there exists a point $((y, \pi_1), (x, a, \pi_2)) \in P$ (By Lemma 4). Again, because $OPT(a)$ is connected, therefore, $\forall\ a \in R,\ OPT(a)$ is contained in the path P. Therefore $\mathcal{N}$ consists of only the path! That's what proposition 2 in the paper tells us.

From now on, we can refer to $\mathcal{N}$ as a single path.

4.3 How E_Γ and Γ are homeomorphism?

To establish a homeomorphism map, we need to transform $(\alpha, x, y) \in E_\Gamma$ with length of $2m + n$ into a vector $\alpha' \in \Gamma$ with size of m. Here directly gives a map between E_Γ and Γ : $f:\ E_\Gamma \to \Gamma$

$$f(\alpha, x, y) = (\beta^T y + \alpha^T x, \alpha_2 - \alpha_1, \alpha_3 - \alpha_1, \cdots, \alpha_m - \alpha_1)^T \tag{4.1}$$

We are trying to prove that function f is a homeomorphism, which means to prove f is bijection, continuous, and f^{-1} is continous as well (recall it in Def 1). It mainly contains several procedures:

(1) Obviously, f is continous.

(2) Consider the first item in f as a function g:

$$Function\ g:\ \mathcal{N} \to R,\ g((y, \pi_1), (x, \lambda, \pi_2)) = \beta^T y + \lambda$$

(3) Prove that $\beta^T y$ and also λ monotonically increases on the directed path $\mathcal{N}$, so that function g is also strictly increases on it. (Use the property of $\mathcal{N}$)

(4) Function g is bijection because of monotonous.

(5) Prove f has a continous inverse f^{-1}:

$$f^{-1} : \Gamma \to E_\Gamma,\ make\ \alpha' \in \Gamma$$

$$g^{-1}(\vec{a}) \text{ gives } x, y\ and\ \lambda$$

α can be determined by solving equation, which has unique solution:

$$s.t. \begin{cases} \forall\, i > 1,\ \alpha_i = \alpha'_i + a_1 \\ \sum_{i=1}^{m} x_i \alpha_i = \alpha'_1 - \beta^T y \end{cases} \tag{4.2}$$

So that $f(\alpha') = (\alpha, x, y)$ and f has a unique solution then.

(6) Finally f is bijection and both f and f^{-1} are continous.

So we get a homeomorphism between Γ and E_Γ with function f in equation 4.1.

V. Algorithms

This paper proposed two algorithms. One is using binary search to find NE on non-degenerate problem, and the other one is to employ simple enumerate to solve degenerate problems. Though they takes different searching method, they all depended on the search of $\mathcal{N}$.

To better explain how the ideas of homeomorphism come into use, here I show the first algorithm in details and the second one can be easily understood along with the core idea.

5.1 What's the general algorithm?

The main thought is to traverse every point in $\mathcal{N}$ using the monotonicity of λ. Here use the conclusions mention in chapter 4 to determine the boundaries of binary searching on $\mathcal{N}$.

The first step is to modify parameters from fraction to interger. A game $(A, B = -A+\alpha\beta^T)$ will not be changed if multiply the same number on A, B. Therefore, denote the multiple c^2 where c is LCM of the denominators of $a_{ij}s, \beta_i s, \alpha_i s$. Finally, we get an equivalent form of game $G(\gamma) = (A, -A + \gamma\beta^T) \in \Gamma = \{(A, -A + \alpha\beta^T)\}$, where A, γ and β are all integers.

An important thing to note is that multiplying the same number will get the scale of input multiplied by at most $O(m^2n^2)$, which is a polynomial increase.

The second step is to determine searching areas. With the help of lemma 2, we know that each Nash equilibrium of the game $G(\gamma)$, there is a unique point in corresponded with in $\mathcal{N}$. According to the definition of Q', the intersection of $\mathcal{N}$ with the hyper-plane $H\ :\ \lambda - \sum_{i=1}^{m} \gamma_i x_i = 0$ gives all the NE point of $G(\gamma)$ (of course this can only happen when $G(\gamma)$ is non-degenerate game.)

Until now, the searching path is clear enough. Since $\forall\, x \in \Delta_m,\ \gamma_{min} \leq \lambda = \sum_{i=1}^{m} \gamma_i x_i \gamma_{max}$, we can traverse along the single path (by proposition 2) to search for all the NE point w. Here gives the binary search algorithm: (Something important to know: the $ISNE(a)$ return just a symbol of whether there's a NE point corresponded with a, instead of directly judging whether a is NE.)

