

## 卒業論文

Q学習を組み込んだエージェントから構成  
されるSugarscapeモデルによる人工社会

指導教員 李 磊 教授

法政大学工学部電子情報学科

平成 23年度

学籍番号 07D4086

氏名 長野 晋一

## 概要

Sugarscape モデルとは、蟻が砂糖を食べるという動きを模倣したものである。従来のモデルは、エージェントが行動選択する際に、設計者がそれに応じて移動ルールを変更していた。しかし、一般に生物は環境の変化に応じて自分の移動ルールを自律的に変更する。本研究では、Q 学習により行動を獲得するような学習エージェントを導入することで、エージェントが自分の行動ルールを獲得することだけでなく、エージェントが進化的に行動ルールを獲得している過程などがシステムに与える影響を社会学的な立場から分析する。また、Sugarscape モデルにおいてのエージェントの交配、寿命、財産相続ルールもさらに実装し、従来のモデルの設計では、生み出すことが出来なかった現象を創発することを試みる。

# 目 次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	はじめに . . . . .	1
1.2	本研究の目的 . . . . .	1
1.3	本論文の構成 . . . . .	1
<b>2</b>	<b>複雑系について</b>	<b>2</b>
2.1	複雑系とは . . . . .	2
2.2	トップダウンアプローチとボトムアップアプローチ . . . . .	2
2.3	創発 . . . . .	2
<b>3</b>	<b>エージェントベースモデルリング</b>	<b>4</b>
3.1	用語解説 . . . . .	4
3.2	エージェントベースシミュレータ artisoc . . . . .	4
<b>4</b>	<b>Sugarscape モデル</b>	<b>6</b>
4.1	環境の局所ルール . . . . .	6
4.2	エージェント . . . . .	6
<b>5</b>	<b>強化学習と Q 学習</b>	<b>8</b>
5.1	強化学習 . . . . .	8
5.2	Q 学習 . . . . .	8
5.3	Q 学習に基づくエージェントの導入 . . . . .	9
<b>6</b>	<b>実験 1</b>	<b>10</b>
6.1	シミュレーション並びに結果 . . . . .	10
<b>7</b>	<b>実験 2</b>	<b>18</b>
7.1	シミュレーション並びに結果 . . . . .	19
<b>8</b>	<b>考察</b>	<b>29</b>
8.1	実験 1 の考察 . . . . .	29
8.2	実験 2 の考察 . . . . .	30
<b>9</b>	<b>まとめと今後の展望</b>	<b>32</b>
9.1	まとめ . . . . .	32
9.2	課題と今後の展望 . . . . .	32
<b>10</b>	<b>謝辞</b>	<b>34</b>

# 1 序論

## 1.1 はじめに

社会の様々な現象の分析にマルチエージェントシステムを利用する試みがなされている。その中でも複雑系理論と人工生命から社会科学のモデルとして Sugarscape モデルがある。これは簡単なローカル・ルールを設定することで、実社会に見られる現象をモデル化し、文化や闘争などの社会ネットワークを分析することに用いられた [3]。

これまでの Sugarscape モデルにおいて、エージェントの行動選択に用いられた移動ルールは設計者によって与えられていた。しかし、一般に人間を含む生物は環境の変化に応じて自分の移動ルールを自律的に変更する。そのため、動的な環境のもとでは、エージェントは自分を取り巻く環境に応じて移動ルールを変更できることが望ましい。柴田らは、ここに着目し強化学習の一つである Q 学習をエージェントに組み込み、エージェントが進化的に行動ルールを獲得している過程などがシステムに与える影響を社会学立場から分析した [1]。

## 1.2 本研究の目的

本稿では柴田ら [1] が提案したモデルを踏まえた上で、エージェントベースシミュレーション用の開発環境 artisoc を用いて、さらに交配ルール、財産相続ルール等を Q 学習をするエージェントに実装し、エージェントが自分の移動ルールを自律的に変更することで、エージェント同士の相互作用により、従来の Sugarscape モデルとは違ったエージェントの振る舞いによる現象が創発されたので従来のモデルと比較考察するとする。

## 1.3 本論文の構成

第 1 章では、本論文の研究目的と概要を簡潔に記載した。

第 2 章では、まず本論文のテーマである人工社会や人工生命などの複雑系と言われる分野の誕生経緯と常用語について説明することにする。

第 3 章では、エージェントベースモデルと本研究で開発環境に用いた artisoc について紹介する。

第 4 章では、複雑系の解明に用いられているエージェントベースモデルにおいて、代表的な Sugarscape モデルを紹介する。

第 5 章では、強化学習及び本実験で用いた Q 学習の紹介をする。

第 6 章では、主に柴田らが提案したシミュレーションと結果をまとめたものを示す。

第 7 章では、第 6 章の実験を踏まえ、更に交配ルールとエージェントの寿命、財産相続ルールも実装し、シミュレーションと結果をまとめたものを示す。

第 8 章では、各実験について考察を述べる。

第 9 章では、本論文の総まとめと見出された課題、今後の展望について述べる。

第 10 章では、本稿の作成に携わった多くの方々へ、この場を借りて謝辞を記載した。

## 2 複雑系について

### 2.1 複雑系とは

複雑系とは多数の要素が相互作用し、個別の要素の振る舞いからは導きだされないような非線形な活動を示すシステムのことである。

例えば、社会科学という学問は“複雑”な科学だと言われている。社会プロセスを経済、人口統計、文化、空間というような因果を局所的に切出すことで全体像を構成しモデル化分析し、それらの結果を集めて社会プロセス全体を解き明かそうとしても、そもそもプロセスを分解することができない。ほとんどの社会学者にとって、これらの分野が人工的であるということに関して異論は認めないが、そのような複数のプロセスを一体として研究するために適切な方法論は存在しないという点でも意見が一致している[2]。

では、このように“複雑”な「複雑系」を解明するために、どのような方法論が有るだろうか、以下で取り上げことにする。

### 2.2 トップダウンアプローチとボトムアップアプローチ

理論化のアプローチの仕方に応じて「トップダウンアプローチ」と「ボトムアップアプローチ」に分類される[5]~[7]。トップダウンアプローチとは、理論を形式化し、それによって実際の現象を説明する方法。即ち、仮説をたてて、その有効性をデータによってテストする。例えば、生物学や物理学では、人間がまるで神になったかのようにモデルや方程式を描きおろし、マクロな物体や生物はこう動く、というような理論を明示的に構築し、それを現実と照らし合わせて検証する。

一方このアプローチで複雑に振る舞う現象を解明するには限界があるため、ボトムアップアプローチによる方法で解決に臨む考え方が出てきた。ボトムアップアプローチとは、現象に基づいて理論を形式化する方法。すなわち、得られたデータを形式化する仮説を立てる。例えばある現象を巻き起こす要素をエージェントとして定義して、その相互作用を観察することで全体の振る舞いを包括的に理解していく等である。本論文のテーマである人工社会はこのアプローチによって複雑に振る舞う現象を解明していくことを試みる。

### 2.3 創発

創発とは、これまでその研究に応じて意味合いは若干異なって使われてきたが、一般には「それ以前には見られなかった（予想出来なかつた）特性の出現」という意味である。また、局所的な複数の相互作用が、個別の要素の振る舞いからは予測できないような特性が出現し、さらにその全体的なシステムの性質がまた個別の要素の振る舞いを決定していくようなこの循環を創発現象と呼ぶ。

元々、この創発という言葉はニュートン以来の西洋科学の基本的なパラダイムである還元主義（簡単に言うならば、物事を細かく分解して、微小単位に分解することで物事を理解しようとする立場のこと。即ち2.2のトップダウンアプローチによる物事の理解しようとする立場のこと）に対するアンチテーゼとして用いられるようになった言葉である。つまり、「ミクロレベルの特徴だけでは必ずしもマクロレベルの特徴は予測できない」、或いは、「部分の総和は全体ではない」という主張が背景

にあり、そのための理論的な基盤を表す用語、或いは概念として、創発が主張された訳である[4].

アリの行列の例を上げて考えてみると、アリが餌を見つけるとフェロモンを出しながら巣に戻る。そして続く2匹目、3匹目のアリもフェロモンの濃い方へ辿りながら餌のある場所に進むが、餌から巣までの距離が遠い場合、アリのフェロモンは揮発性が高いために全く同じ経路で餌に辿り着くことはほぼ無い。そこで後続するアリによって作り出された餌までのフェロモントレイルは、いくつも出来上がるだろう。だが、前述の通りアリのフェロモンは揮発性が高いために、自然と遠回りな経路のフェロモン量は短い経路のフェロモン量より少なくなるため、そして行く行くは多くのアリがより短い経路を通るようになる。このようにして自然と餌と巣を結ぶ最短距離を多くのアリが通り、アリの行列が形成されるようになる。このように、下位レベルのアリの一匹一匹の行動が相互作用し、やがて（ボトムアップ的に）上位レベルの最短経路のアリの行列という特性が新たに形成された。また、形成されたアリの行列はフェロモン情報を通して再び個々のアリの振る舞いを決定づけていくような創発現象が起こる。

### 2.3.1 創発性の尺度

創発は人工生命の中心的なテーマとなっているが、創発性と言われる客観的な尺度に関する議論も行われてきた。ここで、本論文で用いる創発とはどのレベルを指しているかを断っておく。

例えば、人工生命モデルにおいて、一見いかにもそれらしい生命現象が観察できたからといって、単にプログラムに明示的に記述された通りに振舞っただけでは、創発度はゼロである。A. M. Assad[8]らは以下の様に、創発性のレベル付けをした。

**非創発** 記述やルールを調べれば振る舞いは演繹できる。

**弱創発** 振る舞いを観測した後で、記述やルールから後付け的に演繹できる。

**強創発** その振る舞いを演繹することは理論的には可能であるが、そのメカニズムを解明することはひどく困難である。

**最大創発** 記述やルールからその振る舞いを演繹することは不可能である。

このレベル付けに従うならば、本論文で用いる創発とは、弱創発に属していると思われる。

### 3 エージェントベースモデルリング

本論文では、社会プロセスのエージェントベースモデルを人工社会と呼ぶことにする。このアプローチでは、各エージェントの局所的なルールの元で、人工的な環境の中で活動する個体の相互作用から重要な社会構造や群行動を作りだす。以下、エージェントベースモデリングに使われる用語を説明していく。

#### 3.1 用語解説

##### 3.1.1 エージェント

エージェントとは、人工社会に住む“人々”的ことである。それぞれのエージェントはその内部状態と行動ルールを持っている。状態にはエージェントの一生を通じて一定のものもあれば、外部環境や他のエージェントとの相互作用を通じて変化するものもある。これらの移動や相互作用、そして状態の変化は、エージェントや空間の挙動を規定したルールにすべて依存する。

##### 3.1.2 環境

人工社会での生活は、ランドスケープのような何らかの環境の中で展開する。ランドスケープは、エージェントが収穫し消費する再生可能な資源が配置されるマップのように、環境の具体的な有様を表している。このようなランドスケープなら、エージェントに資源を提供する場所を格子状に構成できる。エージェントは環境の中で動作し、環境と相互作用する。

##### 3.1.3 ルール

エージェントや環境には行動ルールがある。エージェントは、出来る限り遠くまで見渡し、食料豊富な所を見つけ、そこへ行き食料を食べるなどというような単純なルールに従っている。これは、エージェントと環境を結びつけるルールなので“エージェント一環境”ルールであるといえる。また、後述する交配ルールなどは、“エージェント一エージェント”の相互作用を支配するルールがある。その他、本論文では実装していないが、ランドスケープのそれぞれの場所は、セルオートマトンのルールのように近隣と結びつけることができる。例えば、ある場所の資源再生率を隣の場所の資源量から計算するようなルールが考えられる。このような場合は“環境一環境”ルールであると言える。

