

松 山 大 学 論 集
第 34 卷 第 2 号 抜 刷
2 0 2 2 年 6 月 発 行

Axelrod 実 験 再 考

安 田 俊 一

Axelrod 実験再考*

安 田 俊 一

1 はじめに

ゲーム理論の分析に遺伝的アルゴリズムを持ち込み、戦略進化の研究を始めたのはおそらく Axelrod (1987) (以下「Axelrod 原論文」) が最初である。

Axelrod についてはその前の氏の論文 (Axelrod (1980a, b)) でのコンピュータトーナメントが有名で、「Tit-For-Tat」(「しっぺ返し」, 以下「TFT」) が囚人のジレンマゲームでの有効な戦略であるとの結論を引き出し、この戦略を有名なものにした。

筆者も一連の遺伝的アルゴリズムと囚人のジレンマゲームの分析において、Axelrod 原論文を検討したことがある (安田俊一 (2001), 以下「前論文」)。ここでは Axelrod 原論文が持つ遺伝的アルゴリズムのコーディングの問題をとりあげ、その解決として「ダーウィンの進化」を反映した方法を検討した。

前論文でも指摘したが、Axelrod 原論文はかなり有名で引用も多いが、彼が用いた遺伝的アルゴリズムの方法自体を検討した論文は筆者の調べた限りではない。

コンピュータトーナメントを含む Axelrod の論文に関しては、ゲーム理論家の Ken Binmore がかなり全面的な批判を行った (Binmore (1994, 1998)) が、主たる批判の対象は Axelrod 原論文の前半にある彼の前著で取り扱ったコンピュータトーナメントの理論的な検討であり、遺伝的アルゴリズムに関する部

* 本稿は 2020 年度松山大学特別研究助成の成果である。

分については対象になっていない¹⁾

Axelrod 原論文は筆者にとって遺伝的アルゴリズムによる囚人のジレンマゲームに関する一連の研究の契機となった論文であり、そこでおこなわれたシミュレーション自体を追試し検討を加えることは必要であると考えていた。前論文の主眼は上述のように Axelrod のコーディングの問題点を取り上げた上で、別の枠組みを提供して比較することになり、Axelrod の実験自体を深く追跡したものではない。

本論文では前論文から現在までの研究の進展や（シミュレーションプログラムを含めた）到達点を踏まえた上で Axelrod が行った実験を再検討することを目的とする。したがって以下では Axelrod 原論文の枠組みをそのまま踏襲したシミュレーション結果を分析する。その結果、（やはり前論文で指摘したように）Axelrod 実験では遺伝子型のコーディング方法が本質的な影響を与えており、彼が行った遺伝的アルゴリズムの結論はおそらく彼が意図したものとは違う結論をもたらすことを明らかにする。

2 Axelrod 実験概要

前論文と重複することになるが、後の議論の必要上、簡単に Axelrod 原論文で行われたシミュレーションを概観しておく。

- 囚人のジレンマゲームにおける「協力（沈黙）」を「0」, 「裏切り（自白）」を「1」と表し、'0', '1' からなる 70 ビットの長さの配列（バイナリ配列）として染色体を定義する。
- 1 回のゲームで「自分の手」「相手の手」が示されるので、その 1 回のゲームの記録は 2 ビットの長さの '0' '1' の組み合わせで表現される。

1) Binmore (1998) では「(別の人物が) 実験を徹底的にやり直して Axelrod の問題点を明らかにしている」旨の記述があり、その部分の Reference に “Forthcoming” として論文が挙げられているが、筆者が調べた限りでは発刊されていない。

1回のゲームの履歴は「自分の手、相手の手」として記録され、例えば“00”は両者が協力したことを示し、“01”ならば「自分は協力したが、相手は裏切った」ことを示す。

- 70ビットの染色体のうち、最初の6ビットを「それまでに行われた直近3回のゲームの記録（これを「**記憶領域**」と呼んでおこう）」として、その6ビットのバイナリ配列を2進数とする。その上で、その2進数を10進数に変換した数、例えば記憶領域が“000000”であれば10進数の「0」、 “001110”であれば「14」、 “111111”であれば「63」を対応させる。この数値はその個体の「次の手」を決定するのに使われる。
- 70ビットの記憶領域以降の64ビットの配列（これを「**戦略領域**」と呼んでおこう）は、記憶領域である6ビット2進数が示す10進数の範囲0から63に対応しており、「3回のゲームの履歴が示す箇所にある‘0’‘1’」がその個体の「次の手」として使われる。例えば記憶領域が“001010”であれば、戦略領域の10ビット目にある文字が次のゲームで出される「手」となる。

以上の巧妙ともいえるコーディングによって、Axelrodはゲームの1プレイヤーを1染色体として表現することに成功した。

こうして遺伝子型（染色体の遺伝子配列）と表現型（ゲームで出される「手」）が決まったので、1染色体（＝個体）を集めた集団を作り、その中で相互に囚人のジレンマゲームを行った。

Axelrodは染色体で表された個体を20個つくり、まず前半でコンピューターナメントで取り扱った8つの戦略と151回のゲームを行い、その平均利得が高い個体を「親」として選択（親選択）、交叉と突然変異の処理をして次世代の集団をつくる遺伝的アルゴリズムを用いて、どのような成績をあげるかを検討した。Binmoreたちゲーム理論家による批判的検討が行われたのはこの部分である。つづいて、対戦相手を固定せずに、集団の中だけでゲームを行ったの

が後半部分になる。

染色体についての以上の定式化からわかるように、Axelrod のシミュレーションは、なんらかの戦略を（固定的に）維持して行動するわけではなく、「直近3回のゲームの履歴」に応じて「次の手」を決めているだけだ。したがって戦略領域にあるパターンが「戦略」ということもできるが、そのパターンは 2^{64} 通りあることになる²⁾。非常に多くのパターンがあるので、配列パターンを用いて何らかの戦略へ分類できるわけではない。前論文でふれたように、「TFT」についてはパターンを特定化できないわけではないが、70ビットの配列が交叉・突然変異によってかき乱される環境ではそのような特定のパターンは検出できない。さらに Axelrod は記憶領域も染色体に含めているため、ゲームのたびに变化する履歴も混じるため染色体パターンによる戦略の分類は現時点では不可能であると思う。

Axelrod は20個体のランダムに発生させた集団によるシミュレーションを10回行って、おおよそ以下のように結論づけている。すなわち、集団は最初完全な裏切り戦略が支配的になることによって集団の平均利得は減少するが、10世代か20世代がすぎると互恵的な戦略をもった個体の利得（すべてのゲームの平均利得）があがるため、徐々にその戦略が集団内に拡散し、互恵的な個体が増えるため集団の平均利得が上がっていくという結論である。

Axelrod はこれ以上詳しい結果を述べていないため、シミュレーションで確認すべき基準となる尺度が示されていない。そこで Axelrod 原論文のシミュレーションをより詳細に進めていくために、以下の指標を定義してシミュレーションを行う³⁾。

2) Axelrod は原論文において、この戦略空間を 2^{70} 通り、としている。これは彼がここでいう記憶領域を染色体に加えており、染色体全体のパターンがそれぞれの戦略であると見なしているからである。

3) 以下のシミュレーションに用いたプログラム言語は Java である (openjdk17.0.1)。標準ライブラリのみを用いて、特殊なライブラリは使用していない。シミュレーションで使ったソースコードは Github で公開している (<https://github.com/shunichiYasuda/AxelrodExperiment2022>)

協力・裏切りへの収束 互恵的な個体かどうかはともかく、集団の平均利得があがっていくにしても、どの程度上がれば、その集団は“協力する個体が多い”といえるのか、その基準を予め定めておく。この基準に関してはこれまで筆者がおこなってきたシミュレーションで用いた、「平均利得 2.76 以上なら協力への収束」とする。これは集団内個体の 8 割が「協力（‘0’を出す）」戦略をとっていることを意味する。

もちろん、協力への収束は一時的（例えば 1 世代でだけ、など）発生することもあり、ある程度の持続性がみられた場合にのみ「集団が協力へ収束した」と判定する。「ある程度」の基準は、協力への収束が連続する 10 世代のうち 6 世代以上でみられることとする⁴⁾。

「裏切りへの収束」も同様に定義でき、「平均利得 1.54 以下なら裏切りへの収束」とする。これは集団内個体の 8 割が「裏切り（‘1’を出す）」戦略をとっていることを意味し、「協力への収束」基準と同様に、連続する 10 世代のうち 6 世代以上で裏切りへの収束が起きた場合に「裏切りへの収束」と定義する。

