The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Reverse Engineering of Spatiotemporal Patterns in Spatial Prisoner's Dilemma

Takuya Ueda and Yoshiteru Ishida
Department of Knowledge-Based Information Engineering
Toyohashi University of Technology
Toyohashi, Aichi, Japan 441-8580

***Abstract*:** Many complex patterns are produced by spatial prisoner's dilemma such as spatial games [Nowak & May 92] and spatial strategies [Ishida & Mori 04]. We have studied an inverse problem of identifying a game by estimating parameters in the payoff of the game from spatiotemporal patterns.

***Keywords*:** spatial prisoner's dilemma, reverse engineering, cellular automata, fractal

1 Introduction

Many spatiotemporal patterns can be found in nature such as ice crystal, a coastal railroad and shells. A spatial game [1] can produce spatiotemporal patterns. This is an extension of well known Prisoner's Dilemma (PD) in a space dimension.

While a game in general is played by two players, a spatial game is played by multiple players. Each player is placed at a cell in a lattice and interacts with the neighbor players. Fig. 1 shows spatiotemporal patterns generated by one dimensional spatial prisoner's dilemma.

This paper deals with an inverse problem of estimating parameters of spatial game from spatiotemporal patterns (generated by the spatial game). When the space is k-dimensional lattice (k is a natural number), a spatial game can be identified as cellular automata. A spatial game process can be estimated as a transition rule of cellular automata from spatial patterns by the rule identification algorithm [3]. Hence strategies of each player can be inferred by analyzing the spatiotemporal patterns.

2 Spatial Prisoner's Dilemma and Spatiotemporal Patterns

2.1 SPD

Iterated prisoner's dilemma (IPD) iterates prisoner's dilemma N steps. Spatial prisoner's dilemma (SPD) [1, 2] is an extended IPD in spatial axis; i.e. The SPD is played by multiple players placed in the neighborhood cells in a lattice space. Here we restrict ourselves to the case of one dimensional square lattice. In our SPD, a player has a strategy and decides an action at each step based on its strategy. Each player plays with neighbor players (including itself). Score of each play will be calculated from payoff matrix (Table 1) for neighbor players within a radius r. For example, when $r = 1$ and the space is one dimensional square lattice, the number of neighborhood players is three. After every interaction, the strategy of the highest score will be copied to its neighbor players.

Table 1: Payoff matrix. ($T > R > P > S$, $2R > T + S$ and $1.0 < b < 2.0$ [2])

		Adversary	
		C	D
Player	C	R(1)	S(0)
	D	T(b)	P(0)

All-D and All-C are the base strategies [1]. All-D (All-C) chooses D (C) every time. In addition to All-D (All-C), we will study spatial strategies such as k-D (k-C) [2]. k-D (k-C) strategy chooses D (C) when the number of players with D (C) in neighborhood exceeds k. We will also use the strategy code. The 2-D strategy, for example, can be expressed by a strategy code CCD (Table 2).

Table 2: Strategy Codes when $r = 1$ and the space is one dimensional square lattice

Number of D in neighborhood	0	1	2
Action	C	C	D

2.2 Spatiotemporal patterns generated by SPD

Spatiotemporal patterns found in SPD can involve many interesting ones including fractal patterns. In fact, Fig. 1 shows spatiotemporal patterns with the parameter b 1.5 in Table 1.

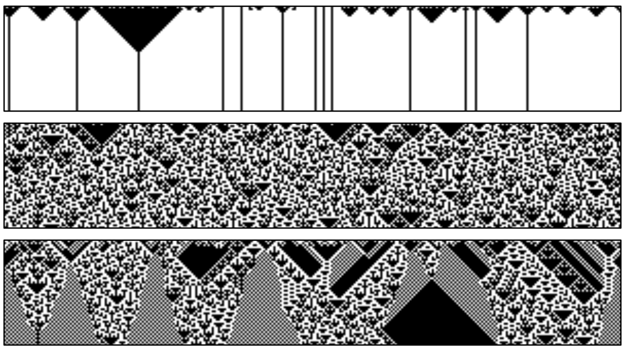

Figure 1: spatiotemporal patterns generated by one dimensional spatial prisoner's dilemma of All-D v.s. All-C. White cells cooperation and black defection. Simulation is carried out in 50 steps with a periodic boundary condition, with the space size is 300 (one dimensional square lattice) and the neighborhood radius : $r = 1$. The pattern above is a result of All-D v.s. All-C, the pattern middle is that All-D v.s. DCD and the pattern below is that of 1-D v.s. DCD. The above pattern is monotonous but the middle and the below involve fractal patterns.

