Some Results on Evolutionary Prisoner's Dilemma Games

Yunshyong Chow

Institute of Mathematics, Academia Sinica
Taipei,Taiwan
Email: chow@math.sinica.edu.tw

10th Workshop Markov Processes and Related Topics
Xidian and BNU
Xian, China
August 14-18, 2014

Prisoner's Dilemma game

Two prisoners are in custody.

	Defect		Cooperation
Defect	1. 6 yrs		1. 3 months
	2. 6 yrs	$>$	2. 10 yrs
Cooperation	1. 10 yrs		1. 1 yr
	2. 3 months	$>$	2. 1 yr

- Defect=Confess. Cooperation=Not Confess
- {Nash Equilibruim} = {(Defect, Defect)}
- **Q** : Any way out of the dilemma?
- 2, n or ∞ players. Repeated play.
  Local or global interaction.
- Strategy updating scheme.

Prisoner's Dilemma Game 2

More generally, payoff paremeters satisfy $b > d > a > c$

	Defect		Cooperation
Defect	1. a		1. b
	2. a	$>$	2. c
Cooperation	1. c		1. d
	2. b	$>$	2. d

Set up: n players sitting around a circle.

- Nearest neighborhood structure $N(i) = \{i - 1, i + 1\}$
- Strategy updating scheme.
  Imitation-best-player. Imitation-best-strategy.
  Best response.

Best response with local interaction

Strategy updating from time t to $t+1$:

- Assume $\vec{s}(t) = (s_1(t), s_2(t), \cdots, s_n(t)) \in \{D, C\}^n$
- $z_i(\vec{s}(t))$ = total payoff that player i will get after playing once with each of his neighbors.
- player i will adopt a strategy that maximum his payoff.
- $z_i(\cdots C, C, C \cdots) = 2d < 2b = z_i(\cdots C, D, C \cdots)$
- Let $r_i(\vec{s}(t))$ be the rational choice for player i under Q_0.
- Then $r_i(\vec{s}(t)) = D$. Hence, $\vec{r}(\vec{s}) = \vec{D}$ under Q_0.
- No way to get out of $\vec{D}$ even with mutation!
- Mutation means $s_i(t+1) = C$ with probability ε^1 indep.
- Sequential updating or parallel updating.

New strategy updating

- For $\cdots C\ C\ C \cdots \longrightarrow \cdots C\ D\ C \cdots$,
  Δ total payoff for these 3 players $= 2b + 2c - 4d$
- change of bond energy
- social consciousness or warrantor or family interest.
- change of state:
  $\vec{s}(t)\|t_i = (s_1(t), s_2(t), \cdots, s_{i-1}(t), t_i, s_{i+1}(t), \cdots, s_n(t))$
- $p_\varepsilon(\vec{s}(t), \vec{s}(t)\|t_i) \approx \varepsilon^{(-\Delta \text{ total payoff})^+} = \varepsilon^{(-U(\vec{s}(t)\|t_i)+U(\vec{s}(t)))^+}$
- Mutation cost is no longer 1.
- Hence, $U(\vec{s}) = -\sum_{i=1}^{n} z_i(\vec{s})$ is a potential function.
- Let P_ε be the Markov transition probability matrix.
- Let $\vec{\mu}_\varepsilon = \vec{\mu}_\varepsilon P_\varepsilon$ be the ergodic distribution.
- Let $\vec{\mu}_* = \lim_{\varepsilon \to 0} \vec{\mu}_\varepsilon$
- Goal: $S_* =$ support of $\vec{\mu}_*$. Long run Equilibria

sequential updating without mutation cost = 1

In the potential case, $S_* = \{\vec{s} : U(\vec{s}) = \min U\}$

Theorem 1

(i) If $b + c < 2d$, then $S_* = \{\vec{C}\}$.
(ii) If $b + c > 2d$, then

$$S_* = \{\vec{A} = CDCD \cdots CD, \vec{B} = DCDC \cdots DC\} \text{ if } n \text{ is even},$$

$$= \{\vec{A}_i = CDCD \cdots CD \overset{i}{C} : 1 \le i \le n\} \text{ if } n \text{ is odd}.$$

(iii) If $b + c = 2d$, then $S_* = \{\vec{s} : \text{all } d_i = 1\} \cup \{\vec{C}\}$.
Here d_i means the length of the i-th D-string.

