

プログラムの自動生成手法を用いた繰り返し囚人のジレンマゲームの行動ルール構築と協調行動の分析

Construction of Action Rule for Iterated Prisoner's Dilemma by Using Program Evolution Method and Analysis of Cooperation

今福啓*

Kei Imafuku

Email: k03082@dokkyo.ac.jp

本論文では、実社会におけるさまざまなジレンマ的な状況を単純なゲームの形式で表現した繰り返し囚人のジレンマゲームにおいて、合理的に行動する主体の行動ルールをコンピュータプログラムの自動生成手法により構築する。この手法では、コンピュータ内部で自動的に作成されたプログラムの形式で行動ルールが表現されるため、行動主体が持つ情報がどのように利用されて協調が生み出されるのかを詳細に分析することが可能となる。また、行動主体をさまざまなネットワークを用いて接続し行動を分析することで、どのようなネットワーク構造が協調を生み出すのかについても明らかにする。そして提案モデルを用いたシミュレーションを行い、得られた結果から有効性について検討する。

In this paper, we propose a method to construct the efficient action rule for iterated prisoner's dilemma game by using program evolution method. In this method, action rule is described by the computer program which is constructed automatically in the computer, therefore we can analyze how the information is used by the individual and becomes to cooperate in detail. We also analyze what kind of network structure will produce cooperation by connecting each individual with various networks. Some simulations are executed, and the effectiveness of the proposed method is considered from the simulation results.

*: 獨協大学経済学部

1. はじめに

たとえば環境問題など、人が合理的に行動するにもかかわらず、社会全体では非合理的な結果となるジレンマ的な状況の生じる問題が社会に存在することが、理論的な分析により示されている。この状況において、ジレンマを回避してより効率的な社会を作り出すためにはどうすれば良いのかについて、多くの研究が行われている。

このようなジレンマ的な問題を一般化して簡略化し、理解しやすい形にしたものとして、囚人のジレンマゲームがある。囚人のジレンマゲームでは、行動を協調と裏切りに限定し、二者が同時に実行する行動を決定する。そして、それぞれが行動に応じた利得を獲得する。この問題では、お互いが協調を実行すれば獲得できる利得の合計は最大となるが、相手の行動を想定して行動を実行する場合、裏切りを実行することが個人にとって合理的であるため、最大の利得を獲得することができないというジレンマが生じる。

実社会では、常に1対1で特定の相手とのやりとりを続けるわけではなく、複数の相手と不特定な回数のやりとりを行うことが一般的である。このような状況を考慮し、囚人のジレンマゲームをさまざまな相手と複数回実行する問題に拡張したものが、繰り返し囚人のジレンマゲームとよばれている。しかし、このゲームでも繰り返しが有限回の場合は、裏切りが合理的な行動となる。

しかし、繰り返し囚人のジレンマゲームにおいて、小規模な集団を構成しての実験を行った場合、必ずしも裏切りだけが実行されるわけではない。有限回の対戦であるにもかかわらず協調するという結果が明らかにされており、コンピュータシミュレーションでもその結果が再現されている⁽¹⁾。これらの研究では、周囲の他者の行動を模倣して自らの行動ルールを決定し、実行される行動がどのように変化していくのかを分析している。その結果、行動主体をつなぐネットワーク構造が協調に影響を与えること⁽²⁾や、行動の模倣を限定的にすることが有効であること⁽³⁾、そしてネットワーク構造と行動の模倣を同時に行うことが有効である⁽⁴⁾といった結果が得られている。

しかし先行研究では、一般的に過去の行動をもとにテーブルを作成し、それを参照して次の行動を決定するといった、非常に単純な手順が用いられてきた。そのため、人が論理的に考えて行動を決定する過程から得られる知見となっているかは明らかではない。また、その結果の多くは行動ルールが協調あるいは裏切りとなったまま安定する。

人の考えや行動は社会の状況に応じて変化するものであり、このような決まった行動のみを繰り返す結果は不自然とも考えられる。一部のシミュレーションでは、戦略が時々刻々と変化する結果が得られているが、そのような結果が得られることはまれであり、特定の行動を繰り返す結果に至ることがほとんどであると述べられている⁽⁵⁾。

本研究では、合理的に行動する主体の行動ルールを詳細に表現し、それをネットワークにより接続してコンピュータ内部に実社会と同様の状況をモデル化し、繰り返し囚人のジレンマゲームを行わせることで、どのように協調が生み出されるのかを分析する。その際、行動ルールをコンピュータプログラムの自動生成手法により作り出すことで、従来研究では扱われていなかった複雑で論理的な行動ルールを構築する。また、実社会と同様の構造を持つネットワークにより行動主体を接続することで、どのような構造が協調に必要なものであるのかについても分析する。

