

繰り返し囚人のディレンマ・ゲームにおけるノイズとメモリーサイズ Noise and Memory Size in Computer Simulation of Iterated Prisoner's Dilemma

高増 明・服部純典

Akira Takamasu and Yoshinori Hattori

大阪産業大学経済学部

1. はじめに

Axelrod(1987)によって開始された繰り返し囚人のディレンマ・ゲームのコンピューター・シミュレーションは近年様々な研究者によって実験が行われている。コンピューター・サイエンスでは、遺伝的アルゴリズムなどを使って、より強い戦略を産み出すことに研究の焦点が置かれているが、社会科学や生物学では、様々な戦略が会って対戦するなかで、どのようにして協調あるいは社会秩序が形成されていくのかという問題の解明に力点が置かれている。そのようなアプローチの代表例として、Lindgren (1991)、Lindgren and Nordahl (1993)、(1996)がある。また高増・服部(1999)は、Lindgren and Nordahlのアプローチをより厳密なかたちで再定式化し、メモリー1の4種類の戦略、すなわち正直者、裏切り者、しっぺ返し(TFT)、反しっぺ返し(ATFT)がランダムに出会って戦うなかで、どの戦略が生き残るのかを利得行列や初期の個体数を変化させて検討している。

高増・服部(1999)の実験では、しっぺ返し戦略が常に好成績を残した。裏切り戦略は、相手を騙したときの利得が協調した場合に比べて大きいとき、正直者の初期の個体数が多いときには勝利したが、そうでないケースについては滅びる場合もあった。正直者や反しっぺ返しは、一般に好結果を得られなかった。しかしながら、この実験では、各プレイヤーは自分の決めた手を実行するにあたって決してミスを犯さない、別の表現を使うとノイズが存在しないと仮定されていた。またプレイヤーのメモリーは1に限定されていたため、戦略は4種類しか存在しなかった。

この論文では、シミュレーションのこうした限定性をはずし、シミュレーションをより一般化していく。そして、ノイズの存在が、高増・服部(1999)の結果にどのような影響を及ぼすのか、またメモリーのサイズを1から2に増加したときに、どのような結果が得られるのかをできるだけ厳密に検討する。

論文の構成を簡単に紹介しよう。第2節では、コンピューター・シミュレーションにノイズを導入する。そして、ノイズがある割合を超えると、しっぺ返し戦略が減るケースがあることを明らかにする。つぎに第3節では、メモリーのサイズを2にし、各プレイヤーが前回相手が選択した手だけではなく、前々回に相手が選択した手についても記憶していると仮定する。そして、そのようなケースについて16種類の戦略が対戦したときに、どの戦略が生き残るのかを実験していく。実験では、初期の個体数を変化させたり、ノイズをいれて、その影響を調べる。その結果は、ノイズが存在しないときには、TFTが勝つか、また裏切り戦略が勝つか、のどちらかになった。一方、ノイズをいれてやると協調を選択する戦略が勝利するケースが存在することが明らかになった。

2. ノイズの効果

ノイズの存在は、繰り返し囚人のディレンマの対戦結果にどのような影響を与えるだろうか。裏切り戦略と協調戦略については、常に選択する手は一定であるから、ノイズによってミスが生じた対戦の利得が変化するだけである。したがって、その影響はそれほど大きくない。しかし TFT の場合には影響は深刻である。1 回のミスがそれ以後の対戦に大きな影響を与えるからである。たとえば、TFT と TFT が対戦し、両者が協調していたとしよう。ある時点で一方がミスをして裏切ったとする。このとき次回に相手は裏切り返すだろう。ミスをしたほうは、つぎの回は協調戦略をとることになるから、以後裏切りと協調が交互に続き、両者の利得は協調を続けたときに比べて低下することになる。このように、ノイズの存在は TFT に不利な影響を与え、TFT は生き延びることがむずかしくなることが予想される。ここでは、その予想が現実妥当なのかをシミュレーションしてみよう。

2.1 コンピューター・シミュレーションの方法

コンピューター・シミュレーションは高増・服部(1999)と同じ、つぎのような方法で行うことにする。ただし、ここでは、ノイズ、すなわちミスをする可能性を与えてやる。ミスの可能性は 0%、10%、15% とする。

00、01、10、11 という 4 種類の戦略（遺伝子タイプ）について初期の個体数を与えてやる。ここでは 00 を 25 個、01 を 45 個、10 を 15 個、11 を 15 個の合計は 100 個にする。

100 個の個体について、第 1 回めの対戦の対戦相手を乱数によって決定する。

第 1 回めに、各戦略（遺伝子タイプ）は、00 は裏切り(0)、01 は協調(1)、10 は裏切り(0)、11 は協調(1)を選択する。

第 1 回めの対戦が行われ、それぞれの利得が決定される。利得行列は、表 1 のように、両者が協調した場合には、それぞれ 5 点の得点、片方が協調し片方が裏切った場合には、裏切ったほうが 6 点、協調したほうが 0 点、両者が裏切った場合には、それぞれ 1 点の得点とする。第 1 回めで、各個体が選択した戦略は、その個体の 1 期前のヒストリーになる。

