Chapter 14
Game Theory

This chapter considers the infinitely repeated game. We refer to [3, 2].

14.1 An Introduction to Game Theory

Game theory is a tool for studying a wide variety of human, and nature behaviors. It has a very long historical background. Gambling, playing cards or chess etc. may be considered as the origin of game theory. The famous Chinese story of the Tian Ji's strategy on horse racing is a typical example of Game theory.

Although some developments occurred before it, the foundation of modern game theory was built by the 1944 book *Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern. Neumann et al were mainly interested in cooperative games. But most important games are uncooperative (competitive) ones. This theory was developed extensively in the 1950s by many scholars. The most significant contribution was the Nash equilibrium, named after John Forbes Nash. It becomes a fundamental tool for uncooperative games.

Game theory was later explicitly applied to biology in the 1970s, although similar developments go back at least as far as the 1930s. Game theory has been widely recognized as an important tool in many fields. Eight game theorists, including Nash, have won the Nobel Memorial Prize in Economic Sciences, and John Maynard Smith was awarded the Crafoord Prize for his application of game theory to biology. Today, "game theory is a sort of umbrella or 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players (computers, animals, plants)" (Aumann 1987).

A game contains at least three basic factors:

(i) Player: Usually, a game contains more than one players. When there is only one player, the game becomes an optimization problem. Throughout this chapter we assume there are only finite players, denoted by $P_1, \dots, P_n$, where $1 < n < \infty$.

(ii) Actions: Each player in the game has some playing options, which are called the actions of this player. Denote the set of actions of P_i by A_i. when A_i is a finite set,

D. CHENG

we denote it by

$$A_i = \left\{a_i^1, \cdots, a_i^{k_i}\right\}, \quad i = 1, \cdots, n.$$

In some books the action is called the strategy. We reserve "strategy" for the way to select actions.

(iii) Payoff functions: The payoff function of player i, denoted by f_i, is the gain of player P_i. It depends on the actions of all players. That is,

$$c_i = c_i(x_1, \cdots, x_n), \quad x_j \in A_j, \quad j = 1, ..., n; \quad i = 1, ..., n.$$

Roughly speaking, Nash equilibrium is the solution for a game. We recall the definition of Nash equilibrium, which was given in Chapter 1:

Definition 14.1. A combined actions $(x_1^*, \cdots, x_n^*)$, $x_i \in A_i$, $i = 1, \cdots, n$, is called a Nash equilibrium, if

$$c_j(x_1^*, \cdots, x_n^*) \geq c_j(x_1^*, \cdots, x_j, \cdots, x_n^*), \quad \forall x_j \in A_j, \quad j = 1, \cdots, n. \tag{14.1}$$

Some examples, including the famous prisoner's dilemma, have been presented in Chapter 1. In the following we give some more examples. We use the following example to introduce the best reaction function.

Example 14.1 (Cournot model of duopoly). Let x and y be the quantities of a product by firms 1 and 2 respectively. Let

$$p(Q) = \begin{cases} a - Q, & Q < a \\ 0, & Q \geq a \end{cases}$$

be the market-clearing price, where $Q = x + y$. Assume the cost for producing unit product is a constant b. Following Cournot, suppose that the firms choose their quantities simultaneously.

Then the payoff functions are

$$\begin{aligned} c_1(x,y) &= [a - (x+y)]x - bx \\ c_2(x,y) &= [a - (x+y)]y - by. \end{aligned} \tag{14.2}$$

Now P_1 want to choose x to maximize c_1. To find such x, calculating

$$\frac{\partial c_i}{\partial x} = a - 2x - y - b := 0$$

yields

$$x = \frac{1}{2}(a - b - y). \tag{14.3}$$

(14.3) is called the best-response function of P_1, which shows for each action of P_2 what the best reaction of P_1 should be. Similarly, we have the best-response function of P_2 as

D. CHENG

$$y = \frac{1}{2}(a - b - x). \tag{14.4}$$

Then (assume $x, y < a - b$) the Nash equilibrium is the solution of best reaction functions

$$\begin{cases} x = \frac{1}{2}(a-b-y) \\ y = \frac{1}{2}(a-b-x). \end{cases}$$

That is,

$$\begin{cases} x = \frac{1}{3}(a-b) \\ y = \frac{1}{3}(a-b). \end{cases}$$

Remark 14.1. In fact, the method for finding Nash equilibrium proposed in Chapter 1 (Example 1.8) is also finding the (discrete) best reaction functions for each players. Then the common set(s) is(are) the solution(s) of the reaction functions.