本研究で用いる手法では、行動ルールがコンピュータ内部でプログラムの形式で作成されるため、獲得した情報がどのように利用され行動が実行されているのかを把握できるという利点がある。また、自らの行動ルールをより高い利得が獲得できるように進化させることで、人が論理的に考えて行動し、それが社会状況に応じてどのように変化していくのかといった、実社会にみられる複雑な過程を分析することが可能となる。

以下では、2章で提案モデルを構築する際に用いる手法の説明を、3章ではシミュレーションの条件と結果の考察を行う。

2. 提案モデルの構築手法

本章では、本論文で行動主体の行動決定に使用するプログラムの自動生成手法と、主体間のつながり方を決定するネットワーク構築手法について述べる。

2.1 構造化プログラム自動生成手法

与えられた機能を満たすプログラムを、人が作成することなくコンピュータ内部で自動的に構築する手法を、プログラム自動生成手法という。本研究では、「Graph Structured Program Evolution (GRAPE)」とよばれるグラフ構造のプログラム構築手法⁽⁶⁾を応用して、行動主体の行動ルールを生成し、他の主体のルールを参照してその内容を進化させる。

GRAPEは、処理内容の置かれた複数のノード

と、複数の数値を格納するデータセットを1つのプログラムを表す個体として、それを複数配置して構成される。図1に示すように、1つのノードは整数を一次元状に並べた内容から構成されており、それを図2のように、決められたルールにもとづいて演算や条件分岐、出力するデータセットの位置といった命令や、使用するデータセットの番号に変換する。また、各ノードには次に処理するノードの位置が記述されており、1つのノードの処理が終了すると指定したノードに移動し、移動先のノードを同様に処理する。出力するデータセットの番号が記入されているノードに達した際には、指定されたデータセットの値を出力し、プログラムが終了する。

ノードの総数と整数値の数はあらかじめ決められた数だけ用意する。ノードの整数値は、命令の種類、移動先のノード番号(2つ)、使用するデータセットの番号(3つ)の順に並んでいる。これらの要素はすべて使用されるわけではなく、命令の種類によっては使用されない場合がある。

GRAPE では、プログラム実行前にデータセットに初期値を格納し、目的となる出力値が得られた際に、正しいプログラムが作成できたものとする。例えば x^2 を計算するプログラムを作成する場合、初期値としてデータセットに1, 2, 3, 4, 5を順に入力し、それぞれの値から1, 4, 9, 16, 25という出力値が得られたならば、正しいプログラムが構築されたこととなる。

GRAPE のノードには、最初ランダムな整数値を一次元状に配置する。どの個体も最初は正しいプログラムとはなっていないため、問題の解にどの程度近い値を出力するのかに応じて、各個体に適合度を与える。そして、遺伝的アルゴリズム(GA)で用いられる高い適合度をもつ個体の選択と、選択した個体を用いた交叉、突然変異を行うことで新たな個体を作成し、さらに高い適合度を持つ個体へと進化させる。GRAPE の持つ個体数と同じだけの新個体を作成した後、それを元の個体と入れ替え世代交代させることで、個体の集団が徐々に正しいプログラムへと進化していく。

GRAPE では多様な個体を用意することでより効率良く目的のプログラムを構築できるとされるため、従来のGAとは異なり、世代交代の方法としてMGG (Minimal Generation Gap) とよばれる手法を採用している。しかし、MGG は、適合度の低い個体の選択を行うことと、親となる個体が次世代に残ることから、大域的最適解への収束の遅れと局所最適解からの脱却の遅れが指摘され

ている。また、GRAPE ではグラフの有効な構造を保持した交叉が出来ないため、交叉の効果が限定的であることも示されている⁽⁹⁾。

そこで本研究では、JGG (Just Generation Gap) とよばれる世代交代モデル⁽⁹⁾を用いてプログラムを進化させる。また交叉は行わず、突然変異のみでプログラムを更新する。JGG では、選択した個体をもとに交叉、突然変異により新たな個体を複数個作製し、その中で最も良い適合度を持つ個体を次世代に残すため、MGG での問題点を解決できるとされる。そして、交叉を行わずに突然変異によりプログラムの一部を変更するだけにとどめ、目的となるプログラムへの収束速度を向上させる。なお、JGG の具体的な手順は3章で述べる。

従来提案されている GRAPE では、出力のノードに至らず、同じ箇所にある命令を実行し続けるという無限ループになることがある。しかし、無限ループとなるプログラムは問題の正しい解とはならない。そのため本研究では、無限ループが生じた場合、それを破棄してプログラムを作成しなおすことで、自動生成の効率化を図る。

