A Formula for Designing Zero-Determinant Strategies
Daizhan Cheng,

Abstract—A formula is presented for designing zero-determinant (ZD) strategies of general finite games, which have \( n \) players and players can have different numbers of strategies. To this end, using semi-tensor product (STP) of matrices, the profile evolutionary equation for repeated finite games is obtained. Starting from it, the ZD strategies are developed for general finite games, based on the same technique proposed by Press and Dyson [14]. A formula is obtain to design ZD strategies for any player directly, ignoring the original ZD design process. Necessary and sufficient condition is obtained to ensure the effectiveness of the designed ZD strategies. As a consequence, it is also clear that player \( i \) is able to unilaterally design \(|S_i| - 1\) dominating linear relations about the expected payoffs of all players. Finally, the fictitious opponent player is proposed for networked evolutionary networks (NEGs). Using it, the ZD-strategies are applied to NEGs. It is surprising that an individual in a network may use ZD strategies to conflict the whole rest network.

Index Terms—Finite evolutionary game, profile evolutionary equation, ZD strategy, networked evolutionary games, semi-tensor product of matrices.

I. INTRODUCTION

Zero-determinant (ZD) strategy was firstly proposed by Press and Dyson [14], which shows that in an iterated prisoner’s dilemma there exist strategies that dominate any evolutionary opponent. Since then it has attracted considerable attention from game theoretic community as well as computer, information, systems and control communities. [10] calls it “an underworld revolution in game theory”, because it reveals that in a repeated game, a player can unilaterally control her opponent’s payoff. The following up works include: (i) ZD strategy stability analysis [13]; (ii) extension to multi-player and multi-strategy case [16], [15]; (iii) games with continuous action spaces [13]; (iv) application to public goods games [17], mining pole games [1], snowdrift game [20], etc. A recent work [19] tried to bypass working out determinant form and find linear restriction relations directly.

Though there are some efforts for extending ZD strategy technique, proposed by the pioneer work of Press and Dyson for prisoner’s dilemma, to general cases [16], [15], to the author’s best knowledge, a systematic description for most general games has not been obtained. The purpose of this paper is to present a general framework for general finite games.

Recently, a new matrix product, called the semi-tensor product (STP) of matrices, was proposed [2], [3], and it has been applied to solve some problems in game theory, including the modeling and analysis of networked evolutionary games (NEGs) [5], providing a formula to verify whether a finite game is potential [4], investigating the vector space structure of finite games and its orthogonal decompositions [7], [9], application to traffic congestion games [21] and diffusion games [12], etc., just to mention a few. Readers who are interested in the STP approach to finite games are referred to a survey paper [8].

Using STP, this paper presents a profile evolutionary equation (PEE) for general finite games, which is essentially the same as the Markov matrix for the memory-one game in [14]. Then a detailed design technique and rigorous proofs are presented for this general case, which are generalizations of those proposed firstly by [14]. Necessary and sufficient conditions are provided for the effectiveness of ZD strategies. As a by product, we also prove that if a player has \( k_i \) strategies she can provide unilaterally \( k_i - 1 \) linear payoff relations using ZD strategies.

The main contribution of this paper is the formula, which provides an algebraic equation for designing ZD strategies, which does not involve any original determinant.

Finally, the ZD strategies for Networked evolutionary games (NEGs) are investigated. By proposing and using fictitious opponent player (FOP) the networked evolutionary games can be formulated as a two player game, where a player, say, player \( i \), plays fights with the FOP, who represents the whole network except player \( i \). The ZD strategies for player \( i \) are designed for \( i \) with FOP.

The rest of this paper is organized as follows: A brief
survey is given in Section 2. Then it is used to develop profile evolutionary equation (PEE) of finite evolutionary games. Finally, some properties of transition matrix of PPE are investigated, which are important for designing ZD strategies. Section 3 deduces a general formula for designing ZD strategies. Necessary and sufficient conditions for the designed ZD strategies to be effective is presented. Some numerical examples are followed to describe the design procedure. The fictitious opponent player (FOP) is proposed in Section 4 for networked evolutionary games (NEGs). Using FOP, the technique of ZD strategies has been applied to NEGs. Section 5 is a brief conclusion.

Before ending this section, a list of notations is presented.

- $\mathcal{M}_{m \times n}$: set of $m \times n$ dimensional real matrices.
- $M^*$: the adjoint matrix of $M$.
- $\sigma(M)$: the set of eigenvalues of $M$.
- $\rho(M)$: the spectral radius of $M$.
- $M > 0$ ($M \geq 0$): all entries of $M$ are positive (non-negative), i.e., $M_{i,j} > 0$ ($M_{i,j} \geq 0$), $\forall i,j$.
- $\kappa$: STP of matrices.
- $\text{Col}(A)$ ($\text{Row}(A)$) : the set of columns (rows) of $A$;
- $\text{Col}_i(A)$ ($\text{Row}_i(A)$): the $i$-th column (row) of $A$.
- $D_k$: $D_k = \{1, 2, \cdots, k\}$.
- $\delta^k_i$: The $i$-th column of identity matrix $I_k$.
- $\Delta_k$: $\Delta_k = \text{Col}(I_k) = \{\delta^k_i | i = 1, \cdots, k\}$.
- $L \in \mathcal{M}_{m \times n}$ is called a logical matrix, if $\text{Col}(L) \subset \Delta_m$.
- $L = [\delta^1_m, \delta^2_m, \cdots, \delta^n_m]$, it is briefly denoted by $L = \delta_m[n_1, n_2, \cdots, n_k]$.
- $\Upsilon^m$: set of $m$ dimensional (column) random vectors.
- $\Upsilon_{nxn}$: set of $n \times n$ (column) random matrices.
- $\mathcal{G}_{[n;k_1,k_2,\cdots,k_n]}$: set of finite non-cooperative games with $n$ players, and player $i$ has $k_i$ strategies, $i = 1, 2, \cdots, n$.

II. MODELING OF FINITE EVOLUTIONARY GAMES

A. A Brief Survey on STP

Definition 2.1: Let $M \in \mathcal{M}_{m \times n}$, $N \in \mathcal{M}_{p \times q}$, and $t := \text{lcm}(n, p)$ be the least common multiple of $n$ and $p$. Then the STP of $M$ and $N$ is defined as

\[
M \otimes N := (M \otimes I_{t/n}) (N \otimes I_{t/p}) \in \mathcal{M}_{(mt/n) \times (qt/p)}. \tag{1}
\]

Remark 2.2:

(i) STP is a generalization of conventional matrix product. That is, if $n = p$, then $M \otimes N = MN$. Hence in most cases the symbol $\otimes$ can be omitted. It is applicable to two arbitrary matrices.