協力の平均持続期間 各実験で上で定義した「協力への収束」が持続した最大期間の平均である。協力へ収束しても実験の中には長い間それが持続したのものもあれば、あまり維持できず、すぐに協力が崩壊する実験もある。この指標は各実験における最も長い協力期間を平均したものであり、協力の持続性を示す。

平均的な最初の協力世代 ランダムな初期状態から出発した集団が、最初に協力への収束を達成した世代を平均したもの。どの程度速やかに協力へ達成したかを示す。

今回のシミュレーションの目的はあくまでも「Axelrod 実験の再考」であるか

4) 安田俊一（2005）以降、囚人のジレンマゲームにおいて集団が協力を達成したとみなす基準として筆者は用いている。

ら、表現型、遺伝子型の設計は上記で説明した Axelrod の方法に従う。その上で、いくつかのパラメータを変化させてこれらの指標を参照しながら、シミュレーション結果を分析していくことになる。

遺伝的アルゴリズムでは、優秀な成績を上げた個体を、その個体をもつ染色体を次世代へ残す「親」として選択する。この「親」選択にあたって、一般的な遺伝的アルゴリズムでは「スケーリング⁵⁾」を行うが、Axelrod 原論文にはその記述はないため今回は使用しない。また、同様に「エリート戦略⁶⁾」も用いない。

四人のジレンマゲームの利得については、Axelrod 原論文で用いられている数値と同じであり、各個体は総当たり戦を行って、それぞれ 151 回繰り返してゲーム⁷⁾を行い、その平均利得を個体が得た利益としている⁸⁾。本稿のシミュレーションもそれにならう。

3 交叉確率と突然変異確率

異性生殖をモデルとするときには、親となる 2 個体の染色体から子となる 2 個体の染色体が作られる場合に「交叉」が行われる。遺伝的アルゴリズムにおいてはこれは基本的な操作であって、「交叉を起こすかどうか」を決定する確率は重要なパラメータである。また、突然変異は親から子へ染色体（の一部）がコピーされるときに、どこかの遺伝子座にある遺伝子に変化を起こすことで

5) 利得の差が極めて小さいと、親として選択する際に不都合が生じるため、各個体間での利得の差を“拡大”する変換を行うこと。一般的には「線形スケーリング」が用いられる (Goldberg (1989))。

6) もっとも優秀な成績を挙げた個体の遺伝子を確実に次世代に残すために、最大利得を挙げた個体の染色体は必ず「親」として採用しておく手法。

7) 四人のジレンマゲームでの利得はよく使われるもので、(協力, 協力) ならば (3, 3), (協力, 裏切り) ならば (0, 5), (裏切り, 協力) ならば (5, 0), そして (裏切り, 裏切り) ならば (1, 1) である。

8) 151 回、という回数は Axelrod 原論文の前半で行われた「代表的な 8 つの戦略」との対戦で使われた回数である。集団の個体同士の対戦については回数の記述はないが、文脈から同じ回数であることは明白である。

ある。ここでの文脈で言えば、染色体である長さ 70 のバイナリ配列のどこかが、'0' から '1'、あるいは '1' から '0' へと「反転する」ことを意味するため、各遺伝子座がどの程度の確率で反転するのかを示す確率も重要なパラメータである。

Axerlod 原論文ではこれが実は曖昧であって、「各染色体、各世代 (per chromosome per generateion)」で「平均的に 1 回の交叉と 2 分の 1 の突然変異 (averaging one crossover and one-half mutation)」となっているが、交叉確率は「その染色体が交叉を起こすかどうか」であるので、各世代で各染色体が 1 回の交叉を起こすというのは、「各世代ですべての染色体が交叉を起こす」ともとれるが、「各世代で平均的に 1 回」ならば、「各世代で平均的に 1 個体が交叉を起こす」と見なした方がつじつまが合うようにみえる。

また、突然変異を「平均的に」述べるのであれば、反転する遺伝子座の数を指定しなければならないが、「各染色体、各世代で平均的に 2 分の 1」の確率でどの程度の反転が起きるのか曖昧である。

もしこれらの交叉確率、突然変異確率の違いが、シミュレーション結果に大きな影響を及ぼすのであればこれらの変化をシミュレーションで確認しなければならない。

そこで、先にこの点を確認しておくために、世代数と集団内の個体数は Axerlod が設定したのと同じく「50 世代、20 個体」として交叉確率と突然変異を以下のケースに想定して結果を比較しよう。

交叉確率 各世代につき、平均的に 1 個体、10 個体、すべての個体（つまり交叉確率 100%）が交叉を起こす 3 ケース

突然変異確率 各世代ですべての個体が平均的に 2 分の 1 回突然変異を起こす（ここまでは Axerlod の記述と同じ）として、突然変異を起こす場合は 70 ビットのうち、平均的に 1 箇所が反転するケース、同様に 3 箇所、7 箇所が反転するケースの 3 ケース

以上のケースについてそれぞれシミュレーション結果が以下である。

表1 収束率

交叉確率	平均1カ所	突然変異確率	
		平均3カ所	平均7カ所
全個体	509	578	491
平均10個体	469	552	435
平均1個体	573	598	603

表2 平均持続期間

交叉確率	平均1カ所	突然変異確率	
		平均3カ所	平均7カ所
全個体	23.08	22.36	21.84
平均10個体	22.9	23.06	23.6
平均1個体	23.92	25.3	24.35

表3 平均初収束世代

交叉確率	平均1カ所	突然変異確率	
		平均3カ所	平均7カ所
全個体	26.54	27.13	28.14
平均10個体	26.69	26.81	26.38
平均1個体	25.43	24.08	25.15

表1は上述の「協力への収束」がみられた実験を100回得るために必要だった実験回数である。協力がどの程度起こりやすいかの指標（以下「収束率」）とする。数が大きいほど収束しにくい、つまり収束率が悪いことに注意しておく。表2は協力の状態がどのくらい継続しやすいかを示し、表3は協力への達成の早さを示す。

以上の結果を見るとどの指標についてもほとんど傾向はみられない。強いていうなら収束率に関しては交叉確率を「平均的に10個体」としたときがもっ

とも高いように見えるがそれにしても違いは5%未満でしかない。したがって、Axelrod 実験の枠組みでは、交叉確率と突然変異確率の違いはシミュレーションの結果に大きな違いをもたらすことがないことが確認できる⁹⁾

以下では交叉確率を「各世代平均的に1個体が交叉（一点交叉）を起こす」、突然変異確率を「各世代・各個体が確率2分の1で突然変異を起こす。突然変異が起こる場合は平均的に染色体の1カ所が反転する」ものとしてシミュレーションを構成する¹⁰⁾

4 個体数と世代数の影響

交叉確率と突然変異確率は以上のように設定したので、ここから Axelrod 実験における他のパラメータ、集団内の個体数と世代数がどのような影響を与えるか検討しよう。

表4 収束率

世代数	個体数		
	20 個体	50 個体	100 個体
50 世代	573	671	717
100 世代	522	422	396
400 世代	412	368	354

表4は上で定義した収束率の一覧である。だいたいにおいて、表の各列を見ていくと、個体数が同じであれば世代数が長くなるにつれて収束率は上がっていく。これは別に不思議なことではなくシミュレーションで用いる世代数が増

9) ただし、後述のように交叉確率は染色体の収束スピードに影響を与える。

10) この機会に、筆者の前論文での記述について触れておく。「5 シミュレーション結果」(p 49)において、筆者は「Axelrod 実験と同じく集団数20, 交叉率0.25, 突然変異率0.01…」と書いているが、Axelrod 実験と同じなのは集団数だけであり、この記述は誤解を招く。本文で述べたように交叉率と突然変異率については Axelrod 実験では曖昧である。ちなみに「突然変異率0.01」は「各遺伝子座が0.01の確率で反転する」との意味であり、前論文でのシミュレーションではそのように設定した。

えれば、集団が「協力へ進化するチャンス」が増える（1回のシミュレーションで進化のためにかける期間が長くなる）からだ。

また、「50世代、50個体・100個体」の欄は少々異なるが、同じ世代数で見れば個体数が増えれば収束率が高くなっている。これも、個体数が増えれば、初期に用意された染色体の多様性が増すため、協力へ進化するチャンスはやはり増えると考えれば説明はつく。そうなると、「50世代、50個体・100個体」での収束率はこれに反しているようにみえるが、この場合はむしろ「世代数の短さ」が問題になっていると考えられる。つまり、50世代という短い時間では、個体数が増えれば多数の個体が協力へ収束するには短すぎたのだと考えることができる。

