

人間同士による繰り返し型囚人のジレンマゲームの実施と結果

大澤博隆[†] 今井倫太[†]

本研究では、シミュレーション上で実施されることの多い繰り返し囚人のジレンマゲームを人間のプレイヤー同士で行わせ、各エージェントの戦略を、実際の人間プレイヤー同士にプログラムさせることで、ゲーム上でのエージェントの戦略の変化を調べる。これにより、実際の人間同士の集団ゲームにおいて、人間がどのように戦略を変化させていくか分析した。本研究では Axelrod の初回の大会の条件を元に、囚人のジレンマゲームの条件を設定し、大学の授業の課題として、授業参加者 72 人の人間を対象とし、25 日間の実験を行った。その結果、Axelrod の初回の大会や、シミュレーションの結果と異なり、上位のものほど記述が複雑化する傾向が発見された。また複数のプレイヤー同士が協力する「主人 - 奴隷戦略」、上記の戦術に対する寄生戦略、自身の勝利を目指さず、他プレイヤーを追い落とす戦略など、一見すると合理的でない、様々な戦略が確認された。

Analysis of Human-generated Strategies in the Iterated Prisoner's Dilemma Game

HIROTAKA OSAWA[†] and MICHITA IMAI[†]

1. はじめに

人間が集団で行動を行う場合、個々の最適な行動と全体としての利益が必ずしも一致しないことがある。このような問題を囚人のジレンマと呼ぶ。この問題のうち、一回だけの取り引きではなく、取り引きが複数回行われる場合には、前回の履歴を元に次の手を考えることが出来るようになる。これを繰り返し囚人のジレンマと呼ぶ。繰り返し囚人のジレンマはゲーム理論や経済学の基礎となる問題の一つである。本論文では、以降、繰り返し囚人のジレンマを IPD と略す。IPD を分析し、どのような解が望ましいかを求めるのは、重要な問題の一つである。

IPD の分析手法として、計算機上のシミュレーションが良く行われる。IPD を用いたゲームでは、Robert Axelrod が 1980 年に行った、計算機上での IPD 大会が有名である¹⁾。この大会では、TFT(しっぺ返し/TIT-FOR-TAT) という、相手の前回の手を真似る戦略が上位を占めたことで知られている。この種の IPD 大会はその後も継続的に行われている²⁾。しかしながら、この形式の大会では、エージェント同士の試合は一度しか行われず、各プレイヤーが他のプレイヤーの戦略

を調べ、それを元にエージェントの戦略を変更することは考慮されていない。

また、進化シミュレーションの手法を IPD ゲームに適用し、エージェントを改良していくことでどのような戦略が有用であるか検討する研究も行われている。Nowak らは、進化シミュレーションを行うことで、相手のエージェントの取った行動を一定確率で別のもので間違えてしまうようなコミュニケーションエラーのある条件下では、最終的に、前回の相手と自分の手が違った場合に相手を裏切る Pavlov 戦略が上位を占めるという結果を導いた⁴⁾。さらに、Beaufils らは、裏切りの回数を覚えておき、その回数と同じ回数だけ自分も裏切りを繰り返す Gradual 戦略が TFT や Pavlov よりも有効であると、理論とシミュレーション結果より指摘している⁵⁾。このように、各エージェントを進化させ、進化シミュレーション上でエージェント間の各戦略の変化を調べた研究は幅広く行われている。しかしながら、これらはいずれもシミュレーションの結果であり、人間のプレイヤー同士が長期間にわたり、お互いの戦略を改良させていく際に、プレイヤー同士でどのように戦略が選ばれていくかを分析した実験はまだ行われていない。

本研究では Axelrod の初回の大会の条件を元に、IPD ゲームの条件を設定した。そして、大学の授業の課題として、授業参加者 77 人の人間を対象とし、エー

[†] 慶應義塾大学理工学部
Keio University

表 1 得点表
Table 1 Iterated Prisoner's Dilemma scoretable

	B:協調	B:裏切り
A:協調	(A:+3,B:+3)	(A:0,B:+5)
A:裏切り	(A:+5,B:0)	(A:+1,B:+1)