第 2 回めの対戦のペア（対戦相手）を乱数によって決定する。第 2 回めの対戦において、各戦略（遺伝子タイプ）が選択する手は対戦相手のヒストリーによって決定される。たとえば 01 という戦略であれば、前期に対戦相手が裏切り(0)を選択した場合には裏切り(0)、協調(1)を選択した場合には協調(1)を選択する。

このようにして 30 回の対戦が行われ、各個体についての利得の合計を戦略（遺伝子タイプ）ごとに集計し、個体数で割って平均を求める。また総平均も求める。利得の平均が総平均より大きい戦略（遺伝子タイプ）は、その個体数がその差に比例して増加し、逆に小さい戦略はその差に比例して減少する。

に戻って対戦を行う。 ~ を繰り返して、各戦略（遺伝子タイプ）の増減をみる。

プレイヤー 2

		協調 (C)	裏切り (D)
プレイヤー 1	協調 (C)	(5, 5)	(0, 6)
	裏切り (D)	(6, 0)	(1, 1)

表 1 囚人のディレンマの利得行列

2.2 シミュレーション

シミュレーションの結果は、以下のようになった。

・ノイズが存在しないとき

このとき、正直者(11)は、その個体数を増やしていき、60 程度で安定する。裏切り者(ALLD)と反しっぺ返し(ATFT)はすぐに滅亡する。これは利得行列と初期の個体数が協調戦略に有利に設定されているからである。

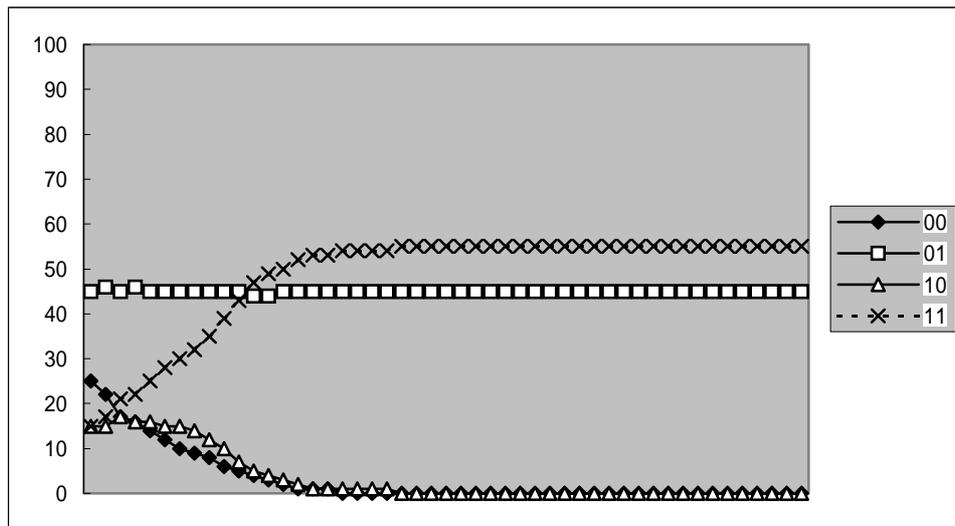


図 1 ノイズが存在しないとき

・ノイズが 10% のとき

ノイズが 10% のときには、ノイズが存在しないときと同様に、TFT と正直者が生き残り、裏切り者と ATFT は絶滅した。ただし、TFT の個体数は、最終的には 35 前後になり、ノイズが存在しないとき (45) と比較して 10 程度小さくなっている。逆に正直者は 65 前後と 10 程度増加していて、ノイズが TFT に不利な影響を及ぼしていることがわかる。

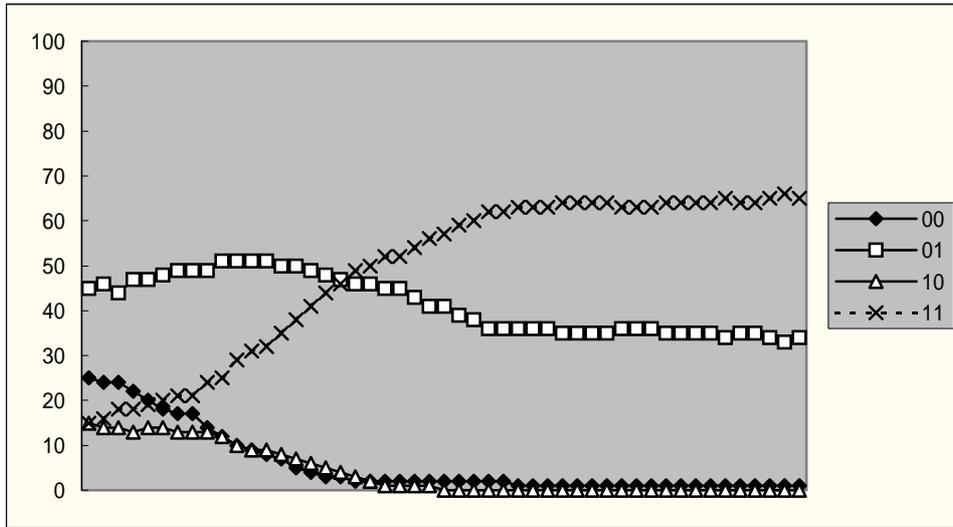


図2 ノイズが10%のとき

・ノイズが15%のとき

ノイズが15%になると、逆に、裏切り者だけが生き残り、他の戦略はすべて滅亡した。ノイズが存在しないときには、TFT は裏切り者と対戦するときには裏切りを選択し、TFT 自身と対戦するときには協調を選択するから、裏切り者に負ける可能性は小さい。しかし、ノイズが存在するときには、裏切り者に対して協調を誤って選択したり、裏切り者が誤って協調を選択したときに、相手を正直者と勘違いして協調を選択してしまう可能性がある。TFT が TFT 自身と対戦するときにも、そのようなことがおきるから、TFT が裏切り者に負ける可能性が大きくなるのである。