続いて表5は平均持続期間である。各行を見ていくと、個体数が増えるに従って協力の平均持続期間は短くなっている。これは個体数が増えれば収束へ到達する期間が長くなり、その結果持続期間も短くなると解釈できる。

この点は次の「平均初収束世代」も含めて検討してみよう。

表5 平均持続期間

世代数	個体数		
	20 個体	50 個体	100 個体
50 世代	23.92	19.07	15.17
100 世代	67.95	57.75	50.39
400 世代	366.98	328.58	312.45

表6 平均初収束世代

世代数	個体数		
	20 個体	50 個体	100 個体
50 世代	25.43	30	33.92
100 世代	27.59	40.14	46.96
400 世代	28.66	49.79	64.24

表6は「協力へ到達する早さ」を示しているが、これは個体数が増えるほど遅く、世代数が長くなるほど遅くなっている。

個体数が増えれば協力への収束が遅くなるのは理解できる。染色体のタイプが多ければ、協力へ収束する可能性は高くなるので収束率はあがるが、多様性が大きいゆえに、協力の状態へ到達するには時間がかかるようになるからだ。

一方、世代が増えるに従って平均初収束世代が遅くなるのは一見説明がつかない。世代数自体は実験者の側が決めた期間であって、個体数が固定されていれば「いつ協力への収束に達成するか」は変わらないはずだからだ。しかし、表4の収束率で見たように、世代数が長いということは協力へ収束するチャンスが増えるということであり、それは「ずっと後の世代になって初めて収束する」ケースが発生することを意味する。そのようなケースでは当然「初収束世代」は大きな値を取るため、世代数が増えれば平均的な初収束世代は大きな値をとる（つまり、「遅く」なる）。

表5の各列をみると、世代数が増えるに従って協力持続期間が延びていることがわかる。これは一見上の理屈と矛盾するように見えるが、そうではない。

世代数の変化は実験者の側で任意に設定した値で例えば100世代と400世代では4倍の差がある。それに対して世代数の違いによる平均初収束世代の違いは大きくて3割程度である。つまり、世代数の差と比べると平均初収束世代はほとんど差があるとはいえず、平均初収束世代はかなり固定的なものだと見ることができる。

そう考えると表5における、400世代が示す非常に長い平均持続時間も説明がつく。平均初収束世代はある程度固定的なので、いったん協力への収束がおこったら、その後は世代がつづく期間だけ、協力への収束状態が続く、ということだ。Axelrodの方法ではいったん協力へ、あるいは裏切りへ集団が収束するとその状態は崩壊しない。このことはAxelrodの染色体の設計に深く関わっており、以下で順次検討していく。

さて、以上みたように、Axelrodのシミュレーションは個体数と世代数パラ

メータの変化に関してもおおきな違いがない安定的な結論をもたらす。以下に上の結論を簡単にまとめておこう。

- 世代数を固定して個体数を増やしていけば、収束率は増大し、平均初収束世代は遅く、平均持続期間は短くなる。
- 個体数を固定して世代数を増やしていけば、収束率は増大し、平均初収束世代は遅くなるが、平均持続期間には本質的な影響を与えない。このことはいったん達成された協力への収束が崩壊しないことを意味している。

これで遺伝的アルゴリズムで重要な代表的パラメータの違いによる結論の違いがわかったので、以下のシミュレーションでは、100世代・50個体のケースをとりあげ、シミュレーションで起こっていることを詳しく分析しよう。

5 個体の性質を識別する

Axelrod は原論文での結論として「互恵的な個体が増えていく（だから集団平均値が上がっていく）」と述べているが、論文の中でそのことを数値的・視覚的に表してはいない。そこで Axelrod 実験の枠組みの中で、「互恵的」とか「裏切り (mutual defect)」とかの「個体の性質」を示すことを考えよう。

筆者は安田俊一 (2014) で、Nowak and Sigmund (1992) が用いた確率的な戦略を組み込んだ囚人のジレンマゲームシミュレーションにおいて、個体の性質を座標領域を用いて識別した。今回用いている遺伝子型の枠組みでは、同じように「座標領域」を用いての設定はできないが、記憶領域のパターンから「個体の性質」を同様に分類することが可能である。

記憶領域の6ビットは直近3回のゲームの履歴であるから、相手の手に対応する自分の手のパターンは個体の性質によって決まってくる。前論文では TFT 個体のパターンのみを抽出したが、今回のシミュレーションでは、安田俊一

(2014) で識別した4つのタイプを検出してみる。

上述のように、Axelrodのシミュレーションでは、「戦略」を遺伝子型から特定することは困難であるが、個体が自分の持つ「戦略」にしたがってゲームを行うのであれば、ゲームの履歴には戦略から導かれた「手」(表現型)が残っている。履歴は1世代1個体がある対戦相手と151回対戦し、総当たり戦で「集団の個体数-1」回それを繰り返すのであるから、履歴はかなりの回数更新され続ける。したがって、1世代での対戦が全て終了した時点での「直近3回」のゲームの履歴は遺伝子型に隠されている「個体の性質」を反映していると考えられても良いであろう。

本稿で識別する個体の性質とその性質が示す記憶領域のパターンは以下の4タイプである。なお以下で示す x, y, z は $x, y, z \in \{0, 1\}$ であり、 x の反対を \bar{x} 、ワイルドカード('0', '1'のなんでもよい)を*で表す。

あまのじゃく 前回相手の出した手と逆の手を出す。したがって記憶領域のパターンは「* $x\bar{x}\bar{x}$ 」であり、最後に相手が出した手に対して逆の手を出すので、記憶領域パターンが示す戦略領域の箇所は「 \bar{x} 」となっている。以下、この表現を「* $x\bar{x}\bar{x} \rightarrow \bar{x}$ 」と記述することにしよう。

裏切り者 相手の手がどうあれ'1'(裏切り)を出すので、「1*1*1* \rightarrow 1」。ただし、「111111 \rightarrow 1」は以下の「互恵主義者」のパターンでもあるため、このパターンは「裏切り者」としてはカウントしない。

お人好し 相手の手がどうあれ、'0'(協力)を出すので、「0*0*0* \rightarrow 0」。上と同様に「000000 \rightarrow 0」は「互恵主義者」のパターンでもあるため、このパターンは「お人好し」としてはカウントしない。

互恵主義者 (TFT) 前回相手が出した手を今回自分が出す¹¹⁾ので、

11) TFTは、一番最初に「協力'0'」を出す、今回の定式化での「直近3回のゲームの履歴」の中では、記憶領域の最初のビットがゲームの途中であるから0ビット目はワイルドカードとしている。最初に「協力」ではなく、「裏切り」から出発する戦略は「Tit-for-Tat」とも呼ばれている。

「* $xyyz \rightarrow z$ 」以上の記憶領域パターンを各世代で抽出するが、「あまのじゃく」のパターンには、「お人好し」「裏切り者」と区別がつかないケースが存在する。

「010101 \rightarrow 0」は「あまのじゃく」パターンであるが（前回の相手の手と反対の手を出すので）、このパターンは「お人好し」パターンと区別がつかない。同様に「101010 \rightarrow 1」は「裏切り者」と区別がつかない。今回のシミュレーションにおいてはこの2つのパターンはそれぞれ「お人好し」「裏切り者」にカウントする。

6 シミュレーション結果

以上のように個体の性質を定義した上で、それぞれのシミュレーションの中で発生した各タイプを検出する。

以下のシミュレーションでは、100世代・50個体のケースにおいて、「協力への収束」「裏切りへの収束」「協力にも裏切りにも収束しない」それぞれの場合を100回集めたシミュレーションの中から代表的なものを取り出して示す。

注目するのは集団の平均値推移、集団を構成する上記各タイプの構成率、そして100世代終了時点での記憶領域・戦略領域の配列パターンである。

配列パターンはAxelrod実験における染色体の設計の影響を強く反映している。遺伝的アルゴリズムの視点から見たAxelrod実験の特徴でもあり、筆者の前論文で指摘した「ラマルクの進化」の問題点を強く示唆するものになっている。

6.1 協力へ収束する場合

4つの性質を持つ個体の世代による分布を、集団平均値の推移と共に図1に示す²⁾これは集団が協力へ収束した場合である。初期状態では全ての性質を持つ個体が存在しているが、「裏切り者」はほとんど増えることなく、30世代でほぼいなくなる。「あまのじゃく」と「互恵主義者」が20%近くの水準で増

