

プログラム自動生成手法によるジレンマ的問題における社会的な公平性の創発についての分析

Analysis of an Emergence of Social Fairness in Various Dilemma Problems by Using the Automatic Program Technique

今福 啓*

Kei Imafuku

Email: k03082@dokkyo.ac.jp

本論文では、実社会において生じるさまざまなジレンマ的問題において、いかにして個人合理的な行動を回避し、社会的に公平な行動を生み出すのかを分析する。そのため、コンピュータ内部に人の行動を模倣する行動主体を複数作成し、その行動ルールをプログラムを自動的に生成する手法の一つである GRAPE(Graph Structured Program Evolution)により構築する。GRAPE では、多様な個体を内部に持つことが、より効率良くプログラムを作成する際に重要とされる。そこで本論文では、GRAPE で個体の多様性を維持する手法を適用し、従来よりも高い効率性を持つ行動ルールの獲得を行う。提案モデルを用いたシミュレーションを行い、得られた結果から手法の有効性について検討する。

In this paper, we analyze how to avoid the individual rationality and yield the fair action in various dilemma problems. For this problem, we create the computer programs automatically that imitate the human's action by using GRAPE(Graph Structured Program Evolution), which is one of the automatic program techniques. To create the computer program efficiently by using GRAPE, diversity maintaining of nodes are important. Therefore, we apply a method to maintain diversity and obtain action rule effectively compared with the existing methods. 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

我々の生活する実社会には、個人にとって望ましい状況を導くように考えて行動しても、それが必ずしも社会全体にとって効率的な結果を導かないジレンマ的な問題が存在する。個人の合理性を超えて社会全体を考え、社会全体において望ましい状況を作るにはどのようにすれば良いのだろうか。人が行動した結果から意思決定の過程を推測し、どのような過程を経て行動を決定するのかを明示することは、非常に困難と考えられる。そのため、従来研究ではジレンマ的な問題を単純化してゲーム的なモデルで表現し、それをを用いた分析が数多く行われてきた。その結果、社会のネットワーク構造¹⁾、行動を模倣する範囲²⁾、ネットワーク構造と行動を同時に模倣すること³⁾が行動に影響することが明らかとなっている。

本研究では、人の意思決定過程を模倣した行動主体をコンピュータプログラムで表現し、それを実社会と同様に他者からの影響を受けて進化、適応させて自動生成することで、ジレンマ的な問題において個人にとって合理的な状況を避け、社会全体にとって望ましい状況を実現可能であるかを分析する。従来研究では、行動決定が過去の行動をもとに作成したテーブルを用いる単純な手順で行われてきたのに対し、本研究ではプログラムの形式で獲得できる点が特徴である。そのため、得られたプログラムを見ることで、どのような手順により行動が決定されているのかを、論理的に分析できるという利点がある。本手法を用いた分析により、我々の社会におけるさまざまな問題を解決するにあたり、どのような行動規範が必要であるのか、どのような制度設計が考えられるのかといった意思決定の指標となると考えられる。

先行研究⁴⁾では、コンピュータプログラムの自動生成手法である GRAPE⁵⁾を用いて、行動ルールを生成する手法を提案した。また、行動主体間を実社会と同様のネットワーク構造で接続し、その構造の違いが行動に与える影響を分析した。先行研究では、くり返し囚人のジレンマ問題を扱い、行動ルールが時間経過とともに変化していく過程を、プログラムの変化を分析することから示した。そして、各主体が実社会と同様のつながりを持つ際、少しの他者とつながりが増えることが社会において望ましい協調行動を生み出す可能性があることを明らかにした。しかし、得られた結果では高い協調率が生じたものの、裏切りが実行される割合も少なからず残り、望ましい社会の持続性に疑問が残る結果となった。

その原因の一つとして、GRAPEにおいてプログラム作成に使用する「個体」の多様性の不足が考えられる。目的となるプログラムを効率良く作成するためには、個体が多様性を持ち、さまざまな内容を表現できることが重要とされる⁶⁾が、先行研究ではそれが十分ではなかったと想定される。

そこで本研究では、GRAPEにおいて個体の高い多様性を維持できる手法⁷⁾を用いて行動ルールの作成を行う。この手法では従来手法と比較して、さまざまなプログラムを高い割合で正しく作成できることから、本

研究においてもより効率良く行動ルールの生成が可能になると考えられる。

本研究ではこのような手法により、いくつかのジレンマ問題において、実社会と同様のネットワーク構造で接続された行動主体が、他者との相互作用から GRAPE により自動生成される行動ルールを時々刻々と変化させる状況で社会的な望ましさを得られるのか、得られるならばどのような要素が必要であるのかを、コンピュータシミュレーションの結果を通じて明らかにする。

以下では、2章で行動ルールを自動生成するための GRAPE と、GRAPE の個体の多様性を維持する手法、行動主体間のネットワーク構築手法、扱うジレンマ問題について述べる。そして3章ではシミュレーションの条件と結果に加え、その考察を行う。

2. プログラム自動生成手法と個体の多様性維持のための手法

本研究で行動主体の行動決定に用いるプログラム自動生成手法である GRAPE と、構築目的のプログラムを効率良く作成するための手法について述べる。また、行動主体間のつながり方を決定するネットワーク構築手法と、本研究で扱うジレンマの問題についても述べる。

