

平成 13 年度知能システム科学専攻修士論文

人工生命によるミーム概念のモデル化と分析

大石 貫之

Conceptualizing Meme on A-Life and its Analysis

OISHI Tsurayuki

提出年月日 平成 14 年 2 月 26 日 (火)

主査教官：山村 雅幸 助教授

審査教官：小林 重信 教授

審査教官：樺島 祥介 助教授

# 人工生命によるミーム概念のモデル化と分析

大石 貫之

## Conceptualizing Meme on A-Life and its Analysis

OISHI Tsurayuki

**abstract:** *Meme* was proposed by R.Dawkins as a new replicator designed for a unit of imitation. It is an exciting idea that provides a new viewpoint for social phenomena such as culture, fashion, and language. Original concept or thought about the meme is pretty abstract, and it is difficult to specify the concrete form. S.Blackmore defined the meme as *whatever it is that is passed on by imitation* and explained various social phenomena as results of gene-meme interaction.

In this paper, at first we propose a basic model of the meme on A-Life. It is based on Multi-agent system where each agent evolves through many generations. The agent's policy is realized by a decision-tree, and we make up meme in some parts of the tree. The tree is updated by Genetic Programming when in generation alternation and by ID3 in imitation. Next, we aim for realizing two phases in it, *gene-drive* and *meme-drive* that were stated by S.Blackmore as typical phenomena evoking by meme. We show the preliminary results and analyses.

### 1 はじめに

ミーム (meme) とは, Dawkins によって提唱された仮想的な因子である。遺伝子が生物を形づくり, その複製の基本的な単位であるように, ミームは文化やアイデアといった情報を形づくり, その複製の基本的な単位となるものとして発想された。ミーム概念は, Dawkins の利己的遺伝子説を情報・流行・文化・宗教などの社会現象にまで発展させる概念として注目されたが, 一方でミームは遺伝子のように確定した形を持たないことから, 実際の現象に適應する際に, その単位や構成を定めることが困難であるという問題があった。

これに対し Blackmore は Dawkins の考えを押し進め, 模倣によって複製されるあらゆるものがミームという自己複製子となりうることを主張した。ミームをそのように定義することで, ミームの構成単位を事象に合わせ柔軟に変えることができ, 様々な進化・社会現象を遺伝子とミームの相

互作用によって説明することが可能になるという。Blackmore によれば, 遺伝子とミームには 2 つの段階の相互作用, 1) 遺伝子の進化によってよりミームが支配的になっていく段階, 2) 支配的になったミームによって遺伝子・ミームに新たな淘汰圧が生まれる段階, があるという。

本研究では人工生命システムを用いてミームをモデル化し, モデル上における現象を観察・分析することを目的とする。

その理由は, 第 1 に, ミーム概念を社会現象に適用することを考えたとき, 社会現象は大規模で実験そのものが容易ではないということである。人工生命システムによるモデルでの現象がすぐさま実際の現象に適用できるということはないが, 実際の現象にミーム概念を適用する際の何らかのヒントを与えることは可能になるかもしれない。第 2 に, 人工生命に交配とは異なる複製オペレーションとして模倣を加えたときに, 新たな創発性を与えることができるのではないかと期待である。

本研究では、まずミームにとって不可欠な模倣オペレーションを人工生命システムに取り入れたモデルを提案する。次にモデルによる実験を行う。実験は、Blackmore の唱える遺伝子とミーム相互作用の 2 つの段階を対象に行い、その中でモデルの持つ創発性に注目しつつ観察・分析する。

## 2 準備

### 2.1 人工生命

人工生命 (artificial life) とは、Langton によって提唱された、自然界の生物特有の振る舞いをする人工システムに関する研究分野である [Langton 89]。情報・計算といった視点から生命をとらえ、コンピュータなどの人工的なメディアで生命の諸現象を合成することで、生命の原理を解き明かすことを目的としている。人工生命システムは以下の特徴を備えている。

- ・ システムは簡単な行動規則 (プログラム) を持った構成要素の集団である。
- ・ 構成要素全体を集中制御する存在はなく、システムとしての振る舞いは、構成要素間の局所的な相互作用によって創発 (emergence) される。
- ・ 構成要素集団が増殖・変異・淘汰のメカニズム、すなわち進化のメカニズムをもつ。(構成要素全体が常に流動的である)

人工生命は、地球上にいる『我々の知る生命 (life-as-we-know-it)』に関する生物学的現象を研究するひとつの方法であるということに加え、生物学の領域を『存在しうる生命 (life-as-it-could-be)』にまで広げる役割も果たしている [Langton 92]。

また工学の面からは、自律性・進化・適応といった生命がもつ特徴を備えたシステムを構築することを目的とした研究がなされている。

本研究では、人工生命の持つ進化のメカニズムに、模倣というオペレーションを加えることでミームが創発されうる環境をモデル化する。

### 2.2 ミーム

ミーム (meme) とは、Dawkins によって提唱された概念である。Dawkins は、自分自身の複製を残すことを目的とした自己複製子 (replicator) として遺伝子を捉え、遺伝子が競合することで生物の進化が起こると説明する利己的遺伝子説 (selfish-gene theory) を唱えた [Dawkins 76]。その中で、遺伝子以外の自己複製子が存在する可能性に触れ、『文化の伝播や模倣の単位となる自己複製子』としてミームを提唱した。

しかしながらミームは、遺伝子に対する DNA のような定まった形をもってはいない。そのためにミーム概念を実社会に適用しようとする場合に、何をもちミームの単位とするのかを明確に定義することが困難であり、そのことがミーム概念に対する批判の対象となっていた。

これに対し Blackmore は『模倣によって伝わるものなら何でも (whatever it is that passed on by imitation)』ミームの単位としてよいと主張した [Blackmore 99]。ミームの単位をこのように定めれば、それは自己複製子の必要条件

- ・ 遺伝 (heredity)
- ・ 変異 (variation)
- ・ 淘汰 (selection)

をみとす。また、場合に応じてミームの単位を変えることで、ミームによってより多くの事象を説明することができる。

さらに Blackmore は、生物の進化における遺伝子とミームの相互作用・共進化に注目し、そこには大きく分けて 2 つの段階があるとしている。

#### 遺伝子による駆動 (genetic drive)

模倣のための身体能力が獲得され、さらに模倣することが進化的に優位性を持つ場合、生物は遺伝子によって模倣がより効果的に働くような身体へと進化する。結果としてミームの複製の精度が増し、遺伝子に対してミームが支配的になっていく段階。

例：模倣することで人間は上手な食べ物の採り

方や道具の使用法といったミームを伝えあうことが可能になった。より上手に模倣することが有利になり、そのことが脳の肥大化を引き起こした。結果としてますますミームが活動しやすくなり、言葉や文字といった高度で人間に欠くことのできない支配的なミームが生まれていった。