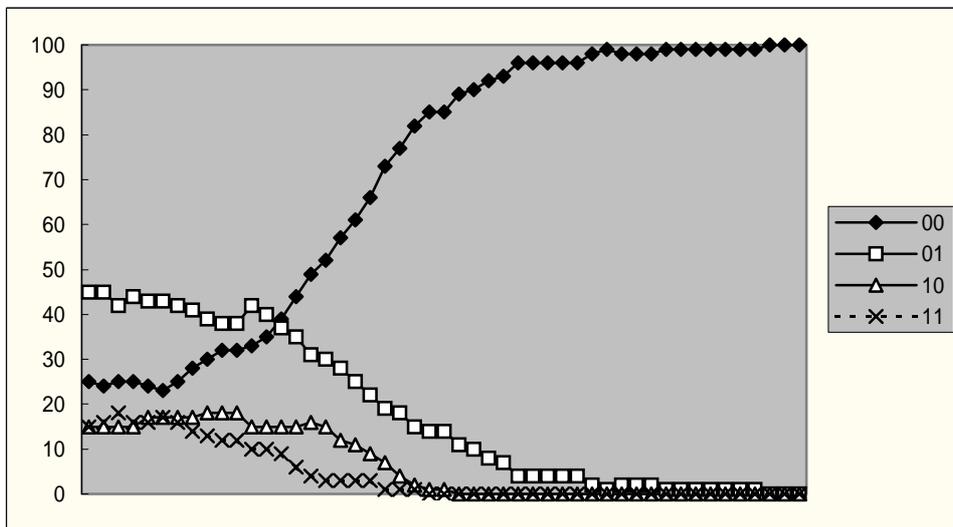


図3 ノイズが15%のとき

2.3 結果のまとめ

この結果から、ノイズ、すなわちミスをする確率が高くなるにつれて、TFT の優位は失われ、裏切

り者が勝利することが明らかになった。このコンピューター・シミュレーションでは、裏切り者が勝つようになり、TFT が滅びるようになるのは、ノイズが 10% と 15% のあいだのときである。

3. メモリー・サイズが増加したときの効果

つぎに、メモリー・サイズが増えていくにつれて、どのようなことが起きるのかをみていこう。

メモリーが 1 のときには、各プレイヤーは相手が前回選んだ手しか記憶していない。そして、前回、相手が裏切ったか、協調したかに応じて、こちらの手を選択する。したがって、戦略の種類は表 2 のように 4 種類しか存在しなかった。4 種類とは、それぞれ、裏切り(00)、しっぺ返し(TFT)(01)、ATFT(10)、正直者(11)である。

	前回対戦相手が取った手	
	0	1
戦略 1 (裏切り)	0	0
戦略 2 (TFT)	0	1
戦略 3 (ATFT)	1	0
戦略 4 (正直)	1	1

表 2 メモリーが 1 のときの戦略の種類

メモリーが 2 になると、前回対戦相手をとった手(0、1 の 2 通り)、前々回対戦相手をとった手(0、1 の 2 通り) のそれぞれの組み合わせ(4 通り)に対して、裏切り(0)、協調(1)を選択することができるから、戦略の数は、以下の表のように 16 種類(2 の 4 乗通り)になる。

	(前々回対戦相手をとった手, 前回対戦相手をとった手)			
	0, 0	0, 1	1, 0	1, 1
戦略 0 : 0000 (裏切り者)	0	0	0	0
戦略 1 : 0001 (2 回続けて相手が協調したときにこちらも協調)	0	0	0	1
戦略 2 : 0010	0	0	1	0
戦略 3 : 0011 (前々回の相手と同じ手をとる)	0	0	1	1
戦略 4 : 0100	0	1	0	0
戦略 5 : 0101 (TFT)	0	1	0	1
戦略 6 : 0110	0	1	1	0
戦略 7 : 0111 (相手が 2 回続けて裏切ったときにこちらも裏切る)	0	1	1	1
戦略 8 : 1000	1	0	0	0
戦略 9 : 1001	1	0	0	1
戦略 10 : 1010 (ATFT)	1	0	1	0
戦略 11 : 1011	1	0	1	1
戦略 12 : 1100	1	1	0	0
戦略 13 : 1101	1	1	0	1