We give another example.

Example 14.2. A Chinese ancient general Tian Ji was gambling with King Qi Wei Wang via three times horse racing. Both Tian and Qi have 3 horses, denoted by $T = \{t_1, t_2, t_3\}$ and $Q = \{q_1, q_2, q_3\}$ respectively. We know that their corresponding velocities satisfy

$$v_{t_3} > v_{q_3} > v_{t_2} > v_{q_2} > v_{t_1} > v_{q_1}.$$

That is, q_1 is the fastest one. The action sets of P_T and P_Q are the same as

$$A_T = A_Q = \{(123), (132), (213), (231), (312), (321)\},$$

where (123) means that P_T chooses the order of the racing horses as t_1, t_2, t_3, (or P_Q chooses this order), etc. Then we have the payoff bi-matrix in Table 14.1.

Table 14.1 Tian Ji Horse Racing

$P_T \backslash P_Q$	(123)	(132)	(213)	(231)	(312)	(321)
(123)	-3,3	-1,1	-1,1	1,-1	-1,1	-1,1
(132)	-1,1	-3,3	1,-1	-1,1	-1,1	-1,1
(213)	-1,1	-1,1	-3,3	-1,1	-1,1	1,-1
(231)	-1,1	-1,1	-1,1	-3,3	1,-1	-1,1
(312)	1,-1	-1,1	-1,1	-1,1	-3,3	-1,1
(123)	-1,1	1,-1	-1,1	-1,1	-1,1	-3,3

It is easy to see that there is no Nash equilibrium.

In Example 14.2, each player must guess the other's strategy. This happens in many other games. In any game in which each player would like to outguess the other(s), there is no Nash equilibrium because the solution to such a game necessarily involves uncertainty about what the players will do. Then we need to consider

D. CHENG

strategies with uncertainty. Such strategies are called the mixed strategy. To distinguish this kind of strategies with the previous strategies, we call the strategies without uncertainties pure strategy. We give a rigorous definition for mixed strategy.

Definition 14.2. Assume in a game a player has his action set as $S = \{s_\lambda \mid \lambda \in \Lambda\}$. A mixed strategy is a probability distribution

$$p_\lambda, \quad \lambda \in \Lambda.$$

Then $p_\lambda \geq 0$ and $\sum\limits_{\lambda \in \Lambda} p_\lambda = 1$. The player taking this strategy means he choose action s_λ with probability p_λ.

We give an example to describe it.

Example 14.3. (Matching Pennies) In this game, each player's action set is $S = \{Head, Tail\}$. The payoff bi-matrix is Then we have the payoff bi-matrix in Table 14.2.

Table 14.2 Matching Pennies

$P_1 \backslash P_2$	Head(H)	Tail(T)
Head(H)	-1,1	1,-1
Tail(T)	1,-1	-1,1

To accompany the payoffs in the bi-matrix, imagine that each player has a penny and must choose whether to display in with heads or tails facing up. If the tow pennies match then player 2 (P_2) wins player 1's (P_1) penney, otherwise, P_1 wins $P_2's$ penny.