減し、「お人好し」が若干残っている状態から、51世代で「互恵主義者」が構成率を急激にのばし、69世代で集団の全てが「互恵主義者」となる。集団の平均利得はその時点で完全な協力を示す3.0へ達し、ずっとその状態が維持されている(図1b)。Axelrodは原論文で『「互恵主義者」が増大していくと集団平均値が上昇する』と述べているが、シミュレーションの結果はこの結論を支持している。

図1で観察されたように集団中の個体が全て「互恵主義者」になり、それと同時に集団が協力へ集中し、それ以降その状態が維持されるのは、協力への収束を確認できるシミュレーションで共通している。つまり、協力への収束に当たっては「互恵主義者」が集団で支配的になり、いったん集団が協力へ収束した場合、その状態が崩されることはほとんどない¹³⁾

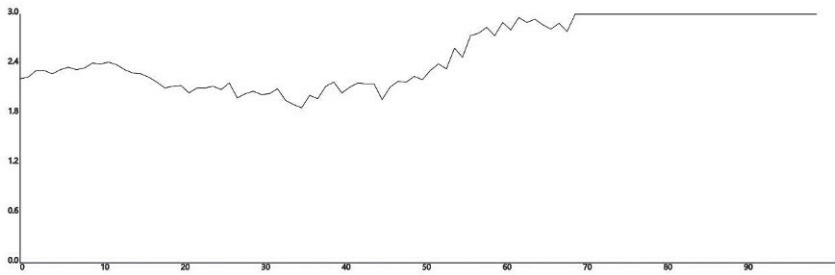
さて、1回のシミュレーションが終了した時点、つまり100世代を経た段階で染色体の配列はどのような状態なのだろうか。

すでに述べたように、Axelrod実験の枠組みでは戦略領域から、通常ゲーム理論で使われる意味での「戦略」を分類することは不可能であるが、集団内でなんらかの特定パターンをもった染色体が多数を占めるようなことがあれば、それは集団が特定の状態に「収束した」ことを意味する¹⁴⁾ これをみるために記憶領域配列、戦略領域配列が何らかのパターンに「集中した」程度を示す「集中度」を定義しよう。

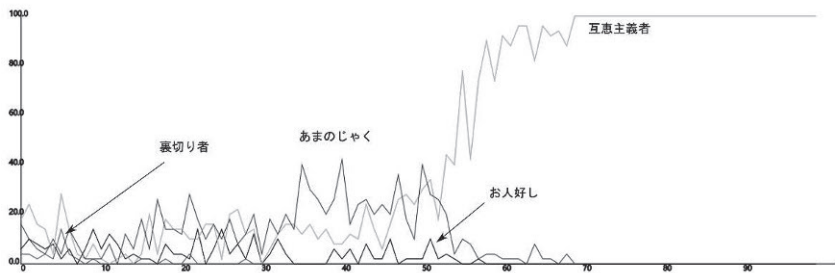
12) グラフはすべて筆者がこのシミュレーション用に開発した分析用アプリケーションによる作図である。開発言語は openjfx.org の `javafx17.0.1`。標準ライブラリ以外は使用していない。ソースコードは [Github](https://github.com/shunichiYasuda/GraphViewer) で公開している (<https://github.com/shunichiYasuda/GraphViewer>)

13) 筆者はこのシミュレーションを数万回走らせているが、中には若干協力の崩壊を示すものもある。しかしそれは400世代のシミュレーションで、ここでの収束基準としている「引き続き10世代のうちの6世代以上」が数回だけ満たされ、いったん基準をしまった後で再び収束し、図1のようにそれ以降は収束状態を維持している。

14) 最適化計算など特定の範囲での「頂上・底」を見いだすために遺伝的アルゴリズムが使われる際には「ハミング距離」(染色体に占めるビットの相違)で測って、染色体が「収束した」状態を探るが、Axelrod実験の場合にはそうしたことを目的としていないため、これまでこの意味での「収束」は問題にしてこなかった。



(a) 集団平均利得



(b) 個体の性質別分布

図1 協力への収束：100世代・50個体

各実験の最後で、記憶領域と戦略領域の配列をチェックし、同じ配列をもつ個体があればそれをカウントしていく。そしてその値が一番多いパターンを各実験で記録する。例えば、50個体の全てが異なるパターンを持っている場合は「1」（そのパターンを持った個体は1つしかない）、50個体全てがまったく同じパターンを持っていれば「50」（そのパターンを持った個体が50個）、20個体があるパターンを持っており、残りの30個体が別のパターンを持っていれば「30」と記録する。その上で、100回実験でのその平均を「平均集中度」と定義する。ここでは個体数が50であるので、平均集中度が50であれば、すべての実験で50個体が1つのパターンに集中したことを意味する。

以下でとりあげる「協力・裏切りへ収束した場合」「どちらにも収束しない場合」について、平均集中度を表7に示す。

表7 平均集中度

	記憶領域	戦略領域	典型的記憶領域パターン
協力へ収束した場合	49.43	45.18	“000000”
裏切りへ収束した場合	45.84	44.39	“111111”
どちらにも収束しない場合	26.94	43.4	“001100”, “110011”等

「協りに収束した場合」の平均集中度は記憶領域で49.43、戦略領域で45.18であり、非常に高い。ほとんどの場合に集団のすべてが同じ記憶領域と戦略領域を持つ、つまり、同じ染色体を持つようになることを意味している。

例えば図1の場合は、最終的に全ての個体の記憶領域が“000000”へ集中し、戦略領域は

“0110110010110110101100101000111111111000101011100010111111000011”

に集中した。

記憶領域が“000000”へ収束しているわけだから、戦略領域の先頭が“0”であれば、この状態がずっと続くと言うことだ。そして戦略領域もただ1つのパターンに集中しているということは、すべての個体が同じ染色体を持っていることを意味するので、交叉を経てもこのパターンは絶対に壊れない。つまりはすべての個体が協力を繰り返すことになる。図1に代表的に示したように、集団がすべて「互惠主義者」となり、かつ集団平均値が3.0の完全な協力にひとたび達すると、それ以降は永遠にその状態が続き、それが崩壊することがないことの原因がこれである。

協力への収束がみられた100回のシミュレーションのうち、多くの場合で記憶領域は“000000”へ集中するが、戦略領域のパターンはすべて同じだとは限らない。例えば別のケースでは記憶領域がすべて“000000”であって、戦略領域

域は4つのパターンに収束した（以下の「:」の後の数字は個体数）。

“01000001011101000000100110101000010000110110000111100000011110100” : 21

“0011010111101000000100110101000010000110110000111100010001011101” : 15

“01000001011101000000100110101000010000110110000111100010001011101” : 10

“0011010111101000000100110101000010000110110000111100000011110100” : 4

この4つのパターンは10ビット目から50ビット目までは同じであり、またそのほかの箇所でも一部分同じパターンが見られる。50個体のうち、21個体で同じパターン、さらに数多くの同じパターンが含まれているということは交叉を経たとしてもそれほどおおきな違いは発生しないであろうことは予想できる。実際このパターンを示したシミュレーションでも集団平均値の推移と個体の性質分布は図1と同じようなグラフを描く。

上の例とは逆に、記憶領域が“000000”ではなく、もっと多様なものが残り、戦略領域がある程度集中したパターンもある。図2にその例を示す。

図2は、記憶領域が

“001100” : 15, “011001” : 12, “000001” : 6, “111001” : 5, “001111” : 4,

“011110” : 3, “100111” : 2, “110110” : 2, “110000” : 1

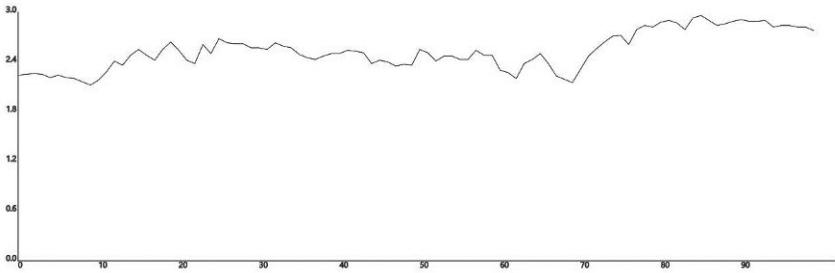
の9パターンに収束し、戦略領域は

“0101010011101110011011010100100000100011000011011000100000110110” : 39

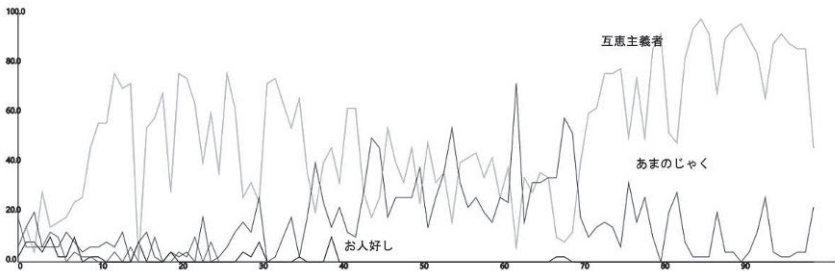
“1001110011101110011011010100100000100011000011011000100000110110” : 11

