

エージェントの進化と問題解決能力

Agents' Evolution and Ability to Solve Problems

間遠伸一郎^{*1}

Shinichiro MADO

^{*1} 宇都宮大学

Utsunomiya University

Abilities to solve problems are very important for intelligent agents to make their evolution of NNTP especially in the case with simulations of human communities. Therefore Massively Parallel Artificial Intelligence: MPAI Agent Simulations should adopt Intelligent Social Evolution Mechanisms: ISEM which use problem solving by each agent to change its non-Neumann type program: NNTP instead of mutations and also use social selections instead of natural selections. Chromosomes are special case of NNTP. Mutations are extremely simplified case of changes by problem solving. And natural selections which depend on natural environment are extremely special case of social selections which depend on social environment. Thereby Genetic Algorithms: GA is an extremely simplified special case of ISEM.

1. 問題の所在

生物や社会などの複雑系に特徴的なことの一つは、それが進化によって形成された系であるということである。生物については、その進化の仕組みが Genome と自然淘汰からよく説明されている。遺伝的アルゴリズムを用いた自然淘汰のコンピュータシミュレーションによる Genome の進化の研究も行われている。しかし、社会については、その進化の仕組みを生物と同様の枠組みで説明することはできない。例えば、社会の Genome は存在しないし、社会の進化は自然淘汰とは異なる仕組みによるものである。

社会を複雑系として取り扱う研究が複数の研究者によって行なわれるようになってきてはいるが、社会の進化を遺伝的アルゴリズムによる社会の Genome の進化として理解しようとする誤った研究が多く見受けられる。これらの、社会を生物に類比した議論は、生物と社会の本質的な差異を見過している。本論では、この生物と社会の本質的な差異を視野に入れて、複雑系としての社会を計算機シミュレーションによって研究するための、社会の進化のモデル化の方法について一つの重要な提案を行なう。

2. 複雑系のシミュレーションと遺伝的アルゴリズム

2.1 遺伝的アルゴリズム GA

遺伝的アルゴリズム Genetic Algorithms: GA は 1960 年代に John Holland によって考案された。GA は、自然界に見られるランダムな変異と自然淘汰による適応現象をコンピュータ上の適応アルゴリズムとして研究する計算機シミュレーションの技法である。

GA は、染色体による集団(例えば、1 または 0 で表現される文字列)から、一種の自然選択と、交叉、突然変異などのランダムな変異操作を用いて新たな別の集団を生成する。

GA による進化は、適合度関数で定義されたランドスケープ上の、染色体の集団の移動と見ることができる。

GA の基本はつぎのようなアルゴリズムである。

1) 長さ l の染色体からなる n 個体 (n は偶数) を持つ集団をランダムに生成する。

2) 集団のそれぞれの染色体に対して適合度 $f(x)$ を計算す

る。

3) 以下の a.~c. のステップを n 個の子孫が生成されるまで繰り返す。

a. 現在の集団から親となる染色体のペアを選択する。そのときの確率は適合度の増加関数である。

b. 一様な交叉確率 p_c により、ランダムに選択された位置で、交叉をして子孫を生成する。このとき、一組の親からは 2 つの子が生成されるものとする。

c. 突然変異確率 p_m によって、2 つの子のそれぞれの遺伝子座において突然変異が行なわれる。

4) 現在の集団を、新たに生成された集団で置き換える。

5) ステップ 2) に戻る。

この処理の一回の繰り返しを世代という。GA では、100 世代以上の繰り返しを行なう。

2.2 GA の問題点

複雑系は進化によって形成されるシステムなので、複雑系のシミュレーションを行なおうとすると、やはり、進化のシミュレーションが肝心な点になってくる。しかしながら、GA は万能ではない。とりわけ、GA を経済現象などの社会現象に適用しようとする、GA の様々な欠点が浮かび上がってきて、GA の拡張の必要性が生じる。GA の欠点としてはつぎのような点が挙げられる:

1) GA は、偶然性に依存した交叉や突然変異によって Genome の変化を作り出しているが、この変化は知能の働きを要しないので、単純である反面、無駄や犠牲が多い。

2) GA は、自然環境による自然淘汰で Genome の変化を淘汰しているが、社会的環境による社会的淘汰を考慮する必要もあり、それが無いのが GA の欠点である。

3) GA は、進化に世代交代を必要としているが、これは、個体の死滅を前提としていようなものである。進化には、個体の死を前提としない進化もあるが、その様な場合に GA は適さない。

4) GA は、適応度関数を用いた定量的適応度で淘汰を行なうが、現実にはエージェント社会の要素主体が適応を求められる場面では、そのような一次元的定量的尺度で適応度を測ることが馴染まない場合も多くある。適応性の基準が質的なものである場合や、社会選択や競争による場合、他の要素主体との複雑な相互作用の結果としての適応などでは、単純な定量的尺度では相応しくないことが多い。この点も、GA の欠点である。

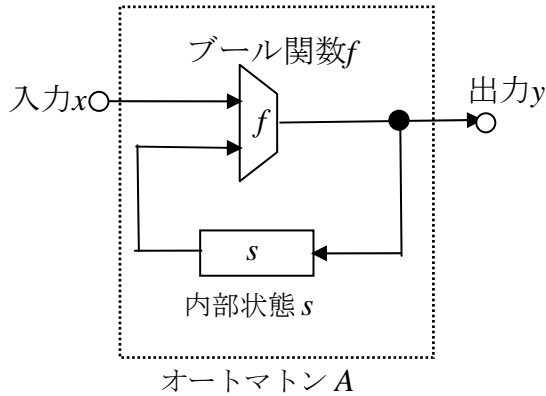
連絡先: 間遠伸一郎, 経済情報科学研究室, 宇都宮大学,
〒321-8508 宇都宮市峰町 350

3. NNTP の概念と ISEM の方法

3.1 オートマトンと NNTP

(1) オートマトン

オートマトンはブール関数に内部状態を介したフィードバックループを加えたものである。



$$\begin{cases} y = f(x, s_{-1}) \\ s \leftarrow f(x) \end{cases}$$

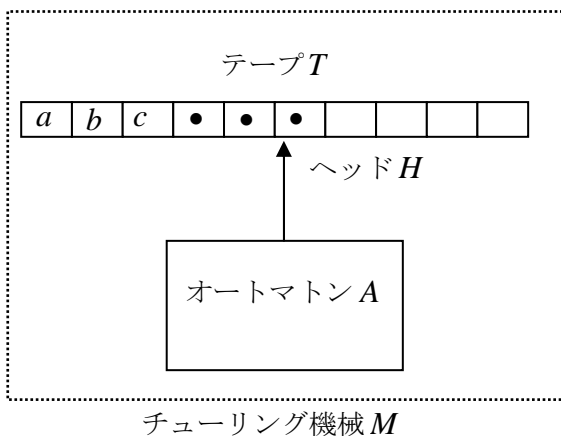
ただし、 s_{-1} は、 s の当該期の書き込みが行なわれる前の値であり、 f の前期の出力が 1 クロックの時間記憶されていたものである。

s の書き込みと読み出しも、 x の読み込みも y の出力も、全てクロックの周期に合わせて行なわれる。

このように、内部状態の記憶 s は、 f の出力から f の次期のフィードバック入力までの、1 クロックの時間だけ、フィードバック通信経路 $f \rightarrow s \rightarrow f$ を遅らせる働きをする。

(2) オートマトンと計算機

オートマトンの概念は計算機の仕組みと深く関わっている。現在の計算機は、その基本的な仕組みをチューリング機械によって行っている。チューリング機械は、オートマトンが、外部のテープを左右に動かしながら、固定されたヘッドからテープ上の情報の読み書きを行ない、データやプログラムを読み込んで、計算した結果をテープに書き出しながら、複雑な計算をする計算機である。このように、チューリング機械には、その中心部分としてオートマトンが含まれているので、オートマトンと計算機の仕組みは切り離せないものである。