It is obvious that there is no Nash equilibrium for pure strategies. Then we consider the mixed strategies. Assume P_1 plays H with probability p and T with probability $1-p$. Correspondingly, P_2 plays H with probability q and T with probability $1-q$. Then the expected payoffs for P_1 and P_2, denoted by E_1 and E_2 are respectively

$$\begin{aligned} E_1 &= -pq + p(1-q) + (1-p)q - (1-p)(1-q); \\ E_2 &= pq - p(1-q) - (1-p)q + (1-p)(1-q). \end{aligned} \tag{14.5}$$

A simple argument shows that the best strategy of P_1, responding to different q, is

$$p = \begin{cases} 0, & q > 0.5 \\ [0,1], & q = 0.5 \\ 1, & q < 0.5. \end{cases} \tag{14.6}$$

We call (14.6) the best-response correspondence. The reason we do not call it the best-response function is that (14.6) is not a function. So the best-response correspondence is a generalization of best-response function. Similarly, we have the best-response correspondence of P_2 is

D.CHENG

$$q = \begin{cases} 1, & p > 0.5 \\ [0,1], & p = 0.5 \\ 0, & p < 0.5. \end{cases} \tag{14.7}$$

Now the only common solution is

$$\begin{cases} p = 0.5 \\ q = 0.5, \end{cases}$$

which is the Nash equilibrium.

The following result is fundamental [3].

Theorem 14.1. *In a finite game G (has finite players, and each player P_i has finite set of actions), there exists at least one Nash equilibrium, possibly involving mixed strategies.*

Theorem 14.1 ensures that a finite game must have at least one Nash equilibrium, but many games have several Nash equilibria. In this case, it is hard to see what the right prediction is. For these, many refinement of Nash equilibria were proposed, such as Pareto-dominant equilibrium, coalition-proof equilibrium and so on [2]. However, in this chapter, we only consider pure strategies of finite games, so the Nash equilibrium may not exist. Then we may look for a "weaker" solution. Next, we give the definition of sub-Nash equilibrium.

Definition 14.3. 1. Given a combined actions $(x_1, x_2, \cdots, x_n)$. Then we can find a non-negative real number $\varepsilon^s \geq 0$, such that

$$\begin{array}{l} c_j(x_1, \cdots, x_j, \cdots, x_n) + \varepsilon^s \geq c_j(x_1, \cdots, x_{j-1}, x'_j, x_{j+1}, \cdots, x_n), \\ \forall x'_j \in A_j,\ j = 1, \cdots, n. \end{array} \tag{14.8}$$

The smallest $\varepsilon^s \geq 0$, satisfying (14.8), is called a tolerance of $(x_1, x_2, \cdots, x_n)$.

2. $(x_1, x_2, \cdots, x_n)$ is called a sub-Nash equilibrium if it has the smallest tolerance.

It is easy to see that a Nash equilibrium is a sub-Nash equilibrium with tolerance 0. Thus, sub-Nash equilibrium is a generalization of Nash equilibrium.

We give some examples to illustrate it.

Example 14.4. Consider a game with two players A and B. The payoff bi-matrix is

Table 14.3 Payoff bi-matrix

$A \backslash B$	1	2
1	2,0	0,2
2	1,2	2,1

D. CHENG

Table 14.4 Tolerances

$A\backslash B$	1	2
1	2	2
2	1	1

It is obvious that there is no Nash equilibrium. It is easy to calculate that the tolerances as Table 14.4.

Hence $(2,1)$ and $(2,2)$ are sub-Nash equilibriums with tolerance 1. However, It is very likely that A may not be satisfied with $(2,1)$ and B may not be satisfied with $(2,2)$.

Example 14.5. Recall the Tian Ji horse racing in Example 14.2. It is easy to obtain the tolerances are

Table 14.5 Tolerances of Tian Ji Horse Racing

$P_T\backslash P_Q$	(123)	(132)	(213)	(231)	(312)	(321)
(123)	4	2	2	4	2	2
(132)	2	4	4	2	2	2
(213)	2	2	4	2	2	4
(231)	2	2	2	4	4	2
(312)	4	2	2	2	4	2
(123)	2	4	2	2	2	4

We can see that the minimum tolerance is 2, and there are many sub-Nash equilibria.

14.2 Infinitely Repeated Games

In the games discussed in last section, the players choose their actions simultaneously and the games are played only once. We call this kind of games the static games. Other games are called the dynamic games which contains the following information:

- the set of players.
- the order of moves.
- the players' payoffs as the function of the actions that were made.
- what the actions the players choose when they move.
- what each player knows when he move.
- the probability distributions over any exogenous events.