Then the average payoff of each player can be computed.

parallel updating without mutation cost $= 1$

$p_\varepsilon(\vec{s},\vec{t}) \approx \varepsilon^{\sum(-\Delta \text{ total payoff at player } i)^+} \overset{def}{=} \varepsilon^{v(\vec{s},\vec{t})}$.
Here $v(\vec{s},\vec{t})$ means the cost jumping from $\vec{s}$ to $\vec{t}$

Theorem 2

(i) If $b+c<2d$, then $S_* = \{\vec{C}\}$.
(ii) If $b+c>2d$, then

$$S_* = \{\vec{A} = CDCD\cdots CD, \vec{B} = DCDC\cdots DC\} \text{ if } n \text{ is even,}$$

$$= \{\vec{A}_i = CDCD\cdots CD\overset{i}{C} : 1 \le i \le n\} \text{ if } n \text{ is odd.}$$

(iii) If $b+c=2d$, then $S_* = \{C,D\}^n$

Sequential updating and parallel updating could be different.

updating rule for player i

$$\overset{b+c+2a}{\cdots CDD\cdots} \longrightarrow \overset{2d+b+c}{\cdots CCD\cdots} \circlearrowleft$$

$$\circlearrowleft \overset{4d}{\cdots CCC\cdots} \longleftarrow \overset{2b+2c}{\cdots CDC\cdots} \qquad \text{if } 2d > b+c$$

$$\circlearrowleft \cdots CCC\cdots \longleftrightarrow \cdots CDC\cdots \circlearrowleft \qquad \text{if } 2d = b+c$$

$$\cdots CCC\cdots \longrightarrow \cdots CDC\cdots \circlearrowleft \qquad \text{if } 2d < b+c$$

$$\circlearrowleft \overset{4a}{\cdots DDD\cdots} \longleftarrow \overset{2b+2c}{\cdots DCD\cdots} \qquad \text{if } 2a > b+c$$

$$\circlearrowleft \cdots DDD\cdots \longleftrightarrow \cdots DCD\cdots \circlearrowleft \qquad \text{if } 2a = b+c$$

$$\cdots DDD\cdots \longrightarrow \cdots DCD\cdots \circlearrowleft \qquad \text{if } 2a < b+c$$

Freidlin-Wentzel Method

General procedure:

Step 1. Find $S_0 = \{\text{ergodic states}\}$ under Q_0

Step 2. $\forall \vec{s} \in S_0$,

find $C(\vec{s}) =$ minimum cost of all spanning trees rooted at $\vec{s}$

Or equivalently, find how states in S_0 reach each other.

Step 3. Then $S_* = \{\vec{s} \in S_0 : C(\vec{s}) = \min C\}$.

Moreover, the order in ε of the waiting time to hit S_* can be obtained

$$\cdots CDD \cdots \overset{2(d-a)}{\longleftarrow} \cdots CDD \cdots \underset{0}{\overset{2(d-a)}{\leftrightarrows}} \cdots CCD \cdots \circlearrowleft$$

Hence, any C-string with length ≥ 2 can hold under Q_0

Case(i) $b+c<2d$ with $2a>b+c$.

- Any C-string with length ≥ 2 can can grow to $\vec{C}$ under Q_0.
  $\circlearrowleft *CCC* \longleftarrow *CDC*$ Otherwise, cost $=4d-2b-2c$.
- $\circlearrowleft *DDD* \longleftarrow *DCD*$ Otherwise, cost $=4a-2b-2c$.
- $\vec{A}=CDCD\cdots CD \longleftrightarrow DCDC\cdots DC=\vec{B}$ if n is even.
- $S_0=\{\vec{C},\vec{D},\vec{A},\vec{B}\}$ if n is even and $S_0=\{\vec{C},\vec{D}\}$ if n is odd.
- $\vec{D} \xrightarrow{4a-2b-2c} \cdot\cdot DDCDD\cdot\cdot \rightarrow \cdot\cdot DCDCD\cdot\cdot \rightarrow \vec{A}$ if n even
  $\rightarrow \vec{C}$ if n odd
- $\vec{C} \xrightarrow{2(4d-2b-2c)}$ at least 2 mutations from C to D to get out.
- Easy to escape from $\vec{A},\vec{B}$ by making 1 mutation from $*DCD*$ to $*CCC*$ at cost $(4a-2b-2c)$.
- $S_*=\{\vec{C}\}$

Case(i) $b + c < 2d$ with $2a \leq b + c$.