#### 3.2 エージェントベースシミュレータ artisoc

本研究で用いたエージェントベースシミュレータ artisoc[6] は以下のターゲットに作成された。

- プログラミング経験があまりない人
- マルチエージェントの世界を学びたい学生
- 日本語を用いたモデルを作成したい人

また、特徴として以下の点が挙げられる。

- 日本語が使える

- GUI 設定画面による、出力やシミュレーションの設定
- 簡易でメジャーなプログラミング言語 VisualBasic に準拠したルール書式
- GUI で配置するモデリングの簡単さ

尚、2012 年 2 月現在では 3D 表現が可能とされる artisoc3.0 がリリースされているが、本研究ではその従来版に当たる artisoc2.6 を用いて実験を行った。

## 4 Sugarscape モデル

Epstein[3] らが提案したエージェントベースモデルの中で代表的な Sugarscape モデルとは、蟻が砂糖を食べるという動きを模倣したものである。そこでは、エージェントが砂糖を収集することにより環境との相互作用が生じる。その結果エージェントの行動は非常に単純なルールに従っているが、大域的には複雑な現象が発生する。彼らはそれらの現象を用いて交配、公害、戦争、疾病、取引、文化、などの様々な社会現象のメカニズムを分析した。

Sugarscape モデルは、エージェントとその食料である砂糖が配置された空間分布である。この空間は格子状のトーラス平面上で表現され、全ての座標において砂糖の最大容量と砂糖の現在量というデータを持っている。

### 4.1 環境の局所ルール

自立したエージェントはこの Sugarscape で資源（砂糖）の収集と消費（代謝）をし続けながら、生きていくことになる。そのため環境変数である砂糖の再生が必要であり、その再生ルールは (4.1) 式によって与えられる。

#### 砂糖の再生ルール $G_\alpha$

資源（砂糖）の現在量を  $g_t$ 、最大容量を  $c$  とし、次の期間  $t+1$  での資源量  $g_{t+1}$  は

$$g_{t+1} = \min(g_t + \alpha, c) \quad (4.1)$$

### 4.2 エージェント

Sugarscape モデルにおける各エージェントは被ることなく（即ち 1 つの座標又はマスに 2 つ以上のエージェントが存在することは無い）、マップを徘徊するものとする。また、各エージェントは砂糖の代謝率と視力などといった遺伝的特性を持つ。代謝率とは、期間ごとにそのエージェントが燃焼する砂糖の量のことである。視力とはエージェントの認識できる範囲を表している。

従来のモデル用いられたエージェントの具体的な移動ルールを以下に示す。

#### エージェントの移動ルール $M$

1. 格子状の直交する 4 方向を視力の限り遠くまで見渡し、最も砂糖が豊富で、他のエージェントがない場所を認識する。（つまりエージェントは斜めなどの対角方向は見えない）
2. そのような場所が複数ある時は、最も近い場所を選択する。
3. その場所に移動する。
4. 新しい場所にある全ての砂糖を収集する。

尚、本研究ではエージェントが立っている座標に砂糖があるかどうか、知覚できないものとした。

エージェントの蓄積財産はここで収集した砂糖の分だけ増加し、同時にそのエージェントの代謝量だけ減少する。エージェントの砂糖財産がゼロ以下になった場合、そのエージェントは餓死したとし

て, Sugarscape 内から取り除かれる. なお, 一期間にエージェントが移動できるのは一度だけであり, その移動距離は 1 マスだけある. 各々のエージェントの移動順はランダムである.

## 5 強化学習と Q 学習

強化学習とその手法の一つである Q 学習について紹介する.

### 5.1 強化学習

時刻  $t(t = 0, 1, 2, \dots)$  における強化学習を以下に示す.

#### 時刻 $t$ における強化学習

1. エージェントは時刻ステップ  $t$  において現在の環境の状態  $s_t \in S$  を観測し, 行動  $a_t$  により状態  $s_{t+1}$  に遷移する.
2. エージェントは遷移によって得られる報酬  $r_{t+1}$  を受け取る.
3. 時刻  $t$  を  $t+1$  に更新して 1. に戻る.

上記の繰り返しにより, エージェントは得られた経験から最適な行動をとる. 強化学習は教師なし学習の一種であると言える.

### 5.2 Q 学習

強化学習の代表的な手法として Q 学習がある. Q 学習は Q 値と呼ばれる Q 関数  $Q(s, a)$  の値が, 実際にエージェントが実行した行動から得られた報酬とともに更新され, エピソード数を重ねるにつれて最適な行動を導く政策に近づいていく. ここで Q 関数  $Q(s, a)$  は状態  $s$  において行動  $a$  を起こす価値を表す関数であり, 全ての環境  $s \in S$  と行動  $a \in A$  に対して値が存在する.

Q 学習のアルゴリズムを以下に示す.

#### Q 学習のアルゴリズム

1. Q 値  $Q(s, a)$  を初期化する.
2. エピソード  $e(e = 0, 1, \dots, e_{max})$  に対して, エージェントの状態を初期化する.
3. 格子状の直行する 4 方向を視力の範囲で見渡し, 現在の時刻  $t$  において, 環境の状態  $s_t \in S$  を認識し, エージェントは Q 値に基づいて決定される政策  $\pi(s_t, a)$  に従って行動を選択する. この行動選択法で代表的なものは,  $\epsilon$ -greedy 法やボルツマン選択などの方法が挙げられる.
4. エージェントは 3. で選択した行動  $a_t$  を実行する.
5. 行動  $a_t$  を実行した結果, 環境の状態は  $s_t$  から  $s_{t+1}$  に遷移し, エージェントは環境から報酬  $r_{t+1}$  を受け取る. その際に用いる更新式を以下に示す.

$$Q(s_t, a_t) \leftarrow (1 - \beta)Q(s_t, a_t) + \beta (\gamma r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a')) \quad (5.1)$$

$\beta$  ( $0 < \beta \leq 1$ ),  $\gamma$  ( $0 < \gamma \leq 1$ ) は, それぞれエージェントの学習率, 割引率を表すパラメータである.

6. 学習終了条件を満たせば, 学習終了. そうでなければ, 時刻  $t$  を更新して 3. に戻る.
7. 最大エピソード数  $e_{max}$  であれば終了. そうでなければ, エピソード数を更新して 2. に戻る.

### 5.3 Q学習に基づくエージェントの導入

従来の Sugarscape モデルでエージェントの移動ルールは, 予め設計者が全て設計していた. だが一般に, エージェントは自分を取り巻く環境に応じて自分自身の移動ルールを変更する. そこで本論文では, 5.2 で説明した Q 学習を以下のように Sugarscape モデルのエージェントに組み込むことで, エージェントが自律的に自己の移動ルールを変更できるようにする.

#### Q 学習を組み込んだ i 番目のエージェントの移動ルール $M_Q$

1. Q 値  $Q(s, a)$  を初期化する.
2. エピソード  $e$  ( $e=0, 1, \dots, e_{max}$ ) に対して, 全エージェントの状態を初期化する.
3. 格子状の直行する 4 方向を視力の範囲で見渡し, 現在の時刻  $t$  において, 環境の状態  $s_t \in S$  を認識し, エージェントは Q 値に基づいて決定される政策  $\pi(s_t, a)$  に従って行動を確率的に選択する. 本論文では行動選択法としてボルツマン選択法を採用した.

$$\pi^i(s_t, a) = \frac{e^{Q^i(s_t, a)/T^i}}{\sum_{a' \in A} e^{Q^i(s_t, a')/T^i}} \quad (5.2)$$

政策  $\pi^i(s_t, a)$  はエージェント  $i$  の状態  $s$  で行動  $a$  を選択する確率であり,  $T^i$  はエージェント  $i$  の温度パラメータである.

4. エージェントは 3. で選択した行動  $a_t$  を実行する.
5. 行動  $a_t$  を実行した結果, 環境の状態は  $s_t$  から  $s_{t+1}$  に遷移し, エージェントは環境から報酬  $r_{t+1}$  を受け取る. なお, 砂糖を収集できなかった場合は, ペナルティとして負の報酬 (-1) を与えた. そして, 状態  $s_t$  において行動  $a_t$  を実行する  $i$  番目 ( $i=1, 2, \dots, N$ ) のエージェントの Q 値は次の式に従って更新される.

$$Q^i(s_t, a_t^i) \leftarrow (1 - \beta^i)Q^i(s_t, a_t^i) + \beta^i(\gamma^i r_{t+1} + \gamma^i \max_{a' \in A} Q^i(s_{t+1}, a'^i)) \quad (5.3)$$

$\beta^i (0 < \beta^i \leq 1)$ ,  $\gamma^i (0 < \gamma^i \leq 1)$  は, それぞれエージェント  $i$  の学習率, 割引率を表すパラメータである.

6. 学習終了条件を満たせば, 学習終了. ここで学習終了条件とは時間  $t = t_{max}$  の時である. そうでなければ, 時刻  $t$  を更新して 3. に戻る.
7. 最大エピソード数  $e_{max}$  であれば終了. そうでなければ, エピソード数を更新して 2. に戻る.

## 6 実験 1

これまでの Sugarscape モデルにおいて、エージェントの行動選択に用いられた移動ルールは設計者によって与えられていた。しかし、一般に生物は環境の変化に応じて自分の移動ルールを自律的に変更する。そのため、動的な環境のもとでは、エージェントは自分を取り巻く環境に応じて移動ルールを変更できることが望ましい。この実験では、Q 学習をエージェントに組み込み、エージェントが進化的に行行動ルールを獲得している過程などがシステムに与える影響を分析する。

### 6.1 シミュレーション並びに結果

ここでは、100 体のエージェントが  $30 \times 30$  の格子状のトーラス平面上に存在する Sugarscape モデルについてシミュレーションを行う。砂糖の再生ルールは  $\alpha = 1$  とする (4.1) 式に従うものとする。 $t_{max} = 3000$  とする 3000 ステップを 1 エピソードとし、最大エピソード  $e_{max}$  は 150 とする。エージェントの視力は全て 1 と設定し、各エピソードごとに 100 体のエージェントをランダムに Sugarscape 上に配置するとする。5.3 のボルツマン選択の温度パラメータ  $T$  について、エピソードが 1~ 100 までなら、(6.1) 式のように各エピソードにおいて更新する。

$$T = 18 - episode/5 \quad (6.1)$$

また、エピソード 101~ 150 までなら、(6.2) 式のように各エピソードにおいて更新する。

$$T = 0.72 - episode/245 \quad (6.2)$$

また、それぞれエピソード数が増え  $T \leq 0$  になったら  $T = 0.1$  とした。

Q 学習を組み込んだエージェントは環境について学習させるために、エピソード 1~ 99 の間では代謝率は 0 とし、100 エピソード以降、エージェントの代謝率を 1~ 2 とした。また、エピソード開始時の初期財産は 15 とし、Q 値更新の際に用いるパラメータである学習率  $\beta_i = 0.3$ 、割引率  $\gamma_i = 0.7 (i = 1, 2, \dots, 100)$  とした。

#### 6.1.1 砂糖（資源）について

砂糖は図 6.2 のように砂糖の山頂から 3, 2, 1, 0 と円状に砂糖は分布している。図の座標  $(x, y) = (0, 0)$  は左下に位置し、そこから右方向が  $x$  座標の正方向、上方向が  $y$  座標の正方向となる。即ち図に示されている領域は第一象限である。また、砂糖の山の円を初期化する式は図 6.1 のように C 言語で記述した。