Here, the point 6 may be hard to understand. For example, consider a game between two players. Assume that they have two payoff bi-matrices, and both of them

D. CHENG

know the probability of each bi-matrix to be used, but only player 1 knows which bi-matrix is used. Then game can be considered as a game that there is a player called "Nature" who will move firstly to choose a payoff bi-matrix randomly, and then player 1 and player 2 will move simultaneously according to what they know (player 1 knows the chosen bi-matrix but player 2 does not).

When it is player i's turn to move, denote H_i the set of possible historical information the player knows and A_i the set of possible actions the player can choose. A strategy s_i of player i is a mapping $s_i : H_i \to A_i$, that is, the way to choose his action according to his knowledge of historical information. Replacing "actions" by "strategies" in Definition 14.1 and 14.3, we get the concepts of Nash equilibrium and sub-Nash equilibrium for dynamic games.

Example 14.6. Recall the Cournot model in Example 14.1. We now suppose player 1 as the "leader", chooses his action first, and then player 2 chooses his own action after observing player 1's action. Thus, player 2's strategies are functions $s_2 : X \to Y$ where X and Y are feasible actions set of player 1 and player 2, while player 1's strategies are simply choosing x from $[0,a]$.

After observing player 1's action x, the best strategies of player 2 is to choose y to maximize $f_2(x,y)$. Thus the best strategy for player 2 is the function

$$s_2(x) = \frac{1}{2}(a-b-x).$$

Player 1 also knows player 2's best strategy, thus he only need to maximize $f_1(x,s_2(x))$. Hence we have the Nash equilibrium

$$\begin{cases} x = \frac{a-b}{2} \\ s_2(x) = \frac{1}{2}(a-b-x). \end{cases}$$

And the outcome of this equilibrium is

$$\begin{cases} x = \frac{a-c}{2} \\ y = \frac{a-c}{4}, \end{cases}$$

which is different from the Nash equilibrium of static Cournot model.

Although it is a very natural way to get the prediction as in the above example, it is not the unique Nash equilibrium (We leave it as an exercise to find another Nash equilibrium). To deal with this problem, a refinement of Nash equilibrium for dynamic games named subgame-perfect equilibrium was proposed. And the equilibrium found in the above example is the unique subgame-perfect equilibrium. Roughly speaking, a subgame-perfect equilibrium is Nash equilibrium for every subgame. Readers can refer to [2, 3] for details.

In the following, we only concern with infinitely repeated games, since it can be described as mix-valued logical (control) networks which can be converted into algebraic form using semi-tensor product as in Chapter 13.

D. CHENG

Definition 14.4. Consider the infinitely repeated game G_∞ of G.

1. A strategies profile is

$$s = (s_1, \cdots, s_n), \tag{14.9}$$

   where s_j is a sequence of logical functions of time, called the strategy of player j, precisely,

$$s_j = \left\{ x_i(0), s_j^t \middle| i = 1, \cdots, n, t = 1, 2, \cdots \right\}$$

   where s_j^t is a function of historical actions, precisely

$$x_j(t) = s_j^t(x_1(0), \cdots, x_n(0), \cdots, x_1(t-1), \cdots, x_n(t-1)).$$

   Denote by S_∞ the set of strategy profiles.
2. The players' payoffs as the function of all the actions that were made. In this chapter, we suppose the payoff functions are the averaged payoffs of all stages

$$J_j(s) = \overline{\lim_{T\to\infty}} \frac{1}{T} \sum_{t=1}^{T} c_j(x_1(t), \cdots, x_n(t)). \tag{14.10}$$

   Thus we can use the results of Section 13.5.2.

In this definition, we assume each player knows all the historical information when he moves. If they have finite memories, that is, each player only remember the information of previous finite steps when he moves, we have the following μ-memory strategies.

