SURVEY ON APPLICATIONS OF SEMI-TENSOR PRODUCT METHOD IN NETWORKED EVOLUTIONARY GAMES

Guodong Zhao¹, Haitao Li¹†, Peiyong Duan², and Fuad E. Alsaadi³

Abstract  Semi-tensor product (STP) method of matrices has received more and more attention from the communities of both engineering and economics in recent years. This paper presents a comprehensive survey on the applications of STP method in the theory of networked evolutionary games. In the beginning, some preliminary results on STP method are recalled. Then, the applications of STP method in many kinds of networked evolutionary games, such as general networked evolutionary games, networked evolutionary games with finite memories, networked evolutionary games defined on finite networks, and random networked evolutionary games, are reviewed. Finally, several research problems in the future are predicted.

Keywords  Semi-tensor product method of matrices, networked evolutionary games, random games, logical networks, algebraic formulation.

MSC(2010)  91-XX, 93-XX.

1. Introduction

The importance of networked evolutionary games (NEGs) has been fully recognized in recent years, and the investigation of NEGs has attracted much attention from physical, social, and engineering communities. The theory of NEGs was established after Nowak and May [89] introduced a network to the classical framework of evolutionary games [2,92]. In the network, nodes and edges denote, respectively, players and interaction relationship among players. In a NEG, players have their own specialized strategy updating rules, which are affected by their neighbors, to update their own strategy. This handling coincides with many practical economic activities, where every person often just plays game with its neighbors, who may be its friends or relatives, rather than plays game with all the other persons, like

¹the corresponding author. Email address:haitaoli09@gmail.com(H. Li)
¹School of Mathematics and Statistics, Shandong Normal University, 250014 Jinan, China
²School of Information Science and Engineering, Shandong Normal University, 250014 Jinan, China
³Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, 21589 Jeddah, Saudi Arabia
*The authors were supported by the Natural Science Foundation of Shandong Province under grants ZR2018BA001 and ZR2015FQ003, the National Natural Science Foundation of China under grants 61903237, 61873150 and 61773246, the Natural Science Fund for Distinguished Young Scholars of Shandong Province under grant JQ201613, and the Young Experts of Taishan Scholar Project.
in normal multi-player games [87]. The earliest work [89] investigated the cooperation emergence and persistence of Prisoner’s Dilemma Game on two-dimensional lattices. With the development of complex networks, many great works have been made toward NEGs.

The most appealing research topic, among the study of NEGs, is to analyze each player’s behavior when the dynamics proceeds. Many results attempted to solve it. [126], which studied the networked adaptive dynamics of Prisoner’s Dilemma Game, proved that there existed a full-defective state or a highly cooperative steady state. [116] simulated animal conflicts on different networks as a finite dynamical game and it delineated that the evolution result was that whether one strategy dominated or two strategies coexisted on the network. In addition to that, the existing works on NEGs are mostly based on the mean-field method [3], and the experiment or computer simulation method [45].

Recently, a new powerful mathematical tool, called the semi-tensor product (STP) of matrices, has been proposed by Cheng [4]. Up to now, this method has been successfully applied to the analysis and control of Boolean networks and mixed-valued logical networks, and many excellent results have been obtained [4, 5, 17, 18, 21–23, 32, 36–38, 40–42, 46, 47, 75–77, 79, 80, 86, 88, 96, 105–107, 110–112, 123, 125]. Especially, [128] studied further results on the controllability of Boolean control networks. [129] investigated the optimal control of Boolean control networks. Pinning control for the disturbance decoupling problem of Boolean networks was considered in [78]. In addition, with STP method in hands, there are many researchers attempting to investigate NEGs in the view of control theory. [6] firstly analyzed NEGs in the view of control theory. An algorithm was proposed by [13] to convert the given NEGs into an algebraic expression based on “myopic best response adjustment (MBRA) rule”. [118] studied the NEGs on finite networks and presented some interesting results. [14] applied STP method to a class of event-triggered control for finite evolutionary networked games. [102] converted weighted potential game to weighted harmonic game. [8] and [35] investigated stochastic stability and stabilization of n-person random evolutionary Boolean games and algebraic formulation and Nash equilibrium of competitive diffusion games, respectively.

This paper gives a comprehensive survey on the applications of STP method in the theory of networked evolutionary games. For different kinds of NEGs, including general networked evolutionary games, networked evolutionary games with finite memories, networked evolutionary games defined on finite networks, and random networked evolutionary games, we delineate the basic applications of STP method in them.

The remainder of the paper is organized as follows. Section 2 contains some necessary preliminaries on STP and game theory. Section 3 presents the description of general NEGs. Section 4 delineates NEGs with finite memories. Section 5 describes the NEG defined on finite networks. Section 6 gives the basic description of control of NEGs. Section 7 recalls random evolutionary games, which is followed by a brief conclusion in Section 8.

Notations. \( \mathbb{R}^{m \times n} \) denotes the set of \( m \times n \) real matrices. \( \mathbb{R}^{+}_{m \times n} \) denotes the set of \( m \times n \) nonnegative real matrices. \( \Delta_n := \{ \delta^i_n | i = 1, 2, \ldots, n \} \), where \( \delta^i_n \) is the \( i \)-th column of the identity matrix \( I_n \). An \( n \times t \) matrix \( M \) is called a logical matrix, if \( M = [\delta^1_n \delta^2_n \cdots \delta^t_n] \), which is briefly denoted by \( M = \delta_n[i_1 \ i_2 \ \cdots \ i_t] \). Define the set of \( n \times t \) logical matrices as \( \mathcal{L}_{n \times t} \). \( \text{Col}_i(L) \) (\( \text{Row}_i(L) \)) is the \( i \)-th column (row) of matrix \( L \). For a set \( E \), \( |E| \) denotes the number of elements in \( E \).
$r = (r_1, \ldots, r_k)^T \in \mathbb{R}_k$ is called a probabilistic vector, if $r_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^k r_i = 1$. The set of $k$ dimensional probabilistic vectors is denoted by $\Upsilon_k$. If $M \in \mathbb{R}_{m \times n}^+$ and $\text{Col}(M) \subseteq \Upsilon_m$, $M$ is called a probabilistic matrix. The set of $m \times n$ probabilistic matrices is denoted by $\Upsilon_{m \times n}$.