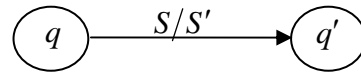
現在使われている大多数の計算機は Neumann 型計算機と呼ばれるものであるが、これは、A. Turing が考案したチューリング機械のテープを有限化して、それを電子的な主記憶として実現し、チューリング機械のオートマトン部分を、算術論理演算部と呼ばれる組み合わせ回路とレジスタ群を含む CPU と呼ばれる電子回路で実現し、それらを通信経路で結んだものである。通信経路を通じて読み書きされるデータまたはプログラムの主記憶上の記憶場所は、その記憶場所のアドレスを、CPU 内部のレジスタの値にセットすることによって制御される。

(3) 作用

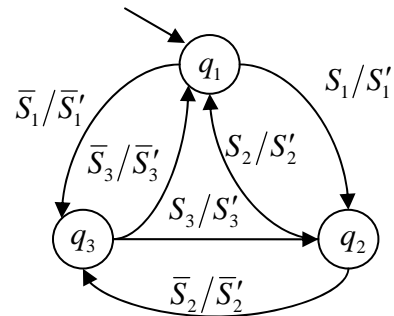
チューリング機械 A において、テープ T からヘッド H を通じて読み出された入力信号 S に対応するオートマトン A の状態変化 $q \rightarrow q'$ には、作用 S' を連動させることができる。

チューリング機械は、4つ組み $qSS'q'$ の集合としても理解できる。ここで S' は、入力信号 S に対する作用を表わしている。

状態遷移図でこの4つ組みを表わすと次のようになる。



一般に、オートマトンは、このような枝が複数結合したグラフで表わされる。つぎの状態遷移図は一つのオートマトンを表わす。



この状態遷移図で表わされるオートマトンを4つ組み $qSS'q'$ の集合で表わすとつぎのようになる。

$$\left\{ \begin{array}{l} q_1, q_1 S_1 S'_1 q_2, q_1 \bar{S}_1 \bar{S}'_1 q_3, q_2 S_2 S'_2 q_1, \\ q_2 \bar{S}_2 \bar{S}'_2 q_3, q_3 S_3 S'_3 q_2, q_3 \bar{S}_3 \bar{S}'_3 q_1 \end{array} \right\}$$

ただし、 q_1 は初期状態である。

(4) CD 変換・IF-THEN ルール・NNTP

記号の認知 Cognitive: C と指令 Directive: D を結びつける要素プログラムを CD 変換と呼ぶ。この場合のプログラムは、ノイマン型計算機の主記憶に格納されたプログラムとは異なり、オートマトンの内部に存在するプログラムである。このような、オートマトンの内部に格納されたプログラムを **非ノイマン型プログラム Non-Neumann Type Program: NNTP** と呼ぶ。それに対して、オートマトンの外部の主記憶に格納されたプログラムを **ノイマン型プログラム Neumann Type Program: NTP** と呼ぶ。

オートマトンの状態変化を考慮するとき、CD 変換は、

$$S \rightarrow S'$$

と表わすより、

$$\langle S, q \rangle \rightarrow \langle S', q' \rangle$$

と、状態の変化を明記して表わす方が良い。これが、非ノイマン型プログラム NNTP を構成する要素プログラムだと考えられるが、これはまた IF-THEN ルールであるとも考えることができる。

すなわち、

$$\text{IF } \langle S, q \rangle \text{ THEN } \langle S', q' \rangle$$

という IF-THEN ルールである。このような IF-THEN ルールが NNTP の要素プログラムであり、NNTP を IF-THEN ルールの集合と考えることもできる。