Definition 14.5. Consider the dynamic game G_∞ of G. A μ-memory strategy is a strategy, where the action $x_j(t+1)$ depends on the past μ historical actions, and s_j^t are time invariant, equals to f_j. Precisely, the strategy s_j is generated by

$$x_j(t+1) = f_j(x_1(t), \cdots, x_n(t), \cdots, x_1(t-\mu+1), \cdots, x_n(t-\mu+1)), \tag{14.11}$$

with initial values

$$x_j(t) = x_j^t, \quad t \le \mu - 1, \ j = 1, \cdots, n. \tag{14.12}$$

Equivalently, we can also denote the set of initial values as

$$X_0 = \left\{ x_1^0, \cdots, x_n^0, \cdots, x_1^{\mu-1}, \cdots, x_n^{\mu-1} \right\}.$$

We can see that if the actions sets $A_i, i = 1, \cdots, n$ are finite sets, (14.11) are essentially the dynamics of mix-valued logical control networks. Identifying

$$a_i^j \sim \delta_{k_i}^j, \quad i = 1, \cdots, n \quad j = 1, \cdots, k_i,$$

D. CHENG

we have $x_i \in \Delta_{k_i}$. Setting $k = \prod_{i=1}^{n} k_i$ and $x = \ltimes_{i=1}^{n} x_i \in \Delta_k$, (14.11) can be converted into their algebraic form

$$x_j(t+1) = L_j \ltimes_{i=0}^{\mu-1} x(t-i), \quad j = 1, \cdots, n, \tag{14.13}$$

where $L_j \in \mathcal{L}_{k_j \times k^\mu}$ is the structure matrix of f_j. Multiplying the equations in (14.13), we have the algebraic form of μ-memory strategies profile

$$x(t+1) = L \ltimes_{i=0}^{\mu-1} x(t-i) \tag{14.14}$$

with initial states X_0. Since (14.11), (14.13), and (14.14) are all equivalent, and it is easy to convert them from one form to another. We simply use $(L_1, \cdots, L_n; X_0)$ or (L, X_0) instead of the corresponding strategies profile $s = (s_1, \cdots, s_n)$.

14.3 Local Optimization of Strategies and Local Nash/Sub-Nash Equilibrium

In Section 13.5.2, we have already investigated the optimization of Boolean control networks. When the scale of network is larger, it is difficult to find out the optimal control. In this section, a distance of strategies will be proposed firstly [1]. Using this distance, local optimization of strategies and local Nash/sub-Nash equilibrium are investigated.

Recall Definition 13.9. The vector distance $D_v(A,B)$ of two Boolean matrices A and B are defined. If $A = (a_{ij}) \in \mathcal{B}_{m \times n}$, we denote

$$\|A\| = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}. \tag{14.15}$$

It is easy to check that $\|\cdot\|$ is a norm.

Next, we define a distance for two Boolean matrices of the same dimension.

Definition 14.6. Let $A = (a_{ij}), B = (b_{ij}) \in \mathcal{B}_{m \times n}$. Then the distance between A and B, denoted by $d(A,B)$, is defined as

$$d(A,B) := \frac{1}{2} \|D_v(A,B)\|. \tag{14.16}$$

The following result is an immediate consequence of the definition, we left it for exercise.

Theorem 14.2. *$(\mathcal{B}_{m \times n}, d)$ is a metric space. That is,*

(i)

$$d(A,B) = 0 \Leftrightarrow A = B, \quad \forall A, B \in \mathcal{B}_{m \times n};$$

D. CHENG

(ii)

$$d(A,B) = d(B,A), \quad \forall A,B \in \mathscr{B}_{m\times n};$$

(iii)

$$d(A,C) \le d(A,B) + d(B,C), \quad \forall A,B,C \in \mathscr{B}_{m\times n}.$$

Using this distance to the set of μ-memory strategy profiles, it follows that $(\mathscr{L}_{k\times k^\mu}, d)$ is a metric subspace of $(\mathscr{B}_{k\times k^\mu}, d)$. Next, the physical meaning of this d on the set of μ-memory strategy profiles is investigated. The following proposition is obvious.