以上の2つに収束した。ここでも集中した2つの戦略領域パターンは先頭の4ビットを除いて5ビットめ以降は同じでありほとんど同じパターンに収束したということができる。

記憶パターンが“000000”である個体は見られないが、5パターンは「互恵主義者」の記憶パターンを示しており、最も多くの個体を持っている[“001100”



(a) 集団平均利得



(b) 個体の性質別分布

図2 協力への収束：記憶領域が集中しないケース

→「1」は「あまのじゃく」パターンである。個体の性質分布は「互恵主義者」が多いが、「あまのじゃく」も一定数以上は生き残っている。そして集団の平均利得は3.0に「張り付く」ことなく若干の変動を繰り返しながら推移している。

ここで示したように、記憶領域が1つのパターンに集中した場合は、集団の平均利得は一定値に収束し、記憶領域がいくつか複数のパターンを維持している場合は集団の平均利得はある範囲で若干の変動を繰り返す。これは以下の「裏切りに収束する」「収束しない」場合でも同じである。

記憶領域と戦略領域のパターンが集中する以上のような傾向は「裏切りへ収

束する場合」「協力と裏切りのどちらにも収束しない場合」にも見られることである。この点については後に触れる。

集団が協力へ収束したこれらのシミュレーションで、最終的に集中が見られた「互恵主義者」は、「お人好し」と区別がつかないことに注意しよう。分類上は「互恵主義者」となっているが、最終的に進化した先は「お人好し」と区別がつかない。

「000000」→「0」は、戦略領域の先頭ビットが「0」であることを意味する。記憶領域が「000000」へ集中すれば、戦略領域は先頭ビットが「0」でありさえすれば、残りの63ビットのパターンがどのようなものであれ、「お人好し」の行動を崩さない。そして、このパターンは「互恵主義者」でもあるので個体の性質分布だけから見れば「互恵主義者（TFT）が進化した・集団中にTFTが繁栄した」とみえる。だが、それは同時に「お人好しが繁栄した」とも言えるわけで、Axelrodのシミュレーションは、その内容を詳しく見るとかならずしも「TFTの勝利」とはいえないのである。

シミュレーションは、必ずしも「協力への収束」をもたらすわけではもちろんない。「裏切りへの収束」が起きる場合もあれば、どちらにも「収束しない」状態で推移する場合もある¹⁵⁾

そうした場合の状況を以下で順次見ていこう。

6.2 裏切りへ収束する場合

図3は集団が裏切りへ収束した場合である。図3においては、10-30世代あたりまで「あまのじゃく」が集団の多数を占める。27世代あたりから「互恵主義者」が集団内で拡大してくるが、集団全体へは広まらない。52世代で「お人好し」が一時的に拡張するがすぐに消滅し、61世代で「互恵主義者」が集団のすべてとなると同時に集団の平均利得は1.0(完全な裏切り)に達して、

15) 「収束率」からいえば、表1が示しているように、協力への収束が起きるのはせいぜい16-20%程度であるから、そうでないケースの方がずっと多い。

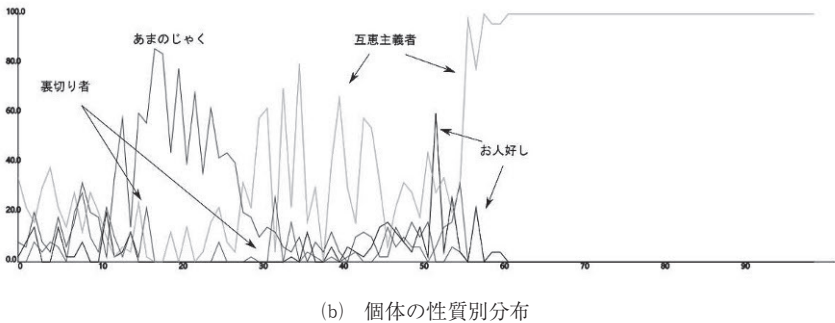
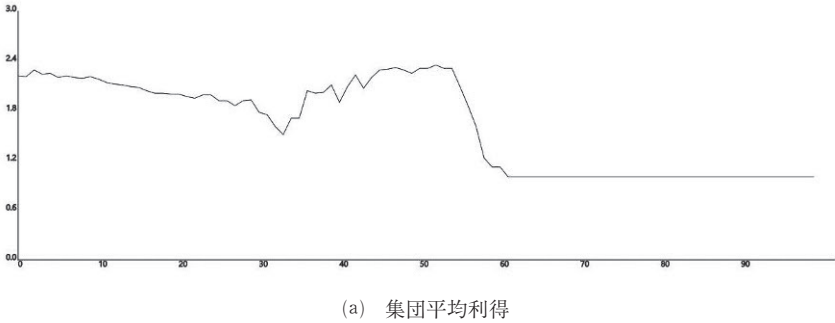


図3 裏切りへの収束：100世代50個体

それ以降はその状態が継続する。

互惠主義者は「報恩者」と同時に「報復者」である。相手の協力に対しては「報恩」で対応するため、集団内で協力が広まると「報恩者としての互惠主義者」が広まる。このことは集団が「礼儀正しい（相手の協力に対して協力する）」個体ばかりになることだから、「（協力，協力）」が集団全体で成立し、集団の平均利得は3.0になる。逆に集団内で裏切りが広がると「報復者としての互惠主義者」が広まることになる。このときは「目には目を（裏切りに対しては裏切りで反応する）」個体が増えて、「（裏切り，裏切り）」が成立して集団の平均利得は1.0になる。

このように、集団が裏切りへ収束した場合は、協力へ収束した場合と同じように、集団内は「互惠主義者」に集中するが、結果は反対になることを図1と図3は示している。そして協力への収束と同様に、集団が互惠主義者に集中し、集団の平均値が完全な裏切りへ達したら、その状態は崩れることがない。

このシミュレーションにおいては、すべての個体の記憶領域が“111111”に集中し、戦略領域は

“110110011110000100101000111110100111000101011100011000101011111111”

に集中した。つまりはすべての個体の染色体がただ一つのパターンに収束したということだ。

裏切りへの収束を記録した100回のシミュレーションでは、記憶領域は“111111”に集中しがちである。また、記憶領域が1つではなくいくつかのパターンに分かれて集中する場合でも戦略領域が一つに集中する場合があるのも協力への収束の場合と同じである。

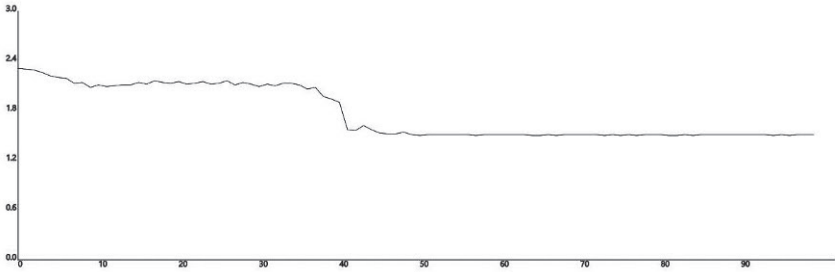
図4はそのようなケースを示している。このケースでは「報復者としての互惠主義者」がまったく成長せず、そればかりか個体の性質を示す他の3つの性質をもった個体も49世代以降姿を消し、集団の平均利得は完全な裏切りを示す1.0よりも少し大きい1.51で安定した。