Algorithm 1 Binary Search on Path of $\mathcal{N}$

Input: Boundaries of γ: γ_{min},γ_{max}

Output: A NE point a

$a_1 \leftarrow \gamma_{min}$; $a_2 \leftarrow \gamma_{max}$
if IsNE(a_1)=0 or IsNE(a_2)=0 **then**
  return Output of IsNE
end if
while True **do**
  $a \leftarrow \frac{\gamma_{min}+\gamma_{max}}{2}$, $flag \leftarrow$ IsNE(a)
  if flag=0 **then**
    break
  else if flag<0 **then**
    $a_1 \leftarrow a$
  else
    $a_2 \leftarrow a$
  end if
end while
return Output of IsNE

The last thing we need to consider is how to design a judging function $IsNE$ that can both tell us whether their is a NE point corresponded to input and caculate accurate position of this NE point.

Recall that when we are analysing the structure of $\mathcal{N}$, we know that $(v_e, \mathcal{E}_{v_e})$, λ increases from $-\infty$ to ∞ along the path from the first edge $(v_s, \mathcal{E}_{v_s})$ to the last edge. Meanwhile, we know that the NE point of $G(\gamma)$ is on hyper plane $H : \lambda - \sum_{i=1}^{m} \gamma_i x_i = 0$. This indicates that we can use OPT to determine a corresponded edge $< u, v >$ on the path of $\mathcal{N}$ (which is feasible cause this path has been completely determined) and then the NE point can be located because it is the intersection of $OPT(a)$ and hyper plane H. If there's no intersection, then we know that we need to search the next point according to the relative position between the edge $< u, v >$ contain $OPT(a)$ and hyper plane H. To better understand what we are doing exactly, see the schematic diagram showed below:

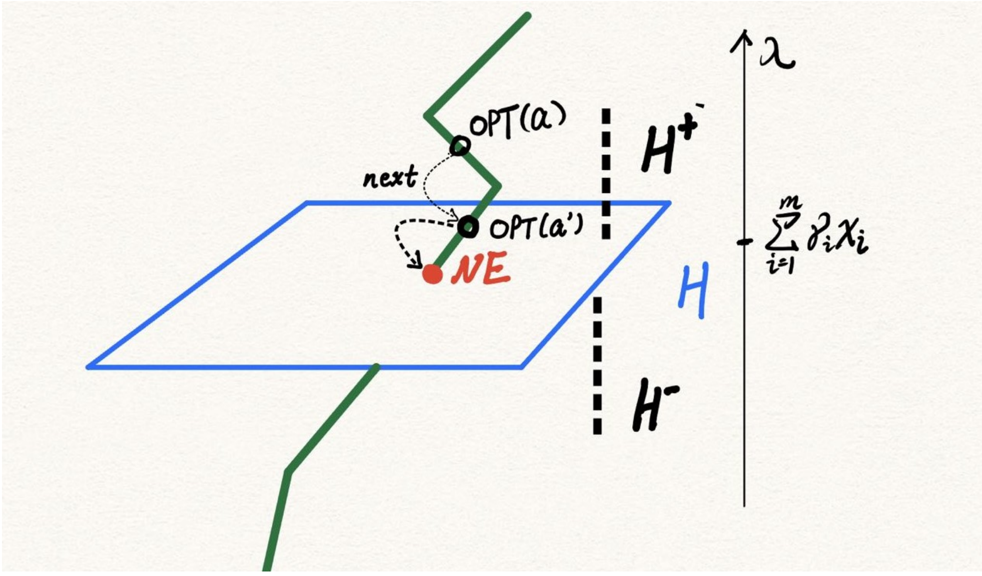

Figure 5-1 How to search on $\mathcal{N}$

Then, we get the algorithm $IsNE$ we want.