Proposition 14.1. *Let* $A,B \in \mathscr{L}_{k\times k^\mu}$. *Then*

$$|\{ i \,|\, \mathrm{Col}_i(A) \ne \mathrm{Col}_i(B) \}| = d(A,B). \tag{14.17}$$

When $\mu = 1$, we have more clear description for the geometric meaning of this distance. The proof of the following proposition is left for exercise.

Proposition 14.2. *Assume* $\mu = 1$. *Then the distance* $d(L_1, L_2)$ *is the number of different edges between the state transfer graphs of the systems* $x(t+1) = L_1 x(t)$ *and* $x(t+1) = L_2 x(t)$.

As for $\mu > 1$ case, since $x(t+1)$ depends on μ historical strategy profiles, we define a path

$$x(t-\mu+1) \to x(t-\mu+2) \to \cdots \to x(t) \to x(t+1)$$

as a compounded edge. The strategy dynamic graph of a μ-memory strategy profile consists of all such compounded edges. Then the following corollary is clear.

Corollary 14.1. *Assume* $\mu > 1$. *Then the distance* $d(L_1, L_2)$ *between two* μ*-memory strategy profiles is the number of different compounded edges between the strategy dynamic graphs of* L_1 *and* L_2.

The results of Theorem 13.9 can be extended to μ-memory mix-valued logical control networks (see [4]). The infinite horizon optimization of μ-memory mix-valued logical control network

$$\begin{aligned}
x_1(t+1) = f_1(\, & x_1(t), \cdots, x_n(t), \cdots, x_1(t-\mu+1), \cdots, x_n(t-\mu+1),\\
& u_1(t), \cdots, u_m(t), \cdots, u_1(t-\mu+1), \cdots, u_m(t-\mu+1))\\
& \vdots\\
x_n(t+1) = f_n(\, & x_1(t), \cdots, x_n(t), \cdots, x_1(t-\mu+1), \cdots, x_n(t-\mu+1),\\
& u_1(t), \cdots, u_m(t), \cdots, u_1(t-\mu+1), \cdots, u_m(t-\mu+1))
\end{aligned} \tag{14.18}$$

where $x_i \in \mathscr{D}_{k_i}$, $u_i \in \mathscr{D}_{s_j}$ can be considered as an infinitely repeated game in which some players $(x_i, i = 1, 2, \cdots, n)$ have fixed their strategies as f_i with initial actions

D. CHENG

$x_i^j, j=0,\cdots,\mu-1$, and other players ($u_j, j=1,\cdots,m$) can choose their strategies freely and the have a common payoff function

$$J=\overline{\lim_{T\to\infty}}\frac{1}{T}\sum_{t=1}^{T}c(x_i(t),u_j(t)), \quad ,i=1,\cdots,n, \quad ,j=1,\cdots,m. \tag{14.19}$$

Setting $x(t)=\ltimes_{i=1}^n x_i(t), u(t)=\ltimes_{j=1}^m u_j(t), k=\prod_{i=1}^n k_i, s=\prod_{j=1} ms_j$, the algebraic form of (14.18) is

$$x(t+1)=Lu(t-\mu+1)\cdots u(t)x(t-\mu+1)\cdots x(t). \tag{14.20}$$

The optimal control has the form of

$$u(t+1)=Gu(t-\mu+1)\cdots u(t)x(t-\mu+1)\cdots x(t). \tag{14.21}$$

Multiplying both sides of (14.20) and (14.21) together and setting $w(t)=u(t)x(t)$, yields

$$w(t+1)=\Psi(G)w(t). \tag{14.22}$$

We leave the expression of $\Psi(G)$ as an exercise.

For every G, we can calculate $\Psi(G)$, and then find the cycles of (14.18) for every initial $u(0),\cdots,u(\mu-1)$. Thus, by comparing the criterions we can find the optimal control. But in general, searching all $G\in\mathscr{L}_{s\times sk}$ to find an optimal solution is unrealistic because of the computation complexity. Using the distance of logical matrices, at each step, we can look for only a local optimal solution. That is, look for optimal solution (G,U_0) ($U_0:=\ltimes_{t=0}^{\mu-1}u(t)$) over a neighborhood

$$B_\varepsilon(G^0,U_0^0)=\left\{(G,U_0)\in\mathscr{L}_{s\times sk}\times\Delta_{s^\mu}\,|\,d(G,G^0)\le\varepsilon, d(U_0,U_0^0)\le\varepsilon\right\}.$$

Set the default $\varepsilon=1$.