エージェントのプログラムをオートマトンで書かせ、25日間の試合中にそのオートマトンを改良させることで、人間同士の勝負でどのような戦略の変化が起きるか検討した。

本論文では、これらの戦略がどのようにして誕生し、発展したか、ログをもとに詳細な分析を行う。本研究での成果を元にして、特に人間と勝負を行う戦略プログラムを設計する際、これらの設計を改良し、より人間の行動に基づいた、人間らしい動作を設計できるのではないかと考えられる。

本論文の構成は以下の通りである。2章では、背景となる IPD について詳しい説明を行う。3章では、Axelrod の試合を踏まえた実際のゲームの設計について述べる。4章では、設計したゲームを元に、シミュレーション上で行った予備評価について述べる。5章では、シミュレーションを元に実際に IPD ゲームを行った過程と結果について述べる。6章では、ゲームの結果に付いての考察を行い、7章で結論と今後の研究について述べる。

2. 背景

本章では、背景となる IPD について説明する。

IPD では、各々のエージェントが自分自身を含めた全てのエージェントと一回ずつ対戦し、一回の試合で複数回の勝負を行う。各エージェントは勝負のたびに協調か裏切りのどちらかの手を選ぶことができる。エージェントが両方とも協調を行えば、両方とも同じだけ点を得ることが出来るが、どちらかが裏切った場合、裏切った方のエージェントはお互い協調したときよりも高い点を得ることができ、裏切られた方のエージェントは点を得ることが出来ない。ただし、両方のエージェントが裏切りを選択する場合には、両者はお互い協調したときよりも低い点しか得ることが出来ない。Axelrod の設定した得点表を表 1 に示す。

IPD では一過性の囚人のジレンマと異なり、勝負が複数回行われる。このためエージェントは前回の相手の手を考慮し、将来的な結果を予想しながら次の手を選ぶことが可能となる。

3. 設計

IPD ゲームは以下の手順で行われる。まず、プレイ

ヤーは対応するエージェントを一体決め、各々のエージェントに戦略をプログラムする。エージェントは自分自身も含め、総当たりで試合を行い、一回の試合で複数回の勝負を行う。勝負の結果、点数表にしたがって各々のエージェントは得点を得る。最後にそれまでの試合で得た全ての得点を合計し、得点の高いものから順番に順位が付けられる。

1980 年に Axelrod が行った、初回の IPD コンテストと大きく異なる点は、以下の通りである。

- 順位決定の回数：毎日 5 回。合計 125 回
順位の決定する勝負は一般的な IPD コンテストと異なり、最後の 1 回だけでなく、毎日 9 時から 21 時まで 3 時間ごと、一日 5 回ずつ行われた。また、試合後の順位結果は全体に公開され、各プレイヤーは web ブラウザを通して、好きなタイミングで自分の戦略を変更することができた。これは、各プレイヤーに対し、常に現在の順位を確認させることで、エージェントの戦略を改良する継続的な動機を与えるためである。
- 参加者への報酬
プレイヤーの報酬となる成績点は順位順でなく、授業への加点の形で、上位 10 名に 30 点、そのほかの参加プレイヤーに一律 10 点という形で与えられる。これは、上位 10 名に対し均等に得点を割り振ることで、より多様な戦略が増え、各プレイヤーの改良が早まるのではないかと考えたためである。
- エージェントの記述方法
各エージェントは一般的なプログラミング言語ではなく、決定性有限オートマトンで記述させる。これは、課題の元になった授業が計算理論の授業であったためである。
- 一回の試合の勝負数：100 勝負
Axelrod による IPD ゲームは、第一回目が 200 回勝負、第二回目が勝負数不定 (0.00346 の確率で勝負が終了) であるが、オートマトンによる記述は一般的なプログラミング言語に比べ、自由度が低い。今回はオートマトンでの記述がいたずらに複雑になりすぎないように、試合ごとの勝負回数を固定し、減らした。
- 常に協調する戦略、常に裏切る戦略、交互に裏切る戦略、3 回に一度裏切る戦略、しつぱ返し戦略など、5 つのデモンストレーションオートマトンを導入する。
これは、特に初期の参加者間において、戦略が固定化されるのを防ぐためである。