このシミュレーションの最終世代では、記憶領域が“111111”：30, “110011”：20の2パターンに収束し、戦略領域はすべての個体が

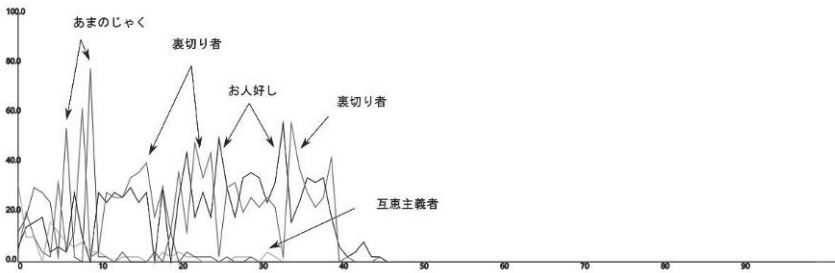
“111110100010110110111010011111001110011001110111100110010011111110”

に集中した。

30個体の記憶領域が“111111”であることから「裏切り者」あるいは「互惠主義者」が多数を占めるように見えるが、戦略領域の63ビット目（最後のビット）は‘0’（協力）であるからそうではない。このパターンは個体の性質を示す4つの分類に当てはまらない。また“110011”は記憶パターンだけを見ると



(a) 集団平均利得



(b) 個体の性質別分布

図4 裏切りへの収束：個体の性質が傾向を示さないケース

「あまのじゃく」であるが、このパターンが示す戦略領域の箇所は51ビット目で、それは「1」であるから、これも「あまのじゃく」ではなく分類にはあてはまらない。

このケースでは集団の平均利得が基準値である1.54未満の水準で安定したため「裏切りへの収束」にカウントされているが、このようにすべての個体が記憶領域“111111”のように典型的な「裏切り者」あるいは「(報復者としての)互惠主義者」にならないケースでも、集団の平均利得が裏切りへ収束する場合もある。

このようなことが起きるのは、やはり染色体が特定のパターンに集中するこ

とが原因である。

協力へ収束するケースで述べたのと同じで、記憶領域が“111111”へ収束し、戦略領域の最後のビット（したがって染色体の最後のビット）が‘1’であればそこからの変化はなくなる。

また、ここでも最終的に集団全体を占める「互恵主義者」は「裏切り者」と区別がつかない。集団平均値が完全な裏切りである1.0に収束する場合には「裏切り者ばかりになった」ともいえるのである。

6.3 収束しない場合

集団が協力へも裏切りにも収束せず、つまり集団の平均利得が基準値である2.76-1.54の間で推移することも、もちろん多い。この場合は集団の性質が（報恩者であれ報復者であれ）「互恵主義者」に収束して協力へ収束した場合の記憶領域が“000000”で集団の平均利得が3.0であるとか、裏切りへ収束した場合の“111111”で1.0であるとか、性質別分布状況と集団の平均利得がある特定の状態に収束するとはいえないが、それでも別の一定パターンに収束しがちである。

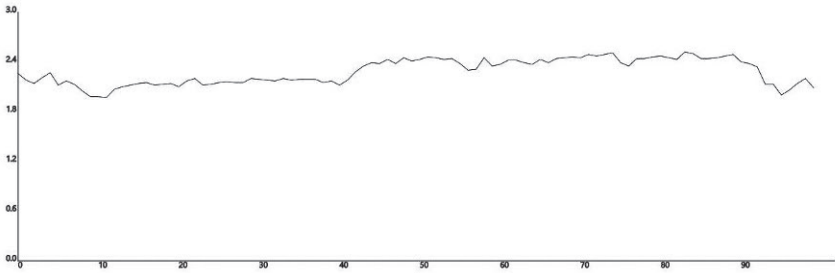
以下ではいくつかのケースを示す。

図5は、集団が協力にも裏切りにも収束しないケースを集めた中でも、集団の平均利得が比較的変動を繰り返しながら推移していく例である。図5bをみると、50世代あたりまでで「裏切り者」「あまのじゃく」がほぼ消滅し、以降は「お人好し」「互恵主義者」が増減しながら推移していることがわかる。集団の平均利得は2.3から2.5あたりまでの間で変動しつつ推移する。

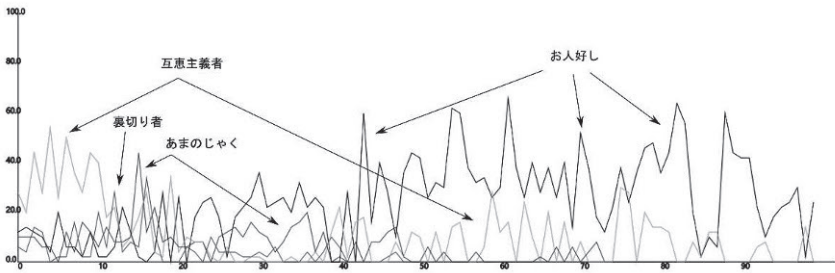
このケースの場合、記憶領域は、最終的に

“010000”：13, “110011”：11, “111111”：10, “001111”：9,
“111100”：4, “100000”：3

の6パターンが残っており、戦略領域は次の3つに収束した。



(a) 集団平均利得



(b) 個体の性質別分布

図5 収束しない：集団平均利得が変動するケース

“1001010001101011001111010001001010111001011011001001001110111010” : 42

“1001010001101011001111010001001010111001011011001001001111000111” : 6

“1001010001101011001111010001001010111001011011001001011111000111” : 2

収束した戦略領域3つのパターンが下位7ビットを除くと全く同じであること、さらに真ん中と一番下のパターンは53ビット目だけが異なるだけであることに注意しよう。ここでも染色体はほぼ似たようなパターンに収束するのである。

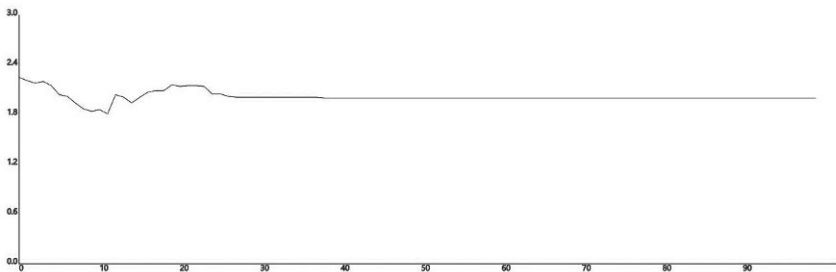
協力にも裏切りにも進化しないということは、一見「不安定」な状態が続く

ようにも思えるが、このように染色体が特定のパターンに集中するため、集団平均値の水準が特定の値に「固着」しない場合でも、ある程度の幅で安定して推移する。

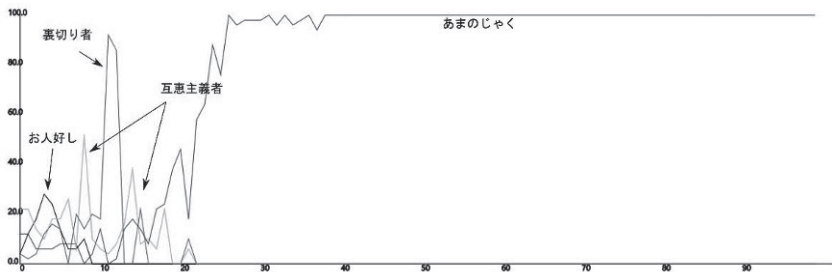
さらに、協力へも裏切りへも集束しない場合であっても、ある特定の値に「固着」するケースもある。

以下でそのようなケースについて、集団が「あまのじゃく」へ集中した場合、および特定の性質が発展しなかった場合を取り上げよう。

図6は集団が「あまのじゃく」に集中したケースである。



(a) 集団平均利得



(b) 個体の性質別分布

図6 収束しない：あまのじゃくへの集中

このとき記憶領域はすべての個体が“110011”へ集中し、戦略領域は
 “0011110100101000011111011110111011100010111100111110100110110001”

へ集中した。そして集団平均値は2.0で一定となりそのまま推移する。

こうした「あまのじゃく」へ集中するケースは一定数見られる¹⁶⁾ その場合、記憶領域は例外なく“110011”あるいは“001100”になっており、戦略領域を見ると「“110011”→‘0’」, 「“001100”→‘1’」となっている。つまり, “110011”の10進表現は51, そして戦略領域の51ビットが‘0’, “001100”の10進表現は12で、戦略領域の12ビット目が‘1’である。

すべての個体が「“110011”→‘0’」であれば、次のゲームが終了した時点で記憶領域は“001100”に更新され、それが指し示す戦略は‘1’であるので、その次のゲームが終了すれば記憶領域は再び“110011”にもどる。したがって、集中した戦略領域のパターンが「12ビット目が‘1’で51ビット目が‘0’」になっていけば、この状態が続くことになる。このゲームの結果は「(裏切り, 裏切り)⇔(協力, 協力)」を繰り返すため、集団の平均利得は2.0となって、変化しない。