(ii) As a generalization, STP keeps all major properties of conventional matrix product available, including associativity, distributivity, etc.

(iii) STP is a fundamental tool in this approach, and all the matrix products used in this paper without product symbol are assumed to be STP. Hence, in this paper the product of two arbitrary size matrices is well proposed.

Let $f : \mathcal{D}_m \rightarrow \mathcal{D}_n$ be a mapping from a finite set to another finite set. Then we can identify $j \in \mathcal{D}_m$ with its vector form $\vec{j} := \delta^j_m \in \Delta_m$. In this way, $f$ can be considered as a mapping $f : \Delta_m \rightarrow \Delta_n$. In the sequel $\vec{j}$ is simply denoted by $j$ again if there is no possible confusion.

**Proposition 2.3:** Let $f : \mathcal{D}_m \rightarrow \mathcal{D}_n$. Then there exists a unique matrix $M_f \in \mathcal{L}_{m \times n}$, called the structure matrix of $f$, such that as the arguments are expressed into their vector forms, we have

\[
f(x) = M_f x. \tag{2}
\]

As a corollary, Proposition 2.3 can be extended into more general form.

**Corollary 2.4:** Let $x_i \in \mathcal{D}_{k_i}$, $i = 1, 2, \cdots, n$, $y_j \in \mathcal{D}_{p_j}$, $j = 1, 2, \cdots, m$, and $x = \kappa_{i=1}^n x_i$, $y = \kappa_{j=1}^m y_j$. Assume

\[
y_j = f_j(x_1, x_2, \cdots, x_n), \quad j = 1, 2, \cdots, m,
\]

which have their vector forms as

\[
y_j = M_f \kappa_{i=1}^n x_i, \quad j = 1, 2, \cdots, m. \tag{3}
\]

Then there exists a unique matrix $M_F$, called the structure matrix of the mapping $F = (f_1, f_2, \cdots, f_m)$, such that

\[
y = M_F x, \tag{4}
\]

where

\[
M_F = M_{f_1} \ast M_{f_2} \ast \cdots \ast M_{f_m} \in \mathcal{L}_{\rho \times \kappa},
\]

and $\rho = \prod_{i=1}^m p_j$, $\kappa = \prod_{i=1}^n k_i$, and $\ast$ is Kratri-Rao product of matrices.

Similarly, we have the following result:

**Corollary 2.5:** Let $x_i \in \Upsilon_{k_i}$, $i = 1, 2, \cdots, n$ and $y_j \in \Upsilon_{p_j}$, $j = 1, 2, \cdots, m$, and

\[
y_j = M_f x, \quad j = 1, 2, \cdots, m, \tag{5}
\]

where $M_f \in \Upsilon_{nxp}$. Then there exists a unique matrix $M_{F_f}$ such that

\[
y = M_{F_f} x, \tag{6}
\]

Let $A \in \mathcal{M}_{m \times n}$, $B \in \mathcal{M}_{l \times n}$. Then the Khatri-Rao product of $A$ and $B$, denoted by $A \ast B$, is defined by

\[
\text{Col}_i(A \ast B) = \text{Col}_i(A) \text{Col}_i(B), \quad i = 1, 2, \cdots, n.
\]
For each player’s payoff function which is also called the structure matrix of the mapping \( F = (f_1, f_2, \cdots, f_m) \).

### B. PEE of Finite Games

**Definition 2.6:** Consider a finite game \( G = (N, S, C) \), where

(i) \( N = \{1, 2, \cdots, n\} \) is the set of players.

(ii) \( S = \prod_{i=1}^{n} S_i \) is the profile, where

\[
S_i = \{1, 2, \cdots, k_i\}, \quad i = 1, 2, \cdots, n,
\]

is the strategies (or actions) of player \( i \).

(iii) \( C = (c_1, c_2, \cdots, c_n) \), where

\[ c_i : S \rightarrow \mathbb{R}, \quad i = 1, 2, \cdots, n, \]

is the payoff (or utility, cost) function of player \( i \).

The set of such games is described by \( \mathcal{G}_{[n; k_1, k_2, \cdots, k_n]} \). A matrix formulation of the evolution game \( G \in \mathcal{G}_{[n; k_1, k_2, \cdots, k_n]} \) is described as follows [5]:

(i) Identifying \( j \in S_i \), with \( \delta^i_k \in \Delta_{k_i} \), then \( S_i \sim \Delta_{k_i} \).

(ii) Setting \( \kappa = \prod_{i=1}^{n} k_i \), then \( S \sim \Delta_{\kappa} = \prod_{i=1}^{n} \Delta_{k_i} \).

(iii) Let \( x_i \in \Delta_{k_i} \) be the vector form of a strategy for player \( i \). Then \( x = \kappa \times x_i \in \Delta_{\kappa} \) is a profile.

(iv) For each player’s payoff function \( c_i \), there exists a unique row vector \( V^c_i \in \mathbb{R}^\kappa \) such that

\[
c_i(x) = V^c_i x, \quad i = 1, 2, \cdots, n.
\]

Now consider an evolutionary game \( G^c \) of \( G \), which stands for (infinitely) repeated \( G \). Then each player can determine his action at \( t + 1 \) using historic knowledge. It was proved in [14] (see also [10]) that: “the shortest memory player sets the rule of the game, which means the long-memory strategies have no advantages over the memory-one strategies”. Based on this observation, the strategy updating rule is assumed Markov-like. That is, the strategy of player \( i \) at time \( t + 1 \) depends on the profile at \( t \) only. Then we have [5]

\[
x_i(t + 1) = L_i x(t), \quad i = 1, 2, \cdots, n.
\]

Two types of strategies are commonly used:

- **Pure strategy:**
  \[ L_i \in \mathcal{L}_{k_i \times \kappa}, \quad i = 1, 2, \cdots, n. \]

- **Mixed Strategy:**
  \[ L_i \in \mathcal{T}_{k_i \times \kappa}, \quad i = 1, 2, \cdots, n. \]

Multiplying (by STP) all equations in (8) together yields

\[
x(t + 1) = L x(t),
\]

where

\[
L = L_1 \ast L_2 \ast \cdots \ast L_n.
\]

In pure strategy case \( L \in \mathcal{L}_{k \times k} \) and in mixed strategy case \( L \in \mathcal{T}_{k \times k} \).

In mixed strategy case \( x(t) \) can be considered as a distribution of profiles at time \( t \). If we take into consideration that \( \delta^i_k \) is used to express the \( i \)-th profile, then \( x(t) \) can also be considered as the expected value of profile at time \( t \).

In this paper we concern only mixed strategy case. Now what a player can manipulate is his own strategy updating rule. That is, player \( i \) can only choose his \( L_i \).