2.1 プログラム自動生成手法

例えば最大公約数を求めるといった、与えられた目的のプログラムを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、それを実現する手法の一つである「Graph Structured Program Evolution (GRAPE)⁵⁾」を用いて、さまざまなジレンマ問題での行動ルール作成を行う。

GRAPE は、図 1 に示すように「処理内容 (命令)」と使用する「データセット (変数)」を示す「ノード」から構成され、これを一次元状に複数つなげた「個体」によりプログラムを表現する。個体は、決められたルールにもとづいて図 2 に示すような演算、条件分岐、使用するデータセットの値といった、グラフ構造のプログラムに変換される。プログラムは最初のノードから指定された順番に従って実行され、出力を示す処理ノードに達すると、そのノードで指定されたデータセットの値を出力して実行が終了する。

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、実行結果が目的の値となった場合、正しいプログラムが作成できたとする。例えば x^2 を計算するプログラムを作成する場合、データセットに 1, 2, 3, 4, 5 を順に代入したとき、プログラムの実行結果がそれぞれ 1, 4, 9, 16, 25 と正しい値になったならば、正しいプログラムが得られたものとする。

個体の良し悪しを表す評価値は、通常の GRAPE ではプログラムの出力値と正しい値との差から計算するが、本研究では 3.3 節で述べるとおり、ジレンマゲームをくり返し行った後に獲得した利得の合計が大きいほど評価値が高いものとする。

プログラムは、ノードの一部に対して突然変異とよ

ばれる変更を加えて複数の個体（子個体）を生成し、元の個体（親個体）と入れ替えることを複数回くり返し進化させる。この操作により、個体群を正解となるプログラムに近い内容に徐々に進化させていく。

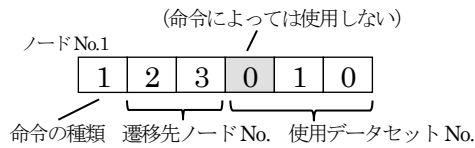


図1 GRAPEのノード

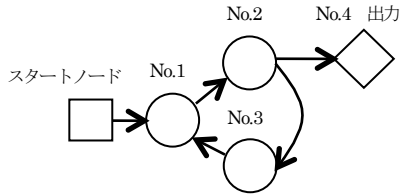


図2 GRAPEのプログラム構造

2.2 個体の多様性維持のための世代交代モデル

プログラム作成において、新たな子個体を生成して親個体と入れ替える手順を「世代交代モデル」と呼ぶ。GRAPEでは、効率的な行動ルールを導くため、多様な命令とデータセットからなる構造が必要となる。効率良く目的のプログラムを作成するには、個体群の内容が多様な要素から構成されるよう維持することが重要とされている。そのための手法として、従来は実数型GAにおいて解を効率良く求めることのできるMGG⁽⁹⁾（Minimal Generation Gap）やJGG⁽¹⁰⁾（Just Generation Gap）が用いられてきた。いずれの手法も、親個体群から一定数の個体を選択し、新たな子個体群を作成した後に、正しいプログラムに近い、すなわち評価値の高い個体を選択して親個体群と入れ替える。しかし、MGGは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。JGGは一部の問題においてMGG以上に高い性能を持つが、時間とともに元の個体群が作成した子個体群で占められていくことが示されている⁽¹¹⁾。そのため、この2つの手法は、個体群の多様性維持よりも、評価値の向上を重視した世代交代モデルになっていると考えられる。

そこで本研究では、GRAPEにおける個体群の多様性が維持できる手法⁽⁷⁾により行動ルールの生成を行う。この手法では、親個体と子個体を入れ替える際、選択した親個体から作成された直系となる子個体群のみを、その親との入れ替え対象とする。これにより、親個体が評価値の高い子個体で占められることがなく、親の

持つ多様性を維持することが可能となる。

手順の概要を図3に示す。選択した親 i ($i = 1, \dots, np$)から、子個体として $子_{i,1}, \dots, 子_{i,nc}$ を作成する。そして、そのうち評価値が最大の個体を選び、それを親 i に代入する。

この手法を用いてさまざまな問題のプログラムを自動生成した際に、正解となるプログラムを作成出来た割合を表1に示す。表1では、一行目は参考文献の番号を表している。従来手法⁽⁶⁾と比較して、GRAPEの多様性を維持する手法⁽⁷⁾により、幅広い問題において高い割合で目的のプログラムを獲得できていることが分かる²⁾。

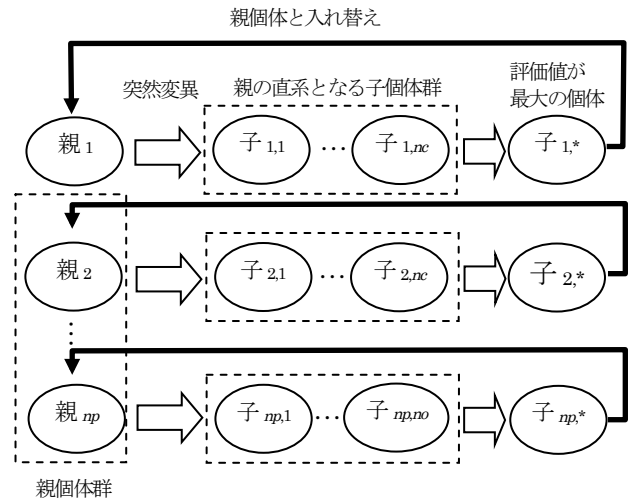


図3 個体の多様性を維持する手法（概要）

GRAPEでは、子個体を作成して評価値を求めるために実行した際、プログラムが出力を示す処理ノードに至らず、同じ箇所にある命令をくり返し実行し続ける「無限ループ」と呼ばれる状態になることがある。無限ループとなるプログラムは問題の正しい解ではないため、本研究では一定の時間経過しても出力を処理するノードに至らない場合、実行を停止してその子個体を破棄し、改めて子個体を作成することで、さらなるプログラムの効率的な生成を図る³⁾。