3.1 カバーストーリー

オートマトン記述の際に、参加者に与えたカバーストーリーは、以下の通りである。

- 対戦型の貿易ゲームを遊んでもらう。複数の島に複数の国家が存在する状況で、各国家の「貿易戦略」をオートマトンで記述する
- 各国は、取引の際に「交易船」か「海賊船」を出すことができる
- 交易船同士が会会うと、交渉はうまく進み、お互いに利益を得ることができる
- 交易船と海賊船が会会うと、交易船を出した国家は損害を被り、海賊船を出した国家が利益を得る
- 海賊船同士が会会うと、お互いに潰しあって、両者に損害が残ってしまう

交易船の提示が IPD での協調行動にあたり、海賊船の提示が IPD での裏切り行動にあたる。

3.2 オートマトンの記述

オートマトンの記述法は以下の通りである

- オートマトンの入力：前回の相手の手 (交易船, 海賊船)
船:0, 海賊船:1
- オートマトンの状態：今回の自分の手 (交易船, 海賊船) 船:偶数番号 (0,2,4...), 海賊船:奇数番号 (1,3,5...)
- 状態はいくつ作っても良い (自分の過去の手を覚えることができる)

ユーザが入力するのは、最初に初期状態を入力し、それ以降に「今の状態, 入力, 次の状態」をカンマで区切って記述したテキスト文章である。下記がオートマトンを表すテキストの例である。このオートマトンは TFT 戦略 (しつぺ返し戦略, TIT-FOR-TAT) を意味する。

```
0
0 0 0
0 1 1
1 0 0
1 1 1
```

デモンストレーションのため導入したオートマトンを図 1 に示す。ID:robovie は全て交易船を返す戦略, ID:roboviem は全て海賊船を返す戦略, ID:robovie2 は、交易船からはじまり、交易船と海賊船を交互に出す戦略, ID:uni は、交易船からはじまり、三回に一回海賊船を出す戦略, ID:robovier は、Axelrod の IPD ゲームで優勝した、前回の相手の手を真似る TFT 戦

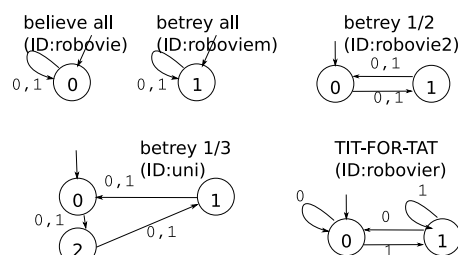


図 1 デモンストレーションのためのオートマトン
Fig.1 Demonstration automata

表 2 得点表
Table 2 Scoretable

	B:交易船	B:海賊船
A:交易船	(A:+100,B:+100)	(A:-150,B:+150)
A:海賊船	(A:+150,B:-150)	(A:-120,B:-120)

略である。

また、試合の際に得られる得点の表は表 2 のように設定した。

4. シミュレーションによる予備実験

上記の条件が適切であるか検証するため、本研究ではまずコンピュータのシミュレーションを行い、5種のデモンストレーションプログラムを入れた場合と入れない場合について検討を行った。

4.1 シミュレーション条件

シミュレーションでは、全て協調しか行わないオートマトン戦略を持った 100 個のエージェントをあらかじめ用意し、一回の試合ごとに総当たりで戦わせた。そして、下位 10 位のオートマトンに対し、オートマトンの長さ・初期条件の変更・飛び先の変更をランダムで 3 回行い、新しい世代のエージェントを作成した。このようにして、エージェントの遷移条件に変化を加え、100 万回の世代交代を重ねた。

シミュレーションの初期条件は、ランダムでの初期化値を変え、5 回の条件で行った。乱数生成アルゴリズムには Mersenne Twister⁶⁾ を使用した。

4.2 シミュレーション結果

シミュレーションの結果、デモンストレーションプログラムを入れた条件下では、しつぺ返しではなく「相手が協調であれば強調し、一度裏切りを受けたら、その後全て裏切りを返す」戦略が安定なることを確認した。これは、各人間のプレイヤーが IPD ゲームについて調査し、エージェントの戦略として単純な「しつぺ返し」を選択するだけでは最上位になれない可能性があることを意味する。