We arrange profiles in an alphabetic order as

\[
S = \{(s_1, s_2, \cdots, s_n) | s_i \in S_i, \quad i = 1, 2, \cdots, n\} = \{(1, 1, \cdots, 1), (1, 1, \cdots, 2), \cdots, (k_1, k_2, \cdots, k_n)\} =: \{s^1, s^2, \cdots, s^\kappa\}.
\]

Denote the probability of player \( i \) choosing strategy \( j \) at time \( t + 1 \) under the situation that the profile at time \( t \) is \( s^\tau \) as

\[
p^\tau_{i,j} = \text{Prob}(x_i(t + 1) = j | x(t) = s^\tau).
\]

Then we have the strategy evolutionary equation (SEE) of player \( i \) as

\[
x_i(t + 1) = L_i x(t),
\]

where

\[
L_i = \begin{bmatrix}
p^1_{i,1} & p^2_{i,1} & \cdots & p^\kappa_{i,1} \\
p^1_{i,2} & p^2_{i,2} & \cdots & p^\kappa_{i,2} \\
\vdots & \vdots & & \vdots \\
p^1_{i,k_i} & p^2_{i,k_i} & \cdots & p^\kappa_{i,k_i}
\end{bmatrix}
\]

In fact, in \( G^c \) we assume the strategies of player \( i \) are

\[
S^\tau_i = \{s^\tau \rightarrow j | s^\tau \in S, \quad r = 1, 2, \cdots, \kappa, j = 1, 2, \cdots, k_i\},
\]

where \( s^\tau \rightarrow j \) means where \( x(t) = s^\tau \) then \( x_i(t + 1) = j \in S_i \).

\( L_i \) comes from the definition [10] immediately.

According to Corollary 2.5 we have PEE as

\[
x(t + 1) = L x(t),
\]

where the transition matrix

\[
L = L_1 \ast L_2 \ast \cdots \ast L_n.
\]

We give a simple example to calculate \( L \).

**Example 2.7:** Consider the repeated prisoners’ dilemma. Let \( p^\tau_{i,j} \) be the probability of player \( i \) taking strategy \( j \in \{C, D\} \sim \}

\[
L = L_1 \ast L_2 \ast \cdots \ast L_n.
\]
\{1, 2\} under the condition \( s' \). Then a straightforward computation shows that
\[
x_1(t + 1) = L_1 x(t),
\]
\[
x_2(t + 1) = L_2 x(t),
\]
where
\[
L_1 = \begin{bmatrix}
p_{1,1}^1 & p_{2,1}^1 \\
p_{1,2}^1 & p_{2,2}^1
\end{bmatrix},
\]
\[
L_2 = \begin{bmatrix}
p_{2,1}^2 & p_{2,2}^2 \\
p_{2,2}^2 & p_{2,2}^2
\end{bmatrix}.
\]
Denote by
\[
p_i = p_{i,1}^1, \quad q_i = p_{i,2}^1, \quad i = 1, 2, 3, 4.
\]
It follows that
\[
p_{1,2}^1 = 1 - p_{1,1}^1 = 1 - p_i, \quad p_{2,2}^2 = 1 - p_{2,1}^2 = 1 - q_i, \quad i = 1, 2, 3, 4.
\]
Then we have
\[
L = L_1 * L_2
\]
\[
= \begin{bmatrix}
p_{1,1}^1 p_{1,2}^2 & p_{1,1}^1 p_{2,2}^2 \\
p_{1,2}^1 p_{1,2}^2 & p_{1,2}^1 p_{2,2}^2 \\
p_{1,1}^2 & p_{1,2}^2
\end{bmatrix}
\]
\[
= \begin{bmatrix}
p_{1,1}^1 p_{1,2}^2 & p_{1,1}^1 p_{2,2}^2 & p_{1,2}^2 & p_{1,1}^2 \cdot p_{2,2}^2 \\
p_{1,2}^1 p_{1,2}^2 & p_{1,2}^1 p_{2,2}^2 & p_{1,2}^2 & p_{1,1}^2 \cdot p_{2,2}^2
\end{bmatrix}.
\]

**Remark 2.8:** It is easy to verify that the transition matrix in PEE (refer to (15)) is essentially the transpose of the Markov matrix for the memory one game in [14]. (Corresponding to the “column order” of [14] there is a “row order” change in [15]. This is because our profiles are ordered in alphabetic as CC, CD, DC, DD, while [14] uses the order CC, DC, CD, DD.

**C. Properties of PEE**

In this subsection we investigate some properties of the transition matrix \( L \) of PEE, which are required for designing ZD strategies. As we discussed in Remark 2.8, \( L \) is the same as the Markov matrix for the memory one game in [14], (only with a transpose). So \( L \) is a column random matrix. This difference does not affect the following discussion. Hence the following argument is a mimic of the corresponding argument in [14]. What we are going to do is to extend it to general case and put it on a solid mathematical foundation.

In the sequel, we need an assumption on \( L \). To present it, some preparation is necessary.

A random square matrix \( M \) is called a primitive matrix if there exists a finite integer \( s > 0 \) such that \( M^s > 0 \) [11].

Some nice properties of primitive matrix are cited as follows:

**Proposition 2.9:** Let \( L \) be a primitive stochastic matrix. Then

\[(i) \rho(P) = 1 \quad \text{and there exists a unique } \lambda \in \sigma(P) \quad \text{such that } |\lambda| = 1.
\]

\[(ii) \lim_{t \to \infty} L^t = P > 0. \quad (16)
\]

Moreover, \( P = uu^T \), where \( Lu = u \), \( u > 0 \), \( L^T v = v \), \( v > 0 \).

We are ready to present our fundamental assumption.

**Assumption A-1:** \( L \) is primitive.

**Remark 2.10:**

\[(i) \quad \text{A-1 is not always true. For instance, consider } \text{(15)} \quad \text{and let } p_{1,1}^2 = 0, p_{1,1}^1 = 0, p_{1,1}^2 = 0, \text{and } p_{1,2}^2 = 0. \text{Then } L \text{ is not primitive.}
\]

\[(ii) \quad \text{If } 0 < p_{i,j}^2 < 1, \forall i, j, \text{then a straightforward verification shows that } L \text{ is primitive. So A-1 is always true except a zero-measure set.}
\]

\[(iii) \quad \text{According to Proposition 2.9 we have the following immediate conclusions: a), if } L \text{ is primitive, then}
\]

\[
\text{rank}(L - I_n) = \kappa - 1.
\]

\[(b) \quad \text{There exists } P = uu^T, \text{where } Lu = u, \ u > 0, \ L^T v = v, \ v > 0, \text{such that (16) holds. That is,}
\]

\[
\lim_{t \to \infty} L^t = uu^T.
\]

**Proposition 2.11:** Let \( L \) be an \( \kappa \times \kappa \) column primitive stochastic matrix. Define \( M := L - I_\kappa \) and denote by \( M^\ast \) its adjoint matrix. Then

\[(i) \text{ rank}(M^\ast) = 1.
\]

\[(ii) \text{ Col}_j(M^\ast) \neq 0, \quad j = 1, 2, \cdots, \kappa.
\]

The proof and all other proofs can be found in Appendix.