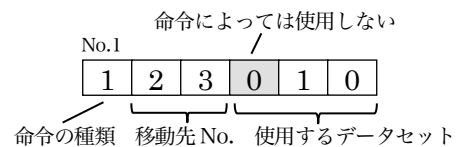


図1 GRAPE のノード

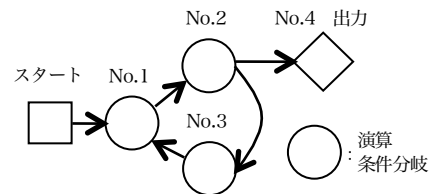


図2 プログラム構造の例

2.2 行動主体間のネットワーク構造

現実の世界におけるネットワークは、小さい頂点間距離と大きいクラスター係数を持つ。頂点間距離が小さいことは、ほとんどのノードが他のノードとのつながりを持たないが、ごく少数のノードが非常に多くのノードとのつながりを持つことを意味する。また、クラスター係数が大きいとは、あるノードとつながりのあるノードどうしがつながっている割合が大きいことを意味する。

現実のネットワークを分析する分野は複雑ネットワークとよばれ、コンピュータ内部にネットワークを構築するさまざまな方法の提案を行っている⁽¹⁰⁾⁽¹¹⁾。代表的な手法として、BAモデルや頂点非

活性化モデルがあるが、それぞれの手法のみでは小さい頂点間距離と大きいクラスター係数を持つネットワークを表現できない。その問題点を解決した手法として、ネットワーク構築の際、BA モデルと頂点非活性化モデルの2つを、指定した確率に応じて使い分けて構築する、SW 頂点非活性化モデルが提案されている。このモデルは、ネットワーク構築の際にノードを追加するとき、BA モデルと頂点非活性化モデルをどの程度の確率で使用するのかをパラメータにより調整することで、小さい頂点間距離と大きいクラスター係数を再現できる。本研究では、この手法により実社会の行動主体間のネットワークをモデル化する。

現実の社会をコンピュータ内部で模倣するには、現実のネットワークと同様の構造を用いることが重要である。ただ、現実とは異なるネットワーク構造からどのような結果が得られるのかを知り、それを現実のネットワークの構築や変更にフィードバックすることも必要と考えられる。

そこで本研究では、さまざまなネットワーク構造を構成し、その構造が行動ルールに与える影響を調べる。具体的には、先のSW 頂点非活性化モデルだけでなく、すべてのノードが均等な数だけ他のノードとの接続を持つ、円環モデルを用いてシミュレーションを行う。

3. シミュレーション

提案手法を用いてシミュレーションを行った。シミュレーションのプログラムは Java version 1.6.0_26 で作成および実行した。なお、本論文での提案手法の有効性を検討するため、予備実験において階乗と最大公約数を実現するプログラムの自動生成を行った。試行を 10 回行ったところ、すべての試行で正しいプログラムを作成できた。GRAPE を使用した他の文献では、階乗が自動生成された回数が 6 割程度⁽¹²⁾である。本論文で GRAPE に採用した JGG と突然変異のみの進化が、従来手法よりも目的とするプログラムを高い確率で作成できることから、提案手法には従来手法と比較して有効性があると考えられる。

3.1 問題とシミュレーションの設定

シミュレーションでは、行動主体の総数を 100 とし、各行動主体間を指定したネットワークにより接続する。各主体には 0-99 の番号がつけられており、それぞれ 1 つの GRAPE の個体を持ち、そのプログラムから行動を決定する。図 1 のように、GRAPE の 1 つのノードは 1 つの番号と 6 つ

の整数値を持ち、それを 50 個用意してプログラムを表現する。

最初、各個体のノードの値はランダムに設定する。囚人のジレンマゲームは、接続のある主体どうしが 30 回繰り返す。行動に応じて主体が獲得する利得は、表 1 のとおりである。行動は、それぞれの主体が持つ GRAPE のプログラムにより決定する。

シミュレーションの 1 ステップでは、最初に全ての主体からランダムに 10 個を選択する。そして JGG の手順にしたがい、選択した主体のプログラムを進化させる。プログラムの進化は、選択した主体と接続する主体のすべてからランダムに 1 つを選び、その個体を持つノードの整数値を 2% の確率で他の数値に変更する（突然変異させる）ことで行う。ただし、突然変異は 90% の確率で実行し、10% の確率で元の個体をそのまま残す。このような新たな個体を 100 個作成し、接続のある全ての主体と 30 回の囚人のジレンマゲームを行い、最も高い利得を獲得した個体を、選択した主体の新しいプログラムとして採用する。これを 1 ステップとして 10000 回繰り返す、得られた結果を分析する。なお、ノードを突然変異させる確率（2%）は文献⁽¹⁾において望ましいとされる値を、突然変異を行う確率（90%）は低い確率で元のノードを残すことを意図している。

本研究では、行動主体の行動が協調か裏切りの 2通りであることから、協調を true、裏切りを false の Boolean 型とよばれる 2 値に対応させる。それにとともに、各命令で使用する値も true と false に限定した。