2. Preliminaries

In this section, we give some necessary preliminaries, which will be used throughout this paper.

Definition 2.1 ([4]). The semi-tensor product of two matrices $A \in \mathbb{R}_{m \times n}$ and $B \in \mathbb{R}_{p \times t}$ is defined as $A \blacktriangledown B = (A \otimes I_p)(B \otimes I_t)$, where $\alpha = \text{lcm}(n, p)$ is the least common multiple of $n$ and $p$, and $\otimes$ is the Kronecker product.

It is noted that the semi-tensor product is a generalization of the ordinary matrix product, and thus we can simply call it “product” and omit the symbol “$\blacktriangledown$” without confusion.

Definition 2.2. Let $M \in \mathbb{R}_{p \times s}$ and $N \in \mathbb{R}_{q \times s}$. Define the Khatri-Rao product of $M$ and $N$, denoted by $M \blacktriangle N$, as $M \blacktriangle N = [\text{Col}_1(M) \blacktriangledown \text{Col}_1(N) \ \text{Col}_2(M) \blacktriangledown \text{Col}_2(N) \ \cdots \ \text{Col}_s(M) \blacktriangledown \text{Col}_s(N)] \in \mathbb{R}_{pq \times s}$.

The semi-tensor product of matrices has the following important properties.

Lemma 2.1 ([4]).
1. Let $X \in \mathbb{R}_m$ and $Y \in \mathbb{R}_n$ be two column vectors. Then, $W_{[m,n]}XY = YX$, where $W_{[m,n]}$ is called the swap matrix. Especially $W_{[n,n]} := W_{[n]}$.

2. (pseudo-commutative property) Let $X \in \mathbb{R}_t$ and $A \in \mathbb{R}_{m \times n}$. Then, $XA = (I_t \otimes A)X$ holds.

Lemma 2.2 ([6]). Assume $X \in \Upsilon_p$ and $Y \in \Upsilon_q$. Define two dummy matrices, named by “front-maintaining operator” (FMO) and “rear-maintaining operator” (RMO) respectively, as:

$$D_f^{p,q} = \delta_p[1 \cdots 1 2 \cdots 2 \cdots p \cdots p]_q$$

$$D_r^{p,q} = \delta_q[1 2 \cdots q 1 2 \cdots q \cdots 1 2 \cdots q]_p$$

Then $D_f^{p,q}XY = X$, $D_r^{p,q}XY = Y$.

An $n$-ary pseudo-logical (or logical) function $f(x_1, x_2, \ldots, x_n)$ is a mapping from $\Delta^n_k$ to $\mathbb{R}$ (or from $\Delta^n_k$ to $\Delta_m$). The following result shows how to express a pseudo-logical (or logical) function into its algebraic form.

Lemma 2.3 ([21]). Let $f: \Delta^n_k \rightarrow \mathbb{R}$ (or $f: \Delta^n_k \rightarrow \Delta_m$) be a pseudo-logical (or logical) function. Then there exists a unique matrix $M_f \in \mathbb{R}_{1 \times k^n}$ (or $M_f \in \mathcal{L}_{m \times k^n}$), called the structural matrix of $f$, such that

$$f(x_1, x_2, \cdots, x_n) = M_f \blacktriangledown_{i=1}^n x_i,$$

where $x_i \in \Delta_k$, $i = 1, 2, \cdots, n$, $\text{Col}_j(M_f) = f(\delta_{k^n}^i)$, and $j = 1, 2, \cdots, k^n$. 
In the following, we recall some notations in game theory.

A normal finite game \((N, S, P)\), considered in this paper, consists of three factors [12]:

(i) \(n\) players \(N = \{1, 2, \ldots, n\}\);

(ii) Player \(i\) has strategy set \(S_i, i = 1, 2, \ldots, n\), and \(S := \prod_{i=1}^{n} S_i\) is the set of strategy profiles;

(iii) Player \(i\) has its payoff function \(p_i : S \rightarrow \mathbb{R}, p_i \in P, \text{ and } i = 1, 2, \ldots, n\).

**Definition 2.3** ([12]). In the \(n\)-player normal form finite game

\[ G = \{S_1, \ldots, S_n; p_1, \ldots, p_n\}, \]

the strategy profile \((s_1^*, s_2^*, \ldots, s_n^*)\) is called a Nash Equilibrium (NE), if for each player \(i\), \(s_i^*\) is (at least tied for) player \(i\)'s best response to the strategies specified for the \(n - 1\) other players, \((s_1^*, \ldots, s_{i-1}^*, s_{i+1}^*, \ldots, s_n^*)\), that is,

\[ p_i (s_1^*, \ldots, s_{i-1}^*, s_i^*, s_{i+1}^*, \ldots, s_n^*) \geq p_i (s_1^*, \ldots, s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_n^*), \]

for every feasible strategy \(s_i \in S_i\), where \(S_i\) is the set of strategies of player \(i\) and \(p_i\) is the corresponding payoff function.

**Definition 2.4** ([6]). (1) A normal game with two players is called a fundamental network game (FNG), if \(S_1 = S_2 := S_0 = \{1, 2, \ldots, k\}\) and player \(i\)'s payoff function is \(c_i = c_i(x, y)\), where \(x\) is player one's strategy, \(y\) is player two's strategy, and \(i = 1, 2\). Namely, \(N = \{1, 2\}, S = S_0 \times S_0\), and \(P = \{c_1, c_2\}\).