ここで、`sugar_max[30][30]` は (4.1) 式の砂糖の最大容量に当たり、`sugar_current[30][30]` は (4.1) 式の砂糖の現在量に当たる二次元配列で、配列の数値が砂糖の山の高さを示す。

```

void initialize_sugar(void)
{
    int x, y;
    int sugar_max[30][30], sugar_current[30][30];
    /* 配列の初期化 */
    for(x=0; x<30; x++){
        for(y=0; y<30; y++){
            sugar_max[x][y]=0;
            sugar_current[x][y]=0;
        }
    }
    /* 砂糖の山の式 */
    for(x=0; x<30; x++){
        for(y=0; y<30; y++){
            if(((x-8)^2 + (y-8)^2) <= 10 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-21)^2 + (y-21)^2) <= 10 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-8)^2 + (y-8)^2) <= 60 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-21)^2 + (y-21)^2) <= 60 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-8)^2 + (y-8)^2) <= 120 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-21)^2 + (y-21)^2) <= 120 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-21)^2 + (y-21)^2) <= 120 ) && (((x-8)^2 + (y-8)^2) <= 120){
                sugar_max[x][y]=sugar_max[x][y]-1;
                sugar_current[x][y]=sugar_max[x][y];
            }
        }
    }
}

```

図 6.1 砂糖の初期化プログラム

図 6.1 に従って、マップを更新すると、図 6.2 のようになる。

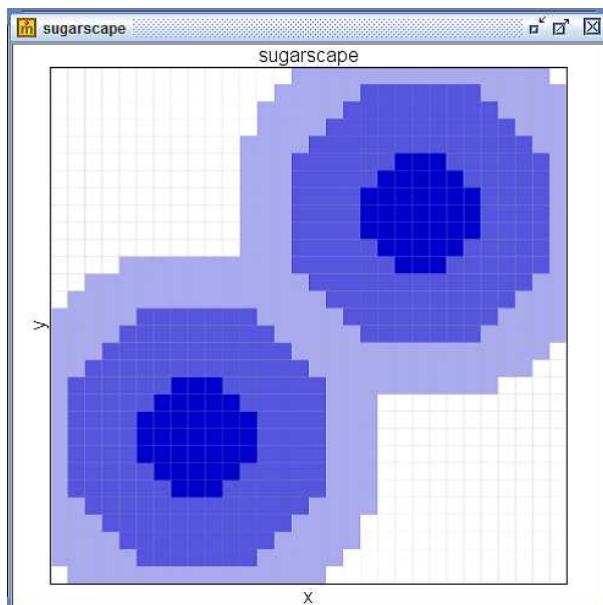


図 6.2 砂糖の山

図 6.3 はシミュレーションの初期画面。図 6.4 はある程度時間が経過し、エージェントが山頂周辺に集まりだした時の画面である。

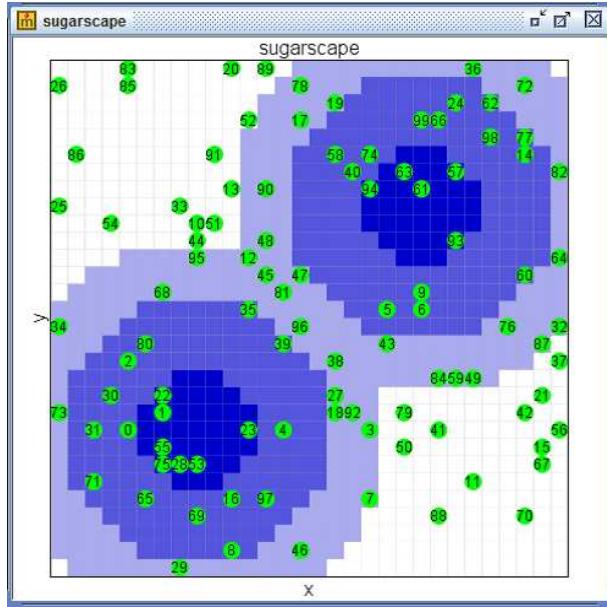


図 6.3 初期画面

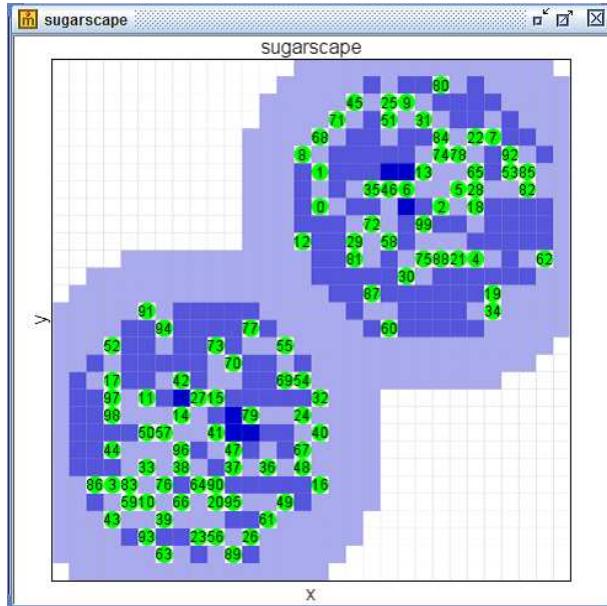


図 6.4 時間経過後の画面

表 6.1 は 90 エピソードの時に、Sugarscape 内からランダムに選んだエージェントの Q テーブルである。state 列にある R, L, U, D は、それぞれ右、左、上、下を意味し、エージェントがその時視認した 4 方向の情報である。0, 1, 2, 3 はそれぞれの場所の砂糖レベルのこと、-1 は他のエージェントがそこに存在することを意味している。Q-value 列にある R, L, U, D, S は、それぞれ右、左、上、下、その

場にとどまることを意味する。Q-value にある数値はエージェントがそれぞれの行動を選択する際に利用する Q 値である。

例えば、No1 はエージェントの右には他のエージェントが存在し、左は砂糖レベルが 1、上は 2、下は 0 である。その時に、エージェントが右、左、上、下に移動する際に使う Q 値はそれぞれ 0.000, 5.476, 6.450, 2.310 である。またその場に止まる Q 値は 5.484 である。この表から砂糖のレベルが高いほどそこに行く確率が高いことがわかる。

表 6.1 ランダムに選んだエージェントの Q テーブル

No	State				Q-value				
	R	L	U	D	R	L	U	D	S
1	-1	1	2	0	0.000	5.476	6.450	2.310	5.484
2	-1	1	2	1	0.000	5.005	6.304	5.192	5.417
3	-1	1	2	2	0.000	4.844	5.879	6.395	5.488
4	-1	1	2	3	0.000	3.430	5.759	8.023	6.100
5	-1	1	2	-1	0.000	4.810	6.545	0.000	5.658

図 6.5 は、各エピソードにおいて全エージェントが獲得した砂糖の合計量の遷移を示したものである。移動ルール  $M$  は従来の Sugarscape モデルのエージェントの移動ルールを指す。移動ルール  $M_Q$  は Q 学習を実装したエージェントの移動ルールである。

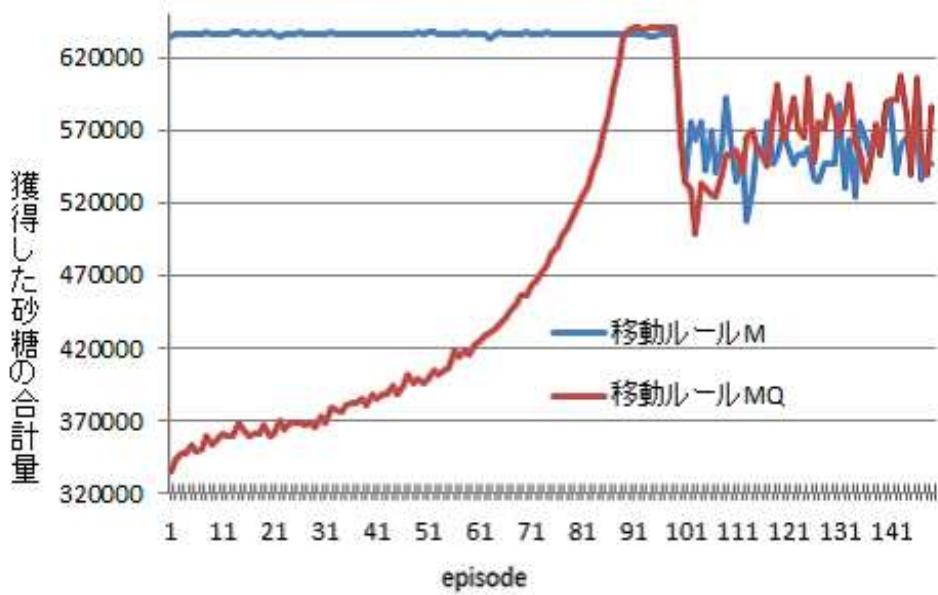


図 6.5 獲得した砂糖の合計量の遷移

図 6.5 より、移動ルール  $M_Q$  のエージェントは、エピソード数が増えるに従って試行錯誤を繰り返し、環境に対応した行動ルールを確立し、全エージェントが獲得した砂糖の合計量が増加しているこ

とがわかる。最終的には従来の移動ルール  $M$  より多くの砂糖を収集できている。尚、この現象は砂糖の分布を多少変えても、同様に従来の移動ルール  $M$  より多くの砂糖を収集することが確認できた。

### 6.1.2 学習残エントロピー

ここで、エージェントの各状態における行動の不確定性がどのように変化しているのかを調べる。これは、学習残エントロピーという量によって評価することができる。平均学習残エントロピー  $I$  は、各エピソードにおいてエージェント  $i$  が経験した状態数  $E^i$  を用いて以下のように定義される。

$$I = \frac{1}{N} \sum_i \left( \frac{1}{E^i} \sum_{S \in E^i} I^i(S) \right) \quad (6.3)$$

ただし、 $I^i(S)$  はエージェント  $i$  の各状態  $S$  に対する行動選択確率  $\pi^i(S, a)$  により定義される。

$$I^i(S) = -\frac{1}{\log 5} \sum_a \pi^i(S, a) \log \pi^i(S, a) \quad (6.4)$$

ここではエージェントのとれる行動は全部で 5 つ（右、左、上、下、止）あるため  $\log 5$  で割っている。 $I^i(S)$  はエージェントの行動が完全なランダムな時、つまり全く学習を行っていない場合には  $\log 5 / \log 5$  となるため 1 である。そして、学習を進めるに従って行動は確定的になり、その値は 0 に近づく。

図 6.6 は、シミュレーションによる移動ルール  $M$  と移動ルール  $M_Q$  の各エピソードにおいて学習残エントロピーの遷移結果である。

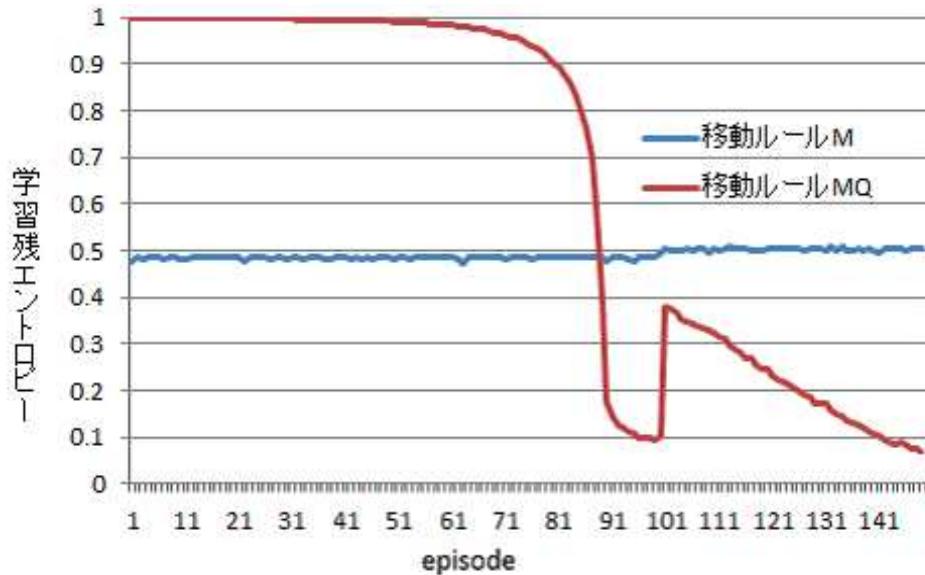


図 6.6 平均学習残エントロピーの遷移

図 6.6 から Q 学習するエージェントは Q 値の更新と温度パラメータ  $T$  が減少するにつれて学習残エントロピーが減少しているのがわかる。100 エピソード後に、急激に学習残エントロピーが跳ね上がったのは、温度パラメータ  $T$  の式が影響しているためである。一方、移動ルール  $M$  のエージェントの学習残エントロピーが 0.5 辺りで一定なのは、エージェントの視界に複数の同じ砂糖量の場所を見つけた場合、その中からランダムに移動場所を選択するためであると考えられる。最も、砂糖の山の分布によってこの値は変動する。