戦略 14：1110 (2回続けて相手が協調したとき にはじめて裏切る)	1	1	1	0
戦略 15：1111 (正直者)	1	1	1	1

表3 メモリーが2のときの戦略の種類

この16種類の戦略が、ランダムに出会って対戦するとき、その個体数はどのように変化するかをコンピューター・シミュレーションを行って調べてみよう。

3.1 コンピューター・シミュレーション

コンピューター・シミュレーションはつぎのような方法で行った。

16種類の戦略(遺伝子タイプ)について初期の個体数を与えてやる。たとえば、各戦略を10個、総計で160個というようにする。

すべての個体について、対戦以前の履歴を便宜的に与えてやる¹。戦略0(0000)、すなわち裏切り者については、すべての個体が、1回前の対戦でとった手が0(裏切り)、2回前の対戦でとった手を0とする。また戦略15(1111)、すなわち正直者については、1回前が1、2回前も1とする。他の遺伝子タイプについては、つぎの表4のような4種類のような履歴タイプをもつ個体数が、総個体数のそれぞれ1/4ずつだとする。

	1回前の対戦	2回前の対戦
履歴タイプ0	0	0
履歴タイプ1	0	1
履歴タイプ2	1	0
履歴タイプ3	1	1

表4 履歴のタイプ

第1回めの対戦相手となる個体のペア(対戦相手)を乱数で決定する。第1回めの対戦で、各戦略(遺伝子タイプ)は、相手の履歴によって、自分のとる手を選択する。たとえば、0111という戦略7は、相手の履歴がタイプ3の場合、1回前の対戦でとった手が1、2回前の対戦でとった手が0だから、表3にしたがって、1(協調)を選択する。

第1回めの対戦が行われ、それぞれの利得が決定される。利得行列は表1のように仮定し、両者が協調した場合には、それぞれ5点の得点、片方が協調し片方が裏切った場合には、裏切ったほうが6点、協調したほうが0点、両者が裏切った場合には、それぞれ1点の得点とする。第1回の対戦で、各個体を選択した戦略は、その個体の1期前の履歴になる。そして、1期前の履歴は2期前の履歴になる。

¹ もちろん対戦前に履歴は存在しないわけだから、ここでの履歴の与え方は恣意的なものである。このようにしないと、はじめの2回の対戦について、各戦略(遺伝子タイプ)のとる手を決定できないからである。このような方法をとるかわりに、各戦略について、はじめの2回の対戦について、とる手をあらかじめ与えてやる方法もある。

このようにして 50 回の対戦が行われ、各個体についての利得の合計を戦略（遺伝子タイプ）ごとに集計し、個体数で割って各戦略の得点（利得）の平均値を求める。また総平均も求める。得点の平均が総平均より大きい戦略（遺伝子タイプ）は、その個体数がその差に比例して増加し、逆に小さい戦略はその差に比例して減少する。比例係数は 1 とする²。

最後の 2 回の対戦で、各戦略（遺伝子タイプ）がとった手のパターンをヒストリータイプ 0 から 3 に分類し、それを戦略ごとに集計する。その比率にしたがって、つぎの個体のヒストリータイプを決定する。

にもどって、対戦を行う。～ のプロセスを 200 回反復して、各遺伝子タイプの増減を調べる。

3.2 シミュレーション

まず、初期の個体数を変化させ、それが戦略の生き残りにどのように影響するのかを調べた。つぎにノイズをいれたときに、それがどのように結果に影響するのかについて検討した。

・シミュレーション 1（各遺伝子タイプの個体数を 10 としたとき）

各戦略の初期の個体数を 10 個として、シミュレーションを行った。このとき、各戦略の個体数は以下の表 5 のように変化した。表 5 で、はじめの行は、それぞれの戦略（遺伝子タイプ）、1 番左の列は対戦回数、各戦略の列は、その戦略の個体数の変化を示している。

このシミュレーションでは、裏切り戦略（0000）は、第 9 回までは個体数が増加している。これは、正直者（1111）など相手の戦略にかかわらず協調という手を選択する個体と対戦することによって高得点を得た結果である。しかし、協調戦略を無条件にとる個体が減少するとともに、得点は減少し 100 回以降に滅亡してしまう。最後まで生存し、個体数が最も増加したのは、TFT（0101）で、200 回の対戦では、60 まで個体数が増加した。

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
2	13	12	12	9	12	10	11	8	12	9	10	8	11	8	9	6
4	17	13	15	9	14	10	12	7	13	8	9	5	11	6	7	4
6	22	15	17	10	16	11	12	7	13	6	6	3	10	4	5	3
8	24	16	19	10	18	12	14	7	14	5	5	1	8	2	4	1
10	24	16	20	11	19	12	14	7	15	4	4	1	7	2	3	1
20	17	13	21	15	20	13	20	6	20	3	2	1	6	1	1	1
30	8	9	19	16	22	19	25	6	24	3	1	1	4	1	1	1
40	4	8	19	15	21	21	28	5	31	2	1	1	3	1	0	0
50	3	7	21	15	17	29	24	4	34	1	1	0	3	1	0	0
60	3	7	23	13	17	30	24	4	33	1	1	0	3	1	0	0
70	2	6	23	12	13	34	29	4	33	1	1	0	1	1	0	0
80	2	2	22	9	13	37	33	3	36	1	1	0	1	0	0	0
90	1	2	23	9	15	38	30	2	37	1	1	0	1	0	0	0

² 各戦略（遺伝子タイプ）の個体数がマイナスにならないためには、比例係数は 1 を超えることはできない。

100	1	2	24	9	17	35	30	2	37	1	1	0	1	0	0	0
150	0	2	29	6	19	41	24	1	36	0	1	0	1	0	0	0
200	0	1	20	6	15	60	17	1	38	0	1	0	1	0	0	0