この結果より、このゲームが Axelrod の IPD ゲー

ムに基づくものとプレイヤーに気づかれたとしても、単純な TFT 戦略をプレイヤーが選ぶことはないだろうと判断し、次章からの実験を行うことにした。

5. 実験

以上のシミュレーション結果を元にして、課題のゲームを実装した。実装は CGI と JavaScript を用い、いわゆる AJAX³⁾ の形式で web アプリケーションとして実装した。

実装結果は下記アドレスの通りである。

<http://www.ayu.ics.keio.ac.jp/automaton2007/>

登録は ID 登録、戦略登録の順で行われ、ID 登録された時点でもっとも単純な、全て交易船を出すプログラムが自動的に登録された。

本研究では 72 人の人間のプレイヤーと 5 種のデモンストラーションエージェントを試合に参加させ、1 日 5 回の更新を行い、約 25 日間、合計 125 回の試合を行った。試合は 2007 年 6 月 8 日から 7 月 2 日まで行われた。参加したのは 77 エージェント (デモンストラーションエージェント含む) であった。

試合の期間中、プレイヤーは任意の時間にオートマトンを編集することが可能である。また、対戦したいオートマトンの ID を指定することで、任意のオートマトン同士を戦わせることが可能であった。ただし、相手のオートマトンの中身を見ることは不可能であった。

5.1 実験結果

最終試合後の各エージェントの順位表を表 3 に示す。要素のうち、記述量はオートマトンの行数、すなわち状態遷移の総数となる。全 77 のエージェントのうち、登録されただけで更新を行っておらず、全て交易船を出すエージェントは、デモエージェントを含め 9 体存在した。これらの順位は 60 位であった。また、TFT(しっぺ返し) 戦略を選択したエージェントはデモエージェントを含め 5 体存在し、この順位は 34 位であった。また、2 回に 1 回海賊船を出す戦略は、デモエージェントのみで 57 位、3 回に一回海賊船を出す戦略を選択したエージェントはデモエージェントを含め 3 体存在し、53 位であった。全て海賊船を出す戦略を選択したエージェントはデモエージェントを含め 3 体存在し、順位は 73 位であった。なお、点数が同順位になった場合には、記述量の少ないもの (= 冗長でないもの) を上位に置く。

また、平均得点・最大得点・最小得点の推移の図を図 2 に示す。横軸は試合回数、縦軸は得点を表す。

表 3 順位表

Table 3 Rank

順位	名前	得点	平均点	記述量
1 位	AGENT01	605870	7971.9	386
2 位	AGENT02	583670	7679.8	258
3 位	AGENT03	524750	6904.6	1776
4 位	AGENT04	517490	6809	498
5 位	AGENT05	509890	6709	250
6 位	AGENT06	506930	6670.1	558
7 位	AGENT07	505220	6647.6	186
8 位	AGENT08	502560	6612.6	1292
9 位	AGENT09	502210	6608	488
10 位	AGENT10	500520	6585.8	72
11 位	AGENT11	500470	6585.1	492
12 位	AGENT12	499160	6567.9	588
13 位	AGENT13	493580	6494.4	416
14 位	AGENT14	489880	6445.8	439
15 位	AGENT15	489570	6441.7	560
16 位	AGENT16	484950	6380.9	300
17 位	AGENT17	484110	6369.8	280
18 位	AGENT18	475230	6253	786
19 位	AGENT19	473650	6232.2	490
20 位	AGENT20	464060	6106	20
21 位	AGENT21	463450	6098	314
22 位	AGENT22	462860	6090.2	108
23 位	AGENT23-24	461950	6078.3	132
25 位	AGENT25	452970	5960.1	224
26 位	AGENT26	429230	5647.7	216
27 位	AGENT27	427250	5621.7	22
28 位	AGENT28	418410	5505.4	48
29 位	AGENT29	412160	5423.1	212
30 位	AGENT30	381220	5016	48
31 位	AGENT31	373260	4911.3	57
32 位	AGENT32	361630	4758.3	396
33 位	AGENT33	360390	4741.9	401
34 位	robovier	359240	4726.8	4
34 位	AGENT34-37	359240	4726.8	4
39 位	AGENT38	324740	4272.9	12
40 位	AGENT39	313520	4125.2	8
41 位	AGENT40	294940	3880.8	6
42 位	AGENT41	291790	3839.3	0
43 位	AGENT42	269580	3547.1	34
44 位	AGENT43	253750	3338.8	12
45 位	AGENT44	226340	2978.1	394
46 位	AGENT45-46	224460	2953.4	4
48 位	AGENT47	223410	2939.6	4
49 位	AGENT48	211940	2788.6	14
50 位	AGENT49	189040	2487.3	281
51 位	AGENT50	28540	375.5	198
52 位	AGENT51	25880	340.5	16
53 位	uni	-89010	-1171.1	6
53 位	AGENT52-53	-89010	-1171.1	6
56 位	AGENT54	-107380	-1412.8	6
57 位	robovie2	-131800	-1734.2	4
58 位	AGENT55	-238540	-3138.6	6
59 位	AGENT56	-241110	-3172.5	6
60 位	robovie	-299750	-3944	2
60 位	AGENT57-64	-299750	-3944	2
69 位	AGENT65	-306390	-4031.4	23
70 位	AGENT66-67	-380680	-5008.9	18
72 位	AGENT68-69	-380680	-5008.9	22
74 位	roboviem	-390090	-5132.7	2
74 位	AGENT70-71	-390090	-5132.7	2
77 位	AGENT72	-395500	-5203.9	22