### 6.1.3 ジニ係数

学習を行う機会が与えられるエージェントと与えられないエージェントが存在する社会において、エージェントの学習の進行が社会の格差に与える影響について議論する。ここでは、経済分野で所得や財産の不平等性を表す量として用いられるジニ係数と呼ばれる要約統計量を検証する。エピソード  $E$  でのジニ係数  $G(E)$  は、ローレンツ曲線が 45 度線からの乖離を表す測度であり、本研究では以下のように計算した。

エピソード  $E$  の最終期間 ( $t = 3000$  の時) において、各々のエージェントが獲得している砂糖量  $S^1(E), S^2(E), \dots, S^N(E)$  を降順にソートする。ただし、 $N(E)$  は Sugarscape 内に存在するエージェント数である。その後、以下の式によってジニ係数  $G(E)$  を求める。

$$S^{i'}(E) = \frac{\sum_{j=1}^i S^j(E)}{\sum_{i=1}^{N(E)} S^i(E)} \quad (6.5)$$

$$G(E) = 1 - \frac{1}{N(E)} \sum_{i=2}^{N(E)} S^{i'}(E) + S^{(i-1)'}(E) \quad (6.6)$$

すべてのエージェントが同じ量の砂糖を獲得した場合、ジニ係数は 0 となり、エージェントが獲得した砂糖の量の格差が広がるに連れ、その値も大きくなる。

図 6.7 では、移動ルール  $M$  と移動ルール  $M_Q$  の各エピソードにおけるそれぞれのジニ係数の遷移を表している。図 6.8 では、代謝率を 1~2 と設定したエピソード 100~150 で Sugarscape 内における各エピソードのエージェントの死亡数の遷移を表している。

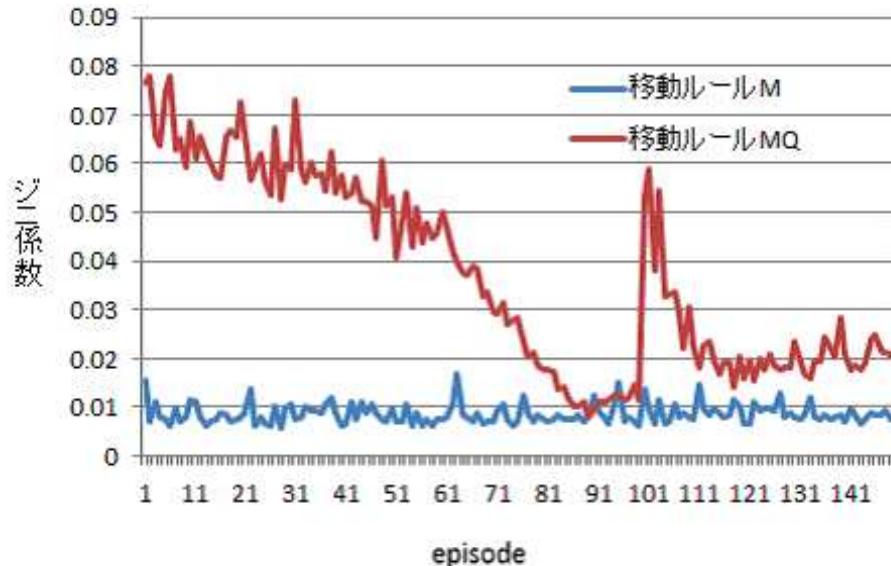


図 6.7 ジニ係数の遷移

図 6.7 より移動ルール  $M_Q$  は 1~99 エピソードの間、学習が進むに連れてエージェント間で砂糖収集量の不平等性が小さくなっていく事がわかる。100 エピソード目で、移動ルール  $M_Q$  のジニ係数が跳ね上がったのは、温度パラメータ  $T$  の式が影響しているためである。

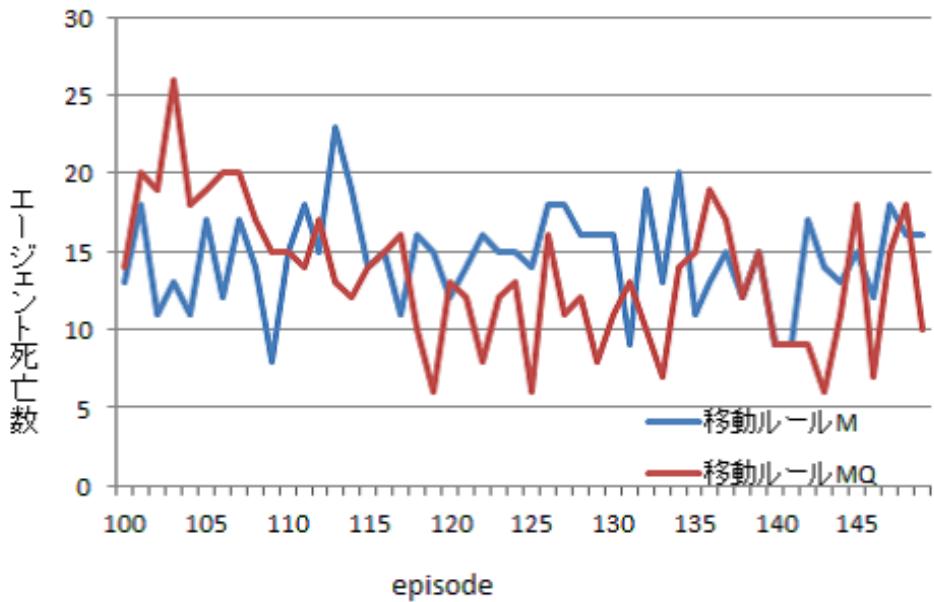


図 6.8 エージェント死亡数の遷移

図 6.8 と図 6.7 より、100 エピソード以後のエージェントの代謝率が 1~ 2 とした時、移動ルール  $M$  と移動ルール  $M_Q$  ではエージェントの死亡数は基本的には変わらないにもかかわらず、移動ルール  $M_Q$  のほうがジニ係数が高くなっていることがわかる。

#### 6.1.4 学習機会の有無を持つ 2 種類のエージェントが存在する社会

更に、エージェントに与えられる学習機会の相違が社会の格差に与える影響を調べるために、Sugarscape モデルに学習する機会が与えられる学習エージェントと学習する機会が与えられない未学習のエージェントの 2 種類のエージェントが存在するモデルで新たに実験をした。50 個のエージェントには  $Q$  値を更新するとし（即ち学習する機会を与え）、残りの 50 個のエージェントには  $Q$  値を更新させず（即ち学習する機会を与えないよう）に、Sugarscape 内に徘徊させるとする。実験のエピソードは 1~ 100 までとし、その他の実験上の条件は全て前実験に準ずるものとする。

図 6.9 では、それぞれ学習する機会の与えられたエージェント、学習する機会が与えられなかったエージェントが各エピソードにおいて、獲得した砂糖の合計量の遷移を示している。図 6.10 では、各エピソードにおいて、全エージェントの獲得した砂糖合計量に対してジニ係数の遷移を示している。

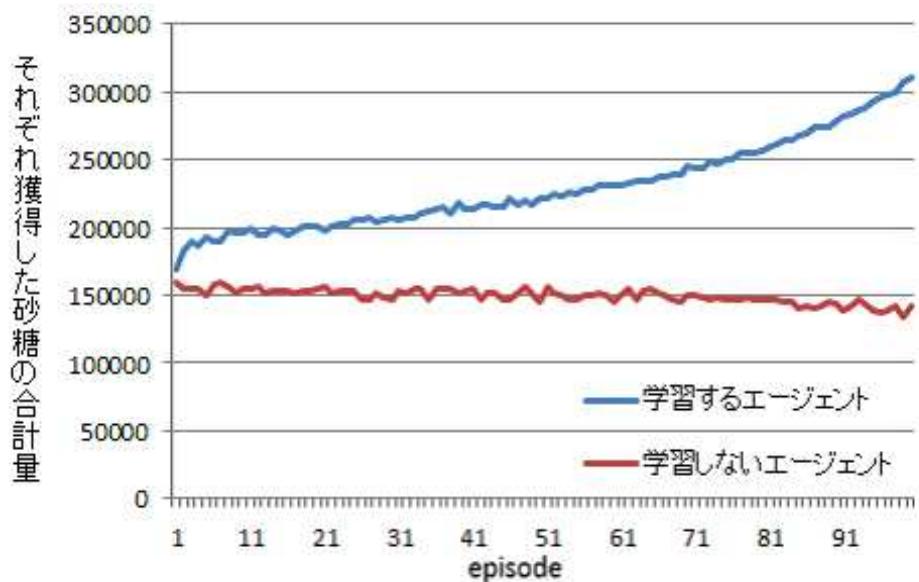


図 6.9 学習機会の異なるそれぞれのエージェントが獲得した砂糖の合計量の遷移

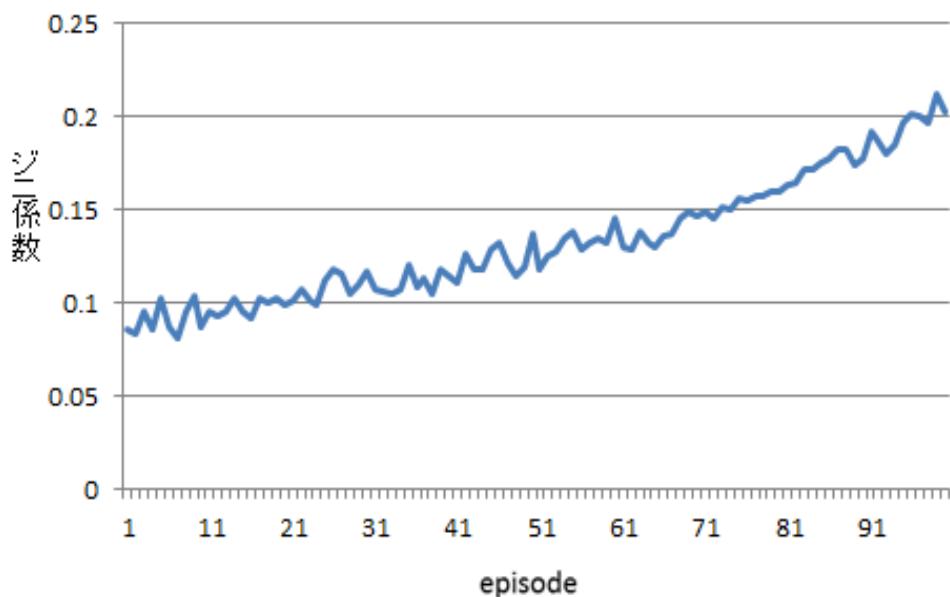


図 6.10 学習機会の異なるエージェントが存在する社会におけるジニ係数の遷移

図 6.9 と図 6.10 から、学習機会の与えられたエージェントがエピソード数を重ねるに従い、エージェントが獲得する砂糖の合計量が増加し、学習機会が与えられないエージェントの砂糖の合計量との差が生じるため、ジニ係数が大きくなっていることがわかる。

