



デジタル生命の進化

和田健之介 著

by Kennosuke Wada



岩波書店



岩波科学ライブラリー

11

IWANAMI

LIBRARY OF SCIENCE

デジタル生命の進化

デジタル生命の進化

和田健之介 著

by Kennosuke Wada

岩波書店

もくじ

1	ネオ・ダーウィニズムを考える……………	3
2	細菌の生き残り戦略……………	14
3	突然変異とポーカーゲーム……………	22
4	進化における交叉のはたらき……………	29
5	遺伝子の翻訳効率と方言……………	35
6	進化システムの夜明け……………	45
7	遺伝的アルゴリズムとはなにか……………	53
8	遺伝的アルゴリズムの長所と短所……………	64
9	遺伝子複製の不均衡仮説……………	72
10	適応能力イコール適応度ではない……………	80

11	遺伝子重複と中立突然変異……………	93
12	紙とエンピツでできた機械……………	99
13	言語を識別する機械……………	107
14	進化する機械を設計する……………	112
15	エピソード……………	128

あとがき

デジタル生命の進化

1

ネオ・ダーウィニズムを考える

対象となるスケールによって異なる進化論

最近はやりの遺伝的アルゴリズムは生物の進化機構を模倣した最適化アルゴリズムですが、そのもととなる進化論には、ネオ・ダーウィニズムや中立進化説、ウイルス進化説、断続平衡進化説、隔離説など諸説があり、現在でも多くの諸派に分かれて互いにしのぎをけずっています。(遺伝的アルゴリズムについては7章で説明します。)

生物の現象のなかでも、特に進化の問題は、歴史的にも科学の領域にとどまらず、宗教、ひいては人間の存在意義そのものにもかかわる重大な問題であるために、昔から多くの人々の関心を引いてきました。また、進化論者は優秀な論客が多く、しばしば排他的であるため熾烈な論争が繰り広げられてきました。

科学は芸術と同じように、同じ対象を扱っても、視点や感性、技法によってまったく

異なつて表現されます。そして、それが従来から存在した表現法の模倣である場合には、価値が低いと見なされるため、それを打ち破るために斬新な展開やまったく新しいパラダイムが模索されます。そのため、現われては消える理論のなから生き残ったものは、それぞれ特徴的な表現を持ち、その個性を主張します。

進化機構も他の生物現象と同じく、対象とする生物や環境、また対象とするスケールによってさまざまな見え方をします。ウイルスを中心に生物界をながめると、高等動物を中心に生物界をながめるとでは当然見え方は異なってきます。また、対象とする単位が遺伝子である場合と、個体である場合、さらに種である場合とでは、進化する実体が異なり、進化という意味もそれぞれの場合で違ってきます。

つまり、集団の中のある特定の遺伝子座に注目して、それぞれの対立遺伝子の頻度の世代変化に興味があるのか、それとも個体の特定の形質の適応的变化に興味があるのか、またある種の行動様式の世代変化に興味があるのかによって、進化という意味がそれぞれ異なり、扱う時間スケールも数世代から数万年まで、まったく異なった時間スケールが用いられます。(ここで用いたいくつかの専門用語については、次章でおいおい説明していきますので、気にせず読み飛ばしてください。)

このように、それぞれの進化論は得意とする固有の領域と材料を持ちます。しかし、

進化機構も含めて、ほとんどの生物学的現象はその機構の複雑さと多様性のために、必ずと言ってよいほど例外を含みます。

このため、単純明解な一般論でさまざまな現象を統一的に解釈するのは、ほとんど不可能に近いのです。したがって、対象が生物である場合には、ある生物の特定の時間スケールでのある機構については、実に矛盾なく説明できても、それを一般的に拡張しようとするとう無理が生じます。

科学の理論は、なるべくシンプルで汎用性が高いほど美しいとされてきたため、進化論もその伝統にそって、きわめて単純なロジックでさまざまな現象を説明しようとしてきました。このために、それぞれ異なった視点から生まれたはずのそれぞれの進化論が、科学的理論としての正当性を主張するために、自身の扱える領域を無闇に拡張しようとするとう無意味な衝突が生じてしまいます。

言葉の意味すら共通の概念を持たないのに、一般論で争っても何も得ることはありません。少なくとも、まず最初に「進化」と言ったときの、対象単位は何か、どのような領域を扱っているのかを明確にしなければ、科学的な議論が成り立ちません。ここでは、それぞれの進化論に立ち入った議論はせずに、人工的な進化システムを設計する際に必要となる概念にしばって考えることにします。まず、進化システムの基礎をなす遺伝的