**Proposition 2.12:** Consider the profile evolutionary equation (13). If \( L \) is primitive, then

\[
x^* := \lim_{t \to \infty} x(t) = u/\|u\|,
\]

where \( u \) comes from (18).

Hereafter, we assume \( u \) has been normalized. Then \( x^* = u \) is the only normalized eigenvector of \( L \) corresponding to eigenvalue 1.

**Proposition 2.13:** Assume \( L \) is primitive, then

\[
\text{Col}_j(M^\ast) \propto u, \quad \forall j.
\]

Combining (19) and (21) yields

\[
\text{Col}_j(M^\ast) = \mu_j u, \quad \mu_j \neq 0, \quad j = 1, 2, \cdots, \kappa.
\]
III. Design of ZD Strategies for Evolutionary Games

A. A Universal Formula for ZD-strategies

Consider the transition matrix \( L \) of PEE. For statement ease, we define two sets of parameters as

\[
\kappa(i) := \begin{cases} 
1, & i = 1, \\
\prod_{j=1}^{i-1} k_j, & i = 2, 3, \ldots, n.
\end{cases}
\]

(23)

\[
\kappa(i) := \begin{cases} 
0, & i = 0, \\
\prod_{j=i+1}^{n} k_j, & i = 1, 2, \ldots, n-1, \\
1, & i = n.
\end{cases}
\]

(24)

Using them, for each pair \( i, j \) \( i = 1, \ldots, n \), \( j = 1, 2, \ldots, k_i \), we define an index set as:

\[
\Phi_{i,j} := \{ s = (\alpha - 1)\kappa(i-1) + (j-1)\kappa(i) + \beta \mid \alpha = 1, 2, \ldots, \kappa(i); \beta = 1, 2, \ldots, \kappa(i) \}.
\]

(25)

Then we have the following result:

Lemma 3.1: Let \( M = L - I_{\kappa} \). Define

\[
\Xi_{i,j} := \sum_{s \in \Phi_{i,j}} \text{Row}_s(M), \quad i = 1, 2, \ldots, n; \ j = 1, 2, \ldots, k_i.
\]

(26)

Then

\[
\Xi_{i,j} = p_{i,j} - \xi_{i,j}, \quad i = 1, 2, \ldots, n; \ j = 1, 2, \ldots, k_i.
\]

(27)

where

\[
p_{i,j} = \left( p_{i,j}^1, p_{i,j}^2, \ldots, p_{i,j}^n \right),
\]

and \( \xi_{i,j} \in B^n \) is defined as follows:

\[
\xi_{i,j}(s) = \begin{cases} 
1, & s \in \Phi_{i,j}, \\
0, & s \notin \Phi_{i,j}.
\end{cases}
\]

(28)

It is obvious from Lemma 3.1 that \( \Xi_{i,j} \) can be used to design \( p_{i,j} \) for player \( i \). A general design formula is presented in the following algorithm:

Algorithm 3.2: Consider an evolutionary game \( G^e \), where \( G \in G_{n; k_1, k_2, \ldots, k_n} \). Assume player \( i \) is aimed at a set of linear relations on the expected payoffs as

\[
\ell_{i,j}(Ec_1, Ec_2, \ldots, Ec_n, 1),
\]

(29)

where \( \ell_{i,j} \) is a linear equation. Then his ZD strategies can be designed as

\[
p_{i,j} = \left( p_{i,j}^1, p_{i,j}^2, \ldots, p_{i,j}^n \right) = \mu_{i,j}\ell_{i,j}(V_1^e, V_2^e, \ldots, V_n^e, 1^T\kappa) + \xi_{i,j},
\]

(30)

where \( \mu_{i,j} \neq 0 \) are adjustable parameters.

Equation (30) is a fundamental formula, which provides a convenient way to design ZD strategies for our preassigned purposes.

Definition 3.3: A set of ZD strategies is rational, if the following rationality conditions are satisfied.

(i) \( 0 \leq p_{i,j} \leq 1, \ j = 1, 2, \ldots, k_i - 1 \).

(31)

(ii) \( 0 \leq \sum_{j=1}^{k_i-1} p_{i,j} \leq 1 \).

(32)

Remark 3.4:

(i) It is obvious that rationality is a fundamental requirement. Non-rational strategies are meaningless.

(ii) It is clear that player \( i \) can unilaterally design at most \( |S_i| - 1 \) linear relations. Because when \( p_{i,j}, j < |S_i| \) are all determined, \( p_{i,|S_i|} \) is uniquely determined by

\[
p_{i,|S_i|} = 1^T - \sum_{j=1}^{k_i-1} p_{i,j}.
\]

(33)

(iii) Of course, player \( i \) need not to design \( |S_i| - 1 \) relations. Is he intends to design \( r < |S_i| - 1 \) relations, equation (32) has to be modified by reducing the summation to \( r \) items.

(iv) The ZD design formula (30) can be used simultaneously by multi-players, or even all \( n \) players.

Even though a set of ZD strategies is rational, it may not be effective. That is, the goal (29) may not be reached. We need the following result:

Theorem 3.5: Consider an evolutionary game, \( G^e \), where \( G \in G_{n; k_1, k_2, \ldots, k_n} \). A set of ZD strategies designed by formula (30) is effective, if and only if,

(i) There exists an \( u \in \mathbb{T}^n \) such that

\[
\lim_{t \to \infty} L^t = u\mathbf{1}^T_{\kappa}.
\]

(33)

(ii) \( \text{rank}(L-I_{\kappa}) = \kappa - 1 \).

(34)

Note that verifying the two conditions in Theorem 3.5 is not an easy job. Hence we may replace them by the following one.

Corollary 3.6: Consider an evolutionary game, \( G^e \), where \( G \in G_{n; k_1, k_2, \ldots, k_n} \). If \( L \) is primitive, then the set of ZD strategies designed by formula (30) is effective.

Remark 3.7:

(i) Even though primitivity of \( L \) is only a sufficient condition, it is almost necessity because only a zero-measure
set of $L$ may not be primitive. That is, all non-primitive $L$ have certain boundary values as $p_{r,i,j} \in \{0,1\}$ for some $(r,i,j)$. So the designer, who intends to use ZD strategies, is better to avoid using such values.