GRAPE で使用するデータセットには 12 個の Boolean 型を用意した。そして、ある行動主体が対戦する際、対戦の前に自己と相手の 2 回前までに実行した行動をデータセットに記憶させる。ただし、それぞれが接続する他の主体は異なるため、対戦開始前のデータセットの値は、前回の対戦時の最後の 2 つの行動と等しいとは限らない。なお、データセットには No.0-11 がつけられており、最初の 3 つ (No.0-2) に自己の 1 ステップ前の行動、次の 3 つ (No.3-5) には相手の 1 ステップ前の行動、その次の 3 つ (No.6-8) には自己の 2 ステップ前の行動、最後の 3 つ (No.9-11) には相手の 2 ステップ前の行動を記憶させた。

GRAPE のノードは、表 2 にもとづいて実行するための命令に変更する。表 2 において「ノードの値」は、ノードにおいて一次元状にならぶ整数の先頭の値を意味する。各ノードの値は、その番

号に応じた命令に変換される。AND, OR, NOT は、ノード中に含まれるデータセットの番号のうち、最初の2つ(NOTは1つ)のデータセットの値を用いて論理演算を行い、3つ目のデータセットに計算値を代入する。そして、1つ目の移動先 No. のノードに移動する。SWAP は、最初の2つで指定するデータセットの値を交換し、1つ目の移動先 No. のノードに移動する。IF は、最初の2つのデータセットの値を比較し、等しければノード中の移動先のうち最初のノード No. に、等しくなければ2つ目のノード No. に移動する。OUTPUT は、1つ目で指定するノードの値を出力する。

なお、本シミュレーションでは行動ルールの作成時には無限ループを排除しているが、自身のルールを変更したことで出力する行動が変化すると、それに応じる対戦相手のルールが無限ループとなることがある。そこで、500個のノードを処理しても終了しない場合には強制的にプログラムを停止し、false を出力させた。これによって協調が続きにくい社会においても、裏切りばかりが実行される状況にならないかを検討する。

表 1 獲得する利得

行動 (自分, 相手)	利得 (自分, 相手)
(協調, 協調)	(3, 3)
(協調, 裏切り)	(0, 5)
(協調, 裏切り)	(5, 0)
(裏切り, 裏切り)	(1, 1)

表 2 GRAPE で使用する命令

先頭の値	命令	使用するノードの値
0	AND	移動先1つ データセット3つ
1	OR	移動先1つ データセット3つ
2	NOT	移動先1つ データセット2つ
3	SWAP	移動先1つ データセット2つ
4	IF	移動先2つ データセット2つ
5	OUTPUT	データセット1つ

3.2 構築するネットワーク

主体間のネットワークには、すべてのノードが同一のノード数を持つ円環モデルと、SW 非活性化モデルを構築し、得られる結果の違いを分析した。円環モデルでは、接続するノード数を 2, 10,

20, 50 として、他の行動主体との接続が増えることの効果を分析した。

SW 非活性化モデルでは、最初に一部の主体間で完全グラフを作成する。そしてノードを追加していく際に、指定した確率に応じて BA モデルか非活性化モデルを選び、指定した数のリンクを作成済みのリンクに接続する。本研究では、パラメータである（完全グラフ数, 接続ノード数, 選択確率）が(4, 3, 0.5)の場合と(10, 5, 0.5)の2通りについてシミュレーションを行った。前者は接続数が多く、後者は少なくなるため、その違いが与える影響を分析する。

3.3 シミュレーション結果

各パラメータごとにシミュレーションを複数行った。その結果は、ネットワークの構造や戦略が毎回変わることから異なるものとなったが、特徴はパラメータごとに同様となることから、任意の回の結果を示す。

円環グラフでの接続ノード数が 2, 10, 20 のときの、1 ゲームあたりの平均利得と全主体中の最大利得を図 3-5 に示す。図 3-5 では、横軸が経過ステップ、縦軸が利得の値で、平均利得を実線で、最大利得を+で表示している。また図 6-8 は、横軸が経過ステップ、縦軸が協調した主体の割合である。接続ノード数が増えるにつれて、最大利得が 3（双方が協調）となることが多くなり、5（一方が裏切り、他方が協調）となる状況が減少していることがわかる。図 6-8 の協調率も同様で、接続ノード数が増えるにつれて、協調率が 1 に近づくケースが多く見られる。