### ミームによる駆動 (memetic drive)

遺伝子に対して支配的になったミームが、遺伝子に対して、またミーム同士で新たな淘汰圧を発生させる段階。

例：人間はしばしば、「顔が好みだ」「趣味が合う」というような、遺伝子的な有利さとは別の価値観によって配偶者を決めており、しかもその価値観は時代・地域によって次々と変わっている。

### 2.3 模倣

模倣にはさまざまな種類のものがある。行動を真似ることも模倣であるし、絵画の模写も模倣である。しかし模倣に共通しているのは、複製元そのものを直接複製するのではなく、複製元の特徴を抽出してその特徴を持つものを再構成するという点である。遺伝子の複製は遺伝子を直接複製することで行われるので、遺伝子とミームとはその複製の仕組みが大きく違う。

このように模倣が間接的な複製方法であるため、ミームには遺伝子にはない以下の特徴を持つ。

- ・ 世代を超えた複製が可能となる（遺伝子は親から子へのみ）
- ・ 不特定多数複製が起こり、ミームは常に混じり合っている（遺伝子は2つの親から1度だけ混じる）

したがってミームをモデル化するにあたっては、間接的な複製であるという模倣の特徴をうまく表現するようなモデルを作ることが重要である。

## 3 人工生命上でのミーム概念のモデル化

本章では、ミーム概念のモデル化を行う。まず全体の枠組みを述べ、次いでその具体的な実装例を示す。

### 3.1 枠組み

まず、多数のエージェントから構成される系を考える。エージェントは、

- ・ 遺伝子  $G$             遺伝子
- ・ 内部状態  $S$         後天的に変化するパラメタ（年齢・体力など）
- ・ 属性  $P$             後天的に変化しないパラメタ（体の色など）
- ・ 行動決定規則  $M$    環境からの入力に対して行動を出力する規則

から成る（図1）。

属性  $P$ 、行動決定規則  $M$  は遺伝子  $G$  から発現した表現形であるとし、エージェントはこの  $G$  をベースとして遺伝的アルゴリズムによる世代交代を行う。 $P$  はエージェントの一生を通じて不変であるが、 $M$  は遺伝子より発現した不変の部分  $M_{static}$  の他に、後天的に可変な要素  $M_{flex}$  を含む。ここで後天的とは、遺伝子によって表現されておらず、したがって親から子へ遺伝しないことを意味する。

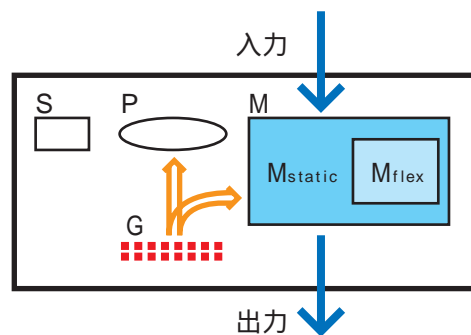


図1: エージェント

このエージェントに、近くにいる他エージェントの動作（入出力）を参照して  $M$  を更新するという機能を持たせることにする。具体的には、現在の自

分の  $M$  と相手の  $M'$  に対して入出力セットのサンプルを取り，その入出力セットに基づいて  $M$  を更新する（図 2）．ここで重要なのは，この操作は  $M$  を直接複製することではなく， $M$  の特徴を抽出し再構成する操作であるということである．つまりこの操作はエージェントに模倣の能力を加えているといえる．

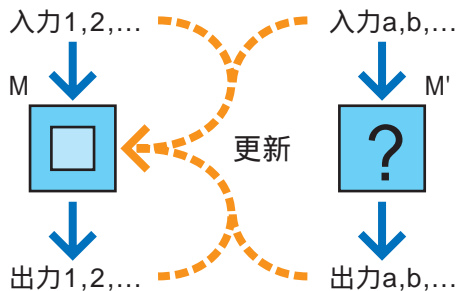


図 2: 模倣オペレーション

模倣オペレーションをこのように設定すれば，行動決定規則  $M, M'$  において， $M_{flex}, M'_{flex}$  が十分な割合をしめる場合， $M'_{flex}$  が模倣を通じて  $M_{flex}$  に複製されたと見ることができる（図 3）．つまり，このような設定の下では， $M_{flex}$  を遺伝子とは独立して模倣によって複製される因子と見ることができ，従って，Blackmore の定義に従い  $M_{flex}$  をミームと呼ぶことができる．

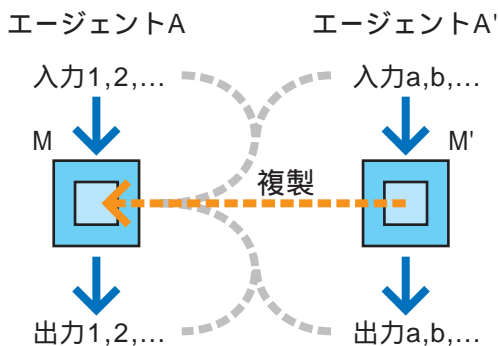


図 3: 模倣による m の複製

### 3.2 実装例

先に述べた枠組みを実装した具体例を示す．なお入出力の組み合わせやパラメータの調整などは，実

験の目的に合わせて変更される．

#### 3.2.1 エージェントと環境

エージェントは，

- ・ 遺伝子  $G$                     実数値・線形決定木
- ・ 内部状態  $S$                 年齢・体力（実数値 / 可変）
- ・ 属性  $P$                       体の色（離散値 / 不変）
- ・ 行動決定規則  $M$         決定木

によって表される．

マス目に切ったトーラス状のフィールドに多数のエージェントと餌を配置する（図 4）．エージェントはフィールドをランダムウォークしつつ，環境からの入力に対して行動決定規則  $M$  に基づいて行動出力を行う．エージェントは内部状態  $S$  として年齢と体力をもち，行動する度に体力を消費し体力が尽きるか寿命を迎えると死ぬ．エージェントは餌に重なると餌を食べ体力を回復する．食べられた餌は一定時間後に回復する．

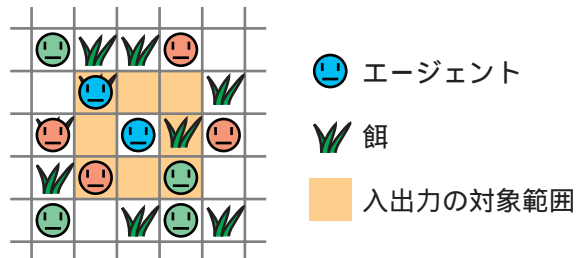


図 4: 環境

#### 3.2.2 入出力

環境からの入力は， $S$  およびエージェントの近傍 8 マスの状態によって決定される．入力としては，

- ・ 自分の年齢
- ・ 自分の体力
- ・ 自分の色
- ・ 近傍エージェントの年齢
- ・ 近傍エージェントの体力
- ・ 近傍エージェントの色

が用意されている．近傍に他のエージェントが複数存在する場合はランダムで 1 匹が選択される．

エージェントの行動出力は,

- ・ 何もしない
- ・ 交配する
- ・ 模倣する
- ・ 餌を調理する (5.2.1 参照)

が用意されている.