(ii) Any player can not unilaterally make the conditions in Theorem 3.3 satisfied. It depends on other players’ strategies. What the player $i$ can do is to do his best, that is, to ensure his designed rows, $\Xi_{r,i,j}$, are linearly independent. (A Chinese idiom says that “Mou Shi Zai Ren, Cheng Shi Zai Tian”, (Man proposes, God disposes. That is the situation for ZD strategy Designer.))

B. Numerical Examples

In the following, we discuss some numerical examples:

Example 3.8:

Consider a $G \in \mathcal{G}_{[3,2,3,2]}$. Since $k_1 = 2, k_2 = 3$, and $k_3 = 2$, using (23) and (24), it is easy to calculate that

$$
\kappa^{(1)} = 1, \quad \kappa^{(2)} = 2, \quad \kappa^{(3)} = 6;
$$

$$
\kappa^{(0)} = 0, \quad \kappa^{(1)} = 6, \quad \kappa^{(2)} = 2, \quad \kappa^{(3)} = 1.
$$

(35)

According to (25), we have

$$
\Phi_{1,1} = \{1, 2, 3, 4, 5, 6\},
\Phi_{1,2} = \{7, 8, 9, 10, 11, 12\},
\Phi_{2,1} = \{1, 2, 7, 8\},
\Phi_{2,2} = \{3, 4, 9, 10\},
\Phi_{2,3} = \{5, 6, 11, 12\},
\Phi_{3,1} = \{1, 3, 5, 7, 9, 11\},
\Phi_{3,2} = \{2, 4, 6, 8, 10, 12\}.
$$

(36)

These parameters depend on the type of games, precisely speaking, they depend on \{n; k_1, k_2, \cdots, k_n\} only. They are independent on particular games.

- Pinning Strategy:

Assume the payoff vectors are:

$$
V_1^c = [-3, -0.5, 6, 9, 8, 7, -4, -4.5, 5, 6.5, 5.7],
$$

$$
V_2^c = [4, -1, -5, 7.5, 2, 3.5, 8, -4, 5, 8, 9, -2],
$$

$$
V_3^c = [9, 5, -6, -5.5, 5.8, 8.5, 5.5, -0, -3.5, 4.5, 7].
$$

Set

$$
p_{2,1} := (0.1) \ast V_1^c - (0.4) \ast 1_{12}^r + \xi_{2,1}
$$

$$
= [0.3, 0.55, 0.2, 0.5, 0.4, 0.5, 0.3, 0.2, 0.15, 0.1, 0.25, 0.1, 0.3]
$$

$$
p_{2,2} := (0.1) \ast V_2^c + (0.3) \ast 1_{12}^r + \xi_{2,2}
$$

$$
= [0.6, 0.2, 0.1, 0.15, 0.25, 0.5, 0.55, 0.25, 0.7, 0.35, 0.15, 0.4].
$$

(38)

Then it is ready to verify that the ZD strategies designed in (38) are rational.

(ii) Extortion Strategy:

Consider a $G \in \mathcal{G}_{[3,2,3,2]}$ again. Assume the payoff structure vectors are as follows:

$$
V_1^c = [16, 11, -4, -8, -2, -10.3, 11.4, 18.5, 1.2, -3, -2.5, 1.5],
$$

$$
V_2^c = [3, 2, -1, 0, 5, -6, 4, 3, 3.1, -1, 7],
$$

$$
V_3^c = [-2.9, 0, 6, 8, 7.1, 2, -9.4, -8.2, 0.4, 4.6, 6.1, -2, 2.3].
$$

Assume player 2 want to design an extortion strategy against both players 1 and 3. He may design

$$
Ec_2 - r = \kappa_1(Ec_1 - r),
$$

$$
Ec_2 - r = \kappa_2(Ec_3 - r).
$$

To this end, he needs to design

$$
p_{2,1} - \xi_{2,1} = \mu_1 \left( [V_2^c - r1_{12}^T] - \kappa_1(V_1^c - r1_{12}^T) \right),
$$

$$
p_{2,3} - \xi_{2,3} = \mu_2 \left( [V_2^c - r1_{12}^T] - \kappa_2(V_3^c - r1_{12}^T) \right).
$$

(37)

Choosing $\mu_1 = 0.05, \mu_2 = 0.1, r = 1, \kappa_1 = 1.1, \kappa_2 = 1.2$, it follows that

$$
p_{2,1} = (0.275, 0.5, 0.175, 0.445, 0.365, 0.2715, 0.178, 0.1375, 0.0890, 0.22, 0.0925, 0.2725],
$$

$$
p_{2,2} = (0.668, 0.22, 0.104, 0.168, 0.28, 0.548, 0.604, 0.272, 0.768, 0.388, 0.16, 0.444].
$$

(39)

The ZD strategies designed in (39) are also rational.

Remark 3.9:

(i) In general, to design a set of rational ZD strategies is not an easy job. To determine related parameters we need to solve a set of linear inequalities.
(ii) To verify Lemma 3.1, we calculate the matrix $M = L - I_k$ for Example 3.3 as follows:

$$
M = \begin{bmatrix}
    p_{1,1} & p_{1,2} & \cdots & p_{1,k} \\
    p_{2,1} & p_{2,2} & \cdots & p_{2,k} \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{k,1} & p_{k,2} & \cdots & p_{k,k}
\end{bmatrix}
$$

Then it is easy to verify that $\Phi_{i,j}$ is the rows with each components containing $p_{i,j}$ as a factor. Moreover, a simple calculation shows that Lemma 3.1 is correct.

(iii) To verify the effective of ZD-strategies in (38), we assume the strategies for player 1 is

$$p_{1,1} = [0.2, 0.3, 0.8, 0.7, 0.5, 0.4, 0.7, 0.9, 0.2, 0.2, 0.1, 0.9];$$

the strategies for player 3 is

$$p_{3,1} = [0.15, 0.2, 0.8, 0.85, 0.2, 0.35, 0.7, 0.9, 0.2, 0.15, 0.55, 0.35].$$