表5 シミュレーション1の結果

・シミュレーション2(TFTの個体数を30、協調を選択する可能性が高い戦略(1100、1101、1110、1111)の個体数を5としたとき)

高増・服部(1999)では、メモリー1のケースについてシミュレーションを行い、TFTの初期の個体数を増やし、正直者(11)の個体数を少なくするときには、TFTと正直者が生き残るという結果が得られた。したがって、ここでも、TFTの数を増やし、協調という手を選択する可能性が高い戦略(遺伝子タイプ)の初期の個体数を抑制したときに、果たして正直者が生き残れるのかをシミュレーションしてみた。その結果は、予想に反して、裏切り者(0000)の個体数が最大になるというものだった。確かに、正直者は、150回までは減びることはなかったが、その結果、裏切り者がゆっくりと増殖していき、結果として、最も勢力を伸ばすことになった。そして、200回終了時の正直者の個体数は100個になった。裏切り者の初期の個体数をもって抑えれば、あるいは、メモリー1のときと類似の状況が生まれるのかもしれないが、そのようなケースをうまく見つけ出すことはできなかった。

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	10	10	10	10	10	30	10	10	10	10	10	10	5	5	5	5
10	13	11	11	9	12	30	11	7	9	11	10	7	6	4	5	4
20	13	12	9	9	10	32	11	7	6	9	11	7	8	7	5	4
30	13	10	9	10	7	33	11	8	9	9	13	5	7	7	5	4
40	19	12	11	10	8	29	10	6	10	9	14	3	7	5	4	3
50	22	15	11	9	7	30	12	6	10	9	11	3	7	3	3	2
60	22	15	9	10	6	33	13	6	9	9	10	3	7	2	4	2
70	21	13	9	11	5	33	11	7	9	11	10	3	8	2	5	2
80	22	12	8	12	4	33	12	8	10	12	9	2	8	2	4	2
90	26	15	8	12	4	30	10	6	11	10	10	2	7	2	4	3
100	34	17	10	9	3	27	10	6	12	7	9	2	7	2	2	3
150	53	26	13	5	2	19	8	4	11	3	4	1	5	2	2	2
200	100	21	12	2	2	12	3	2	2	0	0	0	1	1	1	1

表6 シミュレーション2の結果

・シミュレーション3(各戦略の個体数を10、ノイズを15%としたとき)

つぎに、メモリー2のケースについて、ノイズ、すなわちエラーの可能性が各戦略(遺伝子タイプ)の生き残りにどのような影響を与えるのかを実験してみた。初期の各戦略の個体数を10とし、ノイズを15%として、シミュレーションを行った。このときには、0111という戦略の個体数が200回の後に、最大になった。これは、相手が2度続けて裏切らないかぎり、こちらも裏切らないという戦略である。前にも説明したように、TFTはノイズには強くない。それに対して、0111は、相手が1回間違っても、2回続けて間違わないかぎり、気にしない戦略である。したがって協調的な手を選択する戦略が支配的になるような状況では、TFTよりも強くなるのがわかる。裏切り者(0000)が100回で滅亡した後に、相手が協調しても裏切る戦略、すなわち、0100、0110、1000は個体数を減少させていき、150回には、ともに滅亡することになった。

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	19	14	19	9	20	13	9	9	11	4	7	4	9	5	5	3
20	19	16	19	11	23	15	7	10	12	3	5	3	9	3	3	2
30	16	14	17	12	28	12	15	8	13	3	5	3	7	3	3	1
40	11	10	17	17	30	14	17	7	19	3	3	3	5	1	2	1
50	6	7	17	19	36	12	19	7	22	3	2	3	4	1	1	1
60	4	8	17	21	30	14	22	8	23	3	1	2	4	1	1	1
70	2	6	16	26	27	17	22	8	23	3	1	2	4	1	1	1
80	2	4	16	29	23	19	21	9	24	4	1	2	4	1	0	1
90	1	3	15	29	20	25	23	9	23	3	1	2	4	1	0	1
100	1	3	9	30	14	28	29	10	23	4	1	2	4	1	0	1
150	0	1	0	42	0	39	0	69	0	5	0	2	0	1	0	1
200	0	1	0	43	0	39	0	68	0	5	0	2	0	1	0	1

表 7 シミュレーション 3 の結果

・シミュレーション 4 (6 つの代表的戦略の対戦。ノイズを 15% とする。)

最後に、メモリー 2 という状況において、代表的だと考えられる 6 つの戦略 (遺伝子タイプ) を選択し、それが相互に戦う状況をシミュレーションしてみた。代表的戦略として選択したのは、裏切り者 (0000)、相手が 2 度続けて協調したときだけ相手を信じて協調する慎重な戦略 (0001)、TFT (0101)、相手が 2 度続けて裏切ったときだけ相手を裏切る戦略 (0111)、相手が 2 度続けて協調したときだけ逆に相手を裏切る戦略 (1110)、正直者 (1111) である。他の戦略については、現実社会との関係で、もっともらしい意味付けを見つけることができなかつたため入れなかつた。初期の個体数はすべて 20 とした。