(2) An FNG is symmetric, if \(c_1(x, y) = c_2(y, x)\), \(\forall x, y \in S_0\).

### 3. General Networked Evolutionary Games

#### 3.1. Description of Networked Evolutionary Games

Actually, the following definition of NEGs is very typical and representative [6,13]. In the rest of this paper, all kinds of NEGs are the variations of Definition 3.1.

**Definition 3.1.** A general NEG consists of the following three ingredients:

(1) A network: the network is a connected graph \((N, E)\), where \(N := \{1, 2, \ldots, n\}\) is the set of all the players, \(E = \{(i, j) | \text{ there exists edge between players } i \text{ and } j \text{ in the network }\}\) is the set of edges;

(2) A FNG: if \((i, j) \in E\), then \(i\) and \(j\) play the FNG with strategies \(x_i(t)\) and \(x_j(t)\) at time \(t\), separately. Especially, if the FNG is not symmetric, then the corresponding network must be directed to show that \(i\) is player one and \(j\) is player two;

(3) Players' strategy updating rules: for network \(z\), the rule can be written as

\[ x_i(t) = f_i (x_i(0), x_i(1), \ldots, x_i(t-1), x_j(0), x_j(1), \ldots, x_j(t-1) | j \in N_i) \tag{3.1} \]

where \(x_j(\tau) \in S\) is the strategy of player \(j\) at time \(\tau, \tau = 0, 1, \ldots, t - 1, N_i\) is the neighborhood of player \(i\) in the network \(z\), that is, \(j \in N_i\) if and only if \((i, j) \in E, i \in N\). Obviously, \(i \notin N_i\) and \(j \in N_i \iff i \in N_j\).
In the following, there is an example to illustrate Definition 3.1.

![Network Structures](image)

**Figure 1.** Two typical network topological structures

**Example 3.1.** Firstly, in Figure 1, we illustrate some conceptions of the network topological structures in the definition of NEGs.

(i) For network (a) in Figure 1, we have

- the network topological structure \((N_a, E_a)\), where \(N = \{1, 2, 3, 4, 5, 6\}\) and
  
  \[E_a = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (1, 6)\};\]

- the neighbours of players: \(N_1 = \{2, 6\}\), \(N_2 = \{1, 3\}\), \(N_3 = \{2, 4\}\), \(N_4 = \{3, 5\}\), \(N_5 = \{4, 6\}\), and \(N_6 = \{1, 5\}\).

(ii) For network (b) in Figure 1, we have

- the network topological structure \((N_b, E_b)\), where \(N = \{1, 2, 3, 4, 5\}\) and
  
  \[E_b = \{(1, 2), (2, 3), (3, 4), (4, 5)\};\]

- the neighbours of players: \(N_1 = \{2\}\), \(N_2 = \{1, 3\}\), \(N_3 = \{2, 4\}\), \(N_4 = \{3, 5\}\), and \(N_5 = \{4\}\).

<table>
<thead>
<tr>
<th>Table 1. Payoff bi-matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Player 1</strong> | <strong>Player 2</strong></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td><strong>M</strong></td>
</tr>
<tr>
<td><strong>F</strong></td>
</tr>
</tbody>
</table>

Secondly, we endow the Prisoner’s Dilemma Game, whose payoff bi-matrix is shown in Table 1, to the networks (a) and (b) as the FNG, i.e., if \((i, j) \in E_a/E_b\), then player \(i\) and player \(j\) play the FNG.

Finally, we endow strategy updating rule, myopic best response adjustment (MBRA) for example, to all the players in the networks.

Thus, we construct two NEGs defined on the networks (a) and (b) separately. \(\square\)

Actually, there are many classical strategy updating rules such as MBRA and unconditional imitation. In this paper, we focus on the MBRA strategy updating rule and describe it in the following subsection in details.
### 3.2. Algebraic Formulation of Networked Evolutionary Games

This subsection presents the basic algebraic method to formulate the given NEGs, which are defined as in Definition 3.1. To reach this target, the most important thing is to convert the strategy updating rules into algebraic expressions. [13] gives a concise method to achieve this target.

Firstly, we need to get the payoff functions for all the players in the given NEG. Player $i$ only plays with its neighbors, and its aggregate payoff $p_i : S_{N_i}^{N_i + 1} \rightarrow \mathbb{R}$ is the sum of payoffs gained by playing with all its neighbors, that is,

$$p_i(x_i, x_j | j \in N_i) = \sum_{j \in N_i} p_{ij}(x_i, x_j), \quad x_i, x_j \in S,$$

where $p_{ij} : S \times S \rightarrow \mathbb{R}$ denotes the payoff of player $i$ playing with its neighbor $j$.

Secondly, under the MBRA strategy updating rule, every player holds the opinion that its neighbours will make the same decisions as in their last step, and the strategy at present time is the best response against its neighbors’ strategies in the last step. Based on this assumption, we get

$$x_i(t) \in Q_i := \arg \max_{x_i \in S} p_i(x_i, x_j(t - 1) | j \in N_i). \quad (3.2)$$

Additional, when player $i$ may have more than one best responses, that is, $|Q_i| > 1$. We set a priority for the strategies as follows: for $s_i, s_j \in S$, $s_i > s_j$ if and only if $i > j$. Thus, it guarantees that we can obtain a pure strategy dynamics from this method.

This following algorithm is one of the main results in the [13].

**Algorithm 3.1 ([13]).** The algorithm contains three steps:

1). Calculate the structural matrix, $M_{p_i}$, of the payoff function of each player $i \in N$.

$$M_{p_i} = V_r (A^T (D_r^{k,k})^{n-2} \times \left( \sum_{j<i, j \in N_i} W_{[k^i,k^{n-j-1}]} + \sum_{j>i, j \in N_i} W_{[k^{j-1},k^{n-j}]} \right),$$

where $A^T$ is the structural matrix of the payoff function in the FNG.