表1 さまざまな問題における各手法の成功回数

	(6)	(5)	(8)
階乗	98	69	68
累乗	94	45	99
最大公約数	87	結果なし	結果なし
フィボナッチ数列	67	8	12

2.3 行動主体間のネットワーク構造

人と人のつながりのように、実社会でみられるさま

¹ 遺伝的アルゴリズムのような一般的な進化的計算手法では交叉も用いるが、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が示されている⁽⁹⁾ため、本研究では突然変異のみを用いる。

² 文献⁽⁶⁾では最大公約数を自動生成する問題を扱っていないため、成功回数を「結果なし」とした。

³ ただし、対戦相手の実行する行動により実行結果が変わるため、異なる相手との対戦時には無限ループとなることがある。

さまざまな意思決定主体間のネットワーク構造をモデル化して分析する分野を「複雑ネットワーク¹²⁾」という。複雑ネットワークでは、頂点間距離とクラスター係数という指標を用いて構造の分析を行っている。実社会の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ことから、同様の構造をコンピュータ内部に構築できれば、実社会での主体間のつながりを表現できたといえる。

より実社会に近い状況を想定したシミュレーションを行うため、本研究では「SW 頂点非活性化モデル」により行動主体間を接続する。このモデルは「BA モデル」と「頂点非活性化モデル」という、複雑ネットワークにおける代表的なネットワーク構築手法を確率的に選択して採用することで、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。

2.4 ジレンマ的な問題と行動主体の行動

本研究では、行動主体の行動分析のため、実社会で生じるジレンマ的な社会状況を単純化した問題を用いる。行動主体は、表2~4に示すGame 1~3の各問題⁴⁾においてTRUEかFALSEのいずれかの行動を同時に実行し⁵⁾、それぞれの行動に応じて利得を獲得する。これを「対戦」とよぶ。

各問題において一つの行動主体にとって合理的な行動は、Game 1では双方がFALSEを実行する場合、Game 2では4割の割合でTRUEを実行するか、片方がTRUEで他方がFALSEを実行する場合、Game 3では6割の確率でTRUEを実行するか、双方がTRUEまたはFALSEを実行する場合である。ただし、全行動主体が公平に最も高い利得を獲得できる社会的に効率的な状況は、Game 1では双方がTRUEを出し続ける場合、Game 2では一方がTRUE、他方がFALSEを出し、次の対戦では双方が出す手を入れ替えることを交互にくり返す場合で、Game 3では双方がTRUEを出す場合とFALSEを出す場合をくり返す場合である。このように、Game 2と3では一定の割合でTRUEとFALSEを変化させることが個人的にも社会的にも合理的なため、以下のシミュレーション結果では獲得される利得が変動する。

表2 Game 1における利得表

行動 (自分, 相手)	利得 (自分, 相手)
(TRUE, TRUE)	(3, 3)
(TRUE, FALSE)	(0, 5)
(FALSE, TRUE)	(5, 0)
(FALSE, FALSE)	(1, 1)

⁴⁾ Game 1~3の具体的な問題は、それぞれ「囚人のジレンマゲーム」「タカハトゲーム」「恋人のジレンマゲーム」に相当する。

⁵⁾ シミュレーションプログラムの実行時に、問題ごとに行動を統一的に扱うため、行動をBoolean型のTRUEとFALSEの2値に統一した。

表3 Game 2における利得表

行動 (自分, 相手)	利得 (自分, 相手)
(TRUE, TRUE)	(2, 2)
(TRUE, FALSE)	(0, 5)
(FALSE, TRUE)	(5, 0)
(FALSE, FALSE)	(-2, -2)

表4 Game 3における利得表

行動 (自分, 相手)	利得 (自分, 相手)
(TRUE, TRUE)	(2, 2)
(TRUE, FALSE)	(0, 0)
(FALSE, TRUE)	(0, 0)
(FALSE, FALSE)	(3, 3)

3. シミュレーション

提案モデルを用いたシミュレーションを行った。シミュレーションでは、GRAPEのプログラムにしたがって行動を決定するNo.0~99に番号付けされた総数100の行動主体を用意し、行動主体間を2.3節で述べたSW 頂点非活性化モデルにより接続した。各行動主体は1つのGRAPEの個体を持ち、そのプログラムから行動を決定した。

なお、以下に述べるとおり、初期時刻においてGRAPEの個体をランダムに設定するため、シミュレーション結果は毎回異なるものとなる。そのため、シミュレーション結果は本研究で示す結果と同一のものが毎回得られるわけではない。

3.1 GRAPEのパラメータ

各行動主体の持つGRAPEの個体は、ノードNo.0~No.49の50個と、TRUE, FALSEのいずれかの値を記憶するデータセット\$0~\$11の12個から構成される。