6. 考察

大きな傾向として、30 位までのエージェントでオートマトンの記述量が長くなる傾向が見られ、30 位から 57 位までのエージェントは TFT かその改良型、それ以降のものはあまり改良を行っていない傾向がみら

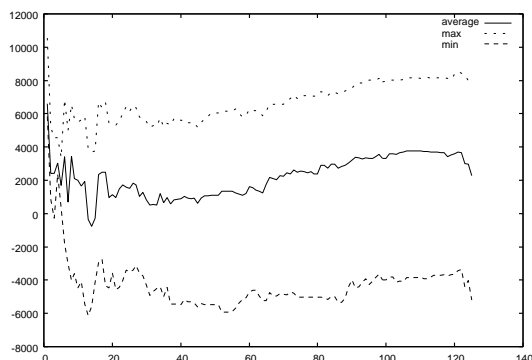


図2 平均・最大・最小得点の推移
Fig. 2 flow of average/max/min score

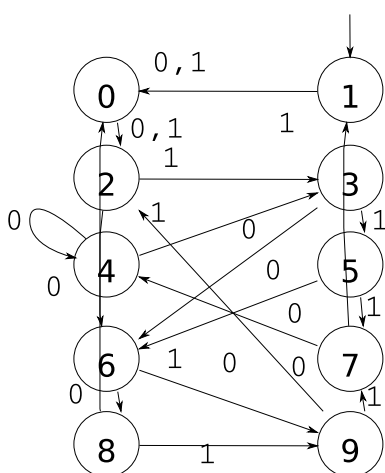


図3 10状態のオートマトン (AGENT20)
Fig. 3 automaton constructed by 10 states(AGENT20)

れた。これは、表3のオートマトンの記述量からもある程度傾向として見られる。下位の順位のものほど、記述量が少なく、エージェントの改良に対し熱心でなかったことが伺える。これは、30位までのものは得点が得られるため、プレイヤーにとってオートマトンを改良する動機が大きかったからだと思われる。一例として、上位30位のうち、状態数が10状態ともっとも少なかったオートマトンの例を図3に示す。この作戦は20位であった。このように、状態数が少ないものでさえ複雑な挙動を示している。

Axelrodの研究では、自分からは裏切らないという性質を持った、NICEと呼ばれるプログラムが上位にあがるということが指摘されている。さらに、優勝した戦略はもっとも単純な戦略であり、そのほかの戦略同士には、順位とプログラムの量に相関関係が無いことが指摘されている。また、シミュレーションによって得られた結果もそれを支持している。

しかし、今回の結果はAxelrodのコンテストとも、シミュレーションによる予備実験とも、極めて異なったものとなった。上位30位のうち、NICEなプログラムは1つしか無く、残りの29エージェントは自分から積極的に裏切る性質を持っていた。また、上位のエージェントほど、プログラムの記述量が多くなる傾向がはっきり見られた。

これは、登録しただけで改良を行っていないエージェントが多く、積極的に裏切りながら利益を得ようとするエージェントの方が、協調を量るエージェントよりも得点が多かったからと思われる。単純なモデルのゲーム理論では、全てのエージェントが合理的に行動することが仮定されるが、特に人間を対象にした実際の状況では往々にして、エージェントが非合理的に行動することがありえる。この現象がこの状況でも観察できる。