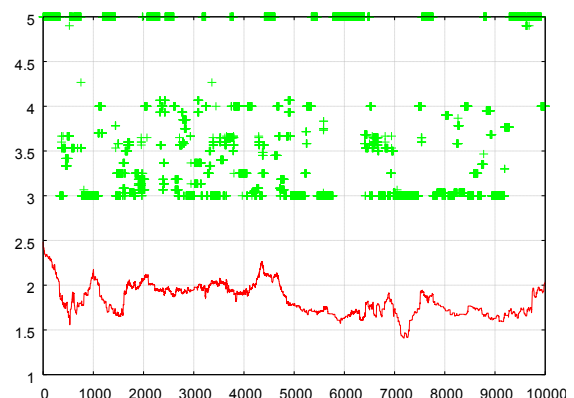


図 3 円環グラフ（接続ノード数 2）の
平均利得と最大利得

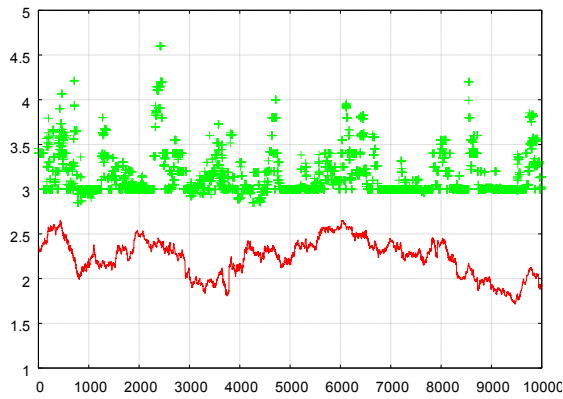


図4 円環グラフ（接続ノード数 10）の
平均利得と最大利得

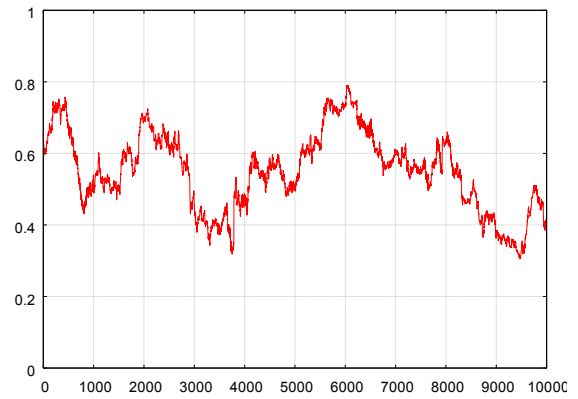


図7 円環グラフ（接続ノード数 10）の協調率

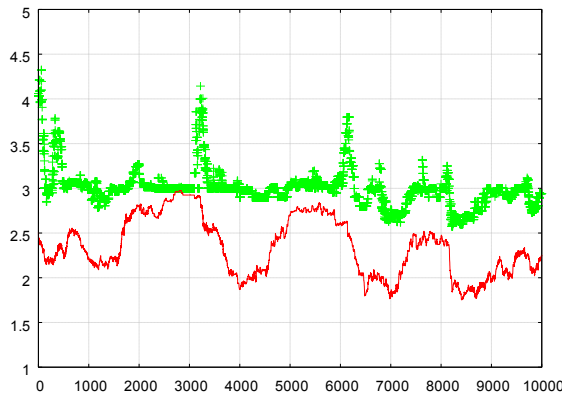


図5 円環グラフ（接続ノード数 20）の
平均利得と最大利得

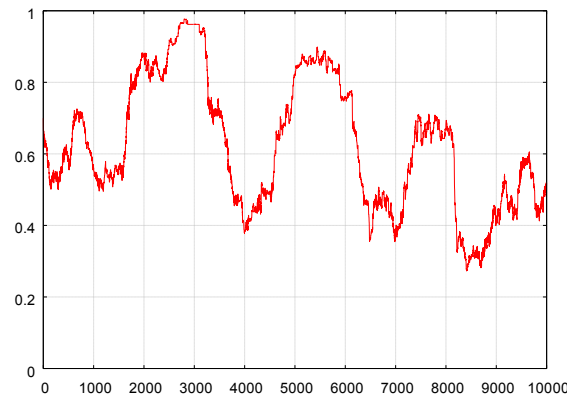


図8 円環グラフ（接続ノード数 20）の協調率

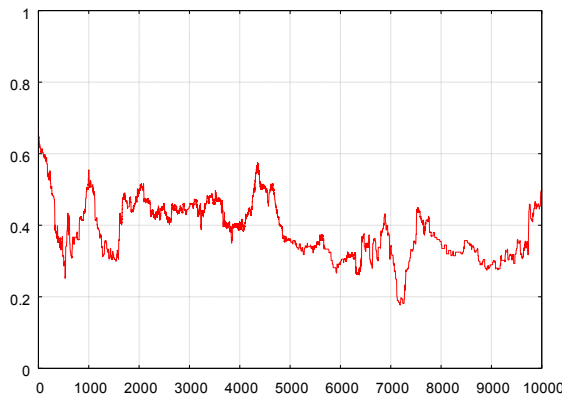


図6 円環グラフ（接続ノード数 2）の協調率

しかし、接続ノード数をさらに増やしていくと、高い協調率を維持する状況だけでなく、低い協調率が続く状況も発生した。図9, 10 に、完全グラフ（全ノードが接続）において高い協調率と低い協調率が生じた際の平均利得と最大利得を示す。図9では平均利得が3に近い状況が非常に多く発生しているが、図10ではほとんど1近くのままとっている。同様の傾向は、接続ノード数を50とした円環グラフにおいてもみられた。