このときには、表 8 のように、正直な戦略だけが生き残り、結果として、すべての戦略が協調を選択するという状況になった。そのときには、ノイズに無関心な正直者が実は最強になり、正直者が最も個体数が増加することになった。

	0000	0001	0101	0111	1110	1111
0	20	20	20	20	20	20
10	6	22	22	22	21	27
20	2	19	24	27	11	37
30	1	18	24	27	1	49
40	1	18	24	27	0	50
50	1	18	24	26	0	51
60	1	18	24	26	0	51
70	0	19	23	29	0	49
80	0	18	23	29	0	50
90	0	18	23	30	0	49
100	0	19	22	31	0	48
150	0	19	23	29	0	49
200	0	20	20	29	0	51

表 8 シミュレーション 4 の結果

3.3 シミュレーションの結果とまとめ

このようなシミュレーションの結果から、どのようなことを読み取ることができるのだろうか。まずノイズが存在しないときには、メモリー 1 の単純な戦略、裏切り者が TFT が勝利した。これは、多

様な戦略が存在する状況のなかでは、複雑な戦略をとることにあまり意味がなくなり、単純な戦略が有利になるということであろう。戦略が淘汰され、少数になったときには、状況は変化してくるが、どの戦略がそれまでに生き残るかは、初期の個体数や利得行列の微妙な違いによって影響され、簡単に決定することはできないことも明らかになった。

つぎに、ノイズをいれてやると、協調を選択する戦略が有利になるという結果がでた。これはメモリー1のケースとは反対の結果である。しかしながら、よく考えてみればわかるように、メモリー1のときには、ノイズの影響は非常に直接的に現れたのに対して、メモリーが2になると影響を中和するような戦略が多数出現することになる。したがって、ノイズの影響がメモリー1とメモリー2で大きく違ってくるのは当然のことであろう。裏切り戦略は、単純に相手を信じる戦略が多いときには、得点を伸ばすことができるが、相手が過去に選択した手をみる戦略が存在するときには、不利になって、やがて滅亡していった。

代表的と考えられる6つの戦略を戦わせた結果は、協調を選択する戦略が勝利するというものだった。両者が協調したときの得点が5点、協調する相手を裏切ったときの得点が6点というように、裏切りによる一時的な利益があまり多くないときには、最終的には、生き残ったすべての戦略が協調を選択しつづける結果になった。

4. おわりに

この論文では、ノイズの存在とメモリーサイズの大きさが、繰り返し囚人のディレンマのコンピューター・シミュレーションにどのような影響を与えるのかを考察した。

多くの興味深い結果が得られたが、ひとつの問題点は、この論文で得られた結果は、すべてコンピューター・シミュレーションによるもので、解析的にどの戦略が勝つのかを分析したものではないという点である。そのような分析として、経済学においては、たとえば Rosenthal(1979)がメモリーサイズが1のケースについて分析を行っているが、そのような分析をこの論文が取り扱っているような状況へ拡張することができ、そこで意味のある結論が得られるのかを検討すべきであろう。

もうひとつの問題点は、戦略それ自体の進化というものを取り扱っていない点である。メモリーサイズは1あるいは2に固定されていたが、現実の生物の世界、経済では、メモリーサイズそれ自身が進化によって大きくなっていくことがある。また遺伝子のタイプも突然変異や交差によって進化していくだろう。そのような戦略の進化を導入するかたちに、今後シミュレーションを拡張していくべきだと考えられる。

参考文献

Axelrod, R. and W. Hamilton (1981), "The Evolution of Cooperation", *Science* 211, pp. 1390-1396.

Axelrod, R. (1984), *The Evolution of Cooperation*, Basic Books. (松田裕之訳『つきあい方の科学：バクテリアから国際関係まで』HBJ 出版局)

- Axelrod, R. (1987). "The Evolution of Strategies in the Iterated Prisoner's Dilemma", in L. Davis ed., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann.
- Langton, C. G. ed. (1989). *Artificial Life*, Addison-Wesley Publishing Company.
- Langton, C. G. ed. (1993). *Artificial Life III*, Addison-Wesley Publishing Company.
- Langton, C. G. ed. (1996). *Artificial Life: An Overview*, The MIT Press.
- Lindgren, K. (1991), "Evolutionary Phenomena in Simple Dynamics", in C. G. Langton, J. D. Farmer, S. Rasumussen and C. Taylor eds., *Artificial Life II*, Addison Wesley.
- Lindgren, K. and M. G. Nordahl (1993), "Artificial Food Webs", in Langton(1993).
- Lindgren, K. and M. G. Nordahl (1996), "Cooperastion and Community Structure in Artificial Ecosystems", in Langton(1996).
- Poundstone, W. (1992), *Prisoner's Dilemma*, Doubleday. (松浦俊輔他訳 (1995) 『囚人のディレンマ』 青土社)
- Rosenthal, R. (1979), "Sequences of Games with Varying Opponents", *Econometrica* 47, pp. 1353-1366.
- 高増・服部「協調の進化：繰り返し囚人のディレンマ・ゲームのコンピューター・シミュレーション」
『大阪産業大学論集 社会科学編』