3 SPD with Single Strategy as DCA

SPD can be identified as deterministic cellular automata (DCA) with D as 1, C as 0; and neighborhood radius r_{CA}. Thus we can apply the the rule identification algorithm [3] on spatiotemporal patterns of SPD. Table 3 is a result of rule identification. We use six patterns (Fig. 2) generated by a single strategy of SPD (without interaction and only action update) with payoff matrix of Table 1 and the parameter b 1.5. Each strategy has been tested assuming DCA and $r_{CA} = 1$.

When two strategies are involved, rules cannot be identified as DCA with a binary state and $r_{CA} = 1$. Only in the case of All-C v.s. All-D, rules can be identified as DCA with a binary state and $r_{CA} = 2$).

Table 3: Results of rule identification when single strategy is used. Rules are represented by Wolfram's numbering [5].

Strategy	2-D	CDC	1-D	DCC	DCD	DDC
Rule No.	160	90	250	5	155	95

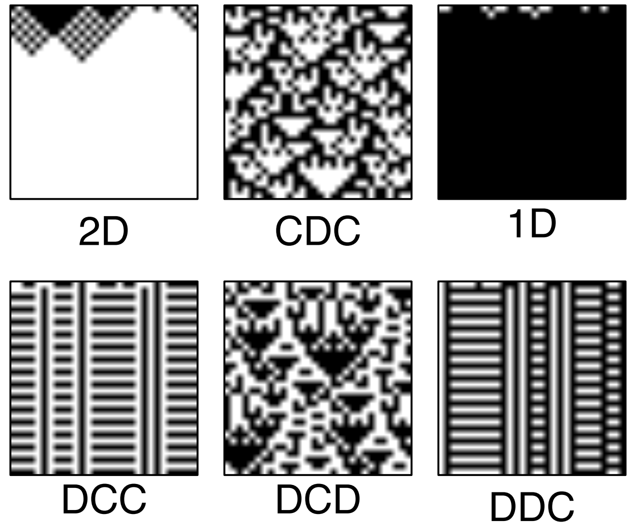

Figure 2: Spatiotemporal patterns generated by a single strategy of SPD with initial random seeds.

4 Estimating the payoff matrix When All-C v.s. All-D

Fig. 3 shows spatiotemporal patterns of All-C v.s. All-D when b is varied. The lattice size is 30 with a periodic boundary condition and simulations are carried out in 30 steps. Cs and Ds are seeded random with an equal probability initially. Table 4 lists the rules identified from the spatiotemporal patterns of Fig. 3. When $1.0 < b \leq 1.5$, rule is identified as $(ECF4EE30)_{16}$ and when $1.5 < b < 2.0$ as $(ECF4EF34)_{16}$. Thus the rule of SPD with All-C v.s. All-D can be identified as either $(ECF4EE30)_{16}$ or $(ECF4EF34)_{16}$ when $r_{CA} = 1$. In other word, we can estimate the range of the parameter b from spatiotemporal patterns generated by All-C v.s. All-D.