交配および模倣は入力された近傍エージェントが出力の対象となる. エージェントが交配を選択すると一定確率で近傍に 1 匹のエージェントが新たに誕生する. 模倣が選択されると近傍のエージェントの行動決定規則を対象として模倣オペレーションが行われる.

入出力は空間の爆発を押さえるために, 問題に合わせて選択して使用している.

### 3.2.3 行動決定規則

行動決定規則  $M$  は,

- ・ 遺伝的アルゴリズムを適用できる
- ・ 遺伝子から発現する先天的な部分  $M_{static}$  と後天的に更新できる部分  $M_{flex}$  を持つ.
- ・ 入出力の教師信号をもとに  $M_{flex}$  を更新できる

という特性をもつ任意の実装が可能である. 今回は決定木を用いる. 決定木を構成するノードを示す (表 1).

判定ノード	<ul style="list-style-type: none"> <li>・ {自分, 相手} の {年齢, 体力} がしきい値より大きいか?</li> <li>・ {自分, 相手} の色がある色であるか?</li> <li>・ {自分, 相手} の体力が増加傾向であるか? (6.3 参照)</li> <li>・ 相手がいるか?</li> </ul>
行動ノード	行動を出力する

表 1: 決定木の構成ノード

これらのノードが線形表現で遺伝子に保持され, 発現して  $M_{static}$  となり,  $M$  の基本構成となる. これに対し, 後天的に更新可能な  $M_{flex}$  は, 別の決定

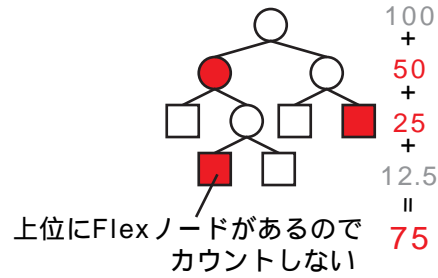


図 5: 算出例

木として  $M$  に保持される.

$M_{static}$  の各ノードには, Flex ラベルというラベルが張られることがある. 遺伝子のみの場合, つまり  $M_{static}$  のみの場合には, この Flex ラベルは何の意味も持たないが, 模倣が有効な場合, つまり  $M_{flex}$  がある場合には, Flex ラベルは  $M_{flex}$  へのリンクの役割を果たす (図 6). Flex ラベルが張られたノードを Flex ノードと呼ぶ.

Flex ノードの数・深さに注目すれば, 模倣がどれだけ働きやすいか, 遺伝子がどれだけ模倣に有利であるかを測定することができる. これを遺伝子の *Flexibility* とし, 自分より上位に Flex ノードを持たない Flex ノード  $N_{flex}$  の深さ  $D_{flex}$  (ルートノードを深さ 0 とする) を用いて式 (1) のように定める. 算出例を (図 5) に示す.

$$Flexibility = \sum_{N_{flex}} \frac{100}{2^{D_{flex}}} \quad (1)$$

初期個体の  $M$  がランダム生成される際に, 各ノードが Flex ノードになるか否かは一定の確率 (*FlexNode probability*) によって決定される (4.1 参照).

### 3.2.4 遺伝的アルゴリズム

遺伝子の表現および交叉の方法を示す (表 2).

表現型	コード	交叉
属性 $P$ (色)	実数値	BLX- $\alpha$
行動決定規則 $M$ (決定木)	LinearGP	SizeFair

表 2: 遺伝的アルゴリズム

$P$  である色は実数によってコード化され, 交叉オ

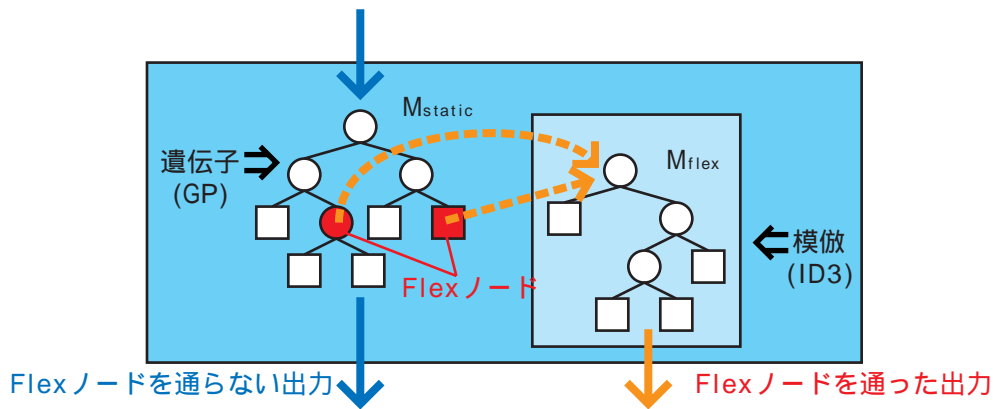


図 6: 行動決定規則

ペレーションには  $BLX-\alpha$  [Eshelman 93] を用いる。

$M$  である決定木には LinearGP [Keith 94] を用いる。LinearGP は決定木を線形化した遺伝子に対して交叉・突然変異を行う手法である。交叉オペレーションにはサイズの近い部分木同士を交叉させる SizeFair 交叉 [Langdon 99] を用いる。突然変異オペレーションは判定ノードのしきい値の変更と行動ノード (Flex ノード含む) の変更を行う。

決定木に対する操作は Flex ノードもその対象となるが、それはあくまで Flex ノード自体の数および位置が変化するということであり、 $M_{flex}$  は決して対象にはならない。

### 3.2.5 模倣オペレーション

模倣は複製元の特徴を抽出し、再構成する操作である。本モデルでは、相手の  $M$  に対して入出力のサンプル (特徴) を抽出し、それに基づいて  $M_{flex}$  を再構成することで模倣をモデル化している。

入出力サンプルを採取するための入力は、年齢・体力などの実数値を実効範囲で離散化し、それらによって作られる入力のすべての組み合わせを用いる (表 3)。これらの入力をランダムに相手の  $M$  もしくは自分の  $M$  に入力し、出力と対して入出力サンプルとする。自分自身からどれだけのサンプルを採取するかは一定の割合 (*self-sampling ratio*) で決める (4.2 参照)。