2). Divide the matrix $M_{p_i}$ as $k^{n-1}$ equal blocks:

$$M_{p_i} = \left[ \text{Blk}_1(M_{p_i}), \cdots, \text{Blk}_{k^{n-1}}(M_{p_i}) \right]. \quad (3.3)$$

For all $l = 1, 2, \cdots, k^{n-1}$, find the column index $\xi_l$, so that

$$\xi_l = \max\left\{ \xi_l | \text{Col}_{\xi_l}(\text{Blk}_l(M_{p_i})) = \max_{1 \leq \xi \leq k} \text{Col}_{\xi}(\text{Blk}_l(M_{p_i})) \right\}$$

holds.

3). Construct the algebraic expression of the game as

$$x(t + 1) = Lx(t), \quad (3.4)$$

where $\text{Col}_i(L) = \text{Col}_i(L_1) \times \cdots \times \text{Col}_i(L_n)$, $L_i = \tilde{L}_i D_r^{k,k} W_{[k^{i-1},k]}$, $\tilde{L}_i = \delta_k[\xi_1, \cdots, \xi_{k^{n-1}}]$, $i \in N$, and $L \in \mathbb{L}_{k^n \times k^n}$. 

Therefore, we can analyze the behaviors of all the players in the given NEG via (3.4). Because the algebraic form (3.4) reveals all the characteristics of the game. In other words, we can investigate the properties of $L$ to analyze the dynamical process of the game. We can obtain the final states of the NEG via the logical network theory easily. The following example illustrates Algorithm 3.1.

**Example 3.2 ([13])**. Consider an NEG with the following items:

- Four players: The player set is $N = \{1, 2, 3, 4\}$, and every player has the same strategy set $S = \{s_1, s_2\}$;
- a network shown in Figure 2;
- a payoff matrix $A = [2 \ 4 \ 0 \ 10]$ for any pair of players on the network;
- the strategy updating rule is the MBRA rule.

With Algorithm 3.1 in hands, we convert the given NEG into the following expression: $x(t+1) = Lx(t)$, where $L = \delta_{16}[1 \ 7 \ 10 \ 16 \ 10 \ 16 \ 10 \ 16 \ 7 \ 7 \ 16 \ 16 \ 16 \ 16 \ 16 \ 16]$.

Then, we can analyze the behaviors of all the players in the given NEG via structural matrix $L$.

### 4. Networked Evolutionary Games with Finite Memories

Note that if we adopt MBRA strategy updating rules, the corresponding NEGs are with one memory, that is, in the NEGs, each individual determines their own strategy choices of the next move only based on their neighbors’ strategies at the last step. However, many practical economic activities imply an obvious fact that every individual can remember more than one strategies of their neighbors. Thus, all the players make their strategy choices in the next move according to their neighbors’ strategies in the last $\tau$ steps with $1 \leq \tau < \infty$. Then, the assumption that all the players in the NEGs can remember their neighbors’ strategies in the past $\tau$ steps is very reasonable. For this situation, [122] presents a new definition of NEGs with finite memories.

**Definition 4.1.** An NEG with $\tau$ memories consisting of the following three ingredients:

1. A network: it is a connected undirected graph $(N, \mathcal{E})$, where $N := \{1, 2, \cdots, n\}$ is the set of all players, and $\mathcal{E} = \{(i, j) \mid \text{there exists interaction between players } i \text{ and } j\}$ is the set of edges;
(2) A FNG: if \((i, j) \in \mathcal{E}\), then \(i\) and \(j\) play the FNG in the network with strategies \(x_i(t)\) and \(x_j(t)\) at time \(t\), separately.

(3) Players’ strategy updating rules: these rules can be expressed as

\[
\begin{align*}
x_i(t + 1) &= f_i(x_i(t - \tau + 1), \cdots, x_i(t)), \\
x_j(t - \tau + 1), \cdots, x_j(t) &\mid j \in \mathcal{N}_i,
\end{align*}
\]

where \(x_j(l) \in \mathcal{S}_0\) is the strategy of player \(j\) at time \(l\), \(l = t - \tau + 1, t - \tau + 2, \cdots, t\). \(\mathcal{N}_i\) is the neighborhood of player \(i\), that is, \(j \in \mathcal{N}_i\) if and only if \((i, j) \in \mathcal{E}\), \(i \in \mathcal{N}\). Obviously, \(i \notin \mathcal{N}_i\) and \(j \in \mathcal{N}_i \iff i \in \mathcal{N}_j\).

We rewrite the item (3) in Definition 3.1 as the item (3) in Definition 4.1. In addition to that, we adopt the \(\tau\)-memory version of *Fictitious Play process* [122] as the strategy updating rules in this section.

For player \(i \in \mathcal{N}\), define *empirical frequency* \(q^1_i(t)\), which is the percentage of stages at which player \(i\) has chosen the strategy \(j \in \mathcal{S}_0\) at time \(t = t - \tau + 1, \cdots, t\), that is,

\[
q^1_i(t) := \frac{1}{\tau} \sum_{l=t-\tau+1}^{t} I\{x_i(l) = j\},
\]

where \(x_i(l) \in \mathcal{S}_0\) is player \(i\)'s strategy chosen at time \(l = t - \tau + 1, t - \tau + 2, \cdots, t\), and \(I\{\cdot\}\) is indicator function. Then, we can define *empirical frequency vector* for player \(i\) at time \(t\) as

\[
q_i(t) = (q^1_i(t), q^2_i(t), \cdots, q^\tau_i(t))^T.
\]