アルゴリズムがベースにしているネオ・ダーウィニズムについて考えてみましょう。

ネオ・ダーウィニズムを考える

よく知られているダーウィンの進化論は、ラマルクの提唱した獲得形質の遺伝の概念を含んでいました。つまり、ある個体が環境に適応する過程で獲得した形質が、次世代の子孫に遺伝するものと考えていました。

しかし、その後、ピカードやヴァイスマンによって、次世代形成のための遺伝情報は生殖細胞にあるのであって、それ以外の細胞、つまり体細胞に生じた変異や適応的变化は次世代には遺伝しないという生殖質の独立説が確立されました。わかりやすく言えば、より環境に適応するために学習したり筋肉を強化したりしても、その効果は一世代限りで、精子や卵子にまでは遺伝情報として伝わらないということです。

そこで、ヴァイスマンはダーウィンの進化論からラマルキズム的要因を完全に取り除き、自然淘汰の万能性を強調したネオ・ダーウィニズムを確立しました。その後、遺伝機構の詳細が明らかになって、生殖質の独立性と連続性は実験的にも確認されています。今日話題になっている遺伝子治療も、治療の効果を一世代限りにするために、対象を体細胞のみに限定し、生殖細胞を操作することは禁じられています。

ネオ・ダーウィニズムの思想は遺伝学の進歩とともに形を変えてはきましたが、遺伝的変異と自然淘汰を中心に据えて、現代的な遺伝学的知見を取り入れた総合学説として、今日でも主流となっています。

では、その中核をなす自然淘汰とはどのような概念でしょうか。進化論で、この淘汰の定義や作用機構について述べるにあたっては、まことに慎重にならざるをえない問題なのですが、ここでは、なるべくダーウィンの主張した説にそって平易に説明します。

まず、生物は基本的に多産性を原則とする、と仮定します。つまり、環境条件さえ許せば、子供は産めるだけ産むものとします。したがって、環境資源が限られていれば、当然、個体間に競争が生じます。（ただし、生息場所は同じでも食物が異なれば、競争が生じない場合があります。また、競争を避けるために、多少栄養価はさがっても捕食効率の良い食物に切り替えたり、生息場所を変えたりすることがあります。このほか、異種の生物が生理的に緊密な関係を保つ、相利共生や片利共生、寄生などが生物進化に重要な働きをしているものと考えられています。しかし、ここではそれらの機構は考えず、一つの種が共通の資源をめぐって競争している場合を考えます。）

子孫を残すときに遺伝子に変異が起きて、各変異個体の間で環境適応能力に差が生じたとします。そして、よりよく環境に適応した個体が生存できて、より多くの子孫を残

し、その変異を次世代に残す確率が高くなる、と考えます。ここで、繁殖という意味での成功率、つまりある個体が次世代に生き延びて生殖年齢に達する子供の期待値をダーウィン適応度といいます。したがって、この適応度という概念には、どれくらいの割合で生き残れるかという生存率と、どれくらいの子供を産めるかという妊性（出産率）の二つの要素が含まれています。また、集団生物学という増殖率と異なり、この適応度という概念は、個体ではなく遺伝子型に対して定義されます。

ここで簡単にゲノムとか、遺伝子型や表現型という言葉の説明しておきましょう。私たちヒトの細胞には、二二対の常染色体と一对の性染色体、合計四六本の染色体が細胞核のなかに入っています。そして、すべての遺伝子を含む染色体を一セットにしたものをゲノムといいます。

したがって、ヒトのような二倍体生物の場合、体細胞には二組のゲノムが、精子や卵子には一組のゲノムが含まれていることになります。染色体のなかには消化酵素や色素など数多くの遺伝子が含まれています。それぞれの遺伝子は何番目の染色体のどの場所というように、アドレスが決まっています。これを遺伝子座と呼びます。ヒトのような二倍体生物の場合は、各染色体が二対ずつあるので、それぞれの染色体の相同な遺伝子座に母親由来の遺伝子と父親由来の遺伝子とが乗っています。

一つの遺伝子座に着目したときに、その遺伝子が異なった複数の遺伝子の形態を持つときに、それぞれを対立遺伝子と呼びます。二倍体生物では、それぞれの染色体の相同な遺伝子座位を占める対立遺伝子の組合せによって、さまざまな遺伝学的構成があり、この組合せを遺伝子型と呼びます。ある遺伝子座について、二つの対立遺伝子と同じ遺伝子であれば、その遺伝子座についてホモ接合であるといい、異なればヘテロ接合であるといえます。