実装では、入出力サンプルからの  $M_{flex}$  の再構成に ID3 [Quinlan 86] を用いる。サンプル入出力のう

ち、実際に Flex ノードに至るものを教師セットとして ID による  $M_{flex}$  の再構成が行われる。ID3 は同一の教師セットに対しては同一の木を生成するため、ID3 自体が多様性を生むことはない。しかし、サンプルに自分の入出力も含めていること、模倣する側の Flex ノードの位置によって実際に ID3 で使われる教師セット数が変わることなど、教師セットに多様性が生じるため、ID3 を用いた模倣オペレーションも多様性を実現する。

入力値	離散値	種類
自分の体力	1, 3, 5, ..., 19	10 種類
相手の年齢	1, 3, 5, ..., 19	10 種類 + 相手無し
相手の色	赤, 緑, 青,	3 種類 + 相手無し
サンプル入力の組み合わせ		310 通り

表 3: サンプル用入力の組み合わせ例

以上のようなモデル設定で、系としての平衡がなるべく保たれるようなアルゴリズムや設定上の工夫を行いシミュレーションを行う。

### 3.3 数理的モデルとしての伝染病との違い

ミームが伝染病 (の病原菌) と異なっているのは、複製の方法である。病原菌は遺伝子ベースで複製を行うため、複製元 (親) と複製先 (子) は遺伝子もその振る舞いも似ている。一方ミームは模倣によって複製される。複製元と複製先は、振る舞いは似て

いるが、その内部表現は必ずしも似ているとは限らない。また、遺伝子は親から子へ方向一度のみの複製であるが、ミームは世代に寄らず、不特定多数の相手との複製・融合が何度でも可能である。

ただし、病原菌の系を数理的モデルとして考える場合、病原菌の世代交代の時定数が極く短い場合には、ミームと伝染病との差は明確ではなくなる。病原菌の世代交代の時定数が十分短ければ、ミームの持つ不特定多数の相手との多数回の複製・融合という特徴と同程度の効果を代替することが可能となると考える。

しかしながら、交配と模倣という複製方法の違いによる多様性（もしくは一極集中）の違いは、最後まで残ると考える。

#### 4 予備実験

本章では、予備実験として 3.2 で述べたモデルの実装における、初期決定木の柔軟性と模倣オペレーションの伝搬能力について予備実験を行い検証する。

##### 4.1 初期決定木の柔軟性

初期個体が生成される時、その行動決定規則  $M$  はランダム生成される。その際に決定木の各ノードが Flex ノードになるか否かは *FlexNode probability* によって決定されると述べた (3.2.3 参照)。ここでは *FlexNode probability* によって、初期個体の柔軟性 *Flexibility* がどのように変化するかを調べる。*Flexibility* は模倣の働きやすさを決める大きな要因であるため、初期個体の *Flexibility* を調整することはシミュレーションを行う上で重要である。

以下に *FlexNode probability* を変化させた場合に、100,000 回ランダム生成した  $M$  の *Flexibility* の分布がどのように変化するかを挙げる（詳しくは付録を参照）。

全体として、*Flexibility* は 0 または 100 に集中する傾向がある。図 8 では平均 *Flexibility* は 45.6 であるが、実際には 4 割近くがルートに Flex ノードを持つことになる。ルートが Flex ノードになる確

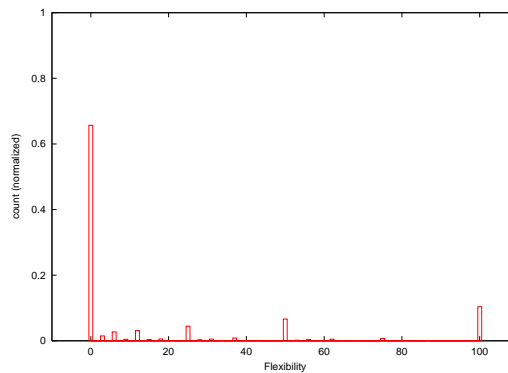


図 7: *FlexNode probability*:0.1 *Flexibility* Avg.:17.7

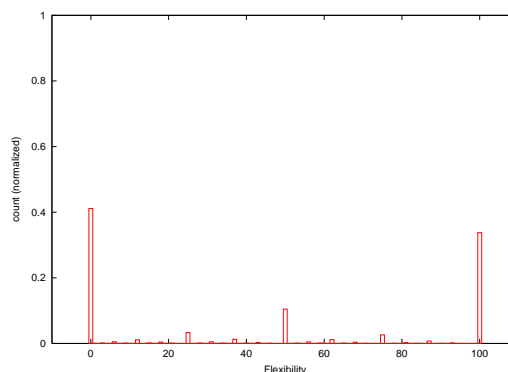


図 8: *FlexNode probability*:0.3 *Flexibility* Avg.:45.6

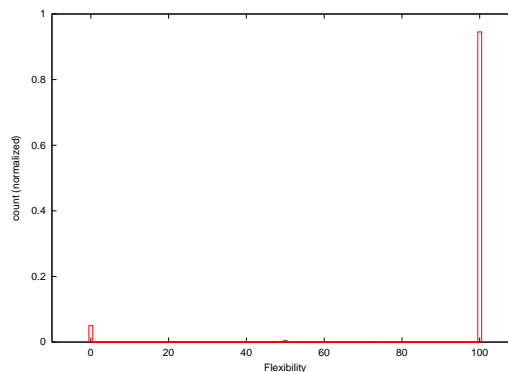


図 9: *FlexNode probability*:0.1 *Flexibility* Avg.:94.8



率が上がる, つまり *FlexNode probability* が上がるにつれて, 0 と 100 以外の *Flexibility* の木はさらに生成されにくくなっていく. *Flexibility* に多様性を持たせるためには, *FlexNode probability* の低いものを使いたい.

#### 4.2 模倣オペレーションの伝播能力

ここでは模倣オペレーションの伝播能力について検証する. ここで模倣の伝播能力とは, 模倣される木 (source tree) の特徴をどれほど模倣する木 (imitation tree) で再現することができるかということである.

どれほど木の特徴を伝えているかは, 入出力サンプルによって測ることにする. すなわち, 模倣オペレーションと同様の組み合わせ (3.2.5 参照) で生成された入力セットを, source tree と imitation tree に入力し, 双方の出力がどれだけの割合で一致しているか (*propagation ratio*) をもって模倣オペレーションの伝播能力とする.

3.2.5 にて, 模倣オペレーションの際に自分自身からどれだけのサンプルを採取するかは *self-sampling ratio* にて決めると述べた. 以下に, *FlexNode probability* および *self-sampling ratio* を変化させたときの *propagation ratio* の変化の様子を挙げる (図 10).