付録 コンピューター・プログラムの解説

以下では、シミュレーションに使用したプログラムについて簡単に解説する。プログラムは F-BASIC で書かれているが、F-BASIC に固有な命令は使っていないため、どのような BASIC でもほぼそのままのかたちで動くはずである。プログラムの構成は以下のようになっている。

7~8行：配列変数の宣言。G は各個体のもつ戦略（遺伝子タイプ）、A は、その個体に割り当てられた番号、H はヒストリー、R は利得行列、N は各戦略の個体数、HH はゲームが開始する以前に便宜的に与えられたヒストリー・タイプ、TR は各戦略の得点の合計である。

9行：各戦略（遺伝子タイプ）の個体数を記録するデータ・ファイル（IPD.DAT）を開く。

16行：利得行列。たとえば $R(1,1)=5$ は、自分が 1（協調）をとり、相手も 1（協調）をとったときの自分の利得 5 を示しているまた $R(1, 0)=0$ は、自分が 1（協調）をとり、相手が 0（裏切り）をとったときの自分の利得 0 を示している。ER はノイズの大きさ、ここでは 15% を示している。

20~27行：各戦略（遺伝子タイプ）の個体数 $N(I)$, $I=0, \dots, 15$ を入力する。ここで個体数の合計は偶数でなければならない。

28~42行：戦略（各遺伝子タイプ） I について、対戦する以前のヒストリーを便宜的に与えてやる。裏切り者(0000)については、すべてが同一のヒストリータイプ 0 であり、前回にとった手が 0、前々回にとった手が 0 であるとする。また正直者(1111)については、すべての個体がヒストリータイプ 4 であり、前回は 1、前々回は 1 とする。他の戦略（遺伝子タイプ）については、ヒストリータイプ 0 ($HH(I, 0)$) からヒストリータイプ 3 ($HH(I, 3)$) が、それぞれ各戦略の総個体数の $1/4$ ずつだとする。

47~50行：各戦略（遺伝子タイプ）の個体数の合計 NN を求める。

51~54行：各個体のペアをつくるための、各個体に数字を割り振っておく。

62~72行：各個体に、その個体がとる戦略（遺伝子タイプ）とヒストリーを与えていく。ここで、 $\%$ は整数を整数で割った商、 mod は整数を整数で割ったときの余りをもとめる命令である。たとえば、 NH というヒストリータイプが 2 のとき、前回とった手は、 NH を 2 で割った商の 1 となり、前々回にとった手は NH を 2 で割った余りの 0 になる。

80~83行： NN 個の個体について、対戦相手のペアを乱数によって決定してやる。

87~94行：各個体は自らの戦略（遺伝子タイプ）にしたがって、対戦相手のヒストリーから、対戦で選択する戦略を決定する。対戦で選択した戦略が次期の対戦のヒストリーになり、今回の得点が各戦略（遺伝子タイプ）のこれまでの総得点に加算される。

95~96行：乱数を発生させ、その乱数の値がノイズ（0.15）より小さければ、各個体は手を誤って選択する。

97行：このような対戦をすべてのペアについて行う。

98行：対戦を 200 回繰り返す。

102~110行：すべての戦略（遺伝子タイプ）について、得点が合計され、総平均得点が求められる。

114~117行：各戦略（遺伝子タイプ）について、ヒストリータイプの個体数が 0 に戻される。

119～122 行：各戦略（遺伝子タイプ）について、最後の二つの対戦についてのヒストリーからヒストリータイプの数がカウントされる。

124～130 行：各戦略（遺伝子タイプ）の得点の平均値を総平均で割った結果に総個体数を乗じて、次期の各戦略の個体数が計算される。すなわち得点の平均値が総平均より大きい戦略については個体数が増え、小さい戦略については個体数が減少すると考える。次期の個体数は、はじめに計算された結果の小数点以下を切り捨てて整数として求められ、その値と元の値との差を記録しておく。

131～139 行：差は大きい順に並べられ、各戦略の次期の個体数の合計が総個体数に等しくなるまで、差の大きい順に、遺伝子タイプの個体数に 1 を加えていく。

140～154 行：各戦略（遺伝子タイプ）について、ヒストリータイプ 0 から 3 までの個体数を集計し、それを個体数の合計で割って、ヒストリータイプの全体に占める比率を求める。その比率を次期の個体数に乗じて、各戦略について、ヒストリータイプの次期の個体数を求めてやる。