そして、ある形質が一定の環境条件のもとで特定の遺伝子型によって主に規定されているとき、その形質をその遺伝子型の表現型といいます。ただし、一つの遺伝子が必ずしも一つの形質に対応しているわけではなく、複数の遺伝子が一つの形質に関与することもあるれば、逆に一つの遺伝子が複数の形質に関係していることもあります。また、一般には遺伝子型の均一な集団から必ずしも同一の表現型が得られるとは限らず、表現型は遺伝子型の効果の浸透の度合いや環境との相互作用に依存して決まります。

簡単な例として、よく知られているABO式の血液型を考えてみましょう。対立遺伝子には、a、b、oの三つがあります。したがって、体細胞での遺伝子型は、aa、ao、bb、bo、ab、ooの六通りがあります。また、赤血球の膜を構成する糖タンパク質の多糖体の末端構造の違いによって血液の凝集の仕方が異なり、よく知られているように、表

現型にはA型、B型、AB型、O型の四つがあります。遺伝子型と表現型の対応は、aaとaoがA型、bbとboがB型、abがAB型、ooがO型、となります。

優性とは、対立形質を持つ遺伝子座がヘテロ接合体であるときに、ある対立遺伝子はその形質の表現に強い効果を持って、他の対立遺伝子の効果を打ち消してしまう性質のことをいいます。逆に、打ち消される弱い遺伝子を劣性といいます。このことから、aはoに対して優性、逆にoはaに対して劣性であることがわかります。bとoについても同様です。また、aとbの対立遺伝子は共に活性を示すので共優性であるといえます。oはa、bのいずれに対しても劣性なので、ホモ接合のときだけO型になります。

ヒトのような二倍体生物の場合、二セットのゲノムを持つので、変異によって生存に必要な遺伝子に異常が生じても、その変異が劣性でヘテロ接合であれば、正常なもう片方の遺伝子が機能するので、その個体は生存できます。一倍体生物の場合は、その変異が直接表現型にあらわれるので、生存に不可欠な遺伝子が損傷すると致死的な結果を招きます。このため二倍体生物のほうが、遺伝子変異に対して強いシステムであるといえます。このことから、二倍体生物が誕生した理由の一つとして、ウイルスなどの侵入に対する防御機構としての役割が考えられています。また、より積極的な意味では、劣性遺伝子を隠れた遺伝情報として用いることができるので、集団中の遺伝的多様性を保持

し、環境変動に対しても適応性が高まることが考えられます。

ここで話を適応度に戻すと、適応度はそれぞれの遺伝子の遺伝子型に対して定められます。注目している遺伝子以外の遺伝子はすべて共通であると仮定して、その注目している遺伝子座の遺伝子型それぞれに対して、その表現型に属する個体の適応度を求めます。したがって、遺伝子型の違いが生殖能力や生存能力に違いをもたらせば、次世代に残せる子孫数に影響を与えて適応度が異なることになります。淘汰によって、適応能力の高い変異個体ほど、より多くの子孫を残し適応度が大きくなるので、集団中の平均適応度は増加します。しかし、このような適者生存の原則は自明なのでしょうか。

適者生存はトートロジー？

例として、鎌型赤血球貧血症を考えてみましょう。ヒトのヘモグロビンの β 鎖をコードしている遺伝子には、正常遺伝子 $Hb\beta^+$ と貧血を引き起こすようなアミノ酸置換の生じた遺伝子 $Hb\beta^s$ の二種類があります。

この $Hb\beta^s$ 遺伝子は正常遺伝子に比べて、六番目のアミノ酸がグルタミン酸からバリンに置換されています。 $Hb\beta^s/Hb\beta^s$ のホモ接合体は、低酸素圧下で赤血球が鎌型に変形し、溶血しやすくなるために激しい貧血症状を示します。 $Hb\beta^+/Hb\beta^+$ の個体は

正常ヘモグロビンを持ちますが、医療や生活条件の厳しいアフリカのある地域では、カを宿主とする寄生虫プラスモディウム・ファルシパルムによって、マラリアに感染しやすくなります。ところが $Hb\beta^s/Hb\beta^s$ のヘテロ接合体は、無症状または軽い貧血症を示しますが、寄生虫のついた赤血球は鎌型に変形し、血液循環によって除去されるので、マラリアには抵抗性を示します。

このようにヘテロ接合体のほうが、重篤な症状を示す両ホモ接合体より適応度が高い値を示すことがあります。ヘテロ接合体の適応度が、いずれのホモ接合体より高い場合を、超優性といえます。このため、超優性の対立遺伝子は集団中にヘテロ接合体を安定に保つことができるので、遺伝的多型を維持するための重要な機構と考えられています。では、このような超優性の遺伝子が複数あったときにはどうなるのでしょうか。