We give an example to illustrate this.

Example 14.7. Consider a Boolean network

$$x(t+1)=Lu(t)x(t),$$

where

$$L=\delta_2[1\ 2\ 2\ 1].$$

Assume

$$c(x(t),u(t))=u'(t)\begin{bmatrix}1&2\\3&4\end{bmatrix}x(t)$$

and $x^0=\delta_2^2$. Choose $G^0=\delta_2[1\ 2\ 2\ 1]$, $u_0^0=\delta_2^2$, we can get that in step 1

$$\mathscr{G}^1=\{(\delta_2[1\ 2\ 2\ 2],\delta_2^1),(\delta_2[1\ 2\ 2\ 2],\delta_2^2)\}.$$

D. CHENG

is the set of optimal strategies in $B_1(G_0;u_0^0)$. We choose $G^1 = \delta_2[1\ 2\ 2\ 2]$, $u_1^0 = \delta_2^1$, then

$$\begin{aligned}\mathscr{G}^2 = \{&(\delta_2[1\ 2\ 2\ 2], \delta_2^1), (\delta_2[1\ 2\ 2\ 2], \delta_2^2),\\ &(\delta_2[2\ 2\ 2\ 2], \delta_2^1), (\delta_2[2\ 2\ 2\ 2], \delta_2^2)\}.\end{aligned}$$

is the set of optimal strategies in $B_1(G^1;u_0^1)$ Thus, $(\delta_2[1\ 2\ 2\ 2], \delta_2^1)$ and $(\delta_2[1\ 2\ 2\ 2], \delta_2^2)$ are local optimal controls. We can check that they are also optimal controls.

Next, we consider the local Nash/sub-Nash equilibrium.

Definition 14.7. For infinitely repeated game G_∞, a μ-memory strategy profile $s = (L;X_0)$ is called a local Nash/sub-Nash equilibrium on $B_\varepsilon(L,X_0)$, if it is a Nash/sub-Nash equilibrium with respect to its ε-neighborhood.

Since there may exits too many Nash/sub-Nash equilibria, in this section we are interested in the initial value independent strategies.We define the common Nash equilibrium.

Definition 14.8. $L^* = ({L_1}^*, {L_2}^*, \cdots, {L_n}^*)$ is called a common Nash equilibrium, if it, combining with any set of initial values, is a Nash equilibrium. Precisely, $\forall\, j = 1, \cdots, n$,

$$\begin{array}{l} J_j({L_1}^*, \cdots, {L_j}^*, \cdots, {L_n}^*;X_0) \geq J_j({L_1}^*, \cdots, L_j', \cdots, {L_n}^*;X_0) \\ \forall L_j' \in \mathscr{L}_{k_j \times k^\mu}, \forall X_0 \in \prod_{i=1}^n \mathscr{D}_{k_i}^\mu. \end{array} \tag{14.23}$$

It is easy to check that a common Nash equilibrium is a subgame-perfect equilibrium. If the common Nash equilibrium does not exist, we may look for a initial-independent sub-Nash solution. To make it precise, we give the following definition.

Definition 14.9. 1. For $L = (L_1, L_2, \cdots, L_n)$, we can find a non-negative real number $\varepsilon \geq 0$, such that, $\forall\, j = 1, \cdots, n$,

$$\begin{array}{l} J_j(L_1, \cdots, L_j, \cdots, L_n;X_0) + \varepsilon \geq J_j(L_1, \cdots, L_j', \cdots, L_n;X_0), \\ \forall L_j' \in \mathscr{L}_{k_j \times k^\mu},\ \forall X_0 \in \prod_{i=1}^n \mathscr{D}_{k_i}^\mu. \end{array} \tag{14.24}$$

The smallest $\varepsilon^s \geq 0$, satisfying (14.24), is called a tolerance of L.
2. L is called a initial-independent sub-Nash equilibrium if it has the smallest tolerance.