Table 4: Rule identification in the case of All-C v.s. All-D when the parameter b varied.

b	1.1	1.3	1.5	1.6	1.7	1.9
Rule No.	$(ECF4EE30)_{16}$			$(ECF4EF34)_{16}$		

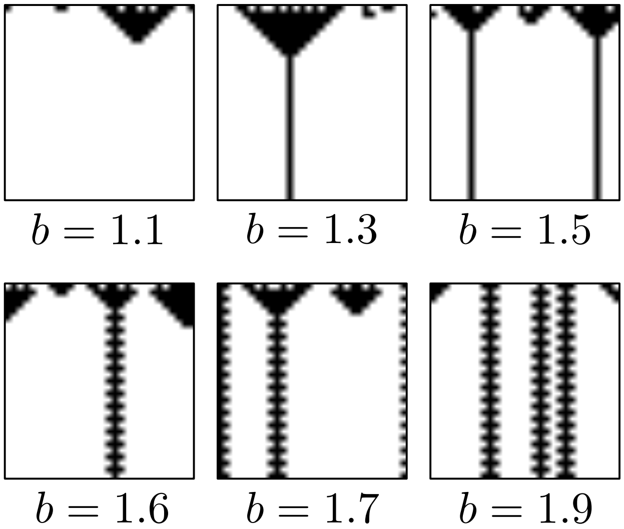

Figure 3: spatiotemporal patterns generated by All-C v.s. All-D in one dimensional lattice with a periodic boundary condition.

5 SPD with Multiple Strategies as PCA

When SPD is analyzed by the probabilistic rule identification algorithm [3], SPD with multiple strategies can be identified as probabilistic cellular automata (PCA) with a binary state and $r_{CA} = 2$ (Table 5).

Table 5: Rule of PCA. p_l indicates the probability of being D.

DDDDD	DDDDC	$\cdots$	CCCCD	CCCCC
p_{31}	p_{30}	$\cdots$	p_1	p_0

Table 6 shows identified rules of SPD as a rule of PCA when All-D v.s. 2-D.

Table 6: SPD identified as a rule of PCA when All-D v.s. 2-D with parameters: lattice size : 500, time steps : 500 steps and b : 1.5. These results are the average of trials of thousand times.

Rule ($r_{CA} = 2$) $p_{31}, p_{30}, \cdots p_0$
1.00, 1.00, 0.48, 0.00, 1.00, 1.00, 0.00, 0.00, 0.49, 0.50, 0.46, 0.49, 0.00, 0.49, 0.00, 0.00, 1.00, 1.00, 0.50, 0.00, 1.00, 1.00, 0.49, 0.00, 0.00, 0.00, 0.49, 0.70, 0.00, 0.00, 0.00, 0.00

6 Estimating the Payoff Matrix with All-D v.s. Other Strategies using PCA

Since SPD with All-C v.s. All-D can be identified as DCA, we can estimate range of the parameter b. But other cases (such as All-D v.s. 1-D) cannot be identified as DCA. However, these cases can be identified as PCA and we can estimate range of the parameter b as well (except the case : All-D v.s. 1-D). Fig. 4, 5 and 6 are binary trees showing stimated range of the parameter b in each game. We can find range of the parameter b by tracing the tree with conditional branches of probability p_l. For example, the parameter b in the case of All-D v.s. 2D with $p_9 \neq 0, p_{18} \neq 0, p_2 = 0, p_8 = 0$ is estimated within the range of $1.0 < b \leq 1.5$.

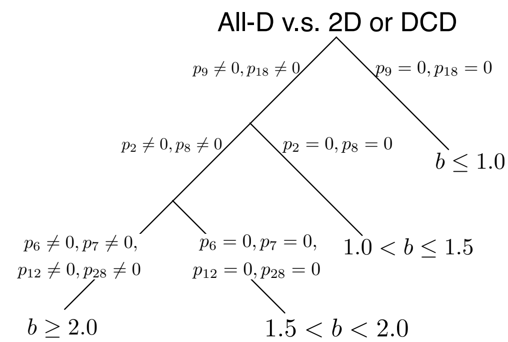

Figure 4: A binary tree showing estimated range of the parameter b when All-D v.s. 2-D or DCD.

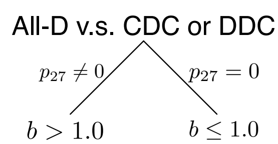

Figure 5: A binary tree showing estimated range of the parameter b when All-D v.s. CDC or DDC.