ノードで使用する命令の種類は表5のとおりである。AND, ORは、ノード(図1を参照)の最初の2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結果を代入する。そして次に処理するノードとして、移動先No.の1つ目で指定するノードに移動する。NOTも同様に、最初のデータセットの値を用いて論理演算を行い、2つ目のデータセットに結果を代入した後、移動先No.の最初のノードに移動する。SWAPは、最初の2つのデータセットの値を交換し、移動先No.の1つ目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、等しければ最初の移動先No.に、等しくなければ2つ目の移動先No.のノードに移動する。=は、2つ目のデータセットに最初のデータセットの値を代入し、移動先No.の1つ目のノードに移動する。OUTPUTは、1つ目で指定するデータセットの値を出力する。

表5の命令において、AND, OR, NOTは論理演算

の完備集合となっており、これを組み合わせることであらゆる入力からあらゆる出力の組み合わせを作ることが出来る。また、AND, OR, NOT だけでは過去の情報をそのまま用いた出力ができないため、SWAP と=によりそれを実現する。そして、過去の状況や計算結果にもとづき状況判断を行い行動を変化させるため、IFを用いる。

各ノードの初期値は、命令の種類は0~6のランダムな値、移動先 No.は0~49のランダムな値、使用するデータセットは0~11のランダムな値を設定した。また、データセットの初期値は全てFALSEとした。

表5 使用する命令

番号	命令	ノードの構成
0	AND	遷移先ノード 1 データセット 3
1	OR	遷移先ノード 1 データセット 3
2	NOT	遷移先ノード 1 データセット 2
3	SWAP	遷移先ノード 1 データセット 2
4	IF	遷移先ノード 2 データセット 2
5	=	遷移先ノード 1 データセット 2
6	OUTPUT	データセット 1

3.2 ネットワークのパラメータ

SW 頂点非活性化モデルを構築する際、最初に一部のノード間をすべて接続した完全グラフを作成する。そして、それ以外のノードからすでに接続を持つノードに、SW モデルか頂点非活性化モデルを確率的に選択して接続していくことを、全ノードにくり返すことでネットワークを構築する。そのため、ネットワーク構築に必要なパラメータは「初期に完全グラフを構築するノード数 n_1 」「接続を持たないノードから他のノードに接続するリンク数 n_2 」「SW モデルを選択する確率 p 」となる。

本研究では、パラメータを $(n_1, n_2, p) = (4, 3, 0.5)$ と $(10, 5, 0.5)$ としてシミュレーションを行った。前者は全エージェントのうち平均的に3%、後者は5%のつながりがあるため、比較的多くの他者とのつながりを持つ状況を想定している。これらのパラメータをさまざまに変化させたときの分析は、今後の課題とする。

3.3 対戦の手順

最初に、対戦を行う行動主体の1つを、全行動主体の中からランダムに選択する。そして、選ばれた行動主体とネットワークを通じて直接つながりを持つ行動主体の中からランダムに1つを相手として選択する。

最初に選択した主体は、自分の持つGRAPEの個体から一定数の子個体を作成する。子個体は、ノードの番号を2%の確率で他の数値に突然変異させ作成する。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。そして、作成した子個体と相手の個体からなるプログラムから行動を決定し、総当たりの対戦を行う。

子個体を用いた対戦では、それぞれの主体は2.4節で述べたGame 1~3のいずれかのゲームで実行できる2つの行動TRUE、FALSEのどちらか一つを同時に提示し、利得表を参照し利得を獲得する。これを30回くり返し行う。対戦の後、各行動主体の子個体が獲得した利得の合計を評価値として、その中で最も高い値を得た子個体を、その親個体と入れ替える。その間、対戦相手は戦略を変更しない。

GRAPEのデータセットには、自分と相手が2回前までに実行した行動を記憶した。データセット\$0-\$2には自分の1ステップ前の行動、\$3-\$5には相手の1ステップ前の行動、\$6-\$8には自己の2ステップ前の行動、\$9-\$11には相手の2ステップ前の行動を記憶させた。ただし、主体が記憶する値は、その主体が直前にゲームを行った相手の行動を記憶したものである。そのため、前回と同じ相手と対戦を行う際でも、その相手が直前の対戦で実行した行動と、記憶する行動が異なることがある。以上をシミュレーションの1ステップとして10000回繰り返す。

シミュレーションでは、行動主体の獲得する利得が時間とともにどのように変化するかを示し、得られたGRAPEのプログラムとあわせて考察を加える。

3.4 シミュレーション結果

Game 1において $(n_1, n_2, p) = (10, 5, 0.5)$ の場合の結果を図4に、 $(n_1, n_2, p) = (4, 3, 0.5)$ の場合の結果を図5に示す。図中の実線(赤色)は、各行動主体に接続する全ての相手と対戦して獲得した利得の平均値⁶、×印(緑色)は各ステップにおける利得の最大値を示す。 $(n_1, n_2, p) = (10, 5, 0.5)$ の場合、4000ステップ付近から全行動がTRUEとなり社会的に望ましい利得3を獲得する結果となったが、 $(n_1, n_2, p) = (4, 3, 0.5)$ の場合、どのステップでもFALSEを実行する行動主体が存在し、TRUEのみが実行される結果には収束しなかった。

他のネットワーク構造がもたらす結果を調べるため、他者との接続数が4の円環グラフにおいてもシミュレーションを行った。その結果を図6に示す。図5と同じく、TRUEのみが実行される状況は生じなかった。