Similar to the local optimization, in each step, we can find the strategy profile with the minimum tolerance in the neighborhood. We describe the algorithm as following

Algorithm 2. • Step 0. Choose an initial strategy profile L^0, set $\mathscr{H} = \{L^0\}$.
• $\cdots$

- Step p. On the neighbor

$$B_\varepsilon(L^{p-1}) = \{L | d(L, L^{p-1}) \leq \varepsilon\}$$

search sub-Nash equilibrium(s), denoted as

$$\mathscr{L}^p = \{L^{p_1}, L^{p_2}, \cdots, L^{p_{k_p}}\}.$$

  – If $L^{p-1} \in \mathscr{L}^p$, choose L^{p-1} as a local sub-Nash equilibrium (the solution) and stop.
  – Else, if $\mathscr{L}^p \cap \mathscr{H}^c = \emptyset$,
    · If $p = 1$, no local sub-Nash equilibrium is found (the algorithm fails) and stop.
    · Else,go back to Step p-1 to choose another L^{p-1} if possible.
  – Else, choose $L^p \in \mathscr{L}_p \cap \mathscr{H}^c$, and add L^p to $\mathscr{H}$.
- $\cdots$

Example 14.8. Consider the infinitely repeated game of prisoners' dilemma, the payoff bi-matrix is

Table 14.6 Payoff bi-matrix

$P_1 \backslash P_2$	1	2
1	3,3	0,5
2	5,0	1,1

Choose $L^0 = \delta_4[1\ 1\ 1\ 3]$, using Algorithm 2, we have

$$\begin{array}{l} L^1 = \delta_4[1\ 4\ 1\ 3],\ L^7 = \delta_4[4\ 4\ 4\ 2] \\ L^2 = \delta_4[1\ 3\ 1\ 3],\ L^8 = \delta_4[1\ 4\ 4\ 2] \\ L^3 = \delta_4[1\ 3\ 4\ 3],\ L^9 = \delta_4[2\ 4\ 4\ 2] \\ L^4 = \delta_4[3\ 3\ 4\ 3],\ L^{10} = \delta_4[2\ 4\ 4\ 4] \\ L^5 = \delta_4[3\ 3\ 4\ 2],\ L^{11} = \delta_4[1\ 4\ 4\ 4]. \\ L^6 = \delta_4[3\ 4\ 4\ 2], \end{array}$$

The algorithm terminates at $k = 11$, L^{11} is a locally Nash equilibrium, which is also a 1-memory Nash equilibrium.

The algorithm can not always find a Nash equilibrium, for example, let $L^0 = \delta_4[2\ 1\ 2\ 3]$, then we can find a locally Nash equilibrium $L = \delta_4[1\ 4\ 2\ 1]$ which is not Nash equilibrium.

Remark 14.2. Since the tolerance of a strategies profile depends on the neighborhood, thus the value of tolerance may not be degressive. Thus, it is possible that the chosen optimal strategy profiles $s_0, s_1, \cdots$, form a cycle, then the algorithm fails. Otherwise, a local sub-Nash equilibrium can be obtained. Theoretically, we have no

D. CHENG

reason to claim that the algorithm will never fail. But in numerical computations, we did not have experience of failing.

Exercise 11

1. Find another Nash equilibrium for Example 14.6.
2. Prove Theorem 14.2.
3. Prove Proposition 14.2.
4. Give the expression of $\Psi(G)$ in (14.22)

References

1. Cheng, D., Zhao, Y., Mu, Y.: Strategy optimization with its application to dynamic games. In: Proc. IEEE CDC'2010, pp. 5822–5827 (2010). DOI 10.1109/CDC.2010.5717060
2. Fudenberg, D., Tirole, J.: Game Theory. MIT Press, Cambridge (1991)
3. Gibbons, R.: A Primer in Game Theory. Prentice Hall (1992)
4. Zhao, Y., Li, Z., Cheng, D.: Optimal control of logical control networks. Preprint (2010)

D. CHENG