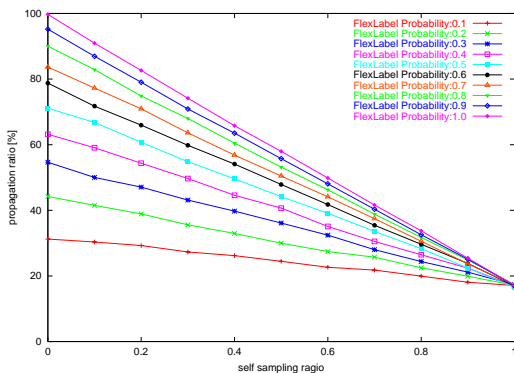


図 10: *propagation ratio*

この実験は, 各設定において 10,000 個体をランダム生成し平均を取ったものをプロットした. 従って, この実験で予測でき得るのは  $M$  がランダム

生成されるシミュレーションの初期状態での様子であるということに注意されたい. Flex ノードの数・位置は, 遺伝子の進化によって変化していくので, シミュレーション中はこの結果に必ずしも従わない.

## 5 シミュレーション

### 5.1 実験目的

今回の実験では, Blackmore の唱える遺伝子による駆動を検証することを目的とする. 遺伝子による駆動とは, 模倣の優位性により模倣がより効果的に働くように遺伝子が進化し, 結果として遺伝子に対してミームが支配的になっていく段階である.

### 5.2 実験計画

Blackmore によれば, 遺伝子による駆動の段階では, 模倣という行為が進化的に優位性をもっていることが前提となる. この実験では次のような仕掛けを入れて実験を行った.

#### 5.2.1 餌の調理

エージェントは餌と重なると無条件で餌を食べるが, ここではそれに加え, 行動出力としてその餌を調理することを用意した. 調理方法は 2 種類用意する. このうちどちらか一方が適した (より美味しくなる) 調理方法で, 通常よりも体力の回復が多くなるとする. ただし, このままでは調理方法を選択することは単なる最適化問題である. そこで, 一定期間ごとに適した調理方法を変更する.

エージェントがどちらが適した調理方法なのかを直接知ることはできない. しかし偶然にも適した調理法を用いているエージェントは多くの体力をもち, 長生きするので, このエージェントと交配するか模倣することで, その他のエージェントは適した調理法を獲得することができる.

ここで, 適した調理方法を変更するタイミングをエージェントの世代交代の期間よりも十分短くすると, 遺伝子の進化ではこの変更に対応できなくなるが模倣では対応できる状態, つまり模倣が優位性を

もつ状態を作り出すことができる。

### 5.2.2 入出力設定

エージェントへの入出力は（表 4）のように設定する。

入力	自分の体力 近傍エージェントの体力
出力	何もしない 交配する 模倣する 調理方法 1,2

表 4: 入出力

入力は「優秀なエージェント = 体力のあるエージェント」という仮定の下、入力空間の爆発を押さえた設定である。

### 5.2.3 環境設定

環境設定を（表 5）に示す。

フィールド	10 × 10
初期個体数	30
初期体力	5
最長寿命	100
行動による体力消費	1.0
餌による体力回復	1.0
交配成功率	10%
餌数	100
餌の回復までの期間	2
BLX- $\alpha$	0.25
<i>FlexNode probability</i>	20%
<i>self sampling ratio</i>	10%
適した調理による付加体力	0.1
適した調理の変更タイミング	1,000step

表 5: 環境設定

### 5.3 予想

この実験の結果として、模倣なし（遺伝子のみ）の場合と模倣ありの場合の違いを予想する。

- ・ エージェント数  
餌によって生死のバランスを安定させているので個体数はほぼ一定で推移
- ・ *Flexibility*  
模倣なしでは Flex ノードが無意味なのでほぼ一定で推移  
模倣ありでは徐々に増加していく（遺伝子による駆動）
- ・ 適した調理を行った割合  
調理法が変わるところで落ち込む  
模倣ありの方が新たな調理法への適応が速い
- ・ 模倣行動  
*Flexibility* の増加に伴って増加する

### 5.4 結果

10,000 ステップの試行を 10 回行った平均を採り、模倣なしの場合と模倣ありの場合を比較した。

エージェント数の推移はどちらも予想通り一定で推移した。*Flexibility* は模倣なしの場合では予想通りほぼ一定で推移したが、模倣ありの場合に増加傾向は見られなかった（図 11）。試行を 30,000 ステップ、100,000 ステップと増やした場合にも増加傾向は見られなかった。適した調理を行った割合は、今回は調理法変更のタイミングを 1,000 ステップに設定したため、1,000 ステップでその割合が下落している様子が分かる（図 12）。ただし、模倣なしの場合とありの場合とで適応速度に目立った違いは認められなかった。模倣行動は *Flexibility* に関わらず常に少なかった（図 13）。

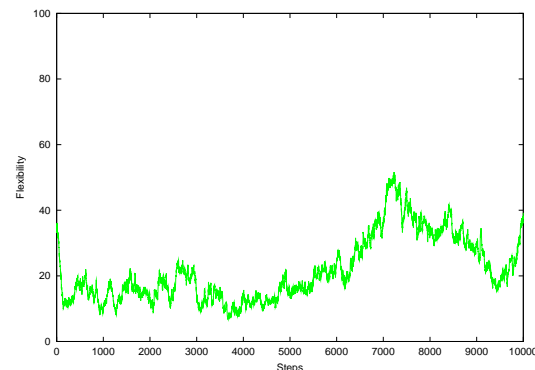


図 11: *Flexibility* (模倣あり)

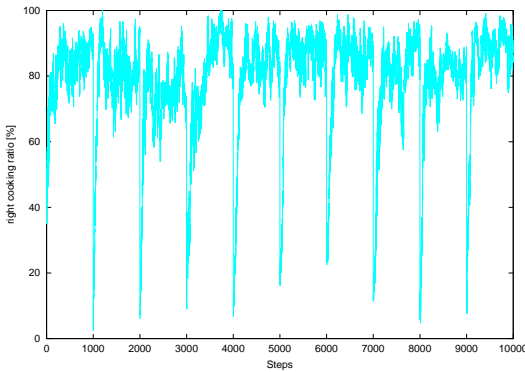


図 12: 適した調理法を行った割合 (模倣あり)

