

# 繰り返し囚人のジレンマにおける Satisficing 戦略

## Satisficing Strategy for Iterated Prisoner's Dilemma

### 1. 序論

どのような意思決定を行えば良いか難しい問題の例として囚人のジレンマがある。ここでは Satisficing という戦略をとるときに、Aspiration Level、ペイオフ利得を変えながらゲームの構造がどのようになるかを検討した。

### 2. モデル説明

囚人のジレンマとは、自分の利得だけを見れば常に裏切の方が得だが、互いに裏切るより互いに協調したほうが互いにとって得であるような状態を表す。

表1 囚人のジレンマの利得行列の例

	B:C(協調)	B:D(裏切)
A:C(協調)	(3,3)	(1,4)
A:D(裏切)	(4,1)	(2,2)

この利得行列のパラメータを変化させることを考える。全体の規模を拡大縮小・平行移動させたものは本質的に同じと考えれば CD のときの利得は(0,1)で固定できる。

表2 囚人のジレンマの一般化

	B:C(協調)	B:D(裏切)
A:C(協調)	( $\sigma a, \sigma b$ )	(0,1)
A:D(裏切)	(1,0)	( $\delta a, \delta b$ )

ただし囚人のジレンマとして成立する条件

$$0 < \delta < \sigma < 1 \quad 0.5 < \sigma \quad (1)$$

このような囚人のジレンマによる得点獲得を繰り返し、高得点を得られる作戦を考える。ここで Satisficing 戦略を使用する。

Aspiration Level  $\alpha$  を設定する。利得  $\pi$  が  $\alpha$  以上ならば(Satisficing)次回も同じ行動を取って現状維持を目指し、 $\alpha$  未満ならば次回は別の行動を取る。 $\alpha$  は利得  $\pi$  に基づいて更新される。 $\alpha$  の変化の速さを制御する変数  $\lambda$  ( $0 \leq \lambda \leq 1$ ) を与える。

$$\alpha[t+1] = \lambda \alpha[t] + (1-\lambda) \pi[t] \quad (2)$$

### 3. 実験

実際に Satisficing 戦略同士で回してみると、CC か DD に収束またはその他のサイクル(ほとんどの場合 DD-DC-DD-CD) で終わることがわかった。1 度でも互いに Satisficing になればそ

こで収束し、互いの AL が  $\sigma$  以下で CC になったら CC 収束し、互いの AL が  $\delta$  以下で DD になったら DD 収束する。DD-DC-DD-CD サイクルは互いの AL が交互に高くなったり低くなったりしていつまでも収束しない。

表3 DD-DC-DD-CD サイクルの例  
( $\sigma=0.8 \quad \delta=0.4 \quad \lambda=0.5$ )

t	A	$\alpha$	$\pi a$	$\pi b$	$\beta$	B
0	C	1.0	0.8	0.8	0.6	C
1	D	0.9	1.0	0.0	0.7	C
2	D	0.95	0.4	0.4	0.35	D
3	C	0.68	0.0	1.0	0.38	D
4	D	0.34	0.4	0.4	0.69	D
5	D	0.37	1.0	0.0	0.54	C
6	D	0.68	0.4	0.4	0.27	D
7	C	0.54	0.0	1.0	0.34	D

パラメータ  $\sigma \cdot \delta \cdot \lambda$ 、 $t=0$  での  $\alpha \cdot \beta \cdot A \cdot B$  をすべて決めれば、その後の 2 人の行動の変化の様子は完全に決定され、CC 収束、DD 収束、その他のサイクルにパターン分けできる。CC 収束を白、DD 収束を黒、その他サイクルをグレーと決める。 $\alpha \cdot \beta$  を変数として、残りは定数として適当な値を入れ、 $\alpha \cdot \beta$  を変化させていきグラフにすると、定数の値によっては次のような模様が描かれる。

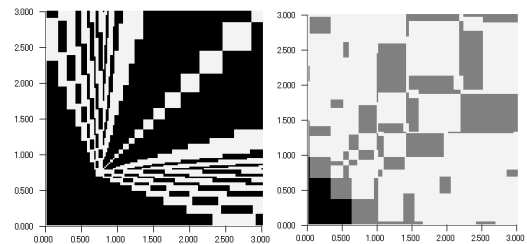


図1 収束の様子

### 4. 結論

Satisficing 戦略空間でどのように CC が生じるかを求めた。Satisficing 戦略は、相手の戦略やペイオフの情報を必要とせず、アルゴリズムが簡単な割には比較的良好な結果を得られた。

### 参考文献

J.Stimpson, et al, IJCAI(2001)