For player \(i\), the chosen strategy at time \(t + 1\) is based on the assumption that the other players are playing randomly and independently according to \(q_j(t)\), where \(j = 1, \cdots, i - 1, i + 1, \cdots, n\). Under this presumption, the expected payoff function for the strategy \(x_i \in \mathcal{S}_0\) of player \(i\) can be written as

\[
U_i(x_i, q_{-i}(t)) := \sum_{x_{-i} \in \mathcal{S}^{n-1}} \left( p_i(x_i, x_{-i}) \prod_{x_j \in x_{-i}} q^\tau_j(t) \right),
\]

where \(q_{-i}(t) := (q_1(t), \cdots, q_{i-1}(t), q_{i+1}(t), \cdots, q_n(t))\) and \(x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)\). In the \(\tau\)-memory version of Fictitious Play process, player \(i\) used the expected payoff (4.3) to select the strategy at time \(t + 1\) from

\[
EP_i(q_{-i}(t)) := \left\{ \tilde{x} \in \mathcal{S}_0 \mid U_i(\tilde{x}, q_{-i}(t)) = \max_{x \in \mathcal{S}} U_i(x, q_{-i}(t)) \right\},
\]

which is called player \(i\)'s best response to \(q_{-i}(t)\), i.e.,

\[
x_i(t + 1) \in EP_i(q_{-i}(t)).
\]

Similarly, we also set a priority to all the strategies to obtain a pure strategy dynamics. Furthermore, the following algorithm is one of the main results in the [122].

**Algorithm 4.1.** This algorithm contains three steps:
Table 2. Payoff bi-matrix

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>(2, 2)</td>
<td>(1, 0)</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>(0, 1)</td>
<td>(3, 3)</td>
<td></td>
</tr>
</tbody>
</table>

(1) Calculate the structural matrix, \( M_i \), of the payoff function of \( i \in N \) by
\[
M_i = M_{c} W_{[k]} \left( \sum_{j<i,j \in N_i} D_{f}^{k_{n}-2} W_{[k_{j-1},k]} + \sum_{j>i,j \in N_i} D_{f}^{k_{n}-2} W_{[k_{j-1},k]} \right);
\]

(2) Construct the structural matrix of expected payoff function of \( i \in N, M_{c} = M_{i} C \), where \( C \) is defined as in Proposition 3.1 in [122]. Then, divide the matrix \( M_{c} \) into \( k(n-1) \) equal blocks into
\[
M_{c, i} = [Blk_1(M_{c, i}), Blk_2(M_{c, i}), \cdots, Blk_k(n-1)τ(M_{c, i})],
\]
and for all \( l = 1, 2, \cdots, k(n-1)τ \), find the column index \( ξ_{i, l} \) such that
\[
ξ_{i, l} = \max \{ξ | Col_ξ(Blk_l(M_{c, i})) = \max_{1 \leq ξ \leq k} Col_ξ(Blk_l(M_{c, i}))\};
\]

(3) Construct the algebraic form of NEGs with \( τ \) memories as
\[
z(t+1) = Lz(t),
\]
where \( L = L_1 * L_2 * \cdots * L_n, L_i = D_{r}^{k_{n}-2} W_{[k_{i(i-1)τ},kτ]} (I_{k^{n-1}} \otimes L_{i}^{ξ}) \Psi_{n, kτ}, L_{i}^{ξ} = \delta_{k}^{[ξ],1, ξ_{i,2}, \cdots, ξ_{i,k(n-1)τ]}, \text{ and } i \in N.\)

Therefore, we can analyze the behaviors of all the players in the given NEG via the (4.4). Because the algebraic form (4.4) reveals all the characteristics of the NEG with finite memories. In other words, we can investigate the properties of \( L \) to analyze the dynamical process of the NEG with finite memories. We can obtain the final states of the NEG via the logical network theory easily. The following example prove that Algorithm 4.1 is very effective.

Example 4.1. Consider an NEG with the following basic factors:

- A network is \( (N, E) \), where \( N = \{1, 2, 3\} \), and \( E = \{(1, 2), (1, 3), (2, 3)\} \);
- The FNG’s payoff bi-matrix shown in Table 2;
- The adjusting rule is \( τ \)-version of FP, where \( τ = 2 \).

With the help of Algorithm 4.1, we convert the above NEG with finite memories into \( z(t+1) = Lz(t) \), where
\[
L = \delta_{64}^{[1 3 1 23 9 27 25 31 1 19 17 23 26 28 26 32 33 39 37 55 42 64 62 64 34 56 54 56 58 64 62 64 1 7 5 23 10 32 30 32 2 24 22 24 26 32 30 32 38 40 38 56 46 64 62 48 38 56 54 56 62 64 62 64]}.
\]
Then, from (4.5), by the theory of logical network, one can get that (i) the fixed points are $\delta_{64}^{1}$ and $\delta_{64}^{4}$, that is, all players adopt the same strategy $M$ or $F$; (ii) two cycles with length 3 are $\{\delta_{64}^{1}, \delta_{64}^{2}, \delta_{64}^{3}\}$ and $\{\delta_{64}^{16}, \delta_{64}^{17}, \delta_{64}^{18}\}$, namely, the strategy profile sequences ($\{M, F, M\}$, $\{M, M, F\}$, $\{F, M, M\}$) and ($\{M, M, F\}$, $\{F, M, M\}$, $\{M, F, M\}$) are two cycles adopted by the three players; (iii) $N_{s} = 0$, $s = 2$ and $4 \leq s \leq 64$.

This section considers the NEGs with finite memories as a dynamic system with time delay. Actually, time delay is a very common situation. Many great works in the community of control and engineering, have reached towards this kind of systems. See in [1, 15, 48, 49, 52–54, 57, 61–71, 90, 91, 94, 97–99, 103, 113, 124, 127, 130, 131] for details.