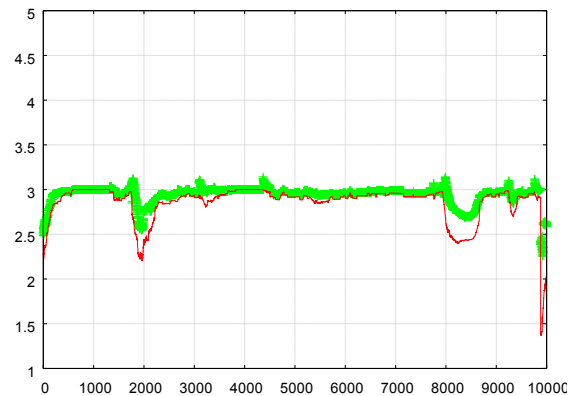


図9 完全グラフの平均利得と最大利得
（高い協調率が続く場合）

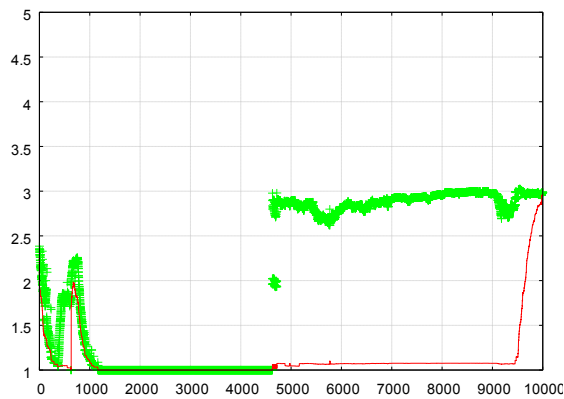


図 10 完全グラフの平均利得と最大利得
(低い協調率が続く場合)

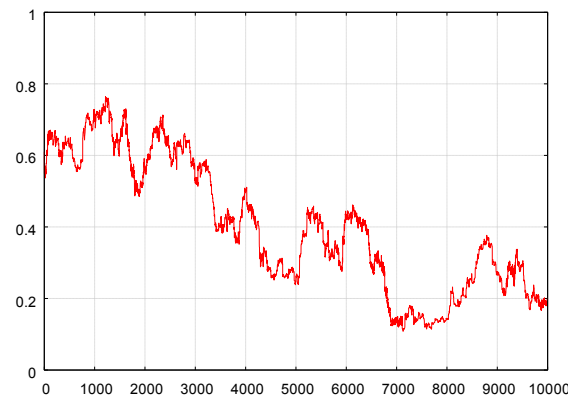


図 12 SW 非活性化頂点モデル(パラメータ(4, 3, 0.5)) の協調率

SW 非活性化頂点モデルのパラメータを(4, 3, 0.5)としたときの平均利得および最大利得は図 11 のようになった。平均利得が 2.5 以下と低く、最大利得も 3 から離れ、裏切りが実行されている状況が多い。協調率は図 12 のとおりであるが、あまり高い値にはなっていない。しかし、パラメータを(10, 5, 0.5)としたときの平均利得および最大利得の結果である図 13 をみると、平均利得が 3 に近づく状況が生じている。また協調率も 1 に近づく状況がみられる。