7 Estimating Strategies from Spatiotemporal Patterns

Since SPD with single strategy can be identified as DCA, we can infer strategies of each player. Fig. 7 above is a spatiotemporal pattern generated by 1-D v.s. DCD. Both below left and right figures in Fig. 7 is a part the pattern above. Table 7 lists a result

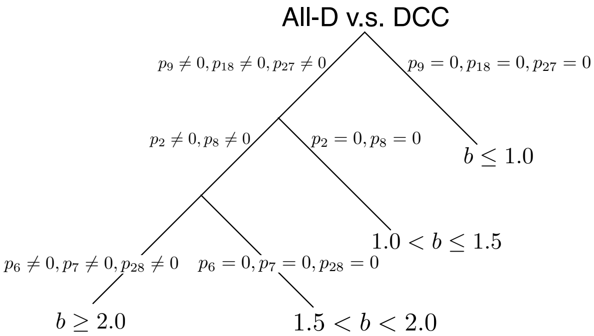

Figure 6: A binary tree showing estimated range of the parameter b when All-D v.s. DCC.

of the rule identification from the strategy cluster-A, and Table 8 from the strategy cluster-B. Upper rows of each table indicate the neighborhood configuration (i.e. CCC) and lower rows indicate the next state of the center. The symbol u (unknown) means that the state cannot be identified from the spatiotemporal patterns. The strategy of cluster-A is 1-D strategy (Table 7), and that of cluster-B is DCD strategy (Table 8). By analyzing a cluster of the spatiotemporal pattern, we can infer possible strategies that could generate the pattern.

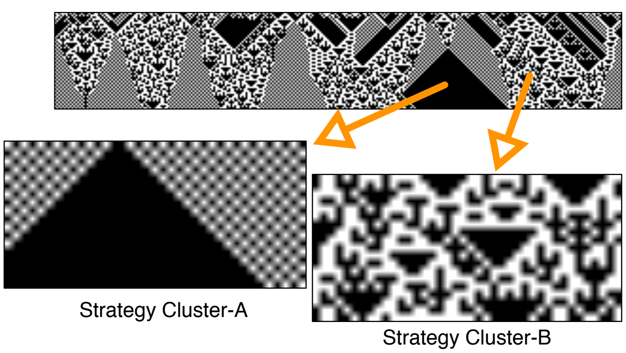

Figure 7: A spatiotemporal pattern generated by 1-D v.s. DCD (above) and its enlarged clusters (below).

Table 7: Identified rule of the cluster-A.

DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
D	D	D	u	D	C	u	u

Table 8: Identified rule of the cluster-B.

DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
D	C	D	C	C	D	C	D

8 Conclusions

Spatial prisoner's dilemma (SPD) may be considered as cellular automata (CA). Since a reverse engineering on spatiotemporal patterns generated by CA allows recognizing the possible rule generating the patterns, we can likewise infer the possible strategy underlying the game by reverse engineering on the spatiotemporal patterns generated by SPD. We regarded SPD with single spatial strategy as deterministic cellular automata (DCA); and SPD with multiple strategies as probabilistic cellular automata (PCA). Spatiotemporal patterns generated by SPD include sufficient information to estimate range of parameters of SPD (hence identifying the game played when an appropriate space-time frame is used).

Acknowledgments

We are grateful to Hisashi Ohtsuki for the important discussion. This work was supported in part by Grants-in-Aid from Toyohashi University of Technology and CASIO Science Promotion Foundation.

References

[1] M. A. Nowak and R. M. May, "Evolutionnary games and spatial chaos," *Nature*, Vol. 359, p.p.826-829, 1992.

[2] Y. Ishida and T. Mori, "Spatial Strategies on a Generalized Spatial Prisoner's Dilemma," *Artificial Life and Robotics*, Vol. 9, p.p.139-143, 2004.

[3] Y. Ichise and Y. Ishida, "Reverse Engineering of Spatial Patterns in Cellular Automata," *Artificial Life and Robotics*, 2009 to appear.

[4] J.J. von Neumann, *Theory of Self-Reproducing Automata*, edited and completed by A. Burks, *University of Illinois Press*, Champaign, 1966

[5] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, Vol.10, pp.1-35, 1984