5. Networked Evolutionary Games Defined on Finite Networks

It is noticed that the network involved in the above mentioned NEGs is one. Actually, many economic activities imply an obvious fact that each participator, who participates in an evolutionary game, will adjust their opponents or neighbours to obtain more as the evolutionary game processes. The structure of network will be changing with the NEG processing. Thus, NEGs defined on finite networks are very meaningful.

**Definition 5.1.** A NEG defined on finite networks consisting of the following four ingredients:

1. A set of finite networks $\mathcal{M} := \{1, 2, \ldots, m\}$: each network is a connected graph $\langle N, E_{z}\rangle$, where $N := \{1, 2, \ldots, n\}$ is the set of players, $E_{z} = \{(i, j) \mid$ there exists interaction between players $i$ and $j$ in network $z\}$ is the set of edges, and $z \in \mathcal{M}$;

2. A FNG: if $(i, j) \in E_{z}$, then $i$ and $j$ play the FNG in network $z$ with strategies $x_{i}(t)$ and $x_{j}(t)$ at time $t$, respectively. Particularly, if the FNG is not symmetric, then the corresponding network must be directed to show that $i$ is player one and $j$ is player two;

3. Players’ strategy updating rules: for network $z$, the rule can be expressed as

   $$x_{i}(t) = f_{i,z}(x_{i}(0), x_{i}(1), \ldots, x_{i}(t-1), x_{j}(0), x_{j}(1), \ldots, x_{j}(t-1) \mid j \in \mathcal{N}_{i,z}),$$

   (5.1)

   where $x_{j}(\tau) \in S$ is the strategy of player $j$ at time $\tau$, $\tau = 0, 1, \ldots, t-1$, $\mathcal{N}_{i,z}$ is the neighborhood of player $i$ in the network $z$, that is, $j \in \mathcal{N}_{i,z}$ if and only if $(i, j) \in E_{z}$, $i \in N$, and $z \in \mathcal{M}$. Obviously, $i \notin \mathcal{N}_{i,z}$ and $j \in \mathcal{N}_{i,z} \iff i \in \mathcal{N}_{j,z}$;

4. Network updating rule: a network selector $z(t)$ and its updating rule is

   $$z(t) = g(x(0), x(1) \cdots, x(t)),$$

   (5.2)

   where $x(\tau) = (x_{1}(\tau), x_{2}(\tau), \cdots, x_{n}(\tau)) \in S^{n}$ is the strategy profile of all players at time $\tau = 0, \ldots, t$.

Because of more than one networks, we assign the selector $z(t)$ to choose the network in the next step. It is the main difference from [13] and [6]. We consider a NEG defined on finite networks as finite NEGs defined on a fixed network.
In this section, we still consider the myopic best response adjustment rule [9]. Then, we get
\[
x_i(t) \in Q_{i,z} := \arg \max_{x_i \in S_{i,z}} p_i, z \ (x_i, x_j(t - 1) \mid j \in N_{i,z}), \ i \in N, \ z \in \mathcal{M}.
\] (5.3)

Note that, via (5.3), players get their expected revenue at time \( t \) relying on the last strategy profile \( x(t - 1) \). Therefore, player \( i \) obtains the expected revenue \( ER_{i,z}(x(t - 1)) = p_i, z(x(t - 1), x(t)) \), where \( i = 1, \cdots, n \), and \( x(t) \in Q_{i,z} \). Thus, one gets a set of networks \( W_i(x(t - 1)) \), where player \( i \) want to attend at time \( t \) to maximize its earning, in the form of
\[
W_i(x(t - 1)) := \arg \max_{z \in \mathcal{M}} ER_{i,z}(x(t - 1)).
\] (5.4)

Thus, by (5.4) and enumerating all the players, we get the number of players who want to participate in network \( z \) at time \( t \)
\[
\delta_z(x(t - 1)) = \left| \left\{ W_i(x(t - 1)) \mid i \in N \text{ and } z \in W_i(x(t - 1)) \right\} \right|,
\]
where \( z \in \mathcal{M} \). Therefore, under strategy profile \( x(t - 1) = \delta_{i,n} \), the selector chooses the network
\[
z(t) \in \mathcal{P}_i := \arg \max_{z \in \mathcal{M}} \delta_z(x(t - 1)),
\]
in which the most players want to participate at time \( t \), to make the evolutionary process of the game proceed. Similarly, when \( |\mathcal{P}_i| > 1 \), we use the aforementioned idea of priority.

Furthermore, the following algorithm is one of the main results in the [118].

**Algorithm 5.1.** [118]

1. Calculate the structural matrix, \( M_{p_i,z} \), of the payoff function of player \( i \in N \) in network \( z \in \mathcal{M} \) as \( M_{p_i,z} = M_i D_r^{k_{n-2},k^2} \times \left( \sum_{j < i, j \in N_{i,z}} W_{[j,k^{n-j-1}]} + \sum_{j > i, j \in N_{i,z}} W_{[k-j-1,k^{n-j-1}]} \right) \);
2. Divide the matrix \( M_{p_i,z} \) into \( k^{n-1} \) equal blocks into \( M_{p_i,z} = [Blk_1(M_{p_i,z}), Blk_2(M_{p_i,z}), \cdots, Blk_{k^{n-1}}(M_{p_i,z})] \), and for all \( l = 1, 2, \cdots, k^{n-1} \), find the column index set \( \Xi_{i,l,z} \), such that
\[
\Xi_{i,l,z} = \max \left\{ \xi_l \mid Col_{\xi_l}(Blk_l(M_{p_i,z})) = \max_{1 \leq \xi \leq k} Col_{\xi}(Blk_l(M_{p_i,z})) \right\};
\]
3. Define \( r_{i,l,z} = |\Xi_{i,l,z}| \geq 1 \), and construct the algebraic expression of the game in network \( z \) as
\[
x_i(t + 1) = L^p_i z(t) x(t),
x(t + 1) = L^p z(t) x(t),
\] (5.5)
where \( L^p = L^p_1 L^p_2 \cdots L^p_r, L^p_i = [L^p_{i,1}, L^p_{i,2}, \cdots, L^p_{i,m}], L^p_i, z = L^p_i D_r^{k,k} W_{[k^{n-1},k]}; Row_{\xi_i}(Col_{\xi}(L^p_{i,z})) = \frac{1}{r_{i,l,z}}, Row_p(Col_{\xi}(L^p_{i,z})) = 0, q \in \Xi_{i,l,z}, p \notin \Xi_{i,l,z}, l = 1, 2, \cdots, k^{n-1}, i \in N, \text{ and } z \in \mathcal{M};
Table 3. Payoff Matrix