⁶ シミュレーションではGRAPEの個体を更新しなかった行動主体も接続する他者と対戦を行うため、ここでは全行動主体が獲得した利得の平均値を示している。

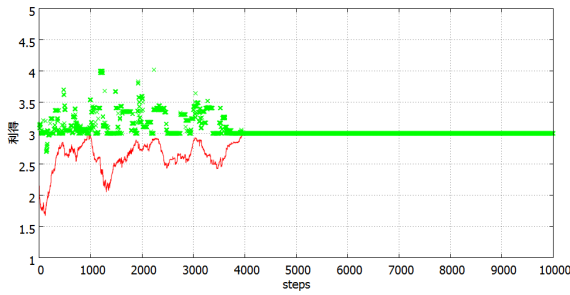


図 4 Game 1 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の最大利得と平均利得

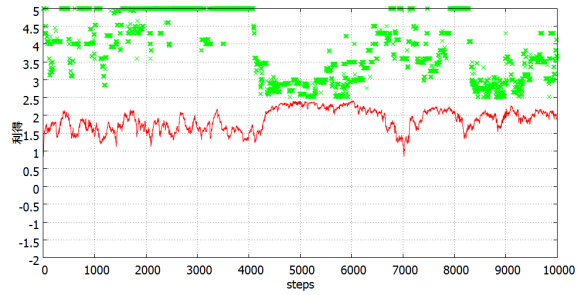


図 7 Game 2 での SW 非頂点活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の最大利得と平均利得

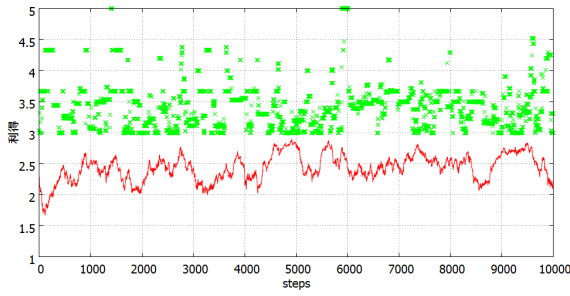


図 5 Game 1 での SW 非頂点活性化モデル $(n_1, n_2, p) = (4, 3, 0.5)$ の最大利得と平均利得

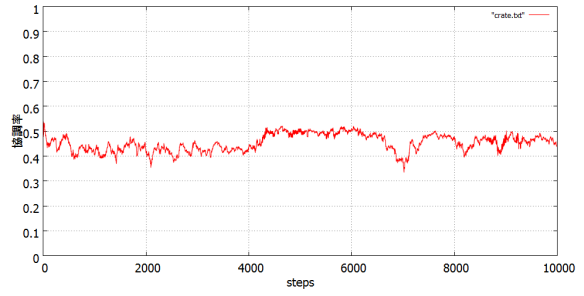


図 8 Game 2 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の TRUE の実行割合

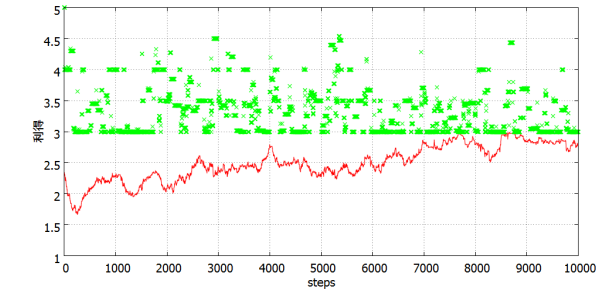


図 6 Game 1 での円環グラフの最大利得と平均利得

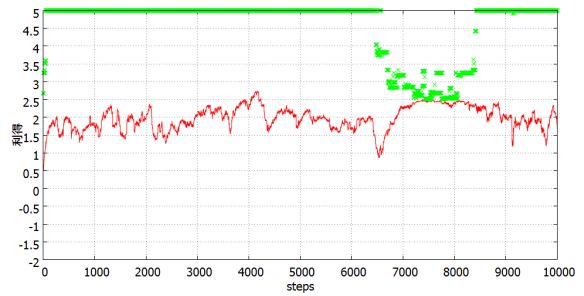


図 9 Game 2 での SW 頂点非活性化モデル $(n_1, n_2, p) = (4, 3, 0.5)$ の最大利得と平均利得

Game 2 の結果を図 7~10 に示す。図 7, 9 はそれぞれ $(n_1, n_2, p) = (10, 5, 0.5), (4, 3, 0.5)$ における利得の平均値と最大値、図 8, 10 は各ステップで TRUE を実行した行動主体の割合である。図 10 より、4500~6500 ステップにおいて、平均利得と最大利得が 2.5、TRUE の実行割合が 0.5 となり、TRUE と FALSE を交互にくり返す混合戦略により社会全体に公平な状況が発生していることがみてとれる。図 9 でも 7000~8000 ステップで同様の状況が生じている。ただし、 $(n_1, n_2, p) = (4, 3, 0.5)$ の際にこの結果が生じるのはまれで、0~6000 ステップのように最大利得が 5 でどちらか一方が FALSE を実行し、かつ TRUE の実行割合が 4 割程度と個人にとって合理的となる状態がシミュレーション期間のすべてで続く結果が多くみられた。