Then the strategy profile dynamics is:

$$x(t + 1) = Lx(t), \quad t \geq 0,$$

where $L$ is

$$L = \begin{bmatrix}
    0.009 & 0.033 & 0.128 & 0.2975 & 0.04 & 0.042 \\
    0.051 & 0.132 & 0.032 & 0.0525 & 0.16 & 0.078 \\
    0.018 & 0.012 & 0.064 & 0.0892 & 0.025 & 0.07 \\
    0.102 & 0.048 & 0.016 & 0.0158 & 0.1 & 0.13 \\
    0.003 & 0.015 & 0.448 & 0.2082 & 0.035 & 0.028 \\
    0.017 & 0.06 & 0.112 & 0.0367 & 0.14 & 0.052 \\
    0.036 & 0.077 & 0.032 & 0.1275 & 0.04 & 0.063 \\
    0.204 & 0.308 & 0.008 & 0.0225 & 0.16 & 0.117 \\
    0.072 & 0.028 & 0.016 & 0.0383 & 0.025 & 0.105 \\
    0.408 & 0.112 & 0.004 & 0.0068 & 0.1 & 0.195 \\
    0.012 & 0.035 & 0.112 & 0.0064 & 0.02 & 0.055 \\
    0.068 & 0.14 & 0.028 & 0.0158 & 0.14 & 0.078 \\
    0.058 & 0.1215 & 0.004 & 0.0075 & 0.0055 & 0.035 \\
    0.042 & 0.0135 & 0.016 & 0.0425 & 0.0045 & 0.1755 \\
    0.2695 & 0.0205 & 0.028 & 0.0105 & 0.0083 & 0.126 \\
    0.1155 & 0.0225 & 0.112 & 0.0505 & 0.0067 & 0.234 \\
    0.225 & 0.048 & 0.008 & 0.012 & 0.0413 & 0.0945 \\
    0.0525 & 0.054 & 0.032 & 0.068 & 0.0338 & 0.1755 \\
    0.042 & 0.0135 & 0.016 & 0.0425 & 0.0045 & 0.1005 \\
    0.018 & 0.0015 & 0.004 & 0.017 & 0.0045 & 0.0195 \\
    0.1155 & 0.0225 & 0.112 & 0.0420 & 0.0723 & 0.014 \\
    0.0495 & 0.0025 & 0.048 & 0.238 & 0.0007 & 0.0260 \\
    0.0525 & 0.054 & 0.032 & 0.048 & 0.3713 & 0.0105 \\
    0.0225 & 0.006 & 0.128 & 0.272 & 0.3037 & 0.0195
\end{bmatrix}$$

It is easy to verify that

$$\text{rank}(L - I_{12}) = 11.$$ 

Moreover, we also have that

$$\lim_{t \to \infty} L^t = u1^T_{12},$$

where

$$u = [0.0731, 0.075, 0.0715, 0.082, 0.126, 0.0775, 0.0434, 0.1002, 0.0475, 0.1278, 0.0683, 0.1077]^T,$$

which is the normalized eigenvector of $L$ with respect to its (unique) eigenvalue 1.

IV. APPLICATION TO NETWORKED EVOLUTIONARY GAMES

This section considers how to design ZD strategies for a player, $i$, in a networked evolutionary game (NEG). We propose a method, called a fictitious opponent player (FOP) conversion.

A. Fictitious Opponent Player

**Definition 4.1:** An NEG is a triple $(N, E), (G, \Pi)$, where $(N, E)$ is a network graph with $N$ as the set of players, $G \in G_{[2:k,k]}$ is a symmetric game with two players, called the fundamental network game; $\Pi$ is the strategy updating rule, which describes how each player to update his strategies using his neighborhood information.

**Remark 4.2:**

(i) $G \in G_{[2:k,k]}$ is symmetric, if $S_1 = S_2 := S_0$ and for any $x, y \in S_0$

$$c_1(x, y) = c_2(y, x).$$

(ii) If $(i, j) \in E$, then players $i$ and $j$ will play game $G$ at each time. (In this paper only the fixed graph is considered.) Since $G$ is symmetric, the other of the two players does not affect the result.

(iii) Such an NEG is denoted by $G^{nc} = ((N, E), (G, \Pi)).$

Let player $i \in N$, and deg$(i) = d$. Then he may consider $N\setminus\{i\}$ as one player, called the FOP of $i$, denoted by $p_{-i}$. Assume $|S_0| = k$, the neighbors’ strategies can be considered as the strategies of $p_{-i}$. That is, $p_{-i}$ has totally $k^d$ strategies.

In fact, we need not to distinguish different neighbors, hence if $S_0 = \{s_1, s_2, \cdots, s_k\}$, then the set of strategies of $p_{-i}$, denoted by $S_{-i}$, is

$$S_{-i} = \{s_1s_1 \cdots s_1, s_1s_1 \cdots s_2, \cdots, s_ks_ks_ks_ks_k\}.$$ (41)

Each $s_a \in S_{-i}$ can be expressed as

$$s_a = (s_1s_1 \cdots s_1 s_1s_2 \cdots s_2s_2 \cdots s_ks_ks_ks_k),$$

where $d_1 + d_2 + \cdots + d_k = d$. Hence, we can also express $s_a$ by $(d_1, d_2, \cdots, d_k)$, which means $s_j$ has been used by $d_j$ neighbors, $1 \leq j \leq k$. Using this notation, we have

$$S_{-i} = \{(d_1, d_2, \cdots, d_k) \mid d_j \geq 0, \forall j; \sum_{j=1}^{k} d_j = d\}.$$ (42)

It is easy to verify that define the strategies of $p_{-i}$ in this way, by ignoring the order of neighbors, the total number of strategies is reduced from $k^d$ to

$$\frac{(k + d - 1)!}{(k - 1)!d!}.$$
As long as the stationary distribution of the overall network exists, ZD strategies are still applicable. Moreover, it is easily designable.

This treatment reduces the computational complexity.

From the point of view of player \( i \), the NEG is equivalent to a game between him and \( p_{-i} \), who has the set of strategies \( S_{-i} \) defined by (42). Let \( s_* = (d_1, d_2, \cdots, d_k) \in S_{-i} \). Then the payoff functions for \( c_i \) and player \( p_{-i} \), denoted by \( c_{-i} \), are

\[
c_i(x_i, s_*) = \sum_{j=1}^{k} d_j(c_i(x_i, s_j),
\]

\[
c_{-i}(x_i, s_*) = \sum_{j=1}^{k} d_j(c_j(x_i, s_j).
\]

(43)

Note that the FOP formulation is particularly suitable for using ZD strategies, because it is not effected by the structure and size of network graph, even though the size might be \( \infty \). As long as the stationary distribution of the overall network exists, ZD strategies are still applicable. Moreover, it is easily designable.

### B. ZD Strategies for NEGs

This section considers how to design ZD-strategies for NEGs. We describe the process through an example.

Consider prisoner’s dilemma, \( G \). The two strategies for both players are cooperation (\( C \)) and defect (\( D \)). Their payoffs are described in Table I where, as a convention, \( T > R > P > S \).

![Networked Prisoners’ Dilemma](image)

| \( p_1 \) \( p_2 \) | \( C \) | \( D \) |
|----------------|-----|-----|
| \( C \) | \( R, R \) | \( S, T \) |
| \( D \) | \( T, S \) | \( P, P \) |

Now consider a networked evolutionary prisoners’ dilemma, denoted by \( G^{ne} \). The network graph, depicted by Fig. 1 is non-homogeneous.