したがって、もしこのようなIPDゲームにおいて、人間より近い振舞いをする人工的なエージェントを作成する場合は、各エージェントに対し、目標への合理的行動の強さ、つまりエージェントの賢さを段階的に設定する必要があると考えられる。

また、いくつかの発展的な戦略のあったことが、ユーザとの対話により分かった。それらについて、以下に詳しく記述する。

6.1 主人-奴隷戦略および寄生戦略

主人-奴隷戦略は、2004年のIPDコンテスト²⁾でSouthampton大学のNick Jenningsらが採用した方法である。この手法では、主人役と奴隷役のエージェントが協調し、最初の数回の取り引きを情報通信手段として使い、あらかじめ決められた手順で協調・裏切りを行う。主人役のエージェントは、決められた手順から奴隷役のエージェントを発見すると勝負が終わるまで裏切り続けるようになり、奴隷役のエージェントは主人を見付けると協調を行い続ける。また、奴隷役のエージェントは主人以外に対して全て海賊船を送ることで、主人以外の相手の点数を可能な限り下げる。

Jenningsらは1チーム・60のエージェントを用意する必要があったが、今回の試合でこの戦略を採用したのは2チーム・それぞれ3つ・合計6つのエージェントであった。主人となったのは、AGENT01,AGENT02であり、奴隷となったのはAGENT66,AGENT67と、AGENT68,AGENT69である。Jenningsらの例と異なり、今回は搾取可能なエージェントが多かったため、多数の奴隷エージェントを揃えなくても順位をあげることが可能となっている。

興味深いことに、AGENT66-67とAGENT68-69

は中身が違うオートマトンである。これらは、以下のように最初に信号を受け取り、この信号を発したものを『主人』と判断して自分が奴隷側に回って搾取される体制をとる。

- AGENT66-67:
開始 0 受信 1/送信 1, 受信 0/送信 0, 受信 0/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0
- AGENT68-69:
開始 0 受信 1/送信 1, 受信 0/送信 0, 受信 0/送信 0, 受信 1/送信 1, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0, 受信 1/送信 0

AGENT01 と AGENT02 は、これらの奴隷を共に利用できるが、この 2 者も異なるオートマトンであった。どちらかの主人に当たるエージェントが、もう一方の戦略に対し、寄生行為を行っていた可能性が考えられる。

このような主人 - 奴隷戦略は、奴隷側のエージェントにあまり利点がないため、発生しないと考えていた。しかしながら、実際には単独で非合理的なエージェントが発生している。この理由として、ゲーム外で奴隷エージェントを作成したプレイヤーに利得があった可能性が考えられる。

6.2 追い落とし戦略

ユーザとの対話により、エージェントの中に、自らの勝利を捨て、上位のエージェントを追い落とすことのみ集中した

51 位の AGENT50, 52 位の AGENT51, 69 位の AGENT65 がこの戦略を採用したとして報告されている。たとえば 69 位のエージェント AGENT65 は、最初の数回で全て海賊船を出し、交易船を出しながら相手の様子を探ろうとする高度な戦略から搾取を行っている。上位陣の中では、このような追い落とし戦略をどのように見分け、防ぐかが重要であったと、該当するユーザは述べている。

このような追い落とし戦略と似た戦略に、Axelrod の論文⁷⁾ に登場する DOWNING と呼ばれる戦略がある。この戦略は Kingmaker と呼ばれ、それ自身が勝利することはなかったものの、搾取可能なものから搾取を行い、そうでないエージェントとは協調するという作戦を持ち、上位と下位のエージェントを振り分け、結果的に TFT 戦略を優勝させる役目を持っていた。今回の追い落とし戦略の動機として、このような Kingmaker を自らつくり出す意図があったことも考えられる。

6.3 全体の傾向

全体の得点の推移は図 2 の通りであり、初期は平均得点が 6000 近くあるものの、すぐに 2000 近くまで下がってしまっている。ただし、それからは徐々に平均得点が回復する傾向がみられる。また、最終試合の数回前に、突然得点の低下傾向が見られるが、これは、その時点で始めてオートマトンを記述し、海賊船を出すことに決めたり、自らの利益を増やすことを求め、海賊船を提出する回数を増やしたためと思われる。