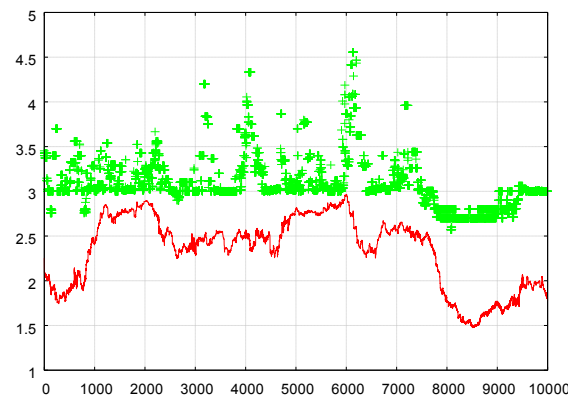


図 13 SW 非活性化頂点モデル (パラメータ(10, 5, 0.5)) の平均利得と最大利得

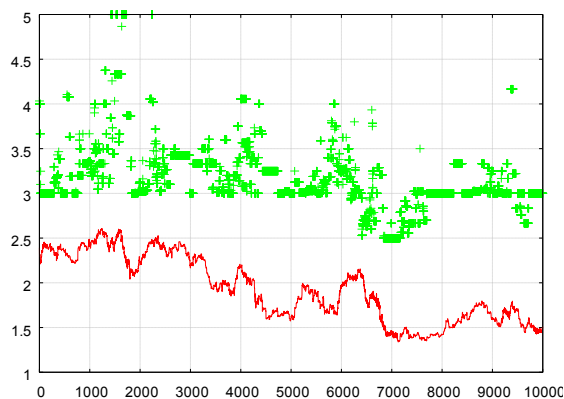


図 11 SW 非活性化頂点モデル(パラメータ(4, 3, 0.5))の平均利得と最大利得

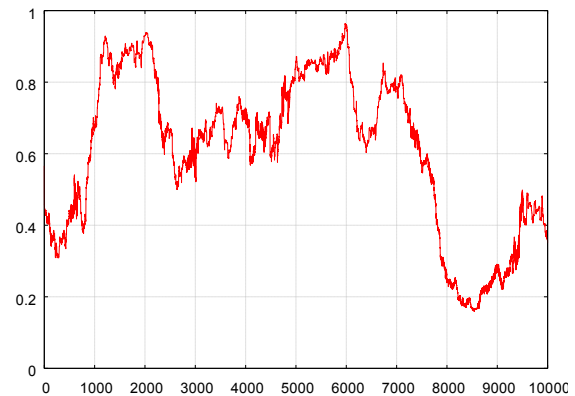


図 14 SW 非活性化頂点モデル (パラメータ(10, 5, 0.5)) の協調率

個々の行動主体がどのように行動ルールを変更しているかを示す。図 15, 16 は、それぞれ GRAPE により構築された、パラメータを(4,3, 0.5)としたときの SW 非活性化頂点モデルにおける行動主体 No.0 と 1 の行動ルールである。図 15 が 3636 ステップ、図 16 が 3637 ステップのルールである。図中の\$に続く値は、使用するデータセットの No. で、行末の()内の値はプログラムの行番号である。

このシミュレーションでは 3635 ステップまで双方が裏切りを実行し続けたが、3636 ステップで主体 No.1 が anti-TFT 戦略を実行するように変化し、それによって双方が協調と裏切りを交互に実行する結果に変化した。それが 3804 ステップまで続き（途中、主体 No.0 の戦略が変化するが対戦の結果は変化しないので割愛する）3805 ステップに主体 No.1 のルールが図 12 のとおり、自分が 1 回前に協調していれば協調するように変化したことで双方が協調するようになり、協調が続く結果となった。この結果の詳細な考察は次節で行う。

```

行動主体 No.0
output $3 (1)

行動主体 No.1
$5=not $9 (1)
$5=$5 or $0 (2)
$7=$10 or $4 (3)
$7=$4 and $8 (4)
swap $1 $9 (5)
swap $2 $3 (6)
if $3==$4 node=12 else node=26 (7)
swap $3 $7 (8)
$5=not $7 (9)
swap $2 $9 (10)
output $5 (11)

```

図 11 構築されたルール (3636 ステップ)

```

行動主体 No.0
output $3 (1)

行動主体 No.1
$5=$9 or $11 (1)
$5=$1 or $0 (2)
$7=$9 or $4 (3)
$6=$3 and $8 (4)
swap $1 $9 (5)
swap $2 $3 (6)
if $4==$6 node=12 else node=26 (7)
swap $1 $10 (8)
if $4==$0 node=15 else node=21 (9)
if $7==$0 node=14 else node=22 (10)
$0=not $4 (11)
$6=$2 or $2 (12)
output $5 (13)

```

図 12 構築されたルール (3805 ステップ)

3.4 考察

図 3-8 で示した円環モデルの接続数 2, 10, 20 の結果から、他の主体との接続が増えるにしたがい協調する割合が増えることがわかる。これは、

裏切りを実行する行動ルールを獲得したとしても、協調を実行する主体からは高い利得を得られるが、つながりが多くなるとお互いが協調を実行しなければ結果として高い利得を得ることができず、戦略を変更するためと考えられる。

しかし、図 9, 10 の完全グラフの結果を見ると、接続が多すぎると、ほとんどの主体が協調あるいは裏切りを実行するといった、極端な傾向となることがわかる。全主体が接続しているため、特定の戦略が社会に浸透しやすいことが原因と考えられる。この傾向は、図 9 の 10000 ステップ近くで急激に平均、最大利得が落ち込んでいることや、図 10 の 9500 ステップあたりから平均、最大利得が大きく上昇しているところからも見てとれる。これは、接続が多い場合、裏切りが続く中である主体だけが協調しようとしても、そのような戦略は高い利得が得られないために社会に浸透せず、裏切りが続く。しかし、偶然一部の主体どうしが協調しあう戦略が同時に生じた場合、その主体どうしの対戦では 90 の利得を獲得でき、他の裏切りのみを実行する主体よりも高い利得が獲得できることからその戦略が模倣され、全体に浸透していくと考えられる。

SW 非活性化モデルにおいてパラメータ（完全グラフ数、接続ノード数、選択確率）を(4, 3, 0.5)とした場合、円環モデルの接続ノード数が 10 や 20 の結果と比較すると、協調する割合が大きく減少する。SW 非活性化モデルでは一部の主体のみ大きな接続ノード数を持ち、ほとんどの主体の接続ノード数が 3 であるため、接続ノード数の多い主体の影響が波及せず、円環グラフの接続ノード数が少ない場合と同じ状況になったためと考えられる。それに対してパラメータを(10, 5, 0.5)とした場合、ほとんどの主体の接続ノード数は 5 と若干増えた程度であるものの、接続ノード数を多く持つ主体の影響が他の主体に影響を与え、円環グラフの接続ノード数が多い場合と同様の状況が生じて協調する割合が多くなったと考えられる。この結果から、実社会でもほんの少し他者と多く接する機会を増やすことが、互いの協調を生み出すことにつながると期待できる。