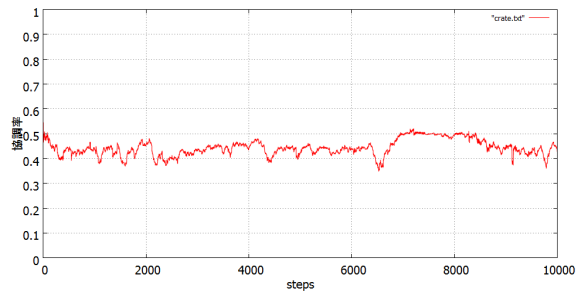


図 10 Game 2 での SW 頂点非活性化モデル $(n_1, n_2, p) = (4, 3, 0.5)$ の TRUE の実行割合

Game 3 において、 $(n_1, n_2, p) = (10, 5, 0.5)$ としたときの平均利得と最大利得を図 11 に、TRUE の実行割合を図 12 に示す。同様に、図 13 は $(n_1, n_2, p) = (4, 3, 0.5)$ のときの平均利得と最大利得、図 14 は TRUE の実行割合である。いずれのネットワークにおいても、早い時刻から全主体が FALSE のみを実行する結果に収束している。初期時刻に選択された主体は FALSE を出

したほうがTRUEを出すよりも高い利得が得られるため、FALSEを出すプログラムを早い時点で獲得し、逆にその主体に接続する他の対戦相手はFALSEを出さなければ利得が得られなくなることから生じている。

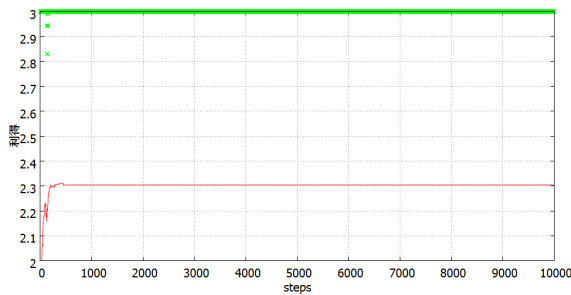


図 11 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の最大利得と平均利得

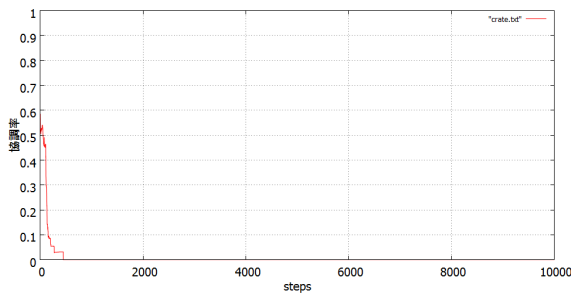


図 12 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の TRUE の実行割合

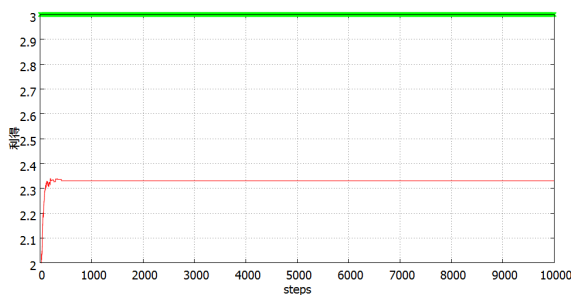


図 13 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (4, 3, 0.5)$ の最大利得と平均利得

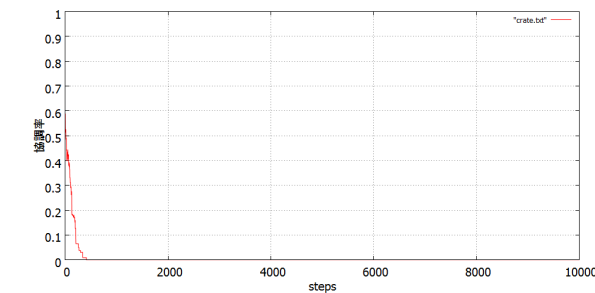


図 14 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (4, 3, 0.5)$ の TRUE の実行割合

3.5 考察

Game 1 の結果を先行研究⁴⁾と比較すると、図 4 より SW 頂点化非活性化モデルで TRUE を実行する割合が大幅に向上していることがわかる⁷。これは、GRAPE の個体の多様性を維持したことが影響していると考えられる。このことから、他者の行動の模倣を効率良く行うことが、より望ましい社会を構築する際には重要になるといえる。

図 5, 6 では、利得の最大値が 3 の場合とそれ以上となる場合とで常に変動している。これは、FALSE を含む行動を実行するプログラムが高い利得を獲得し、そのプログラムが模倣されるが、FALSE が高い割合で実行されると今度は逆に TRUE を高い割合で実行するルールを持つ主体どうしが高い利得を獲得するようになり、それが模倣されるといった変動がくり返されるためと考えられる。しかし図 4 では、ネットワーク構造から FALSE を実行する行動が高い利得を獲得できない状況になっていると考えられる。

このように、図 4 と図 5, 6 の比較から、実社会で構築されるネットワーク構造、特にある程度の大きさのクラスターを形成することが個人の合理性を考えた行動の蔓延を防ぎ、社会全体の公平性につながっているといえる。これは先行研究の結果と同様で、他者とのつながりを平均的に多く持つことが、効率的な社会を構築する上で重要な役割を担っていると考えられる。さらには円環グラフのように画一的なつながりではなく、SW 頂点非活性化モデルのようにつながりに粗密がある社会のほうが、社会全体の合理性を実現する上では望ましいと想定できる。