## 7 実験 2

Joshua M. Epstein らは、交配ルールと寿命ルールを導入することで、エージェントが性的に相互作用して子を産み、人口のダイナミクスの創発を目的とする実験を行なってきた。また、財産権のような“社会制度”と“生物進化”的過程にどのような関係があるか調べるために、財産相続ルールを導入した[3]。この実験では実験 1 の Q 学習をするエージェントを取り入れ、更に交配、財産相続、寿命ルールを入れることで、従来の手法と如何に異なる振る舞いをするか、また、従来の手法では創発することが出来なかった現象を比較考察することを目的とする。

以下、実験 1 と比べ、実験 2 で新たに導入した各々のルールを説明する。

まず、子供をもうけるエージェントには、交配可能な年齢範囲があることが求められる。次に、両親は新生児に何らかの初期財産を与えてやらなければならない。全く砂糖を持たないまま Sugarscape に投げだされれば、新生児はすぐに死んでしまう。そこで、父と母はそれぞれ自分が生まれたときに授かった砂糖の半分を子供に与えることにする。従って、新生児は両親の（普通は異なった）贈与量の和に相当する砂糖を持って Sugarscape に誕生する。同様に Sugarscape に配置される初期メンバーも後に示す各ルールに従って贈与量が設定される。また、親になるエージェントは子供を生む時に、少なくとも自分が生まれたときにもっていた量の砂糖を蓄えていなければならない。

この年齢条件と財産条件を満たしたエージェントは“交配可能”なエージェントとされ、交配ルール  $S$  を実行する。

### エージェントの交配ルール $S$

1. 近傍のエージェントをランダムに選択する。
2. その近隣が交配可能な異なる性別のエージェントで、しかも少なくとも一方のエージェントの近隣に子供を配置するための空の場所があれば、一人の子を産む。
3. すべての近隣についてこれを繰り返す。

生殖の際ににおける、子の代謝率、最大寿命などの遺伝的特性は、メンデルの法則を使って親の遺伝的特性から決定される。これは片方が親の遺伝子型 ( $m, v$ ) で、もう一方の親の遺伝子型が ( $M, V$ ) だとすると、彼らの子供の遺伝子型は表 7.1 に示した 4 種類の組み合わせが等しく起こりうるものである。

表 7.1 生殖における遺伝的特性の交差

寿命	代謝率	
	$m$	$M$
$v$	( $m, v$ )	( $M, v$ )
$V$	( $m, V$ )	( $M, V$ )

## エージェントの財産相続ルール I

- あるエージェントが死ぬと、財産は生きている彼あるいは彼女の子供の間で等しく分配される。

その他、実験 2 のジニ係数算出について、実験 1 のジニ係数算出が異なるので、以下に変更点を述べる。

実験 1 ではジニ係数をエピソード E の最終期間 ( $t = 3000$  の時)において、各々のエージェントが獲得している砂糖量を対象に算出したが、実験 2 では、各ステップにおいて、各々のエージェントが保持している財産を対象に算出した。前者はそのエピソードの間に獲得した砂糖の累計を対象にしていて、自分の財産（体力）を対象にしていない。

### 7.1 シミュレーション並びに結果

マップの大きさを  $30 \times 30$  から  $50 \times 50$  に変更し、エピソードでカウントをせずに、時間 3000 ステップでシミュレーションをした。

#### シミュレーション共通ルール

下記に示すのは、実験 2 の共通ルールである。

- 交配可能期間は雄雌ともに 12 歳から 15 歳の間から始まる。
- 雄雌共に 60 から 100 歳で寿命が尽き、死亡する。
- Sugarscape に配置される初期メンバーの初期財産は 50 から 100 とする。
- Sugarscape に配置される初期メンバーの数は 350 とする。
- エージェントの代謝率は 1~4 とする。

この共通ルールを踏まえた上で、その他 3 つのルールを用意し、比較実験をする。以下その 3 つのルールについて説明する。

#### 実験 2において 3 つのルール

##### ルール 1

- 雌の交配活動期間は 40 から 50 歳の間で終わる
- 雄の交配活動期間は 50 から 60 歳の間で終わる
- 初期メンバーの砂糖贈与量は 40 固定

##### ルール 2

- 雌の交配活動期間は 40 から 50 歳の間で終わる
- 雄の交配活動期間は 50 から 60 歳の間で終わる
- 初期メンバーの砂糖贈与量は 5~15

### ルール 3

- 雌の交配活動期間は 30 から 40 歳の間で終わる
- 雄の交配活動期間は 30 から 40 歳の間で終わる
- 初期メンバーの砂糖贈与量は 5~ 15

ルール 2, 3 の初期メンバーの砂糖贈与量がルール 1 より低いということは、ルール 2, 3 の方がルール 1 より子供を少ない砂糖（財産）で産めることに他ならず、これは子供を産むコストが低いといえる。

#### 7.1.1 Q 学習を取り入れたモデル

実験 2においての移動ルール  $M_Q$  について以下に述べる。

予めエージェントを 1400 体固定で 90 エピソードまで学習させた後、その中から 100 個のエージェントを選択し、その  $Q$  値の平均を取った値を初期メンバーの初期  $Q$  値とする。この間の代謝率は 0 とし、交配、寿命、財産相続ルールも適用しないものとした。その後、各ルールに従ってシミュレーションするものとする。

学習中のボルツマン選択の温度パラメータ  $T$  は (7.1) 式に基づいて各エピソードにおいて更新する。その後、各ルールを適用したシミュレーションでは各時刻において一律に  $T = 0.15$  とした。

$$T = 10 - \text{episode}/9 \quad (7.1)$$

この他、学習率や割引率といったパラメータなどは全て実験 1 と同じ値を用いた。

尚、エージェントを 1400 体固定に設定したのは、後に示す図 7.4 のように、エージェントは比較的密な状態で餌を取り合うことになるため、エージェントが多い状態を想定し学習させた。

#### 7.1.2 砂糖（資源）について

実験 1 の砂糖の条件と異なる点は砂糖の山の分布が違うだけで、その他砂糖の再生ルール等の条件は全て実験 1 に準じるとする。

砂糖の山の円を初期化する関数は図 7.1 にて、C 言語で記述した。

```

void initialize_sugar(void)
{
    int x, y;
    int sugar_max[50][50], sugar_current[50][50];
    /* 配列の初期化 */
    for(x=0; x<50; x++){
        for(y=0; y<50; y++){
            sugar_max[x][y]=0;
            sugar_current[x][y]=0;
        }
    }
    /* 砂糖の山の式 */
    for(x=0; x<50; x++){
        for(y=0; y<50; y++){
            if(((x-17)^2 + (y-17)^2) <= 100 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-37)^2 + (y-37)^2) <= 100 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-17)^2 + (y-17)^2) <= 250 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-37)^2 + (y-37)^2) <= 250 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-17)^2 + (y-17)^2) <= 300 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if(((x-37)^2 + (y-37)^2) <= 300 ){
                sugar_max[x][y]=sugar_max[x][y]+1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if((((x-37)^2 + (y-37)^2) <= 250 ) && (((x-17)^2 + (y-17)^2) <= 250)){
                sugar_max[x][y]=sugar_max[x][y]-1;
                sugar_current[x][y]=sugar_max[x][y];
            }
            if((((x-37)^2 + (y-37)^2) <= 300 ) && (((x-17)^2 + (y-17)^2) <= 300)){
                sugar_max[x][y]=sugar_max[x][y]-1;
                sugar_current[x][y]=sugar_max[x][y];
            }
        }
    }
}

```

図 7.1 実験 2 の砂糖の初期化プログラム

図 7.1 に従って、マップを更新すると、図 7.2 のようになる。図 7.2 の砂糖は山頂から 3, 2, 1, 0 と円状に砂糖は分布している。

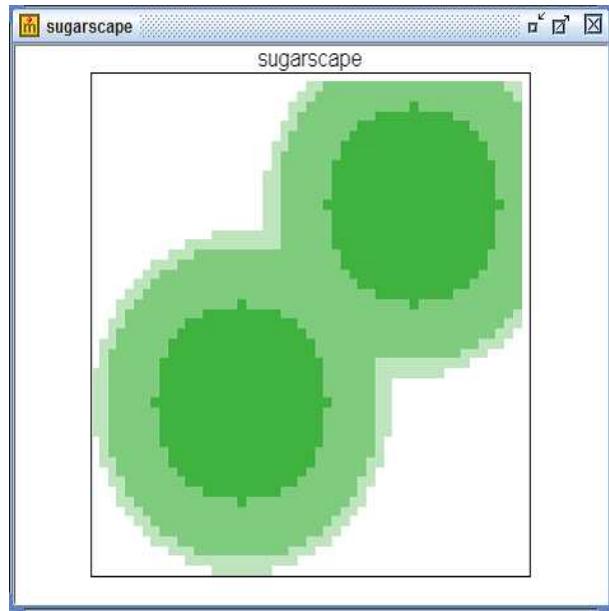


図 7.2 実験 2 の砂糖の山

図 7.3 は上述の移動ルール  $M$  のシミュレーション初期画面 (時間がおよそ 10 ステップ経過後), 図 7.4 は十分に時間が経過し, 移動ルール  $M$  のエージェントが山頂周辺に集まっている時の画面である. 同じく図 7.5, 図 7.6 は移動ルール  $M_Q$  のシミュレーション画面を撮ったものである.

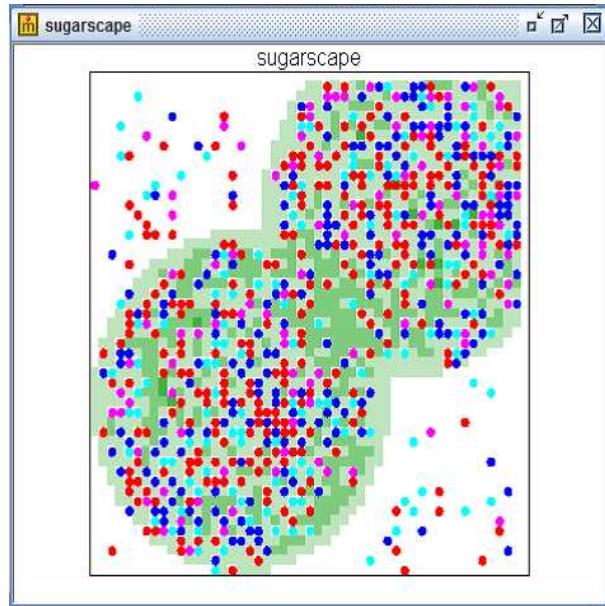


図 7.3 移動ルール  $M$  のシミュレーション (初期)

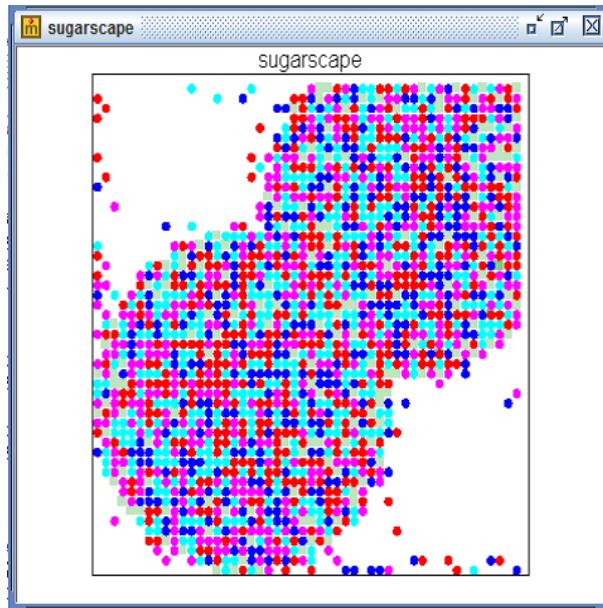


図 7.4 移動ルール  $M$  のシミュレーション (時間経過)