- Any C-string with length ≥ 2 can can grow to $\vec{C}$ under Q_0.
- $\cdot\cdot DDD \cdot\cdot \rightarrow \cdot\cdot DCD \cdot\cdot$ ⟲ allowed.
  Hence, Any C-string can can grow to $\vec{C}$ under Q_0.
- $\vec{D} \longrightarrow \vec{C}$. In fact, any $\vec{S} \longrightarrow \vec{C}$ in one step under Q_0.
- $S_0 = \{\vec{C}\}$ and then $S_* = \{\vec{C}\}$.

Case(ii) $b + c = 2d > 2a$

- Any C-string with length ≥ 2 can can grow to $\vec{C}$ under Q_0.
- $\circlearrowright \cdot\cdot CCC \cdot\cdot \leftrightarrow \cdot\cdot CDC \cdot\cdot \circlearrowleft$ with equal probability.
  Hence, $\vec{C}$ can reach $\vec{S}$ in one step under Q_0.
- $\cdots CDD \cdots \longrightarrow \cdots CCD \cdots \circlearrowleft$
- $\cdot\cdot DDD \cdot\cdot \longrightarrow \cdot\cdot DCD \cdot\cdot \circlearrowleft$
  Hence, any $\vec{S}$ can reach $\vec{C}$ in one step under Q_0.
- $S_0 = \{C, D\}^n = S_*$

Note that $S_* = \{\vec{s} : \text{all } d_i = 1\}$ in sequential updating.

Case(iii) $b + c > 2d$

$\cdots CDD \cdots \longrightarrow \cdots CCD \cdots \circlearrowleft$
$\cdots CCC \cdots \longrightarrow \cdots CDC \cdots \circlearrowleft$
$\cdots DDD \cdots \longrightarrow \cdots DCD \cdots \circlearrowleft$

S_0 is complicated.

- Any D-string with length $\geq 2 \longrightarrow$ C-string. Hence, $\vec{D} \leftrightarrow \vec{C}$.
- $\vec{A}, \vec{B}$ (if n even) are absorbing states under Q_0
- $\vec{A}_i$ (if n odd) are absorbing states under Q_0
- String CDC remains unchanged under under Q_0.
- String $CDC(C)DC$ remains unchanged under under Q_0.
- $\cdots DCCCD \cdots \longrightarrow \cdots * CDC * \cdots$
  Hence, no C-string with length 3 can exist in $\vec{s} \in S_0$.

Case(iii) $b+c>2d$ continued

- $\cdots C\overbrace{D\cdots D}^{\geq 2}\overbrace{\underline{C}C\cdots CC}^{\geq 4}D*\cdots \to \cdot\cdot CC\cdots C\underline{C}D\cdots DC*\cdot\cdot$
- $\cdot\cdot(C)C\cdot\cdot$ will be eliminated unless it is $\cdot\cdot CD(C)CDC\cdot\cdot$

$$\cdots C\overbrace{D\cdots D}^{\geq 2}(C)CDD\cdots \to \cdots CC\cdots C(C)CCC\cdots$$
$$\to \cdots * D\cdots D(D)DD*\cdots$$

$S_0 = S_a \cup S_f$, where
$S_a = \{\vec{s} : c_i = 1 \text{ or } 2, \text{ and } d_i = 1\}$
$S_f = \{\vec{s} \leftrightarrow \vec{t} :$ all $c_i \neq 3$ and some $c_i \geq 4$ in $\vec{s}$ on in $\vec{t}$,
$d_{i-1} = d_i = 1$ in case $c_i < 3\}$.
Note both $\vec{D}, \vec{C} \in S_f$ as $\vec{D} \leftrightarrow \vec{C}$.