<table>
<thead>
<tr>
<th>Player 1 \ Player 2</th>
<th>Mum</th>
<th>Fink</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mum</td>
<td>(2, 2)</td>
<td>(1, 0)</td>
</tr>
<tr>
<td>Fink</td>
<td>(0, 1)</td>
<td>(3, 3)</td>
</tr>
</tbody>
</table>

(4) Use

\[ x_{-i}(t) = D^k_{i,k} W_{[k_{i-1}, k_j]} x(t) := \delta^{i_j}_{k(n-1)} , \]

and

\[ w_{i,j} := \arg \max_{z \in \mathcal{M}} \left\{ Col_{i,j,z}(Blk_{j_l}(M_{p_{i,z}})) \mid z \in \mathcal{M} \right\}, \]

to construct

\[ \mathcal{W}_l = \sum_{i \in \mathcal{N}} \sum_{\eta \in w_{i,j}} \delta_{\eta}^l m, \]

and for all \( l = 1, 2, \cdots, k^n \), find the row index set \( \Theta_l \), such that

\[ \Theta_l = \max \left\{ j \mid Row_j(\mathcal{W}_l) = \max_{1 \leq i \leq m} Row_i(\mathcal{W}_l) \right\} ; \]

(5) Define \( \mu_l = |\Theta_l| \geq 1 \) and construct the algebraic expression of the network adjusting rule as

\[ z(t + 1) = L^p z(t) x(t), \tag{5.6} \]

where \( L^p = L^p_1 L^p_2 \cdots L^p_{k^n} \), \( L_p = [\hat{L}_1, \hat{L}_2, \cdots, \hat{L}_{k^n}] \), \( Row_q(\text{Col}_{l}(L^p_q)) = \frac{1}{\mu_l} \), \( Row_p(\text{Col}_{l}(\hat{L}^p_l)) = 0 \), \( q \in \Theta_l, p \notin \Theta_l \), and \( l = 1, 2, \cdots, k^n \). Then, by (5.5) and (5.6), one has

\[ X(t + 1) = L^p X(t), \tag{5.7} \]

where \( X(t) = z(t) \times x(t) \) and \( L^p = L^p_2 * L^p_2 \) is called the transition matrix of the given NEG.

The following example illustrates Algorithm 5.1.

**Example 5.1 ( [118])**. The given NEG defined on finite networks consisting the following basic ingredients:

- Three network, denote by \((N, \mathcal{E}_z)\), where \( z \in \mathcal{M} = \{1, 2, 3\}, N = \{1, 2, 3, 4\}, \mathcal{E}_1 = \{(1, 2), (3, 4)\}; \mathcal{E}_2 = \{(1, 2), (2, 3), (3, 4)\}, \) and \( \mathcal{E}_3 = \{(1, 2), (1, 4), (2, 3), (3, 4)\} ; \)
- The FNG is Prisoner’s Dilemma Game with the payoff bi-matrix shown in Table 3;
- The evolutionary rule is the myopic best response adjustment rule;
- The network selector’s updating rule is the majority voting system.
According to Algorithm 5.1, it is easy to obtain the following algebraic expression:

\[ X(t+1) = LX(t), \]

where \( X(t) = z(t) \bowtie x(t) \), \( x(t) = \bigotimes_{i=1}^{n} x_i(t) \), and

\[ L = \delta_{48}[33 \ 35 \ 36 \ 41 \ 42 \ 44 \ 37 \ 39 \ 38 \ 40 \ 45 \ 47 \ 48 \ 37 \ 39 \ 38 \ 40 \ 47 \ 48 \ 33 \ 35 \ 34 \ 44 \ 43 \ 43 \ 44 \ 38 \ 48 \ 38 \ 48 \ 48 \ 48 \ 48 \ 48]. \]

Thus, by the theory of logical network, one can get two final states: (i) the fixed points are \( \delta_{33}^4 \) and \( \delta_{48}^4 \), namely, all players adopt the same strategy Mum or Fink in network 3; (ii) the unique cycle with length 2 is \( \{ \delta_{38}^4, \delta_{43}^4 \} \), which means that the profiles \( \{ \text{Mum, Fink, Mum, Fink} \} \) and \( \{ \text{Fink, Mum, Fink, Mum} \} \) are adopted alternately by the four players; (iii) \( N_s = 0, 3 \leq s \leq 3 \).

6. Control of Networked Evolutionary Games

When we are investigating the control and optimization of NEGs, without loss of generality, we usually set player 1 as a pseudo-player to the game, which can be regarded as an external control input. Using Algorithm 3.1, Algorithm 4.1, or Algorithm 5.1, we can get the desired expression for the given NEGs in the control form as

\[ y(t+1) = Lu u(t)y(t), \]  

where \( u(t) = x_1(t) \) and \( y(t) = x_2(t)x_3(t) \cdots x_n(t) \).