162 行：54 行に戻って再び対戦が繰り返される。このような繰り返しを 200 回行う。

```

1  *****
2  ' Iterated Prisoner's Dilemma(IPD)
3  '   Computer Simulation
4  '   Case of Memory Size 2
5  '   1999.01.23.
6  ' *****
7  dim G(1000),A(1000),H(1000,2),PR(1000)
8  dim R(2,2),N(16),HH(16,4),TR(16),D(16)
9  open "IPD.DAT" for output as #1
10 *****
11 '   Payoff Matrix
12 '   0: Player 1   1: Player 2
13 '   0: Defecting  1: Cooperating
14 '   ER: Probability of Making Error
15 *****
16 R(1,1)=5:R(1,0)=0:R(0,1)=6:R(0,0)=1:ER=0.15
17 *****
18 '   Number of Players for Each Strategy: N(I)
19 *****
20 for I=0 to 15
21   print using "Type ## = ";I,
22   input N(I)
23 next I
24 for I=0 to 15
25   print #I, N(I),
26 next I
27 print #I
28 *****
29 '   Initial "History" of Each Player: HH(I,J)
30 '   HH(I,0): H1=0  H2=0
31 '   HH(I,1): H1=0  H2=1
32 '   HH(I,2): H1=1  H2=0
33 '   HH(I,3): H1=1  H2=1
34 *****
35 HH(0,0)=N(0):HH(0,1)=0:HH(0,2)=0:HH(0,3)=0
36 HH(15,0)=0:HH(15,1)=0:HH(15,2)=0:HH(15,3)=N(15)
37 for I=1 to 14
38   HH(I,0)=int(N(I)/4)
39   HH(I,1)=int(N(I)/2)-int(N(I)/4)
40   HH(I,2)=int(N(I)*3/4)-int(N(I)/2)
41   HH(I,3)=N(I)-int(N(I)*3/4)
42 next I
43 *****
44 '   Iteration of Simulation
45 '   NN: Total Number of Players
46 *****
47 NN=0
48 for K=0 to 15
49   NN=NN+N(K)
50 next K
51 for K=1 to NN
52   A(K)=K
53 next K
54 for I=1 to 200
55 *****
56 '   Genotypes (Strategies)
57 '   0:"0000"  1:"0001"  2:"0010"  3:"0011"
58 '   4:"0100"  5:"0101"  6:"0110"  7:"0111"
59 '   8:"1000"  9:"1001" 10:"1010" 11:"1011"
60 '   12:"1100" 13:"1101" 14:"1110" 15:"1111"
61 *****
62 M=0

```

```

63   for NG=0 to 15
64     for NH=0 to 3
65       H1=NH * 2:H2=NH mod 2
66       while HH(NG,NH)>0
67         M=M+1:HH(NG,NH)=HH(NG,NH)-1
68         G(M)=NG
69         H(M,1)=H1:H(M,2)=H2
70       wend
71     next NH
72   next NG
73   *****
74   ' Sequences of Matches
75   *****
76   for J=1 to 50
77   *****
78   ' Matched Opponents
79   *****
80   for K=1 to NN/2
81     M1=NN+1-2*K :M2=int(M1*rnd(1))+1
82     Y1=A(M1+1) :Y2=A(M2)
83     A(M2)=A(M1) :A(M1)=Y2
84   *****
85   ' Match
86   *****
87     if H(Y2,2)=0 then P1=G(Y1) * 4 else P1=G(Y1) mod 4
88     if H(Y1,2)=0 then P2=G(Y2) * 4 else P2=G(Y2) mod 4
89     if H(Y2,1)=0 then P1=P1 * 2 else P1=P1 mod 2
90     if H(Y1,1)=0 then P2=P2 * 2 else P2=P2 mod 2
91     H(Y1,2)=H(Y1,1):H(Y2,2)=H(Y2,1)
92     H(Y1,1)=P1 :H(Y2,1)=P2
93     if rnd(1)<ER then P1=1-P1
94     if rnd(1)<ER then P2=1-P2
95     TR(G(Y1))=TR(G(Y1))+R(P1,P2)
96     TR(G(Y2))=TR(G(Y2))+R(P2,P1)
97   next K
98 next J
99 *****
100 ' Fitness
101 *****
102 TT=0
103 for K=0 to 15
104   TT=TT+TR(K)
105 next K
106 for NG=0 to 15
107   if N(NG)=0 then AR=0 else AR=TR(NG)/N(NG)
108   print using " ###.##";AR/(TT/NN),
109 next NG
110 print
111 *****
112 ' Rearrangement of Population
113 *****
114 for NG=0 to 15
115   for NH=0 to 3
116     HH(NG,NH)=0
117   next NH
118 next NG
119 for K=1 to NN
120   NH=H(K,1)*2+H(K,2)
121   HH(G(K),NH)=HH(G(K),NH)+1
122 next K
123 NT=0
124 for NG=0 to 15
125   D(NG)=TR(NG)/TT*NN
126   N(NG)=int(D(NG))
127   D(NG)=D(NG)-N(NG)
128   NT=NT+N(NG)
129   TR(NG)=0
130 next NG
131 while NT<NN
132   KK=0
133   for LL=1 to 15
134     if D(KK)<D(LL) then KK=LL
135   next LL
136   N(KK)=N(KK)+1
137   NT=NT+1
138   D(KK)=0
139 wend
140 for NG=0 to 15
141   if N(NG)=0 then
142     for K=0 to 3
143       HH(NG,K)=0
144     next K
145     goto *FEND
146   else
147     NHH=HH(NG,0)+HH(NG,1)+HH(NG,2)+HH(NG,3)
148     HH(NG,0)=int(N(NG)*HH(NG,0)/NHH)
149     HH(NG,1)=int(N(NG)*HH(NG,1)/NHH)
150     HH(NG,2)=int(N(NG)*HH(NG,2)/NHH)
151     HH(NG,3)=N(NG)-HH(NG,0)-HH(NG,1)-HH(NG,2)
152   end if
153 *FEND
154 next NG
155 for K=0 to 15
156   print using " ###";N(K),
157   print #1, N(K),
158 next K
159 print
160 print #1
161 stop
162 next I
163 stop
164 close #1
165 end

```