Step 2. How states in S_0 communicate with each other

- $\delta = \min(2d-2a, b+c-2d)$ = min. cost to escape.
- Any mixed state in S_f can reach $\vec{A}$ or $\vec{B}$ at cost δ.
- $\vec{s} = \cdot\cdot C \overbrace{D\cdot\cdot D\underline{D}}^{\geq 2}\overbrace{CC\underset{\bullet}{C}\cdots CC}^{\geq 4} D\cdot\cdot \leftrightarrow C \overbrace{C\cdots C}^{\geq 2}\overbrace{CDD\cdots D}^{\geq 4} *$
- $\vec{s} \xrightarrow{2d-2a} C \overbrace{C\cdots C\underline{D}}^{\geq 2}\overbrace{CD\cdots DC}^{\geq 4} *$
- $\vec{s} \xrightarrow{b+c-2d} C \overbrace{C\cdots C}^{\geq 2}\overbrace{CD\underset{\bullet}{C}D\cdots D}^{\geq 4} *$
- $\vec{C}$ can reach out at cost $b+c-2d$.
- $\vec{C} \leftrightarrow \vec{D} \xrightarrow{b+c-2d} CDCC\underset{\bullet}{C}\cdot\cdot C = \vec{t} \leftrightarrow CDC\underline{D}D\cdot\cdot D = \vec{s}$

$$\vec{C} \xleftarrow{b+c-2d} \vec{s} \xrightarrow{2d-2a} CDC\underline{D}C\cdot\cdot C \text{ for } n \geq 5$$

$$\vec{t} \xrightarrow{b+c-2d} CDCD\underset{\bullet}{C}DD\cdot\cdot D$$

States in $S_a \setminus \{\vec{A}, \vec{B}\}$ for n even: $c_i = 2, c_{i+1} = 1$

String CC can be moved to either side at cost δ.

- $\vec{u} = *CDCC\,\underset{\bullet}{D}\,CDC* \xrightarrow{b+c-2d} *CDCC\,\underset{\bullet}{C}\,CDC* \leftrightarrow \vec{t}$

  $\vec{t} \leftrightarrow *CDCC\,\underset{\bullet}{C}\,CDC* \xrightarrow{b+c-2d} *CDC\,\underset{*}{D}\,CCDC*$

- $\vec{u} \xrightarrow{2d-2a} \vec{t} = *CDC\,\underset{\bullet}{D}\,DCDC* \xrightarrow{2d-2a} *CDC\,\underset{\bullet}{D}\,CCDC*$

Hence, all strings CC can be moved together at cost δ.

States in $S_a \setminus \{\vec{A}, \vec{B}\}$ for n even: $c_i = c_{i+1} = 2$

These states can reach each other and $\{\vec{A}, \vec{B}\}$ at cost δ.

- $\vec{u} = *CDC\ \underset{*}{C}\underset{\bullet}{D}\ CCDC* \xrightarrow{b+c-2d} *CDCC\ \underset{\bullet}{C}\ CCDC* = \vec{s}$

  $*CDCD\ \underset{\bullet}{D}\ DCDC* \leftrightarrow \vec{s} \xrightarrow{b+c-2d} *CDCD\ \underset{\bullet}{C}\ DCDC*$

  $$\xrightarrow{b+c-2d} \cdots \xrightarrow{b+c-2d} \cdots \longrightarrow \vec{A}$$

- $\vec{u} \xrightarrow{2d-2a} *CDC\ \underset{*}{D}\ DCCDC* \longrightarrow *CDC\ \underset{*}{C}\ CCCDC* \leftrightarrow \vec{w}$

  $\vec{w} = *CDC\ \underset{\bullet}{D}\ DDCDC* \xrightarrow{2d-2a} *CDC\ \underset{\bullet}{D}\ CCCDC*$

  $$\longrightarrow *CDC\ \underset{\bullet}{D}\ CDCDC*$$

$\{\vec{A}, \vec{B}\}$ for n even

Min. cost to reach out is $b + c - 2a > b + c - 2d \geq \delta$.