この IF-THEN ルールは、上で述べたオートマトンの 4 つ組表現と 1 対 1 に対応する。

3.2 オートマトンによる Agent のモデル

(1) Agent の内部プログラム

エージェント Agent とは多主体システム Multi-Agent System の要素主体 Element のことである。エージェントは自律的な主体であり、自ら行動する。要素主体が複数集まって複合主体を形成する場があるが、複合主体をエージェンシー Agency と呼んでいる。

多主体システムの全体、あるいは、エージェント全体の集団を単に社会 Society と呼ぶ。ここで、注意する必要があるのは、エージェントの社会 Society は、必ずしも人間社会 Community を意味するものではないということである。単に、エージェントの集団を全体として言及する必要があるとき、それを社会 Society と呼んでいくにすぎない。人間社会を問題にする場合もあるが、エージェントという概念自体は、本来、人間社会とは無関係である。エージェント社会 Society もまた、本来、人間社会 Community とは無関係である。

エージェントをモデル化するには、オートマトンが適している。オートマトンによって自律的に行動する主体を表現することができるからである。

(2) NNTP のプログラム可能性と Agent の条件

ところで、オートマトンには内部プログラムが存在する。エージェントをオートマトンで表現したとき、オートマトンの内部プログラムとエージェントの内部プログラムはどこが違うのだろうか。

ダニのように単純なエージェントの場合、ダニの全ての行動は本能に基づくものであると考えられる。このような場合、エージェントの内部プログラムは固定されていると考えられるが、この場合でも、オートマトンの内部プログラムがダニの行動プログラムを表現するようにプログラミングする必要がある。つまり、エージェントの場合、オートマトンの内部プログラムがプログラム可能であることが必要なのである。

(3) 超並列 Agent シミュレーション

超並列エージェントシミュレーションとは、おおよそ 100 個以上のエージェントを用いて行なわれるエージェントシミュレーションのことである。100 個以上だから 1000 個でも 10000 個でも良いのだが、使用する計算機と計算機プログラミングの方法によって制約される。

(4) MPAI Agent Simulation

Massively Parallel Artificial Intelligence Agent Simulation (MPAI Agent Simulation) とは、

- 1) エージェントの個数が 100 個以上で、かつ、
 - 2) 個々のエージェントが各々の問題解決能力を有している
- エージェントシミュレーションである。

3.3 進化と NNTP

エージェント社会の進化は、エージェントの NNTP の進化を基礎としている。また、NNTP は、基本的に、IF-THEN ルールという単純な構造の集積である。

(1) 社会の Agent の NNTP と生物の Genome の類似性

生物の Genome も、生物の個体を Agent と見れば、NNTP に他ならない。Genome が生物を決定すると同時に、Genome の進化が生物の進化そのものである。同様に、社会の Agent の場合でも、Agent の NNTP の進化が Agent の進化だと考えられる。

(2) 社会の Agent の NNTP と生物の Genome の差異

生物の Genome は、生物の個体が生まれてから死ぬまでの間不変であるが、一般に、Agent の NNTP は、必ずしも不変とは限らない。むしろ、Agent の NNTP は可変であるのが普通である。それによって、学習や、Agent の性格の変化が可能になる。社会の Agent の進化には、世代交代を必要としないので、個体の平均寿命よりずっと短い時間で適応や進化を達成することが可能である。それに対して、生物の進化は、ずっと長い時間を必要とする。

また、特に、人間社会 Community の Agent シミュレーションの場合には、人間という知能を持った構成主体の精神が NNTP に相当する。この場合も、NNTP は可変であり、個人の生存中に、学習によって変化したり、教育によって書き換えたりすることが可能である。

しかし、それだけでなく、個人の NNTP は、問題解決によって変化する。問題解決の前に、問題認識が必要であるが、問題認識 Problem Cognition: PC と問題解決 Problem Solution: PS の両方もが知能の働きである。