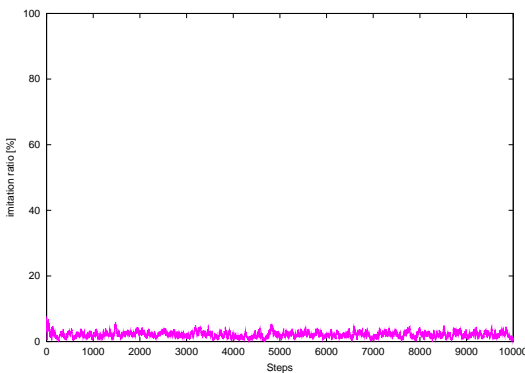


図 13: 模倣を行った割合

## 5.5 考察

*Flexibility* において遺伝子による駆動の様子が観察できなかったのは、模倣の優位性がうまく発揮されていないことによるものと考えられる。これは模倣行動そのものが少なく、また遺伝子の進化の時定数と模倣による進化の時定数が接近しすぎているために、適応能力で模倣が遺伝子を上回ることができないためだと考えている。模倣行動そのものが少ないのは、系を安定させるためにシビアな設定になっているため、生死に直接関わらないような行動（模倣）が阻害されてしまうということのようである。実際に長く生き延びたエージェントの決定木を調べてみると、ルートノードに調理行動があるものが多かった。

## 6 シミュレーション 2

5の実験では、模倣の優位性を確保することができなかった。そこで本章では、模倣の優位性を確保するための条件を整理し、設定の改善し実験を行う。

### 6.1 2種類の最適化機構

遺伝子による駆動の実験は、行動最適化問題における遺伝子と模倣との関係性を調べるものである。つまり本実験には、

- 遺伝子による行動最適化
- 模倣による行動最適化

という、2種類の最適化機構が混在しており、遺伝子による駆動とはこの2つの最適化機構の相互作用を調べるということである。

これら2種類の最適化機構は、 $M_{static}$ 、 $M_{flex}$  がともに決定木によって実装されていることから、同程度の行動決定規則（決定木）を獲得し得る。さらに、 $M_{flex}$  の使われやすさ（*Flexibility*）は  $M_{static}$  によって決定されるため、遺伝子が模倣による行動最適化の強さを制御しているという特徴がある。

### 6.2 模倣の優位性

2種類の最適化機構の混在し、さらに遺伝子が模倣による行動最適化の強さを左右している本実験の環境において、模倣が優位性を保つためには、模倣が以下のことが満たす必要がある。

- 短い時定数 遺伝子による行動最適化の時定数に対して十分短い時定数で最適化を行う
- 動的追従性 環境の動的な変化に対して適応することができる

まず、遺伝子と模倣とによる2種類の最適化機構は、ともに同程度の行動決定規則を獲得し得るので、遺伝子による最適化と模倣による最適化の効果を明確に分けるためには、両者の時定数にはっきりと差をつける必要がある。現実世界では一般に1回の世代交代に対して無数の模倣が行われることを

考え、遺伝子による最適化の時定数に対して模倣による最適化の時定数を十分に短く設定することにする。

2種類の最適化の時定数に差を付けることで、動的追従性についても遺伝子と模倣とに差を付けることができる。本実験において動的追従性とは、一定時間毎に変わる「適した調理法」に適応することができるということである。適した調理法を変更するタイミングを、遺伝子による最適化の時定数と模倣による最適化の時定数の間に設定することで、模倣による最適化のみが動的追従性を獲得することができる。

### 6.3 体力変化

模倣の優位性には動的追従性が重要となる。つまり、一定時間後とに変わる「適した調理法」を行うことができているか否かが重要となる。しかしながら、適した調理法が変わった場合に、ある個体が適した調理法を行っているか否かは、体力・年齢からでは測ることができない。そこでエージェントの内部状態  $S$  として、新たに体力変化を設ける。これは、過去の一定時間において、体力が増加傾向にあるかまたは減少傾向にあるかを示すパラメタである。

行動による体力消費・餌による体力回復・適した調理による体力回復を調整することによって、この体力変化を(表6)のように設定することができ、体力変化によって適した調理を行っているか否かを測ることができる。

	体力変化
通常	減少傾向
適した調理をした場合	増加傾向

表6: 適した調理法と体力変化

### 6.4 強制的な模倣

5の実験では、模倣を行動出力の1つとしていた。従って、模倣が優位性をもつためには、効率良く生きている相手を模倣するということが行動決定とし

て獲得される必要があった。しかしながら、遺伝子の最適化の時定数が短かったこと、環境設定がシビアであったことなどから、模倣行動は初期の段階で遺伝子によって阻害されてしまった。

そこで、今回は模倣を行動出力に加えず、体力変化が増加傾向にあるエージェントと出会ったら強制的に模倣することとした。もちろん本来であれば模倣を行動出力に加えるべきであるが、まずは、模倣によって適した調理法への最適化が行われるかどうかを確かめる意味で、強制的に模倣を行う設定とした。

## 6.5 設定

### 6.5.1 入出力設定

エージェントへの入出力は(表7)のように設定する。

入力	自分の体力 自分の体力変化 近傍エージェントの体力変化
出力	何もしない 交配する 調理方法 1~6

表7: 入出力

模倣は行動出力から外された。調理方法は、遺伝子による最適化の時定数を大きくしたことに伴って6種類に増やした。

### 6.5.2 環境設定

環境設定を(表8)に示す。

## 6.6 結果

30,000 ステップを10試行した平均を示す(図14,15,16,17)。

*Flexibility* は、遺伝子のみの場合にはほぼ横ばいに推移している(図14)。模倣ありの場合には緩やかではあるが増加傾向がある(図15)。ただし値は45付近に収束している。50,000 ステップまで試行を続けた場合もそれ以上の上昇は確認できなかった。

フィールド	10 × 10
初期個体数	70
初期体力	5000
最長寿命	100000
行動による体力消費	1.0
餌による体力回復	1.0
交配成功率	100%
餌数	100
餌の回復までの期間	5
BLX- $\alpha$	0.25
<i>FlexNode probability</i>	20%
<i>self sampling ratio</i>	10%
適した調理による付加体力	2.0
適した調理の変更タイミング	1,000step

表 8: 環境設定

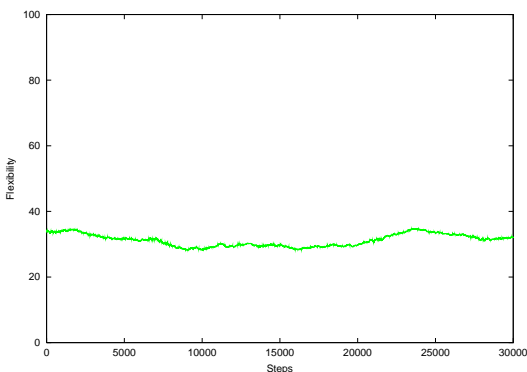


図 14: *Flexibility* (遺伝子のみ)