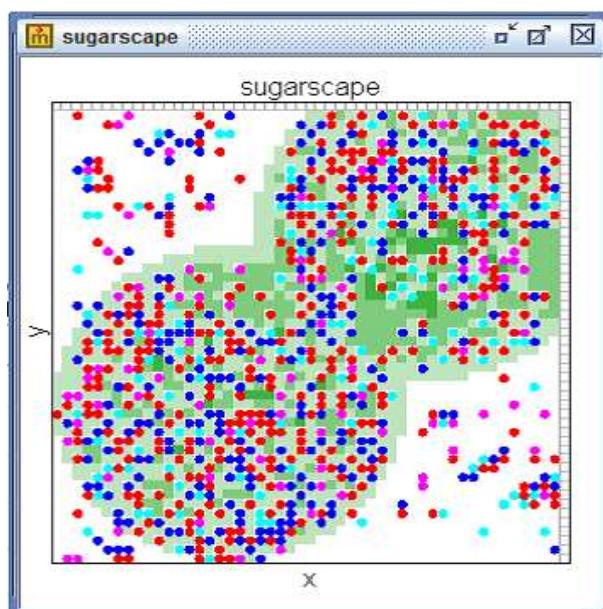


図 7.5 移動ルール  $M_Q$  のシミュレーション (初期)

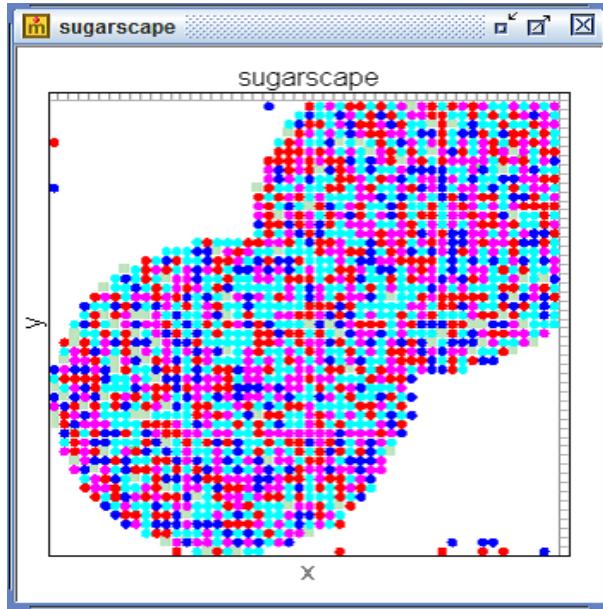


図 7.6 移動ルール  $M_Q$  のシミュレーション (時間経過)

表 7.2 はエージェントを 1400 体固定で 90 エピソードまで学習させた後、その中から 100 個のエージェントを選択し、その Q 値の平均を取った Q テーブルである。

表 7.2 実験 2 でのエージェントの Q テーブル

No	State				Q-value				
	R	L	U	D	R	L	U	D	S
1	-1	1	2	0	0.000	3.544	4.675	1.036	4.023
2	-1	1	2	1	0.000	3.619	4.552	3.619	4.232
3	-1	1	2	2	0.000	3.734	4.586	4.586	4.374
4	-1	1	2	3	0.000	3.910	5.035	6.168	4.822
5	-1	1	2	-1	0.000	3.544	4.545	0.000	4.159
6	-1	-1	-1	0	0.000	0.000	0.000	0.512	2.757
7	-1	0	-1	-1	0.000	0.416	0.000	0.000	2.672
8	-1	0	0	0	0.000	-1.537	-1.160	-1.221	-0.998
9	0	0	0	-1	-1.025	-1.252	-1.576	0.000	-0.998

表 7.2 の No1~ 5 では前実験の表 6.1 と比べると、Q 値に偏りが無く全体的に低い。No6, 7 を見るとエージェントが周りにいる場合、その場にとどまる Q 値がやや高い。また、No8, 9 を見ると、周りに餌が無くてエージェントが隣にいる場合、なるべくそのエージェントから離れないような、その場に留まる選択の確率が高い。

図 7.7, 図 7.8, 図 7.9 はそれぞれルール 1, 2, 3 を適用させた時、時間ステップごとの移動ルール  $M$

と移動ルール  $M_Q$  のエージェント数の遷移を表している。

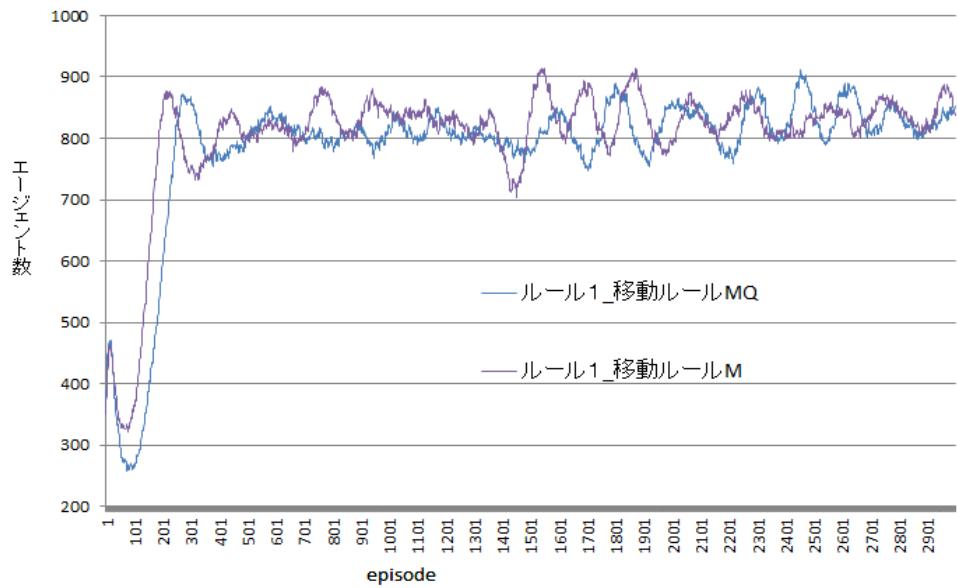


図 7.7 ルール 1 適応時のエージェント数の遷移

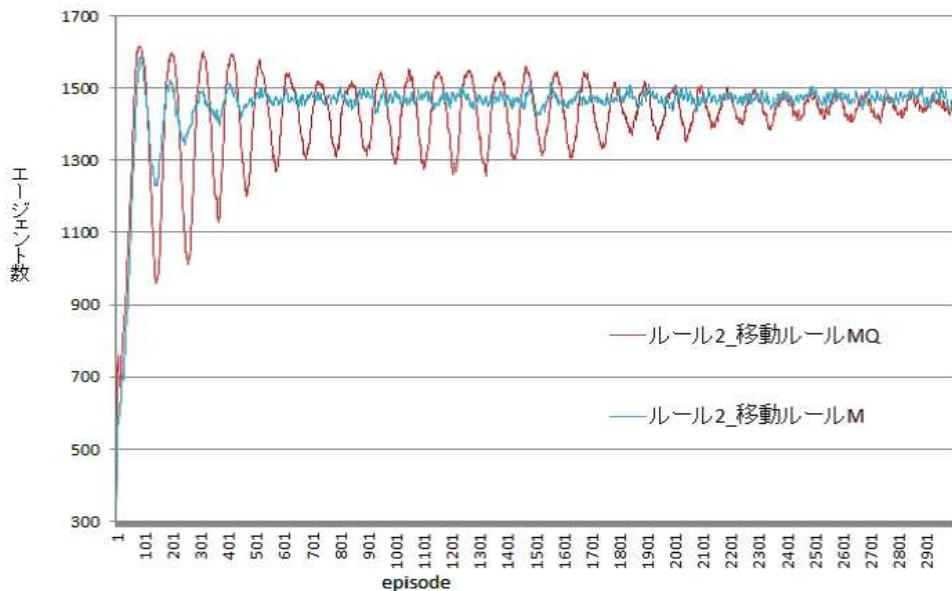


図 7.8 ルール 2 適応時のエージェント数の遷移

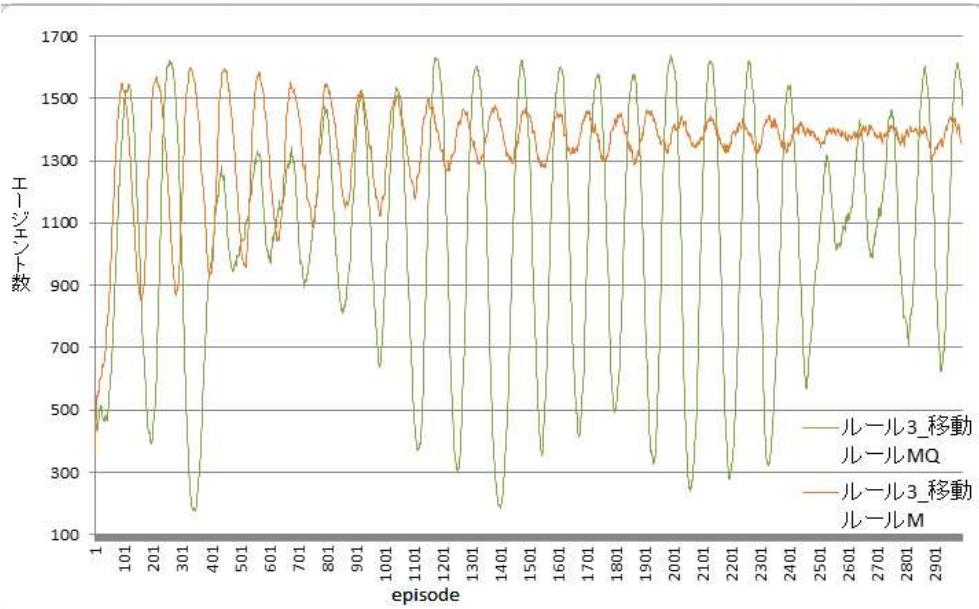


図 7.9 ルール 3 適応時のエージェント数の遷移

図 7.7, 図 7.8, 図 7.9 に共通して, 移動ルール  $M_Q$  のほうが移動ルール  $M$  よりエージェント数が周期的にかなり変動していることが読み取れる.

図 7.10 は, 時間ステップごとにおいて, 各ルールのジニ係数の遷移を表している.