裏切りのみ実行する状況から協調を実行する状況に変化する過程は、協調と裏切りを交互に実行する状況に移行することがきっかけとなる。これは、一方のルールが変化せずとも、他方が戦略を実行することから始まっている。図 11 では、主体 No.1 の(1)で相手の 2 回前の行動と逆の行動を求め、(2)でそれと自分の 1 回前の行動のうち一方

でも協調であれば(11)で協調を出力する、anti TFT とよばれるプログラムである¹。これによって、裏切りのみ実行される状況から、主体 No.1 が協調を実行するようになる。主体 No.0 は TFT 戦略であるため、主体 No.0 が協調なら No.1 は裏切り、主体 No.0 が裏切りなら No.1 が協調となる状況が続く。これによって、裏切りを続けるより高い利得を得るようになるため、主体 No.0 と No.1 以外の主体にもこのルールが模倣される。

協調と裏切りを繰り返す状況から協調のみを実行する状況への変化は、図 12 から把握できる。図 12 において主体 No.1 のルールは、(2)で自分の1回前の行動をデータセット No.5 に入れ、それを(13)で出力するプログラムとなっている²。それに対して主体 No.0 は TFT 戦略であるが、対戦前に記憶される前回の主体 No.0 の行動は裏切り、No.1 の行動は協調であることから、対戦の最初の行動は双方共に協調となる。それ以降は双方が協調を出し続けるため、協調が続いていく。

これらの結果は、表 2 に示した GRAPE で使用する命令を変更して構成されるプログラム内容が変わると異なるものになることも予想される。また、プログラム生成の際に、構成されるプログラムの長さを調節することも行動ルールに影響を与えると想定できる。上記の点についての検証が今後の課題である。

4. おわりに

本論文では、さまざまなネットワークを通じて接続された複数の行動主体の行動ルールを、コンピュータ内部でプログラム自動生成手法である GRAPE により構築し、繰り返し囚人のジレンマゲームによる対戦を行った。そして、提案モデルによるシミュレーション結果から、どのようにして協調が生じるのか、また異なるネットワークを用いた結果からどのような要素が協調を生み出すことに有効であるのかを分析した。

得られた結果から、実社会におけるネットワークで、少しの他者とのつながりが協調を生み出す可能性と、裏切りのみが実行される状況から協調が生じる過程へのプログラムの変化とその詳細な分析結果を示した。

参考文献

- (1) 川越敏司：「実験経済学」、東京大学出版会 (2007.10)
- (2) 塚本鋭、内田誠、白山晋：“協調の進化に与える初期ネットワーク構造の影響” 人工知能学会論文誌、24 巻、5 号、pp.438-445 (2009)
- (3) 脇山宗也、谷本潤、相良博喜、萩島理：“社会構造が持続可能型社会創発可能性に及ぼす影響” 日本建築学会論文集、73 巻、628 号、pp.831-838 (2008.6)
- (4) 谷本潤、相良博喜：“戦略とネットワークの共進化による協調の創発と Assortative mixing” 信学技報、AI2007-49、pp.31-36 (2008.3)
- (5) 星野力：“進化論は計算しないとわからない”、共立出版 (1998.7)
- (6) 白川真一、長尾智晴：“Graph Structured Program Evolution による複雑なプログラムの自動生成とその解析” 情報処理学会研究報告、2007-MPS-66、pp.21-24 (2007.9)
- (7) 電気学会 G A 等組合せ最適化手法応用調査専門委員会編：“遺伝的アルゴリズムとニューラルネット”、コロナ社 (1998.2)
- (8) 石堂真大、白川真一、長尾智晴：“グラフ構造プログラム自動生成手法のための子個体生成方法の提案” FIT2009、A-001 (2009)
- (9) 小林重信：“実数型 GA のフロンティア” 人工知能学会誌、24 巻、1 号、pp.128-143 (2009.1)
- (10) 増田直紀、今野紀雄：“複雑ネットワークの科学”、産業出版 (2005.2)
- (11) 今野紀雄、町田拓也：“よくわかる複雑ネットワーク”、秀和システム (2008.12)
- (12) 石堂真大、白川真一、長尾智晴：“グラフ構造のプログラム自動生成手法への ADF の導入” 電子情報通信学会総合大会、D-8-8 (2009)

(2011 年 9 月 30 日受付)

(2011 年 12 月 21 日採録)

¹ (1),(2),(11)以外の行は出力には影響を与えないため、実行しても意味のない行である。

² (2),(13)以外は、実行しても出力に影響しない行である。