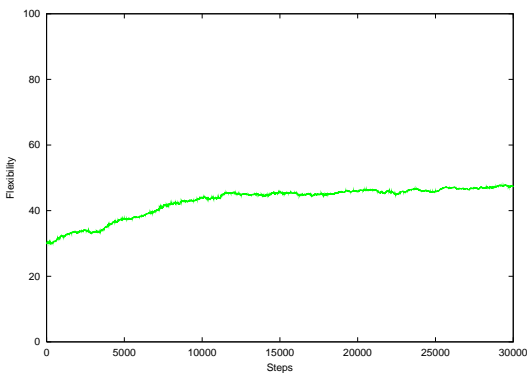


図 15: *Flexibility* (模倣あり)

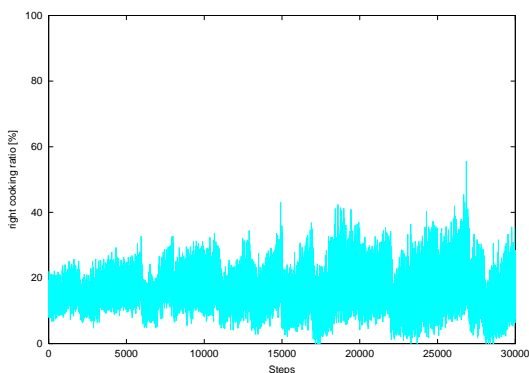


図 16: 適した調理法を行った割合 (遺伝子のみ)

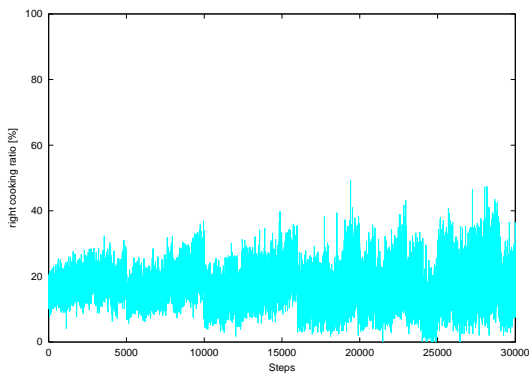


図 17: 適した調理法を行った割合 (模倣あり)

適した調理を行った割合は、遺伝子の場合と模倣ありの場合ではっきりとした差を見ることはできなかった。どちらの場合も次第に振幅が大きくなっていく傾向が見られた(図 16,17)。

## 6.7 考察

今回の実験では、遺伝子による最適化の時定数を大きく設定した(エージェントの平均寿命は 6,300steps~6,500steps)ため、遺伝子による最適化はかなり緩やかに行われていた。しかしながら、実験全体を通じて模倣による最適化が弱いと感じた。そのことは、図 15 のように *Flexibility* が低い段階で収束してしまうこと、図 16,17 のように遺伝子と模倣とで適した調理法への最適化にはっきりした差が見られないこととして現れていると考える。模倣条件および模倣オペレーションの実装を見直すことで、模倣による最適化の効率が図れるのではないかと考えている。

## 7 おわりに

本研究では、模倣によって伝わる自己複製子であるミーム概念を、遺伝子とミームの相互作用を観察・分析すべく人工生命システムを用いてそのモデル化を試みた。人工生命においてミームを創発させるために、「複製元の特徴を抽出してそれを基づいて再構成する」という性質を再現する模倣オペレーションを提案し実験を行った。実験では、Blackmore がの唱える 2 つの段階の相互作用のうち、まず遺伝子による駆動の段階を検証することを目的とした。実験の結果、遺伝子による駆動に沿う現象が観察された。しかしながら模倣による最適化が弱く、遺伝子の場合との明確な差を確認するには至らなかった。

今後は模倣条件および模倣オペレーションの実装を見直し、遺伝子による駆動の検証を続けるとともに、ミームによる駆動の検証実験も行いたい。

## 謝辞

本研究を行うにあたり、終始暖かいご指導頂いた山村雅幸助教授に深く感謝致します。

また、不出来な後輩に懇切丁寧に指導して下さった田中文英様、吉良聡様始めとして山村研究室の皆様にご感謝致します。

## 参考文献

- [Blackmore 99] Blackmore,S., *The Meme Machine*, Oxford University Press, 1999.
- [Dowkins 76] Dawkins,R., *The Selfish Gene*, Oxford University Press, 1976.
- [Eshelman 93] Eshleman,L.J. and Schaffer,J.D., *Real-Coded Genetic Algorithms and Interval-Schemata*, *Foundations of Genetic Algorithms 2*, pages187-202, 1993.
- [Keith 94] Keith,M.J., and Martin,M.C., *Genetic Programming in C++:Implementation*

*Issues*, chapter13 of *Advances in Genetic Programming*, edited by Kinneer,K.E,Jr., pages285-310, MIT Press, 1994.

[Langdon 99] Langdon,W.B., *Size Fair and Homologous Tree Genetic Programming Crossovers*, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages1092-1097, 1999.

[Langton 89] Langton,C.G., *Artificial Life, Artificial Life*, pages 1-47, Addison-Wesley, 1989

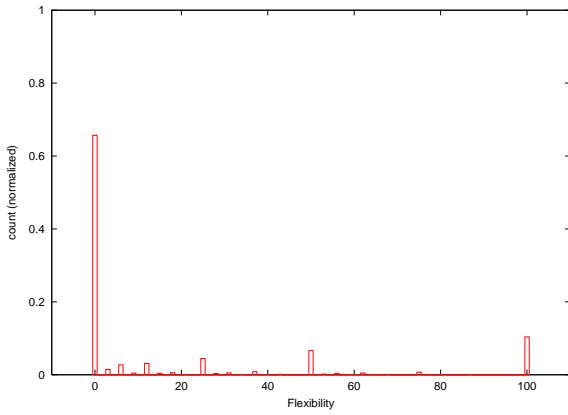
[Langton 92] Langton,C.G., *Preface, Artificial Life II*, pagesxiii-xviii, Addison-Wesley, 1992.

[Quinlan 86] Quinlan,J.R., *Induction of Decision Trees*, *Machine Learning*, Vol.1, No.2, pages81-106, 1986.