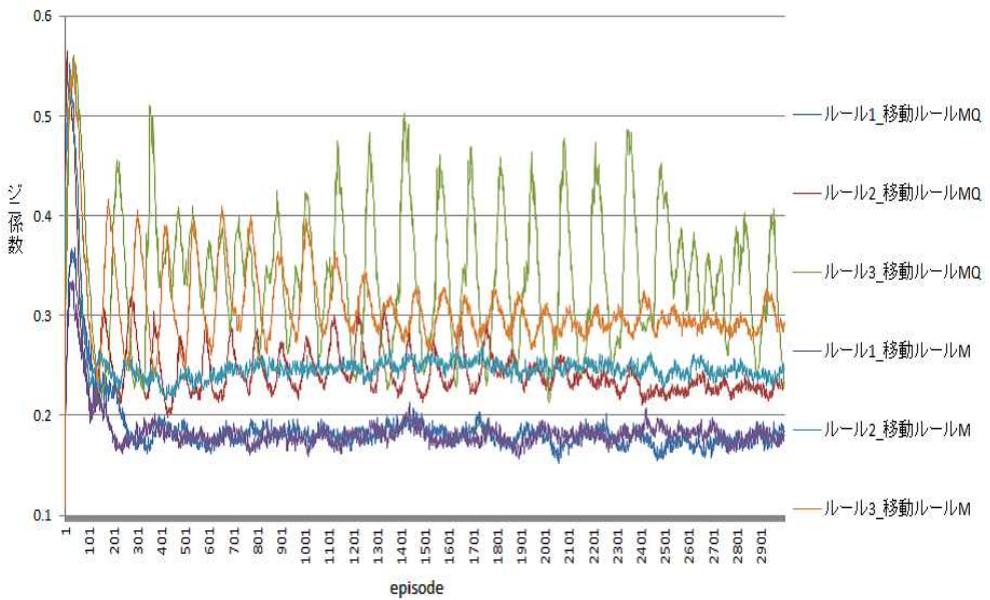


図 7.10 それぞれのジニ係数の遷移

図 7.10 から, 各ルールに共通してエージェント数が増えるに従ってジニ係数が減り, エージェント数が減少するとジニ係数が増える傾向があるとわかる.

図 7.11 は, 財産相続ルール適用後, 各時間ステップにおいて各ルールのエージェント数の遷移をま

とめた図である。

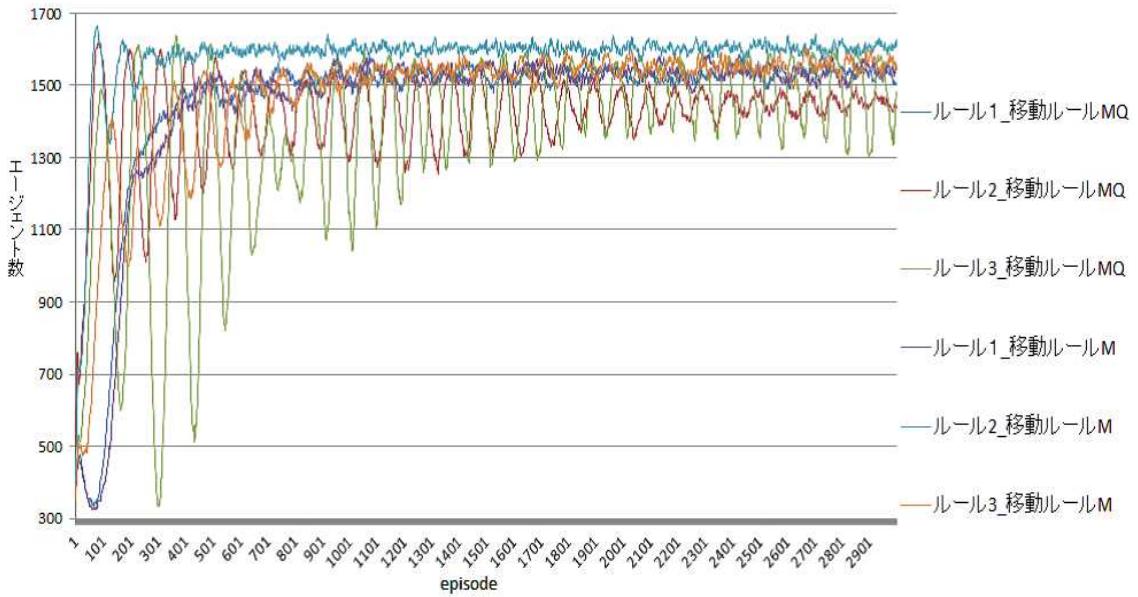


図 7.11 それぞれのエージェント数の遷移 (財産相続ルール適用)

図 7.11 を見ると、基本的に、財産相続ルールを適用していないモデルに比べて、エージェントの数が多い状態のまま維持されている。子供が生みにくいモデルのルール 1 では最終的にエージェント数はおよそ 1500 のままで一定となっていて、図 7.7 と比べてかなりエージェントの数が多いまま維持されている。

図 7.12 は、財産相続ルール適用後、時間ステップごとにおいて各ルールのジニ係数の遷移をまとめた図である。

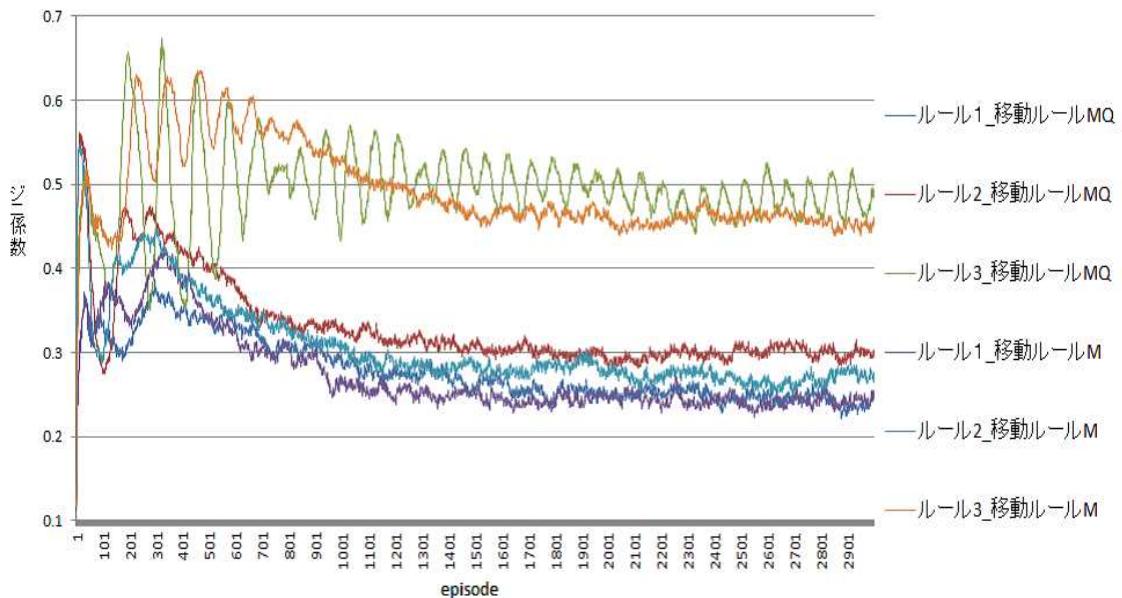


図 7.12 それぞれのジニ係数の遷移 (財産相続ルール適用)

図 7.12 と財産相続ルール無しのモデルである図 7.10 と比べると、各ルールのジニ係数が全体的に高いことがわかる。

図 7.13 は、移動ルール  $M_Q$  のルール 1 を適用し、財産相続ルールを適用したモデルと財産相続ルールを適用しないモデルのエージェントが、時間ステップごとにおいてジニ係数の遷移を表している。尚、ルール 2, 3 を適用したモデルや移動ルール  $M$  のモデルでも同様な傾向があることが確認できたため、ここでは、特にルール 1 を適用した移動ルール  $M_Q$  のモデルのシミュレーション結果を示した。

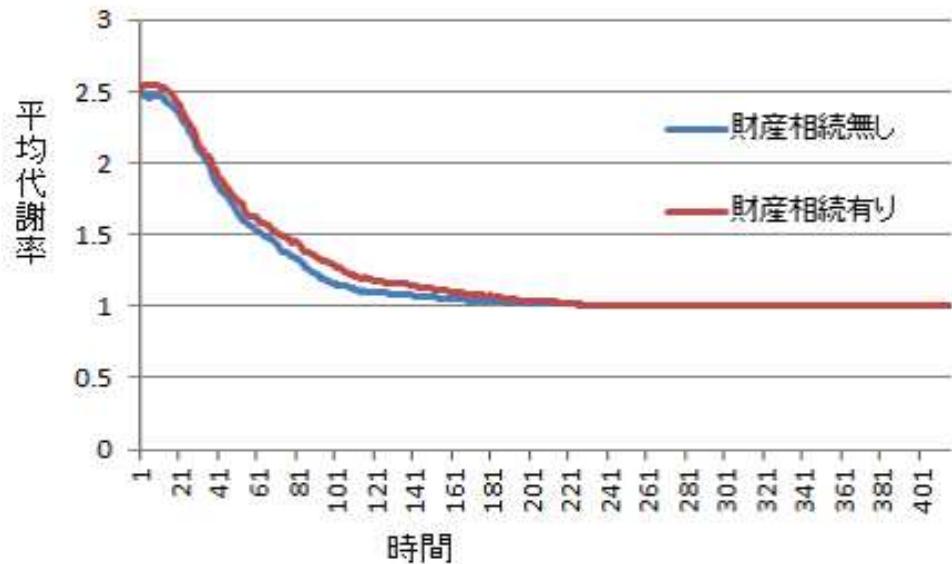


図 7.13 代謝率の平均値の進化

図 7.13 から、財産相続が無いモデルの方が、代謝率が高いエージェントが早く淘汰されていることがわかる。その後、代謝率の時系列を示す 2 曲線は同じ値に落ち着いている。

## 8 考察

ここでは、実験 1 と実験 2 の結果について、それぞれの考察を述べる。

### 8.1 実験 1 の考察

表 6.1 より、エージェントが学習を行った結果、砂糖の量がより多い場所に移動するという 3.1.3 の“エージェント-環境”に当たるルールをエージェント自ら獲得することを示した。そして、図 6.6 より、Q 値が十分に更新されるにつれ、エージェントが移動ルール  $M$  の従来の Sugarscape モデルに比べて学習残エントロピーが減少していることを示した。エージェントが Q 学習を利用して移動ルールを獲得することで、従来の Sugarscape モデルでは議論不可能だった学習残エントロピーに対して、創発する現象について議論可能となる。

図 6.7 より移動ルール  $M_Q$  は 1~99 エピソードの間、学習が進むに連れてエージェント間で砂糖収集量の不平等性が小さくなっていく事がわかる。さらに図 6.5 の結果から、この間に形成された社会とは、エージェントが獲得した砂糖量は増加している事がわかるため、平等でなおかつ全体収穫量の多い社会となっていることがわかる。従来の移動ルール  $M$  では、エージェントの移動ルールが固定されているため、砂糖レベルが多い所を見つけると、直ぐその場に移動しようとする、言わば貪欲なエージェントと言える。一見、単純に砂糖が多い場所にエージェントを移動させる移動ルール  $M$  の方が効率的に見えるが、最終的に砂糖を多く収集できたのは Q 学習を組み込んだ移動ルール  $M_Q$  のエージェントだった。最も砂糖が多い所に必ずしも移動しないエージェントの行動が、結果的に多くのエージェントにとって効率よく収集活動を促すことになった。

この移動ルール  $M_Q$  と移動ルール  $M$  のエージェントの行動について、さらに細かく分析する。

砂糖の多く含む山が円形である以上、砂糖レベル 3 の円にエージェントが密集すると、過密状態となり、逆に中心にいる幾つかのエージェントが身動きが取れなくなり、効率的な砂糖収集ができない現象が起こる。即ち、砂糖レベル 3 に移動したエージェントはその次のステップで 4 方向にエージェントに囲まれてしまい、次のステップではその場に留まるしかなく、新たに再生された砂糖レベル 1 を収集することしかできなくなる。これに対して、移動ルール  $M_Q$  のエージェントは砂糖レベル 3 の場所を視認しても、必ずしもそこに移動することが、次のステップに多くの砂糖を収集できることに直結しないことを Q 学習アルゴリズムによって学習している。この根拠は表 6.1 からも見て取れる。エージェントは砂糖レベル 3 の場所を視認しても、2 に移動する可能性とその場にとどまる可能性が高い。よって結果的に砂糖レベル 3 の場所に行かないエージェントも幾つか存在するため、エージェントが過密な状態を回避し、砂糖の山の上でそれぞれ砂糖を効率的に収集できていると考えられる。