Algorithm 2 IsNE(δ)

Input: The point δ on $\mathcal{N}$

Output: A flag showing whether there's an NE point

1: Find $OPT(\delta)$ by solving $LP(\delta)$ and caculate plane H

2: $< u, v >\leftarrow$ The edge containing $OPT(\delta)$

3: $\mathcal{H} \leftarrow \{w \in < u, v > | w \in H\}$

4: **if** $\mathcal{H} \neq \varnothing$ **then**

5:     Output $\mathcal{H}$

6:     **return** 0

7: **else if** $< u, v >\in H^+$ **then**

8:     **return** 1

9: **end if**

10: **return** -1

5.2 Complexity analysis

Firstly, because every step in one binary search circulation is constant, so that we only need to consider the total number of circulation steps in binary search. For binary search, make max and min are two boundaries, d is the last distance of cursor and t is total loops and obviously, we have equation below:

$$2^t = \frac{max - min}{d}$$

We have $max = \gamma_{max}$ and $min = \gamma_{min}$ in the algorithm, so we only need to determine the last distance $d = a_2 - a_1$ then. Because the algorithm can only terminate when $OPT(a_1)$ or $OPT(a_2)$ and $OPT(a)$ are in the opposite sides of H and these two point are within a same edge, the last edge $(v, \mathcal{E}_v) \in \mathcal{N}$. Set the range of λ for any edges in $\mathcal{N}$ is $[\lambda_1, \lambda_2]$. Then we have

$$d \leq \lambda_2 - \lambda_1$$

Note that λ_1 and λ_2 correspond to the two vertices of $\mathcal{E}_v \in Q'$, and Q' has $m+2$ equations for every vertex of it. Therefore λ_1 and λ_2 are rational numbers with denominator. Set the denominator D, we have $D \leq (m+2)!(|B|)^{m+2}$, where $|B|$ is the biggest integar among elements of A, β, γ. Though it is weird to estimate lower bound by this way, we know that $d = \lambda_2 - \lambda_1 \geq 1/D^2$ and this surely works.

So, we get:

$$\begin{aligned} t &\leq log_2 \frac{\gamma_{max} - \gamma_{min}}{1/D^2} \\ &= 2log_2 D + log(\gamma_{max} - \gamma_{min}) \\ &\leq O(mlog_2 m + mlog|B| + log(\gamma_{max} - \gamma_{min})) \end{aligned}$$

It is obvious that $|B|, \gamma_{max} - \gamma_{min}$ are within polynomial of m, n and the max bit length of input, so that total time taken by BinSearch is polynomial. And surprisingly we get a polynomial time algorithm for Rank-1 Bi-matrix Game then!

VI. Promotion of core idea

6.1 Extension of other case

In paper [1], the author proposed an approach used to extend homeomorphism between the subspace of rank-1 games and it’ s Nash equilibrium correspondence to the subspace with **rank-k games**. The main thought is to expand every rank and manipulate the rank-k with k’s rank-1. Firstly, we need to redefine game spaces here.

Game (A, B) can be written as $(A, -A + \sum_{l=1}^{k} \gamma^l (\beta^l)^T)$. The corresponding game space $\Gamma^k = \{(A, -A + \sum_{l=1}^{k} \alpha^l (\beta^l)^T) \in R^{2mn} | \forall\ l \leq k,\ \alpha^l \in R^m,\ \beta^l \in R^n\}$. Correspondingly, the Nash equilibrium space of Γ^k is

$$E_{\Gamma^k} = \{(\alpha, x, y) \in R^{km} \times \Delta_m \times \Delta_n | (x, y)\ is\ NESP\ of\ G(\alpha) \in \Gamma^k\}$$

And set of polytope Q' will be changed to: $Q'^k = \{(x, \lambda, \pi_2) \in R^{m+k+1} | \ x_i \geq 0, \ \forall \ i \in [m]; \ x^T(-A^j) + \sum_{l=1}^{k} \beta_j^l \lambda_l - \pi_2 \leq 0, \ \forall \ j \in [n]; \ \sum_{i=1}^{m} x_i = 1\}$, and this time λ turns to a vector with length of k. We still construct $\mathcal{N}$ for rank-k games:

$$\mathcal{N}^k = \{(v, w) \in P \times Q'^k | \ L(v) \cup L(w) = \{1, \cdots, m+n\}\}$$

Also, we have linear program $LP^k(\delta)$ and its optimal solution set $OPT^k(\delta)$:

$$LP^k(\delta): \qquad max \sum_{l=1}^{k} \delta(\beta^T y) - \pi_1 - \pi_2$$

$$s.t. (y, \pi_1) \in P; \ (x, \lambda, \pi_2) \in Q'^k; \ \lambda_l = \delta_l, \forall \ l \leq k$$

Prepared with new notations, we can construct a **homeomorphism** $f^k : \ E_{\Gamma^k} \rightarrow \Gamma^k$ as below: (it is quite similar with rank-1, which just turns the elements to a size of k's vector)

$$f^k(\alpha, x, y) = (\alpha'^1, \alpha'^2, \cdots, \alpha'^k)$$
$$where \ \alpha'^l = (\alpha^{l^T} x + \beta^{l^T} y, \alpha_2^l - \alpha_1^l, \cdots, \alpha_m^l - \alpha_1^l]), \forall \ l \leq k$$