1) Consider player A. Since \( \deg(A) = 2 \), The set of strategies of \( p_{-A} \) is

\[ S_{-A} = \{ (CC), (CD), (DD) \}; \]

Using (43), the payoff vectors for \( c_B \) and \( c_{-B} \) are, respectively,

\[
V^c_B = (2R, R + S, 2S, 2T, T + P, 2P),
\]

\[
V^c_{-B} = (2R, R + T, 2T, 2S, S + P, 2P).
\]

It is easy to calculate that \( \kappa = 6 \), and

\[
\Phi_{1,1} = \{1, 2, 3\},
\]

\[
\chi_{1,1} = (1, 1, 1, 0, 0, 0, 0, 0).
\]

The design of ZD strategies is similar to the one for A.

2) Consider player B. Since \( \deg(B) = 3 \), The set of strategies of \( p_{-B} \) is

\[ S_{-B} = \{ (CCC), (CCD), (CDD), (DDD) \}; \]

Using (43), the payoff vectors for \( c_B \) and \( c_{-B} \) are, respectively,

\[
V^c_B = (3R, 2R + S, R + 2S, 3S, 3T, 2T + P, T + 2P, 3P),
\]

\[
V^c_{-B} = (3R, 2R + T, R + 2T, 3T, 3S, 2S + P, S + 2P, 3P).
\]

We also have \( \kappa = 10 \) and

\[
\Phi_{1,1} = \{1, 2, 3, 4, 5\},
\]

\[
\chi_{1,1} = (1, 1, 1, 1, 1, 0, 0, 0, 0, 0).
\]

3) Consider player C. Since \( \deg(C) = 4 \), The set of strategies of \( p_{-C} \) is

\[ S_{-C} = \{ (CCCC), (CCCD), (CCDD), (CDDD), (DDDD) \}; \]

Using (43), the payoff vectors for \( c_C \) and \( c_{-C} \) are, respectively,
The design of ZD strategies is similar to the one for $A$, or $B$.

V. Conclusion

This paper extends the technique of ZD strategies proposed by Press and Duson to general finite games. Using STP, a fundamental formula is presented to design ZD strategies for general finite games. In addition to the generality, it causes the design procedure very easy. Condition for the rationality of ZD strategies is presented. Then, necessary and sufficient condition for the effectiveness of the designed ZD strategies is also obtained, which put the ZD technique on a solid foundation. Some numerical examples are presented to demonstrate the efficiency of the method proposed in this paper.

Finally, the NEGs are considered. A new concept, called FOP, is proposed as the opponent for a preassigned player $i$. Using it, the ZD strategies for player $i$ is designed for the game between himself and his FOP. It is a surprising fact that one single player may be able to challenge whole network by using ZD strategies, no matter how large the network is.

Some important issues remained for further study, such as:

(i) How to design rational ZD strategies for a complicated game?

(ii) How to predict the efficiency of ZD strategies for a player?

(iii) What will happen when ZD strategies are used to against ZD strategies?

REFERENCES

[1] M. Cao, C. Tang, Y. Liu, F. Lin, Z. Chen, Application of ZD strategy in mining pool game, Proc. 38th CCA, 880-885, 2019.

[2] D. Cheng, H. Qi, Z. Li, Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, London, 2011.

[3] D. Cheng, H. Qi, Y. Zhao, An Introduction to Semi-tensor Product of Matrices and Its Applications, World Scientific, Singapore, 2012.

[4] D. Cheng, On finite potential games, Automatica, Vol. 50, No. 7, 1793-1801, 2014.

[5] D. Cheng, F. He, H. Qi, T. Xu, Modeling, analysis and control of networked evolutionary games, IEEE Trans. Aut. Contr., Vol. 60, No. 9, 2402-2415, 2015.

[6] D. Cheng, T. Liu, K. Zhang, On decomposed subspaces of finite games, IEEE Trans. Aut. Contr., Vol. 61, No. 11, 3651-3656, 2016.

[7] D. Cheng, T. Liu, Linear representation of symmetric games, IET Contr. Theory Appl., Vol. 11, No. 18, 3278-3287, 2017.

[8] D. Cheng, Y. Wu, G. Zhao, S. Fu, A Comprehensive Survey on STP Approach to Finite Games, Sys. Sci. Compl., http://arxiv.org/abs/2106.16086, (submitted).

[9] Y. Hao, D. Cheng, On skew-symmetric games, Journal of the Franklin Institute, Vol. 355, 3196-3220, 2018.

[10] D. Hao, Z. Rong, T. Zhou, Zero-determinant strategy: An underway revolution in game theory, Chin. Phys. B, Vol. 23, No. 7, 078905, 2014.

[11] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge, Cambridgve Univ. Press, 1986.

[12] H. Li, X. Ding, Q. Yang, Y. Zhou, 2018. Algebraic formulation and Nash equilibrium of competitive diffusion games, Dyn. Game Appl., Vol. 8, 423-433, 2018.

[13] A. Mcvoy, C. Hauert, Autistic strategies for iterated games with arbitrary action spaces, Proc. Natl. Acad. Sci., Vol. 113, doi:10.1073/pnas.1520163113, 2016.

[14] W. Press, F.J. Duson, Iterated prisoner’s dilemma contains strategies that dominate any evolutionary opponent, Proc. Natl. Acad. Sci., Vol. 109, No. 26, 10409-10413, 2012.

[15] X. He, H. Dai, P. Ning, R. Dutta, Zero-determinant strategies for multiplayer multi-action iterated games. IEEE Signal Proc. Lett., 23, No. 3, 311-315, 2016.

[16] C. Hilbe, B. Wu, A. Traulsen,A. M.A. Nowak, Cooperation and control in multiplayer social dilemmas, Proc. Natl. Acad. Sci., Vol. 111, No. 46, 16425-16430, 2014.

[17] L. Pan, D. Hao, Z. Rong, T. Zhou, Zero-determinant strategies in iterated public goods game, Sci. Rep., Vol. 5, 13096, 2015.

[18] A. Szolnoki, M. Perc, Evolution of extortion in structured populations, Phys. Rev. E, Vol. 89, No. 2, 022804, 2014.

[19] R. Tan, Q. Su, B. Wu, L. Wang, Payoff control in repeated games, Proc. 33rd Chinese CDC, 997-1005, 2021.

[20] J. Wang, J. Guo, H. liu, A. Shen, Evolution of zero-determinant strategy on iterated snowdrift game, Actu Plus. Sin., Vol. 66, No. 18, 180203, 2017.

[21] J. Zhang, J. Lou, J. Qiu, J. Lu, Dynamics and covergence of hyper-networked evolutionary games with time delay is strategies, Information Sciences, Vol. 563, 166-182, 2021.