では特定の性質が発展しない場合はどうであろうか？

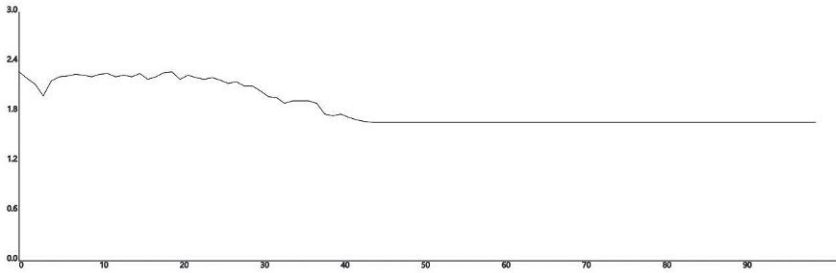
図7は、前節の図4と同様に、なんらかの特定の性質が発展することなく、しかし不安定にも推移せず、集団の平均利得が一定値(1.67)に収束した例である。このとき、最終的にはすべての個体が、記憶領域が“111100”に、戦略領域が

“1011000110001000110100101110110011010011011011001101011111101111”

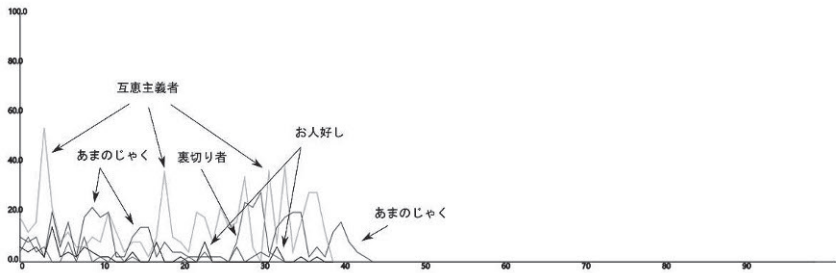
に集中した。そして、集団平均値は一定値1.67で推移する。

こうなるロジックも明白だ。

16) こうしたケースは協力へも裏切りにも収束しなかった100回のケースのうち13回を観察した。



(a) 集団平均利得



(b) 個体の性質別分布

図7 収束しない：集団平均利得が一定値に集束するケース

記憶領域が“111100”であるならば、それが示す60ビット目の値が‘1’。すると、次のゲームが終わった状態で集団のすべてが“110011”の履歴を持ち、その次のゲームで51ビット目にある‘1’を出す。したがって次の履歴は“001111”。戦略領域の15ビット目が‘0’ならばその次に再び“111100”に戻る。ゲームの結果は「(裏切り, 裏切り)→(裏切り, 裏切り)→(協力, 協力)」を繰り返すことになる。したがって、平均利得は $(1.0+1.0+3.0)/3=1.666\cdots \approx 1.67$ へ収束する。上の戦略領域はこの条件を満たしている。そしてすべての個体がこの染色体を持っている以上、この過程が壊れる可能性は極めて低い。

7 Axelrod 実験の帰結

以上のシミュレーション結果が示唆することは、Axelrod 実験においては最終的に染色体が一つのパターンに収束しやすいということだ。

上のシミュレーションはすべて個体数 50, 世代数 100 で行っている。遺伝的アルゴリズムの過程によって、染色体が一定パターンに進化するのであれば、世代数を増やせば（進化可能な期間を延ばせば）1つのパターンへの集中度は増えるはずだ。実際、シミュレーション期間を延ばして、400 世代とした場合の集中度を表 8 に示す。

表 8 400 世代の場合の平均集中度

	記憶領域	戦略領域	典型的記憶領域パターン
協力へ収束した場合	48.53	50.0	“000000”
裏切りへ収束した場合	44.79	49.15	“111111”
どちらにも収束しない場合	32.5	49.78	“001100”, “110011” 等

表 7 と比較すると記憶領域については「協力・裏切りへ集束した場合」には若干の違いがあるが、せいぜい 1% 程度の違いであるからほとんど違いがないといってもよい。注目すべきは戦略領域の集中度で、簡単に言えばほぼすべての個体が 1つのパターンに収束している。これは「協力・裏切りへの収束、収束しない」に無関係に、すべてのシミュレーションで最後には染色体が一つに収束することを意味している。つまり、Axelrod 実験においては、アルゴリズムを続けていけば記憶領域に若干違いが残ることはあるものの、戦略領域は一つに集中してしまうということである。つまり、やはり染色体は特定のパターンに集中する。

さて、そうなると第 3 節で検討した交叉率について再度見直す必要が出てくる。第 3 節では「協力への収束」を基準としてその影響を見たわけだが、上のシミュレーション結果で明らかになってきたように Axelrod 実験では「染色体

の収束」が焦点になる。すると染色体パターンを破壊する可能性がある交叉・突然変異過程が重要になり、これらのパラメータについて、今度は染色体の収束、「集中度」の視点から確認しておかなければならないだろう。

ここまで行ってきたシミュレーションでは、交叉確率は「各世代平均的に1個体」であったから50個体のケースで言えば2%である。これを100%、つまり各世代ごとにすべての個体が交叉を起こすようにしたときの平均集中度を表9に比較して示す。

表9 交叉率を変えた場合の平均集中度

	交叉率 2 %		交叉率 100 %	
	記憶領域	戦略領域	記憶領域	戦略領域
協力へ収束した場合	49.43	45.18	49.23	10.36
裏切りへ収束した場合	45.84	44.39	46.15	11.46
どちらにも収束しない場合	26.94	43.4	26.43	12.2

交叉率 2%と 100%の場合を比較すると、戦略領域の平均集中度の違いは劇的である。100 世代の終わりにおいて平均集中度は約 5 分の 1 になる。

ではこの事実は集団の進化をなにか劇的に変化させるのであろうか？

そのようなことはない。各シミュレーションでの集団の平均値の推移、個体の性質別分布などを確認すると、特に何も変化していないことが確認できる。

戦略領域のパターンが同一であるかどうかは「すべてのビットが同じかどうか」で判断している。64 ビットのうちの 1 ビットだけ違っていてもそれらの染色体は同じもののだとはみなしていない。しかし前節でみたように多少の違いがあっても他の部分では全く同じである戦略領域は多い。

記憶領域が少ないパターンに固まっている（平均集中度が高い）ならば、それが指し示す個体の行動を決める戦略領域の箇所は少数の箇所に特定されていき、そのほかの場所にある遺伝子（‘0’、‘1’）は個体の行動に影響を与えない。したがって、個体の行動に影響をあたえるそのような特定の箇所を含む部分的

なパターンが同じであれば、他の部分のパターンが違っていても個体の行動は変わらない。その結果、毎回同じ行動をとる個体が増え、記憶領域のパターンに対応した戦略領域が固定されてくる。

つまりは、記憶領域が特定のパターンに集中するかどうかが重要なのだ。表9では、記憶領域への平均集中度にほとんど変化がない。第6.1節で述べたように、例えば記憶領域が“000000”に集中すれば、戦略領域のパターンが多少多様になったところで、戦略領域の先頭にあるビットが‘0’であれば、集団の平均値は完全な協力を示す3.0で一定となる。これと同じように、記憶領域が特定のパターンに集中すれば集団の平均値は一定値に収束する。収束したその値が3.0（ここでは基準値の2.76を超える値）であれば「完全な協力」へ進化したことになるし、1.0（ここでは1.56以下の値）であれば「完全な裏切り」へ進化したことになる。また、それらの中間値に収束した場合はやはり集団内の個体は、単純ではないかもしれないが特定の行動を繰り返すことになり、集団の平均利得はやはり一定値に収束する。

戦略領域のパターンが集中するかどうかは単に時間、世代数の問題である。遺伝的アルゴリズムは交叉・突然変異で染色体を変化させながら「よりよい利得」をもたらす染色体を探索していく。交叉率が高い場合は染色体が次世代で破壊される確率は増えるが¹⁷⁾それでも時間をかけていけばよりよい染色体が集団に拡散していく。したがって表8のように世代数を延ばせば、やはり戦略領域も一定のパターンに収束してくる。実際、交叉率100%としても、世代数を400世代までに延ばせば表9のどの場合においても戦略領域の平均集中度は上昇する¹⁸⁾

17) ここで、今回のシミュレーションではエリート戦略をとっていないことに注意しておこう。交叉率100%ではもっとも利得が高い個体の染色体も交叉過程に入るため、一般的に考えて、進化のスピードは遅くなる。