For system \( \dot{x}(t) = f(x(t)) \), we hope that \( x(t) \) would convergence the equilibrium point \( x = 0 \). For the NEG (6.1), we hope that \( y(t) \) would convergence the Nash equilibrium \( y_e \) under the input sequence \( \{u(t)\}_{0 < t < \infty} \).

\[ [13], [118] \] and \[ [122] \] adopted the same method to solve this problem. Firstly, we should determine the Nash equilibrium \( y_e \) for the given NEG.

Then, we rewrite (6.1) as

\[ y(\tau) = Lu u(\tau - 1)y(\tau - 1) = Lu u(\tau - 1)Lu u(\tau - 2)y(\tau - 2) = Lu(I_k \otimes Lu)u(\tau - 1)u(\tau - 2)y(\tau - 2) = Lu(I_k \otimes Lu) \cdots (I_{k-1} \otimes Lu)(\tau - 1) \cdots u(0)y(0) \]

\[ := Lu(\tau - 1)u(\tau - 2) \cdots u(0)y(0) = y_e. \]  

(6.2)

From (6.2), we design a proper input sequence \( \{u(t)\}_{0 < t < \infty} \) to reach the target.

Actually, when we use STP method to investigate the NEGs, we convert the given NEGs to the logical networks and apply the theory of logical networks to them. Therefore, readers can see details in \[ [10, 11, 22–31, 33, 34, 39, 81–85, 119–121, 133] \] for more theory of logical control networks.

7. Random Networked Evolutionary Games

In addition to the pure strategy evolutionary game, the investigation of mixed strategy evolutionary games is also a heated topic. This section introduces one kind of mixed strategy evolutionary games, i.e., \( n \)-person random evolutionary Boolean games (REBG) \[ [8] \].
An $n$-person REBG which has the fixed strategy updating rule can be represented as follows:

$$
\begin{align*}
    x_1(t+1) &= f_1(X(t), \omega_1(t,p_1), y(t)), \\
    x_2(t+1) &= f_2(X(t), \omega_2(t,p_2), y(t)), \\
    &\vdots \\
    x_n(t+1) &= f_n(X(t), \omega_n(t,p_n), y(t)); \\
    y(t) &= h(X(t)),
\end{align*}
$$

where $X(t) = (x_1(t), x_2(t), \cdots, x_n(t)) \in \mathcal{D}^n$ is the strategy of every player at time $t$, $\omega_i(t,p_i) \in \mathcal{D}$ with $\mathbb{P}\{\omega_i(t,p_i) = 1\} = p_i$ ($0 \leq p_i \leq 1$) denotes a random variable which is the possibility of making right choice for each player, $y(t) \in \mathcal{D}$ is the game result, and $f_i : \mathcal{D}^{n+2} \to \mathcal{D}$ for $i = 1, \cdots, n$ and $h : \mathcal{D}^n \to \mathcal{D}$ are Boolean functions which are determined by the strategy updating rule.

In the following, there is the definition of stability with probability one for the $n$-person REBG (7.1).

**Definition 7.1** ([8]). The $n$-person REBG (7.1) is said to be globally stable at $X^* \in \mathcal{D}^n$ with probability one, if there exists a positive integer $\tau$ such that

$$
\mathbb{P}\{X(t) = X^* | X(0) = X_0\} = 1
$$

holds for any initial strategy $X_0 \in \mathcal{D}^n$ and any integer $t \geq \tau$.

Our target is to analyze stability of the REBG. Therefore, [8] converted system (7.1) into the algebraic form of the $n$-person REBG (7.1) as follows:

$$
\begin{align*}
    x(t+1) &= L\omega(t)x(t), \\
    y(t) &= Hx(t).
\end{align*}
$$

Then, one can analyze the above equation (7.2) by the following result.

**Theorem 7.1** ([8]). The $n$-person REBG (7.1) is globally stable at $x_e = \delta_{2^n}$ with probability one, if and only if there exists an integer $1 \leq \tau \leq 2^n$ such that

$$
M^\tau = \delta_{2^n}[\alpha \cdots \alpha],
$$

where $x_e = \delta_{2^n}$ is the canonical vector form of $X^*$.

8. Conclusion

In this survey, we have reviewed a number of applications of STP method in the investigation of some kinds of NEGs, including general NEGs, NEGs with finite memories, NEGs defined on finite networks, control of NEGs, and random NEGs. By adopting STP method, constructive and precise analysis from the perspective of mathematics has been reflected in these applications. With the rapid development of science, we have faith in that STP method will have more wider applications in the theory of NEGs in the future.
Based on the literature review, some related topics for potential applications in NEGs are given as follows: (1) Many classical strategy updating rules, unconditional imitation updating rules for example, need to be algebraically formulated to complete the theory of NEGs; (2) Although we can analyze the given NEGs theoretically via STP method, how to reduce the computational complexity of calculations for STP method is still a problem in the future work; (3) The mix-valued strategy dynamics of the NEGs should be considered in the future work. We need to find more mathematical tools to deal with mix-valued dynamics; (4) Finally, when the evolutionary games are defined on the complex networks, how can we analyze the NEGs in the view of graph theory? (5) Many related results in control theory, such as [7, 16, 19, 20, 43, 44, 50, 51, 55, 56, 58–60, 72–74, 95, 100, 101, 104, 108, 109, 114, 115, 117, 132], could be extended to the theory of NEGs.

References


[34] H. Li, G. Zhao, M. Meng, and J. Feng, A survey on applications of semi-tensor product method in engineering, Science China Information Sciences, 2018, 61(1), 010202.


Survey on applications of semi-tensor product method...


[71] X. Lv, R. Rakkiyappan, and X. Li, $\mu$-stability criteria for nonlinear differential systems with additive leakage and transmission time-varying delays, Nonlinear Analysis, Modelling and Control, 2018, 23(3), 380–400.