問題認識の結果、その問題が、既知の問題であり、しかも、以前に解決したことのある問題であれば、同じ解法を試み、解決を得ることができる。

逆に、問題認識の結果、その問題が全く新しく遭遇したものであり、解法を知らない場合には、その問題の問題解決のために、新たな思考を必要とする。

つまり、Solvent という一つの知識が、個人の NNTP の構成部分であり、問題 x に対して、

$$PC(x) \rightarrow PS(x)$$

というプログラムが NNTP に含まれていれば、それが Solvent(x) であり、それによって解決 PS(x) を得ることができる。

しかし、Solvent(x) を知らない場合には、Solvent(x) を見出す思考が必要になる。

この場合、問題解決には次の 2 種類がある：

- 1) 既知の Solvent(x) を適用する問題解決、
- 2) 未知の Solvent(x) を見出す問題解決

である。

既知の Solvent を適用して解決できる場合には NNTP は変化しない。未知の Solvent を見出して問題解決を行なった場合に、見出された新しい Solvent が NNTP に追加されることによって NNTP が変化する。

新たな Solvent による NNTP の変化は、生物の Genome に例えれば、突然変異 Mutation に当たる。生物の進化の場合には、突然変異は自然淘汰によって淘汰されなければならないが、社会を構成する個人の NNTP の場合、自然淘汰ではなく、社会淘汰によって淘汰される。

社会的淘汰には様々なものがあるが、代表的なものは法的規制である。警察 Agent の組織 Agency の働きにより、個人 Agent の行動を観察し、法的適合性を持たない行動を起こした

個人 Agent を捉え、その Agent の NNTP を書き換えさせることによって法的適合性を持たない行動を淘汰する。法的規制の根拠となる法律 Law は、個人 Agent にとっては社会環境である。

結局、NNTP と生物の Genome を比較すると、つぎの 1), 2), 3) のような差異が認められる。

- 1) 社会の Agent の NNTP は問題解決という知能の働きによって変化するのに対して、生物の Genome は偶然性によって変化する点が異なる。
- 2) 社会の Agent の NNTP は社会的淘汰によって淘汰されるのに対して、生物の Genome は自然淘汰によって淘汰される。
- 3) 社会の Agent の NNTP は個人の生存中に書き換えられて変化するのに対して、生物の Genome は、個体の生涯にわたって変化しないで、世代交代を通じてはじめて変化する。

3.4 問題解決による NNTP の進化

(1) 知的社会進化メカニズム ISEM

知的社会進化メカニズム Intelligent Social Evolution Mechanism: ISEM は GA の欠点を克服して、問題解決能力を持った知的な Agent を構成主体とした多主体システムにも適用できるようにした、GA より高度な複雑系の進化メカニズムである。

ISEM は、GA のように、偶然性に依存した突然変異によってではなく、Agent による問題解決で NNTP の変化を作り出している。しかし、見方によっては、GA は、知能を要しない方法で問題解決を行なっている特殊な場合と考えることもできる。

ISEM は、自然環境による自然淘汰ではなく、社会環境による社会的淘汰で NNTP の変化を淘汰している点が GA とは異なる。しかし、見方によっては、自然環境を社会環境の特殊な場合と見ることができるとともに、自然淘汰を社会的淘汰の特殊な場合と考えることができる。

ISEM は、Agent の NNTP の書き換えを行なうので、適者生存または世代交代を特に必要としない。しかし、見方によっては、GA は、要素主体の NNTP の書き換えを、世代交代という方法で行なう特殊な場合と考えることもできる。

ISEM は、GA のように、適応度関数を用いた定量的適応度で淘汰を行なうことは普通せず、むしろ、適応性の基準として、論理的適合性などの、質的なものを採用する。また、他の要素主体との複雑な相互作用の結果としての、いわば社会的適応などの場合は、適応度関数が馴染まないで、GA よりも ISEM の方が優れている。

(2) Agent の NNTP の変化と社会的規制が作り出す Agent の進化

- 1) 個人の知能の働きによる NNTP の書き換えと、
- 2) その結果として起こる個人の新たな行動に対する社会的規制をとおしての NNTP の社会的淘汰

によって、個人と社会の進化が進行する。
この進化の仕組みは、生物の進化の仕組みと似ているようでも、実は、大きく異なるものである。

GA は、生物進化のシミュレーションとしては有効であっても、社会変化のシミュレーションとしては、不十分であり、社会変化のシミュレーションには、ISEM が必要となる。