APPENDIX

1) Proof: of Proposition 2.11

(i) Since $L$ has unique eigenvalue 1, $\text{rank}(M) = \kappa - 1$. Hence there exists at least one $(\kappa - 1) \times (\kappa - 1)$ minor of $M$, which is nonsingular. Hence, $M^* \neq 0$. Observing that

$$MM^* = \text{det}(M) = 0,$$

and $\text{rank}(M) = \kappa - 1$, it follows that $\text{rank}(M^*) = 1$.

(ii) Assume there exists $1 \leq j \leq n$ such that $\text{Col}_j(M^*) = 0$. Consider $M \setminus \{\text{Row}_j(M)\}$, which is obtained from $M$ by deleting its $j$th row. Then all its $(\kappa - 1) \times (\kappa - 1)$ minors have zero determinants. That is, $M \setminus \{\text{Row}_j(M)\}$ is row-dependent. To get contradiction, we show that any $\kappa - 1$ rows of $M$ are linearly independent.
Proof: of Proposition 2.12
First we show that the limit exists. Since \( \{x(t) \mid t = 1, 2, \cdots \} \subset \Upsilon^k \) and \( \Upsilon^k \) is a compact set, there exists a subsequence \( \{x_{t_i} \mid i = 1, 2, \cdots \} \) such that
\[
\lim_{i \to \infty} x_{t_i} = x^* \in \Upsilon^k.
\]
Note that \( \lim_{t \to \infty} L^tP = P \), denote by \( x^0 = Px^* \), we claim that
\[
\lim_{t \to \infty} x_t = x^0. \tag{45}
\]
Given any \( \epsilon > 0 \), there exists \( N_1 \) such that when \( t_i > N_1 \)
\[\|x_{t_i} - x^*\| < \sqrt{\epsilon};\]
and there exists \( N_2 > 0 \) such that when \( t > N_2 \)
\[\|M^t - P\| < \sqrt{\epsilon}.\]
Choose an element \( t_{i_0} > N_1 \) from the subsequence and set
\[N_3 = t_{i_0} > N_1.\]
Assume \( t > N_2 + N_3 \), then
\[x(t) = M^{t-N_3}x(t_{i_0}).\]
Since \( t - N_3 > N_2 \) and \( N_3 = t_{i_0} > N_1 \), it follows that
\[\|x(t) - x^*\| < (\sqrt{\epsilon})^2 = \epsilon. \tag{45}\]
(45) follows. It is also clear that \( x^0 = x^* \). Moreover, \( Lx^* = x^* \) and \( Px^* = x^* \). Now since \( P = uu^T \in \mathcal{T}_{\kappa \times \kappa} \), without loss of generality, we can normalize \( u \) by replacing \( u \) by \( u/\|u\| \). Then \( v = 1_k. \) Moreover, \( x^* = Pxx^* = uu^Tx^* = u. \)

Proof: of Proposition 2.13
Note that \( MM^* = \det(M) = 0, \) and \( M u = 0. \) Since \( \text{rank}(M) = \kappa - 1 \), the solutions of equation \( Mx = 0 \) is a one-dimensional subspace. Now each column of \( M^* \) is a solution, the conclusion follows.

Proof: of Lemma 3.1
Recall the PEE (13), using (14), a straightforward computation shows that
\[
M = L - I_k = [p^1 - \delta_k^1, p^2 - \delta_k^2, \cdots, p^\kappa - \delta_k^\kappa], \tag{46}
\]
where
\[
\begin{align*}
p_{1,1}^1 & \quad p_{2,1}^1 \quad \cdots \quad p_{n,1}^1 \\
p_{1,2}^1 & \quad p_{2,2}^1 \quad \cdots \quad p_{n,2}^1 \\
\vdots & \quad \vdots & \quad \ddots & \quad \vdots \\
p_{1,n}^1 & \quad p_{2,n}^1 \quad \cdots \quad p_{n,n}^1
\end{align*}
\]
\[
p^r = \begin{bmatrix}
p_{1,1}^r \\
p_{1,2}^r \\
\vdots \\
p_{1,n}^r
\end{bmatrix}, \tag{47}
\]
It can be seen by a careful observation that
(i) \( \Phi_{i,j} \) is the rows where \( p_{i,j}^r \) appears.
(ii) Let \( s \in \Phi_{i,j} \). Denote
\[
\text{Row}_s(M) = (c_{i,j}^1(s)p_{i,j}^1, c_{i,j}^2(s)p_{i,j}^2, \cdots, c_{i,j}^\kappa(s)p_{i,j}^\kappa),
\]
where \( c_{i,j}^\ell(s) \) is used to express the coefficient of \( p_{i,j}^\ell \), which is the product of some \( p_{p,q}^r \), \( (p, q) \neq (i, j) \). Then
\[
\sum_{s \in \Phi_{i,j}} c_{i,j}^\ell(s) = 1, \quad \ell = 1, 2, \cdots, \kappa.
\]
(iii) If \( s \in \Phi_{i,j} \), the in the row summation \( \Xi \) a constant \( -[\delta_{i,j}^\ell]^T \) is added. Adding all such constants together yields \( \xi_{i,j} \).
The conclusion follows from (i), (ii), and (iii).

Proof: of Theorem 3.5
(Necessity) It is obvious that to make ZD strategies workable the existence of stationary distribution is necessary. That is,
\[
\lim_{t \to \infty} L^t x_0 = u, \quad \forall x_0 \in \Upsilon^\kappa. \tag{48}
\]
We first prove \( \lim_{t \to \infty} L^t \) exists. Since \( \Upsilon^\kappa \times \kappa \) is a compact set, if \( \lim_{t \to \infty} L^t \) does not exist, there must be at least two subsequences \( \{L^{n_i}\} \), and \( \{L^{m_i}\} \), such that
\[
\lim_{i \to \infty} L^{n_i} = P_1, \quad \lim_{i \to \infty} L^{m_i} = P_2,
\]
and \( P_1 \neq P_2 \). Say, \( \text{Col}_s(P_1) \neq \text{Col}_s(P_2) \), choosing \( x_0 = \delta_{k, i}^\ell \), then it violates (48).
Hence we have
\[
\lim_{t \to \infty} L^t = P.
\]
Again, because of (48) \( P \) should have the form that \( P = [u, u, \cdots, u] \), the conclusion is obvious.

As for the condition (ii), if \( \text{rank}(L - I_\kappa) < \kappa - 1 \), then \( M^* = 0 \) is a zero matrix. Then (29) fails. Hence (29) can never be obtained from (30), and ZD strategies do not work.

(Sufficiency) Replacing any row \( s \in \Phi_{i,j} \) of matrix \( M = L - I_\kappa \) by \( \Xi_{i,j} \), then condition (ii) ensures (22). Using (30) and expanding the determinant via replaced row, (29) follows. \( \blacksquare \)