6.4 次回実験への検討

今回のゲームで観察された参加者の行動として、ゲームの更新前までダミーのオートマトンを用意し、更新直前にこれをすり替える行為があった。このような手段が行われると、各プレイヤーに十分な検討が行えなくなってしまう。この手段を防ぐ方法として、今回は、更新時のオートマトンを保存し、このオートマトンと自分で編集したオートマトンを戦わせるよう、ゲームの仕様を変更する予定である。

さらに、今回の条件では授業に対する加算点が、プレイヤーに対して与える利得が不明確であったため、主人 - 奴隷関係のような、単独行動ではありえない協調行為が発生した。この点に関し、今後プレイヤーにどのような利得を与えるか考えることでより正確な実験を行いたいと考えている。

また、今回設定したオートマトンによる記述では、自由変数を使いがたいため、複雑な条件を記述できない。たとえば、相手の裏切りの回数分だけ自分が裏切る、Gradual 戦略⁵⁾ のような戦略を記述するのは複雑である。よって、今後実験を設定する際には、より自由度の高いプログラミング手法を取り入れることで新しい戦略を発見できる可能性がある。

7. 結 論

本研究では、72 人の人間を対象とし、エージェントのプログラムをオートマトンで書かせる IPD ゲームを行うことで、人間同士の勝負でどのような戦略の変化が起きるか検討した。本研究で使用した IPD ゲームは、一定時間ごとに試合が行われることで、人間が改良する余地がある。その結果、戦略の賢さに差が出来、合理的でないエージェントを搾取するため、上位の戦略が複雑化し単純なしっぺ返し戦略が上位に来ないことが発見された。また、主人 - 奴隷戦略やそれに対する寄生戦略など、他のプレイヤーと密接にコミュニケーションを取る戦略も見られ、奴隷側、追い落とし戦略など、合理的でない目標を独自に発見し、それに向けて改良を行うプレイヤーも見られた。

本研究の成果により、人間と似たエージェントを設計するには、合理的な行動に対する欲求を、エージェント毎に変えることが望ましいのではないかと思われる。また、設定したゴール以外のゴールを自動的に設定し、それに向かって改良を行う人間の性質も発見できた。将来的には、全てのプレイヤーが与えられたゴールに満足せず、自ら新しいゴールを設定すると言う、メタゲームの要素をどのようにゲーム理論で扱うかが今後重要になるとと思われる。

今後は、今回のゲームで得られた評価を元に、より厳密なゲームの定義を行い、シミュレーション上のエージェントと人間の行動で何が異なるか、ひきつづき実験を行いつつ検討していく予定である。

8. 謝 辞

IPD ゲームのバージョンテストに参加してくれた研究室の諸氏、計算理論の IPD ゲームに参加してくれた、慶應義塾大学情報工学科の学生諸氏、特に、ゲームの詳細な感想を送ってくれた野口大介君に、この場を借りて謝意を表します。

参 考 文 献

- 1) Axelrod, R.: *The Evolution of Cooperation*, Basic Books, New York (1984). (松田裕之訳: つきあい方の科学, CBS 出版 (1987)).
- 2) The Iterated Prisoner's Dilemma Competition: Celebrating the 20th Anniversary. <http://www.prisoners-dilemma.com/> (2004.)
- 3) Jesse James Garrett, Ajax: A New Approach to Web Applications <http://www.adaptivepath.com/ideas/essays/archives/000385.php> (2005.)
- 4) Nowak MA, Sigmund K : *A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game*. Nature 364: pp. 56-58.
- 5) Bruno Beaufils, Jean-Paul Delahaye and Philippe Mathieu: *Our Meeting with Gradual: A Good Strategy for the Iterated Prisoner's Dilemma*. Proceedings of the Fifth Int'l Workshop on the Synthesis and Simulation of Living Systems, pp.202-209, MIT Press, 1997.
- 6) M. Matsumoto and T. Nishimura, Mersenne twister: *A 623-dimensionally equidistributed uniform pseudorandom number generator*, ACM Trans. on Modeling and Computer Simulations, 1998.