- $\vec{A}$ has no CCD or CDD string.
- $\vec{A} \xrightarrow{b+c-2d} *CDC\ \underset{\bullet}{C}\ CDC* \longrightarrow \vec{A}$ no escape!
- $\vec{A} \xrightarrow{b+c-2a} *CDCD\ \underset{\bullet}{D}\ DCDC* \leftrightarrow *CDCC\ \underset{\bullet}{C}\ CCDC*$ a mixed state in S_f.
- Hence, $S_* = \{\vec{A}, \vec{B}\}$.

The case n that is odd can be dealt with similarly.

$S_* = \{\vec{A}_i = CDCD \cdots CD\ \overset{i}{C} : 1 \leq i \leq n\}$.
The proof is more complicated.

Ellison's Radius and Co-radius Theorem is useful.
The basin of $\{\vec{A}, \vec{B}\}$ is deeper.

Parallel updating with mutation cost $= 1$

- Previously, $p_\varepsilon(\vec{s}, \vec{t}) \approx \varepsilon^{\sum(-\Delta \text{ total payoff at player } i)^+}$.
- In literature, $p_\varepsilon(\vec{s}, \vec{t}) = (1 - \varepsilon)^{n - dist(\vec{s},\vec{t})} \varepsilon^{dist(\vec{s},\vec{t})}$.
  Here $dist(\vec{s}, \vec{t}) = \#\{1 \le i \le n : t_i \ne r_i(\vec{s})\}$
  and $r_i(\vec{s})$ is the rational choice for player i at state $\vec{s}$.

Theorem 3

(i) If $b + c < 2d$, then $S_* = \{\vec{C}\}$.
(ii) If $b + c > 2d$, then $S_* = S_a \cup S_f$ defined in Theorem 2 (ii).
(iii) If $b + c = 2d$, then $S_* = \{C, D\}^n$.

- For (i), it takes 2 mutations for $\vec{C}$ to reach out.
- For (ii), 1 mutation for any state in S_0 to reach each other.
  Too cheap to make a mistake.
  Confucius says think thrice...
- For (iii), all states can reach each other under Q_0.

Sequential updating with mutation cost $= 1$

Theorem 4

(i) If $b + c < 2d$, then $S_* = \{\vec{C}\}$.
(ii) If $b + c > 2d$, then $S_* = S_a$ defined in Theorem 2 (ii).
(iii) If $b + c = 2d$, then $S_* = \{\vec{C}\} \cup \{\vec{s} : \text{ all } d_i = 1\}$.

- $S_a = \{\vec{s} : \text{ all } d_i = 1, c_i = 1 \text{ or } 2\}$.
- Results in (ii) and (iii) are better than those in Theorem 3.
- Theorem 3 (ii). If $b + c > 2d$, then $S_* = S_a \cup S_f$ defined in Theorem 2 (ii).
- Theorem3 (iii). If $b + c = 2d$, then $S_* = \{C, D\}^n$.
- Being a potential function in Theorem 1, we get $S_* = \{\vec{s} : U(\vec{s}) = \min U\}$ without doing anything.
- Now we have to sweat a little.

References

- Ellison, Learning, local interaction, and coordination, *Econometrica* **61** (1993), 1047-1071.
- Ellison, Basins of attraction, long-run stochastic stability, and the speed of step-by-step evolution. *Review of Economic Studies* **67** (2000), 17-45.
- Eshel, Samuelson, and Shaked, Altruists, egoists, and hooligans in a local interaction model, *American Economic Review* **88** (1998), 157-179.
- Freidlin and Wentzell, Random Perturbations of Dynamical Systems. Springer-Verlag, New York, 1984.
- Chiang and Chow, A limit theorem for a class of inhomogeneous Markov processes, *Ann. Probab.* **17** (1989), 1483-1502.
- Chiang and Chow, Optimal Ventcel graphs, minimal cost spanning trees and asymptotic probabilities, *Applicable Analysis and Discrete Mathematics,* **1** (2007), 265-275.