図 4 で得られた GRAPE のプログラムの進化について考察する。プログラムは、行動主体 No.0 と 1 の持つ内容を示す。図 15 は、195 ステップで実行されたプログラムである。双方が TRUE を実行する状況が 594 ステップまで続いた。いずれの内容も、相手の 2 回前（データセット \$3 と \$5）と 1 回前（データセット \$9）の行動が TRUE のとき、相手の 2 回前（データセット \$10, \$11）の行動、すなわち TRUE を出力する。よって、双方の 2 回前の行動が共に TRUE であれば TRUE を出し続けるが、1 つでも FALSE が含まれるとこの関係は崩れる。

実際 595 ステップにおいて、行動主体 No.0 の過去 2 ステップの行動が、行動主体 No.1 以外の相手との対戦を経て共に FALSE となり、595 ステップから 928 ステップまでは一方が TRUE、他方が FALSE を実行する結果が続いた。595 ステップにおいて、行動主体 No.0 および No.1 のプログラムは、ともに相手の 1 回前の行動（データセット \$4, \$5）を出力するという、いわばしつぱ返し戦略となっていた。

929 ステップでは、行動主体 No.1 の記憶する過去 2 回分の行動が、No.0 以外との対戦を通じて TRUE となった。このとき、行動主体 No.0 のプログラムに大きな変化はなかったが、行動主体 No.1 のプログラムは図

⁷ 先行研究⁴⁾では、TRUE が協調、FALSE が裏切りに対応する。

16のように進化していた。ただし、図16では出力に影響のない不要な行を削除している。これは、自分の2回前の手が等しければ相手の1回前の行動(データセット\$5)を、そうでなければ相手の2回前の行動(データセット\$9)を反転して出力する内容である。そのため、行動主体No.1の行動は、最初の対戦では相手の1回前の行動であるFALSEとなるが、2回目は相手の2回前の行動を反転したTRUEとなり、それ以降は双方がTRUEを出す状況が続く。

行動主体はこのようにプログラムを進化させつつ、最終ステップでは再び一般的なしつぺ返し戦略を獲得していた。Game 1ではしつぺ返し戦略を基軸にしつつ、その時々で対戦相手の行動に適応して高い利得を獲得するため、巧妙にプログラムを進化させている状況が見て取れる。

```
【行動主体 No.0】
0:$3=$9 and $5
2:if $3=$2 node=9 else node=39
9:output $11
```

```
【行動主体 No.1】
0:if $9=$3 node=33 else node=19
33:$3=$5 and $6
22:$2=$6
8:$10=not $2
31:output $10
36:output $4
```

図 15 195 ステップのプログラム

```
【行動主体 No.1】
5:if $7=$1 node=23 else node=13
33:output $5
13:$9=not $3
34:output $9
```

図 16 929 ステップのプログラム

Game 2では、Game 1のように安定した公平な状態に収束したまま持続することはなかったが、ネットワーク構造に依存することなしに、TRUEの行動割合が0.4程度の個人的に合理的な状態から割合0.5の公平な状態に変化する。よって、提案モデルのように効率良く他者の行動ルールを模倣することで、ジレンマ的な状態を解消できる可能性があることがわかる。

Game 2におけるGRAPEのプログラムの変化を示す。図7では、4043ステップまでは共に相手の1回前の行動を反転したものを出力し、行動主体No.0がTRUE、No.1がFALSEを出し続ける状況が続く。ところが、行動主体No.1以外との対戦を経て、4044ステップで行動主体No.0の記憶する過去2回分の行動が共にFALSEとなった。そのため、それ以前のステップと同じ行動ルールでは高い利得を獲得できなくなり、プログラムが進化する。しばらくステップが進むと、一方がTRUE、他方がFALSEとなる状況になる。このときのプログラムは、共に相手の2回前の行動を反転

したものを出力する内容であった。

その後7426ステップで、双方の2回前の行動がFALSE、1回前がTRUEとなったことをきっかけに(それ以前は互いに逆となる行動を実行していた)対戦の結果が変わり、共にTRUEを実行するようになる。図17は、7426ステップにおけるプログラムである。行動主体No.0は自分か相手の1回前の行動がTRUEならばTRUEを出力するため、自らの行動はTRUEとなる。行動主体No.1はしつぺ返し戦略であるが、1回前の相手の行動がTRUEであるために最初はTRUEを出力し、それ以降は相手か必ずTRUEを実行するために、出力はTRUEのままとなる。

Game 2においても、相互作用を通じてプログラムが進化する過程が見て取れる。

```
【行動主体 No.0】
44:swap $1 $10
26:$10=$4 or $9
12:output $10
```

```
【行動主体 No.1】
33:output $5
```

図 17 7426 ステップのプログラム

Game 3では、シミュレーション結果の通り、先に行動する機会を得た行動主体が高い利得を獲得し続ける社会になってしまう。これを回避できるかを検討するため、3.3節における行動ルール生成の方法を変更する。3.3節では、対戦の際、ランダムに選択された一方の行動主体のみが子個体を作成して相手と対戦しGRAPEの個体を更新したが、これを「自分および対戦相手が、双方のGRAPEの個体から一定数の子個体を生成し、自分および相手の作成した子個体どうしで総当たりで行い、双方で評価値の最も高い子個体を用いて同時にプログラムを変更する」と変更してシミュレーションを行った。