3.5 Agent の問題解決と問題解決戦略

ISEM では、Agent は問題認識と問題解決を行なうが、問題解決に当たって、Agent は**問題解決戦略 Problem Solving Strategy: PSS** を用いる。PSS は Agent によって異なる。良い PSS を持っている Agent は「賢い」とか「知能が高い」ということになるだろう。

ISEM では、Agent の知能は、近視眼的な問題解決能力だけでなく、社会的淘汰を予想して、適切な戦略を選択する能力を持ちうると考える。

また、ISEM では、Agent は自身の PSS 自体を改良する能力を有するとも考える。つまり、Agent は、自らの努力によって、より「賢く」なることも可能である。

4. 結論

生物の進化と社会の進化には本質的な差異がある。複雑系としての社会を Agent ベースの計算機シミュレーションによって研究する際には、社会の進化が次のような進化モデルに従うと考えることが有益である。

すなわち、Agent の進化は Agent の内部の非ノイマン型プログラム NNTP の進化に他ならず、Agent が社会システムの揺らぎから生じる不確実性に直面する際に遭遇する問題を解決する際に Agent の NNTP が改良されるが、この Agent の NNTP の変化がもたらす Agent の行動の変化が、さらに社会的な強制を受けることによって制限または助長され、Agent の主体的な NNTP の改良と社会的な強制による Agent の NNTP の書き換えとが相互作用することによって、トータルな Agent の NNTP の進化のダイナミクスが形成されるという進化モデルである。

進化を Agent の問題解決による NNTP の進化としてモデル化する Agent Simulation を、複雑系としての社会システムのシミュレーションとして採用すると、Agent には、問題認識能力とその問題に対する何らかの問題解決能力が要求されることになる。

一般に、知能とは問題解決能力のことであるから、Agent Simulation の Agent には知能が要求されることになる。こうして、Agent の問題解決によって NNTP が進化するタイプの Massively Parallel Agent Simulation は、Massively Parallel Artificial Intelligence にならざるを得ないのである。

結局、MPAI Agent Simulation では、Agent の問題解決による NNTP の変化と社会的淘汰を用いる知的社会進化メカニズム ISEM が採用されなければならない。GA は ISEM の特殊な場合である。

参考文献

- [Axelrod 1997 年] Robert Axelrod: *The Complexity of Cooperation*, Princeton University Press, 1997 年. (邦訳: 寺野隆雄監訳『対立と協調の科学…エージェント・ベース・モデルによる複雑系の解明…』ダイヤモンド社 2003 年)
- [Garzon 1995 年] Max Garzon: *Models of massive parallelism: analysis of cellular automata and neural networks*, Springer-Verlag, 1995 年.
- [Holland 1992 年] John H. Holland: *Adaptation in Natural and Artificial Systems*, MIT Press, 1992 年. (邦訳: 嘉数侑昇監訳『遺伝アルゴリズムの理論…自然・人工システムにおける適応…』森北出版 1999 年)
- [Iba 2005 年] 伊庭斉志: 『進化論的計算手法』, オーム社, 2005 年.
- [Mado 2005 年] 間遠伸一郎: エージェントと非ノイマン型プログラム, 2005 年度人工知能学会全国大会(第 19 回)論文集, 3E1-06, JSAI, 2005 年.
- [Minsky 1985 年] Marvin Minsky: *The Society of Mind*, Simon & Schuster, 1985 年.
- [Mitchell 1996 年] Melanie Mitchell: *An Introduction to Genetic Algorithms*, MIT Press, 1996 年. (邦訳: 伊庭斉志監訳『遺伝的アルゴリズムの方法』東京電機大学出版局 1997 年)