# 付録 Flexibility の分布

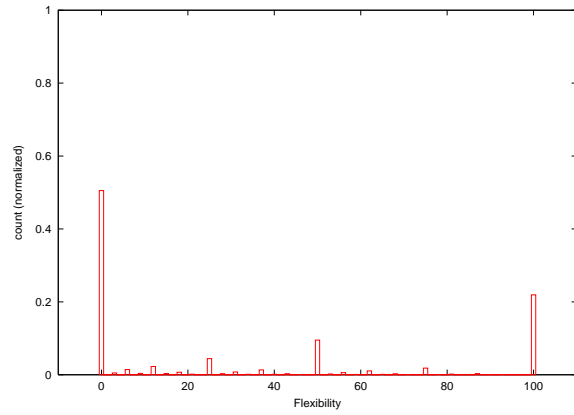
FlexLabel probability:0.1

FlexUnitRank Avg.:17.7



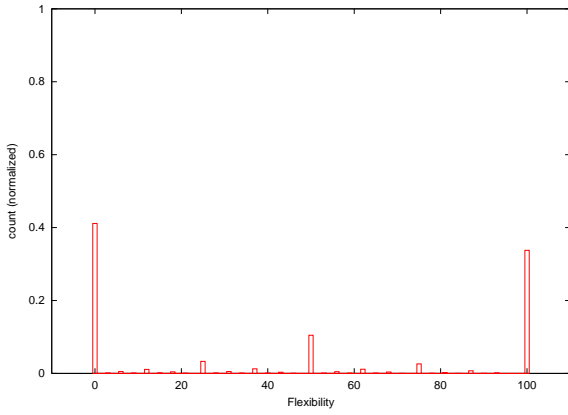
FlexLabel probability:0.2

FlexUnitRank Avg.:32.9



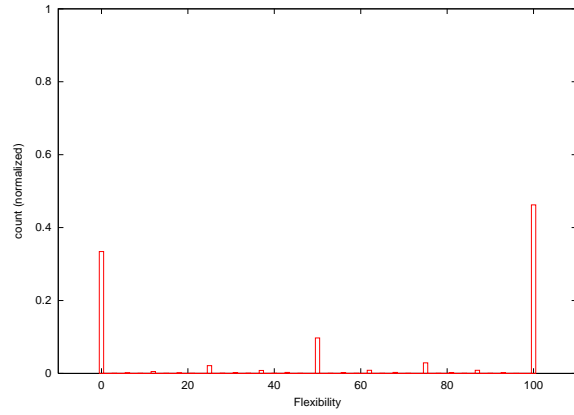
FlexLabel probability:0.3

FlexUnitRank Avg.:45.6



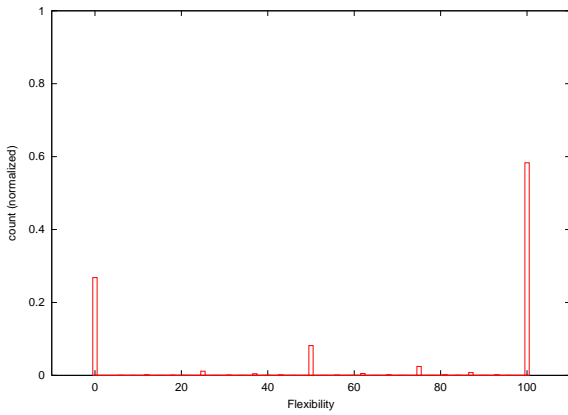
FlexLabel probability:0.4

FlexUnitRank Avg.:56.7



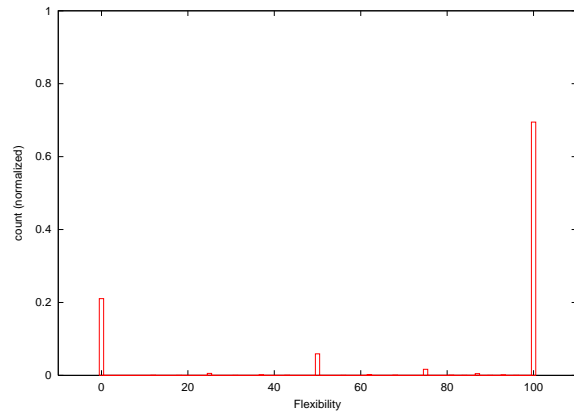
FlexLabel probability:0.5

FlexUnitRank Avg.:66.4



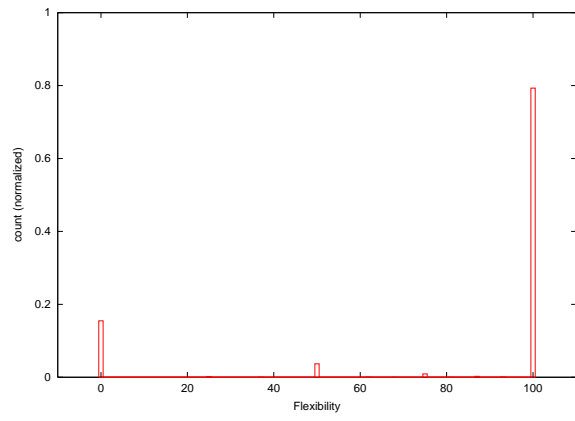
FlexLabel probability:0.6

FlexUnitRank Avg.:74.8



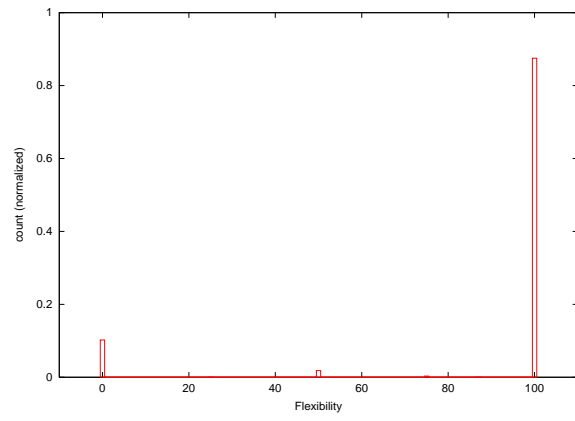
FlexLabel probability:0.7

FlexUnitRank Avg.:82.2



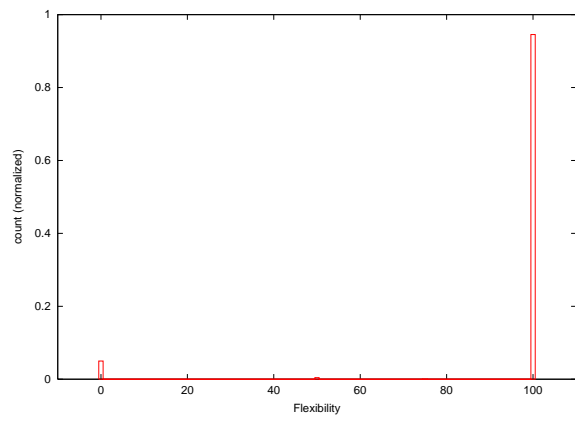
FlexLabel probability:88.7

FlexUnitRank Avg.:



FlexLabel probability:0.9

FlexUnitRank Avg.:94.8





## 修正部分

- 数理的モデルとしての伝染病都の違い  
樺島先生からのご質問がございましたので書き加えました。(3.3)
- シミュレーション  
提出時から発表時までに行った実験に基づいて修正致しました。(5)
- シミュレーション 2  
発表時以降に行った実験に基づいて書き加えました。(6)
- その他  
全体を通して文章，図版の修正を行いました。