18) 交叉率100%で世代を400まで延ばした場合、「協力への収束」「裏切りへの集束」「収束しない」各ケースでの戦略領域での平均集中度は、それぞれ、49.47、48.98、49.11に跳ね上がる。これらの数値は個体数50の場合、ほとんどのシミュレーションで集団が一つの戦略領域に集中したことを示す。

さて、交叉率とは無関係に記憶領域が高い集中度（「収束しない」ケースでも3割近くは集中している）を示すのはなぜだろうか。

これは Axelrod の染色体の設計に原因がある。

Axelrod はゲームの履歴をそのまま染色体に含めてしまった。そして 70 ビットの染色体のうち、先頭 6 ビットを「直近 3 回のゲームの履歴」としている。

交叉がたとえ 100% の確率（すべての個体が各世代で交叉を起こす）だとしても、交叉する場所は一様分布の乱数で決定されるため、「先頭 6 ビットの間」に一点交叉のポイントがくる（記憶領域が破壊される）確率と「7 ビット以降に交叉ポイントがくる（記憶領域はそのまま残る）確率」では 10 倍以上の差がある。つまり、記憶領域は（染色体の他の位置に比較すれば）交叉によって破壊されにくい。したがって、ある世代から次の世代へ染色体が（交叉・突然変異を経て）渡される場合に、多くの個体で前の世代のゲームの記憶が引き継がれることになる。これが筆者の前論文で指摘した「ラマルクの進化」である。

そして記憶領域が進化の結果として一つのパターンに集中するということは、世代が変わってもすべての個体と同じゲーム履歴の下で行動を決定することであり、その集団のすべての個体と同じ行動（あるいは同じ繰り返し行動）を行うことを意味するので、集団の平均値は一定値に収束する。

Axelrod のコーディングは集団行動の固定化をもたらしやすいのである。

8 結 語

以上見てきたように、Axelrod 実験はパラメータの設定はどうあれ、最終的に集団がある特定の染色体に集中する傾向が極めて強い。

これは遺伝的アルゴリズムの本来の性質であって、利得が高い個体が「親」として遺伝子を次世代に伝えていくことで、そして、その過程で交叉・突然変異を繰り返して、そこで発生するより環境に適合的な個体が持つ染色体が集団に拡散していくことで、最後には集団のすべての個体と同じ染色体を持つにい

たる。最適値探索においてはこの性質が利用されているのであり、Axelrod 実験の結果は遺伝的アルゴリズムを用いた実験の帰結としてはごくまっとうなものである。

しかしながら、Axelrod の意図としては、遺伝的アルゴリズムのこの性質をもちいて、多様な可能性の中から最適な戦略を進化させ、囚人のジレンマゲームにおける社会的な最適解である「(協力・協力)」状態、そしてその場合の利得 (3.0, 3.0) の実現へ収束させることであつたらう。それを達成する戦略が「互恵的」、すなわち TFT であることを示したかたに違いない。

しかし、Axelrod 実験で収束した先の染色体のパターンは多様である。ここでは「協力・裏切りの達成」の基準、それが持続することによる「集団が協力(裏切り)へ収束する」基準を設けてシミュレーション全体を分類しており、その結果の収束先は「“000000”→‘0’」「“111111”→‘1’」、もたらされる集団平均利得 3.0 と 1.0 であるが、これらを「両極端」とすれば、その間には、「あまのじゃく」への収束で見たような「“001100”→‘1’」などのようなパターンも存在する。「毎回協力する」「毎回裏切る」ような単純な戦略だけではなく、記憶領域パターンと戦略領域パターンの組み合わせによって「循環」が起きて、その結果として集団の平均利得が一つの値、1.0 と 3.0 の間に位置する値に収束する場合もある。

Axelrod 実験の到達先は一つではない、ということだ。たしかに集団が協力へ到達する場合は存在する。だが同時に裏切りにも、協力と裏切りとの中間にも同じようにいきつく可能性がある。行き着く先は予測できない。

さらにいえば、Axelrod が個体数 20、世代数 50 のシミュレーションを 10 回走らせて観察した「互恵的な個体の増大が集団の平均値を上昇させる」現象は、観察したシミュレーション回数が少なかったことからくる誤解である可能性が高い。「裏切り」へ収束した場合には集団の平均利得が低下していくこともあり得たはずなのである。

今回のシミュレーションでわかるのは、かならずしも「協力への収束」のみ

が起きるわけではないことである。表1でわかるように、Axelrod 実験では協力への収束はたかだか20%程度でしか起こらない。ちなみに、今回のシミュレーションを単純に1万回走らせてみると、そのうち協力へ収束したのは24.4%である。裏切りへの収束は20.0%、そして協力にも裏切りにも収束しなかったのは55.6%である。そしてそれぞれのケースで染色体の収束が起り、集団の平均利得は一定値、あるいはある程度の幅の範囲での変動を示す。つまりどのような状態であるにせよ、すべてのシミュレーションは「行き着くところへ行き着く」。

Axelrod 実験の最終的な染色体が、さまざまなパターンに収束することは、長さ70のバイナリ配列が示す 2^{70} の可能性の一つ一つが「戦略」だと考えたとして、それら戦略が構成する空間にはさまざまな（おそらく非常に多数の）アトラクターが存在していることを意味する。したがって、TFTのような何か特定の戦略はその中の一つに過ぎないのであって、繰り返し囚人のジレンマゲームにおいて遺伝的アルゴリズムが進化させる戦略がTFTのみであるわけではないのだ。

私見では囚人のジレンマゲームに遺伝的アルゴリズムを応用するのは、なんらかの特定の戦略が進化してきて、それが「よい戦略」であることを示すためではない。本稿で取り扱ったように、進化の過程でどのような性質をもった個体が発展・消滅し、また、集団の平均利得がどのように変化し、集団内で協力が（あるいは裏切りが）発生してくるのかを観察するためである。その意味ではAxelrod 実験において、最終的に進化してくる戦略自体は多くの可能性があって特定できないものの、集団が協力へ収束する場合がある程度あること、そしてそのようなときに集団内での個体の性格別分布がどのように移り変わってくるのかなどを観察できたことは無意味ではない。集団が協力へ収束するようなケースにおいては、Axelrod が指摘したように、確かに「互惠主義者」が集団内に拡大していくのである。

今回 Axelrod 実験を改めて検証することで、これまでに触れられていない

くつかの事実を確認できた。これらの帰結をもたらした Axelrod の染色体設計はやはり独特のものであり、設計自体にこうした帰結をもたらす原因が含まれているように見える。今後 Axelrod が用いた「過去の履歴から次の手を決める」タイプの「決定論的戦略」を用いる際は、前論文で示したように、記憶領域を染色体に含めることなく取り扱うことが「記憶領域の収束」によるバイアスを避けるために必要であろう。

参 考 文 献

- Axelrod, Robert (1980a) "Effective Choice in the Prisoner's Dilemma," *Journal of Conflict Resolution*, Vol. 24, No. 1, pp. 3-25, March.
- (1980b) "More Effective Choice in the Prisoner's Dilemma," *Journal of Conflict Resolution*, Vol. 24, No. 3, pp. 379-403, September.
- (1987) "The Evolution of Strategies in the Iterated Prisoner's Dilemma," in Dabis, Lawrence ed. *Genetic Algorithms and Simulated Annealing*: Pitman, pp. 32-41.
- Binmore, Ken (1994) *PLAYING FAIR: Game Theory and the Social Contract I*: MIT Press.
- (1998) "Review The Complexity of Cooperation," *Journal of Artificial Societies and Social Simulation*, Vol. 1, No. 1, <http://jasss.soc.surrey.ac.uk/JASSS.html/1/1/review1.html>.
- Goldberg, David E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley.
- Nowak, Martin and Karl Sigmund (1992) "Tit for tat in heterogeneous populations," *Nature*, Vol. 355, pp. 250-253, January.
- 安田俊一 (2001) 「GA による囚人のジレンマ実験—ダーウィンの手法とラマルクの手法—」, 『松山大学論集』, 第 13 卷, 第 2 号, 37-62 頁.
- (2005) 「どのような集団が協力を生み出すか?—GA による囚人のジレンマ—」, 『進化経済学論集』, 第 9 卷, 217-226 頁.
- (2014) 「集団的な囚人のジレンマゲームにおける戦略推移について—「報恩」と「報復」を軸として—」, 『松山大学論集』, 第 26 卷, 第 3 号, 21-48 頁.