結果的には、エージェントが学習をするに連れて砂糖レベル 3 を見つけてもそこに移動しないというこの行為は、あたかもその砂糖の場所を他のエージェントに譲っているように見える。このモデルでは設計者が明示的にエージェントにこの行動をするように設定したわけではないが、結果的に譲りあうような社会が創発されたと言える。従来の移動ルール  $M$  のような貪欲なエージェントだけが存在する Sugarscape モデルでは、本モデルで起こった創発現象がそもそも議論出来なかつたが、学習をするエージェントを取り入れることで、環境に対して最適な行動を取るエージェントが新たに生む創発現象の議論が可能となることを、本実験で示した。

また、図 6.8 より 100 エピソード後のエージェントの代謝率が 1~2 とした時、移動ルール  $M$  と移動ルール  $M_Q$  ではエージェントの死亡数は基本的に変わらず、また、学習残エントロピーも低くなっているにもかかわらず、移動ルール  $M_Q$  のほうがジニ係数が高くなっている。また図 6.5 よりエージェントが獲得した総砂糖量も両者それほど変わらない結果であった。これは 99 エピソードまで 100 エージェント一定でシミュレーションしていたが、その後はモデル内のエージェントがやや少ない状態になったため、かつての  $Q$  値の参照による行動が必ずしも最適であったとは限らないためだと思われる。環境が変わると、最適な行動をエージェントに取らせるためには再度十分に  $Q$  学習をさせる必要がある。

6.1.4 では、その他学習機会の有無をもつ二種類のエージェントが存在する社会におけるジニ係数の変化を調べ、図 6.9 と図 6.10 から、学習機会の与えられたエージェントが学習を進めるに従い、社会の格差が発生することを示した。従来の Sugarscape モデルによる人工社会において、社会の不平等性の分析には、視力や代謝率などエージェントの先天的な遺伝的特性の違いに焦点を当てられていた。しかし、それらの相違がほとんど見られない社会においても、本手法を用いる事により、エージェントが後天的に獲得する特性の違いが社会の不平等性に与える影響の分析が可能になると考えられる。

## 8.2 実験 2 の考察

表 7.2 の No1~5 では前実験の表 6.1 と比べると、 $Q$  値に偏りが無く全体的に低いことが確認できる。これはエージェントの分布が過密な下ではエージェントがそのような状態に遭遇する場合が少なく、学習をする機会が少なかったからだと思われる。No6, 7 を見るとエージェントが周りにいる場合、その場にとどまる  $Q$  値がやや高い。エージェントが周りにいる場合はその場に餌がある場合が高いためである。また、No8, 9 を見ると、周りに餌が無くてエージェントが隣にいる場合、なるべくそのエージェントから離れないように、その場に留まる選択の確率が高い。これらの結果が反映される様子は図 7.6 からも見て取れるだろう。エージェントは学習するにつれて近隣にエージェントが多くいる場合、その場に留まることで、砂糖をとれる確率が高い事を学ぶようになる。これにより、移動ルール  $M$  のエージェントたちより、移動ルール  $M_Q$  のエージェントたちの方が固まって行動し、群れて固まっているような現象が創発されたことが確認できる。

従来の移動ルール  $M$  では、仲間の位置情報によって、餌を取るという議論は出来なかった。そこでは、ただ砂糖の多い場所にエージェントを移動させるという簡単にして、設計者が明示的にルール設計をしていた。しかし、この結果から、移動ルール  $M_Q$  を導入することで、エージェントは仲間の位置情報を参照し、3.1.3 にて記述した“エージェント-エージェント”ルールに当たる移動ルールも、エージェント自身の学習によって自ら獲得できたと言える。また、仲間が周りにいる場合は、その場に砂糖があること可能性が高いことを、本来設計者が明示的に設計しなかったが、エージェントが学習により移動ルールを獲得し、エージェントたちが固まって行動し、群れて固まっているような現象が創発されたといえる。

図 7.7、図 7.8、図 7.9 に共通して、移動ルール  $M_Q$  のほうが移動ルール  $M$  よりエージェント数が周期的にかなり変動していることが読み取れる。これは移動ルール  $M_Q$  を導入したモデルで、エージェントが過密になりすぎた状態では子供も産める空間もなく、その場で餌をとり続け、ただ年齢を浪費す

るため、一斉に死滅し、その後一気に子供を生むという結果が招いたものである。また、財産相続ルールを適用したモデルである図 7.11 と、財産相続ルールを適用していないモデルの図 7.7、図 7.8、図 7.9 を比べて、基本的にエージェントの数が多い状態のまま維持されている。特に子供を産むコストの高いモデルのルール 1 では最終的にエージェント数はおよそ 1500 のまま一定となっていて、図 7.7 と比べてかなりエージェントの数が多いまま維持されている。これは老後を迎えたエージェントの財産が、その子供に当たるエージェントによって保存されたためである。

図 7.12 と財産相続ルール無しのモデルである図 7.10 と比べると、各ルールのジニ係数が全体的に高いことがわかる。これは参考文献 2 でも言及されているが、一般的に財産相続がある社会では、財産相続が無い社会と比べ、財産の格差が生じ、ジニ係数が増加する傾向がある。

図 7.13 から、財産相続が無いモデルの方が、代謝率が高いエージェントが早く淘汰されていることがわかる。その後、代謝率の時系列を示す 2 曲線は同じ値に落ち着いている。わずかな変化ではあるが、相続は選択を遅くするという効果があることを示している。社会ダーウィン主義者の中には、選択圧が希薄されずに作用するのなら、その種にとっては最良だという根拠のもとに、裕福な人々から貧しい人々へ財産移転に反対する人がいる。だがこの結果から、選択圧が希薄する要素には、裕福な人々から貧しい人々へ財産移転だけでなく、裕福な人々からその子孫へ財産を相続させる行為も、全体の選択圧を希薄させる効果があることがわかる。[2]

移動ルール  $M_Q$  を採用したモデルでも、従来の Sugarscape モデルで見られる現象は続けて創発することは可能であり、またその際に、従来の創発現象に対して、得られた見解から大きく外れることもないことがわかる。

## 9　まとめと今後の展望

### 9.1　まとめ

実験 1 では、エージェントが学習を行った結果、砂糖の量がより多い場所に移動するという 3.1.3 の “エージェント-環境” に当たるルールをエージェント自ら獲得することを示した。そして、学習が十分に行われ、Q 値が十分に更新されるにつれ、エージェントが移動ルール  $M$  の従来の Sugarscape モデルに比べて学習残エントロピーが減少していることも示した。さらに、その他学習機会の有無をもつ二種類のエージェントが存在する社会におけるジニ係数の変化を調べ、学習機会の与えられたエージェントが学習を進めるに従い、社会の格差が発生することを示した。

また、実験 1 では元々エージェント自身が最終的に多くの砂糖を獲得するために、他のエージェントに砂糖の最も多いその場所を譲るという行動が見られた。そもそもこの行動自体は我々設計者が直接的にエージェントに組まなかつたが、結果的にエージェントがそのように振舞っているように見える創発現象をこの実験で確認できたかと思う。従来のエージェントの移動ルールの設計手法では、設計者がエージェントが環境に対して最適な行動をとるように設計しているとは言い難く、またそれによって作り出された現象の創発度も弱いと思われる。それに対してエージェントが自ら学習し、最適な移動ルールの獲得をさせることを目的とする本手法は、より創発性の強い創発現象を作り出せると言える。

実験 2 では、実験 1 の提案手法を交配ルールが存在する Sugarscape モデルに応用することで、エージェントが砂糖の量がより多い場所に移動する移動ルールを獲得させるだけでなく、エージェントたちが密な状態下で如何に砂糖を収集するか試行錯誤する過程で、仲間の位置情報を利用するという 3.1.3 にて記述した “エージェント-エージェント” ルールに当たる移動ルールも自ら獲得されていく様子が観察できた。これにより従来の移動ルール  $M$  とは異なった振る舞いをし、エージェントたちがより固まって群れている創発現象が生み出されたことを示した。また、財産相続ルールを組み込んだモデルでは、財産相続によってジニ係数が増加する現象など、従来の Sugarscape モデルでも共通して見られる現象を示した。

本研究の実験 1, 2 を通して、従来のものと比較して、学習をするエージェントが創発性の強い創発現象を新たにうむことの可能性を示し、またエージェントベースモデリングによる方法で、社会科学等の複雑系の解明に役立てられることを、ここに示す。

### 9.2　課題と今後の展望

実験 2 の課題として挙げられることは、交配による絶えず人口が変動するモデルでは、過密な状態下で Q 学習させたエージェントの行動は必ずしも、常に最適であるとは言えないことである。それは、エージェントたちが死滅し、その後砂糖の山にいるエージェントが少ない状態の下で、エージェントたちの取った行動は事前に学習をさせていないために、最適であるとは言えない。今後、このようなエージェントの数が変動する状況下でもエージェントが常に最適な行動をとるよう学習させるモデルを設計すること出来れば、従来のモデルとは違った新たな創発性の強い創発現象を生むことができると予想できる。

また、強化学習には欠点として、マルチエージェント環境でモデルの設計する際、状態数が極端に多くなる次元の呪いにはまりやすいという点がある。本研究でもこれが原因になり、大分シミュレーション時間を費やす実験があった。また、airtisoc ではメモリの確保と開放の操作ができないため、度々メモリ不足に陥り、発展編としての実験を断念したモデルもあった。

その他、今回はエージェントの視力を 1 と固定としたが、従来の Sugarscape モデルでは視力を 1 以上にして、シミュレーションを行い、創発された現象を分析してきた。もし、視力を 2 以上で、なつかつ Q 学習をエージェントをモデルに組み込むことができるなら、それは、より非明示的にエージェントを設計したモデルとして、従来の Sugarscape モデルより創発性の強い創発現象を議論可能とすることができるだろう。

今後、さらなる要素の多い複雑なマルチエージェントベースモデルを設計する際には強化学習ないしは機会学習のアルゴリズムの改良、またそれらを如何にモデルに組み込むかがモデル設計において肝要であると思われる。

## 10 謝辞

本研究を遂行するにあたり、多大なるご指導を頂きました法政大学教授李 磊教授に厚くお礼を申し上げます。また、本研究室の大学院生の諸先輩方、並びに4年生諸氏にも感謝いたします。

また、シミュレーション作成にあたり、様々なご助言をいただきました株式会社構造計画研究所の皆様、神戸学院大学柴田 淳子 教授に御礼申し上げます。

最後に、学費と生活の面倒をみていただいた両親に心より深く御礼申し上げます。

## 参考文献

- 1) 柴田淳子, 奥原浩之, 石井博昭, 塩出省吾, “Q学習を組み込んだエージェントから構成される Sugarscape モデルによる人工社会”, 電子情報通信学会論文誌, Vol.J90-A, No.1, pp.35-40, 2007
- 2) Joshua M.Epstein, Robert Axtell 著, 訳服部 正太, 木村 香代子訳, 人工社会: 複雑系とマルチ エージェント・シミュレーション, 共立出版, 1999
- 3) J.M.Epstein and R.Axtell, Growing Artificial Societies, Brookings Institution Press and MIT Press, 1996
- 4) 有田 隆也, 人工生命 改訂2版, 医学出版, 2002
- 5) 赤間 世紀, 人工生命入門, 工学社出版, 2010
- 6) 構造計画研究所, MAS コミュニティ, <http://www.mext.go.jp/>, 確認: 2012年2月
- 7) 星野 力, 人工生命的夢と悩み -コンピュータの中の知能と行動の進化-, 裳華房出版, 1994
- 8) A.M.Assad and N.H.Packard, “Emergent Colonization in an Artificial Ecology”, Proc.of the First European Conference on Artificial Life, pp.143-152, 1992