Meanwhile, there might exists a way that can solve NE in Rank-k Games by using a fixed point formulation. Suppose a rank-k game $G(\gamma)$:

Let $\gamma_{min} = (\gamma_{min}^1, \cdots, \gamma_{min}^k)$ and $\gamma_{max} = (\gamma_{max}^1, \cdots, \gamma_{max}^k)$. Denote 'box' $\mathcal{B} = \{a \in R^k | \ \gamma_{min} \leq a \leq \gamma_{max}\}$ ('$\leq$' is for every elements in the vector). Define a piecewise linear function $t : \mathcal{B} \rightarrow \mathcal{B}$ such that

$$t(a) = (\sum_{i=1}^{m} \gamma_i^1 x_i, \cdots, \sum_{i=1}^{m} \gamma_i^k x_i), \text{ where } (x, \lambda, \pi_2) = \{w \in Q'^k | (v, w) \in OPT^k(a), v \in P\}$$

If there exists a fixed point such that $f(a) = a$, then $\lambda = (\sum_{i=1}^{m} \gamma_i^1 x_i, \cdots, \sum_{i=1}^{m} \gamma_i^k x_i)$, which means there's a corresponded Nash equilibrium of game $G(\gamma)$. So, solving any rank-k games in polynial time reduces to **finding a fixed point of a polynomially computable piecewise linear function** $t : \mathcal{B} \rightarrow \mathcal{B}$. Unluckily, until now finding a fixed point is PPAD-complete and people have not find any polynomial algorithm yet. But this extension surely contributed a new approach to find NE of bi-matrix games.

6.2 Some weaknesses

Though works in paper [1] promotes a polynomial algorithm on rank-1 games, it can only find one NE point once. However according to works in paper [4] and paper [2], the number of equilibria of a rank-1 game may be exponential. So if we want to find the optimal NE point by searching every Nash equilibrium, the total time consumption maybe exponential as well.

A recent new work [2] published in 2021 (if you want more details and unsolved information, you may find it in this paper), proposed a much more faster algorithm on Rank-1 Bimatrix Games. And meanwhile the author proved that computing all the NE solutions at once is equivelant to $P = NP$, which seems impossible. And another promotion in work [2] proved *RANK-PRESERVING STRUCTURE* which explained a more general homeomorphism exists in bi-matrix games.

VII. Summary

In this report, I first sorted out the definition of symbol representation and rank-1 bi-matrix game in the literature for the convenience of readers to read the report or refer to the original literature [1]. Secondly, I sorted out the lemmas involved in the proof of the paper according to the steps in origin paper, and attached my own understanding of the proof or significance of the lemmas. Then I rearranged the core theoretical structure of the original literature on the basis of reading and understanding, meanwhile I deduced the main conclusions according to my understanding of core ideas, and explained the meaning of the theorem with a more visual description. Finally, combining with the latest research results [2] on the same topic in 2021, this report expounds the promotion and limitations of this literature.

In general, the main conclusions in the original text include the following:

(1) There is a homeomorphism between an m-dimensional affine subspace Γ of the bimatrix game space and it' s Nash equilibrium correspondence E_Γ, where Γ contains only rank-1 games.

(2) All equilibria of a rank-1 game can be found by following a piecewise linear path. However, such a path-following method finds only one equilibrium of a bimatrix game.

(3) One equilibrium of a rank-1 game can be found in polynomial time.

(4) For a game of rank-k, the set of its Nash equilibria is the intersection of a generically one-dimensional set of equilibria of parameterized games of rank-1 with a hyperplane.

References

[1] Adsul B , Garg J , Mehta R , et al. Rank-1 Bi-matrix Games: A Homeomorphism and a Polynomial Time Algorithm[C]. ACM, 2011.

[2] Bharat Adsul, Jugal Garg, Ruta Mehta, Milind Sohoni, Bernhard von Stengel. Fast Algo-

rithms for Rank-1 Bimatrix Games. Operations Research, 2021

[3] Lemke, Howson. Equilibrium points of bimatrix games. SIAM Journal on Applied Mathematics, 1964

[4] J. Bulow and J. Levin. Matching and price competition. American Economic Review, 96:652–668.