$(n_1, n_2, p) = (10, 5, 0.5)$ とした際の平均利得と最大利得を図18に、TRUEの割合を図19に示す。社会の一部ではFALSEを実行して利得3を獲得し続ける行動主体が存在するが、FALSEのみが実行されるのではなく、積極的にTRUEを実行してジレンマを回避する状況となっている。この結果から、対戦する2つの行動主体が行動を変更する、つまりGame 1, 2よりも早く行動ルールの変更が行われることがジレンマの解消につながるものと考えられる。

図18の結果では、1299ステップまでは行動主体No.0, 1の双方がFALSEを実行し続けるが、1300ステップにおいて行動主体No.0の過去2回分の行動がTRUEとなり、図19(出力に影響しない行は省略)に示す内容に双方のプログラムが同時に進化した。行動主体No. 1は相手の1回前の行動を実行するしつぺ返し戦略だが、No.0は自分の1回前と相手の1回前が同じ行動ならばTRUEを、そうでなければ0行と23行の間で無限ループとなりFALSEを出力する。その結果、

行動主体 No.0 は FALSE, FALSE, TRUE を、No.1 は TRUE, FALSE, FALSE を順番にくり返す。

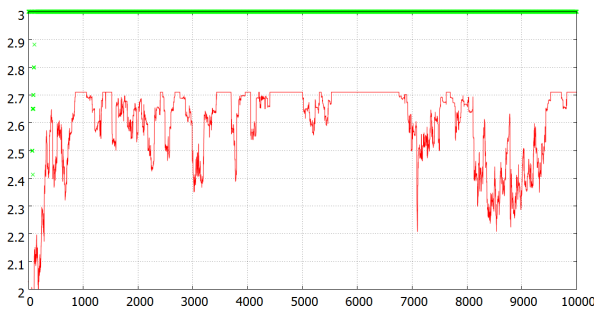


図 18 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の最大利得と平均利得

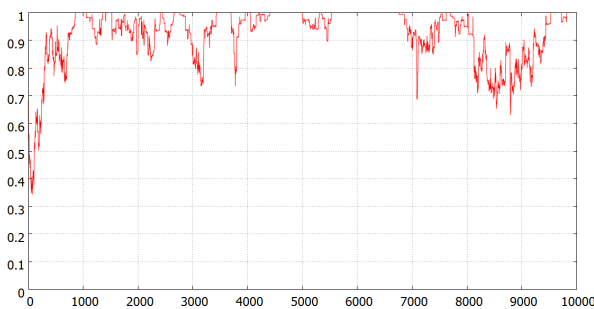


図 19 Game 3 での SW 頂点非活性化モデル $(n_1, n_2, p) = (10, 5, 0.5)$ の TRUE の実行割合

【行動主体 No.0】

```
0:$7=$11 or $1
23:if $2=$3 node=35 else node=0
14:output $7
```

【行動主体 No.1】

```
17:output $3
```

図 20 1300 ステップのプログラム

シミュレーション結果とその考察から、各問題でジレンマ解消に必要となる要素は、Game 1 では他者の行動ルールの効率的な模倣(本研究では GRAPE が多様な個体を持つこと)と一定数の他者とのつながりがあるネットワーク構造を持つこと、Game 2 では他者の行動ルールの効率的な模倣、Game 3 では行動ルールの変更が効率良く、早いサイクルで行われることと考えられる。

4. おわりに

本研究では、社会でみられるさまざまなジレンマ的問題において、コンピュータプログラムの自動生成手法である GRAPE により行動ルールを生成し、個人にとっての合理的な行動を避け、社会全体として望ましい行動が得られるかを分析した。GRAPE では、プログラムを構成する個体の多様性を維持することが効率的なプログラム作成に重要である。そのため本研究では、GRAPE において個体の多様性を維持できる手法を用

い、従来手法よりも効率良くプログラム作成を行った。

提案モデルを用いたシミュレーション結果から、行動主体間のネットワーク構造や行動ルールの更新速度を変更することで、さまざまなジレンマ問題においてジレンマを解消できる可能性があることを示した。

参考文献

- (1) 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)
- (2) 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)
- (3) 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)
- (4) 今福啓“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”情報学研究所紀要(獨協大学)、Vol. 1、pp.5-13 (2012.2)
- (5) Shinichi Shirakawa, Shintaro Ogino, and Tomoharu Nagao: Graph Structured Program Evolution, Proceedings of the Genetic and Evolutionary Computation Conference 2007 (GECCO '07), Vol.2, pp.1686-1693, London, England, 7-11 July (2007)
- (6) 石堂眞大、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)
- (7) 今福啓“プログラムの自動生成手法における個体の多様性の維持”情報処理学会研究報告、Vol.2012-CE-115、No.5 (2012.7)
- (8) 石堂眞大、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)
- (9) 佐藤浩、小野功、小林重信“遺伝的アルゴリズムにおける世代交代モデルの提案と評価”人工知能学会誌、Vol.12、No.5 (1997.9)
- (10) 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)
- (11) 秋本洋平、永田祐一、佐久間淳、小野功、小林重信：実数値 GA における生存選択モデルとしての MGG と JGG の解析、人工知能学会論文誌、Vol.25、No.2、(2010).
- (12) 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)

(2012 年 9 月 21 日受付)
(2012 年 12 月 19 日採録)