ナジラキが集団遺伝学を用いて明らかにしたことは、たとえば二つの遺伝子座が超優性で、各遺伝子の適応度への寄与が乗法的である場合には、最終的に平均適応度が極大値以外の所に落ち着いてしまうことがあるということです。したがって、複数の遺伝子が相互作用している場合には、適者生存の原則は思ったほど自明ではないことになりました。

また、このような膠着状態を脱するための駆動力の一つに突然変異があるわけですが、

そのように都合の良い変異が生じてくれるものなのでしょうか。これも集団遺伝学によって明らかになっていることですが、複数の遺伝子について、淘汰と突然変異を考慮したモデルを考えたときに、それぞれの遺伝子頻度が周期的に変動してしまう場合があることが知られています。つまり、ジャンケンのように各遺伝子が三すくみ状態となって、集団の遺伝子頻度が各遺伝子の間をぐるぐると回ってしまうのです。つまり、淘汰によって適応度の高い個体ほど子孫を残しやすいという自然淘汰の機構が、必ずしも最も適応度の高い遺伝子型に集団を収束させるわけではないことがわかります。

2

細菌の生き残り戦略

突然変異はわるもの？

ガンや多くの遺伝病が、遺伝子の変異によって引き起こされることから明らかなように、基本的には突然変異は個体にとっては有害なものです。このため生物は、DNAの複製時に生じたエラーを修復する機構を持っています。

一方、ダーウィンの進化論では、突然変異が少しずつ蓄積して環境に適応していく過程を進化と捉えています。しかし、淘汰に対して有利となるような変異が集団中にそんなに頻繁にあらわれるのか、という点に関しては疑問が残ります。

ヒトのように進化の過程がかなり進んでしまった生物は、環境に適応する過程で、きわめて複雑な代謝機構や調節機構を身につけてきました。それらの機構がベストとは言えないまでも、このような生物にとっては、変異することによって、その精緻な機構が

狂ってしまうか、淘汰に対して有利でも不利でもない中立な場合がほとんどではないかと考えられます。逆に、ある生物が進化初期の段階にあれば、変異によって生存に有利な機構を獲得する確率が少しは高いだろうと考えられます。

しかし、現存する生物でも、置かれた環境によっては、生存に有利な変異を次々と獲得して生き残っているものがあります。

突然変異を利用して生きのびる細菌

最近、社会問題にもなっている院内感染の元凶は、MRSAというブドウ球菌です。細菌感染が起きると、化学療法として抗生物質が投与されますが、しばらくすると薬剤耐性を持った菌が現われて効かなくなってしまうます。細菌は酵素を産生して、アセチル化、リン酸化、アデニル化、加水分解などさまざまな修飾を行なって、抗生物質を不活化してしまいます。

たとえば、ブドウ球菌に対してペニシリンを用いているうちに、ペニシリンを分解するペニシリナーゼという酵素を持つ菌が現われます。そこで、さらにこのペニシリナーゼによって分解されない抗生物質が開発されました。そして、この強力な抗生物質の一つであるメチシリンに対しても、耐性を獲得してしまったブドウ球菌がMRSAです。

次々とイタチごつこのように抗菌スペクトルの広い抗生物質が開発されて行きますが、そのたびにますます強力な細菌が現われてしまいます。細菌側からみれば、すさまじく厳しい生育環境にさらされていることになりますが、このように非常に強い淘汰圧のせいで、より細菌の適応能力を加速度的に増強させることになってしまいます。

このような短い期間での適応的变化を進化と呼ぶかどうかは、それぞれの進化論の立場によって異なります。とくに動植物の形態の機能的適応に重点を置く古典的な進化論の立場では、進化の不可逆性を主張することがあります。進化の過程で退化したり消失した器官が、その後復帰することはなく、仮に環境の変化によって元の器官が生存に必要になっても、別の器官が発達してその役目を代替するという意味です。

「進化は繰り返さない」というこの主張にも、もちろん例外はありますが、ある目的を達するための実現方法が無数にあるときには、何回かの試行に再現性がないのは当然と言えます。しかし、この主張をさらに推し進めて、可逆的な適応的变化は、いくら遺伝子に変化が生じていても進化とは呼ばない、という立場もあります。

細菌の多くの変異は一塩基の置換であり、このような変異型はもう一度置換して、もとの塩基に戻る復帰突然変異が生ずると、最初と同じ野生型にもどるので、この場合は可逆的な変化が起きていることになります。しかし、これを進化と呼ぶかどうか、とい

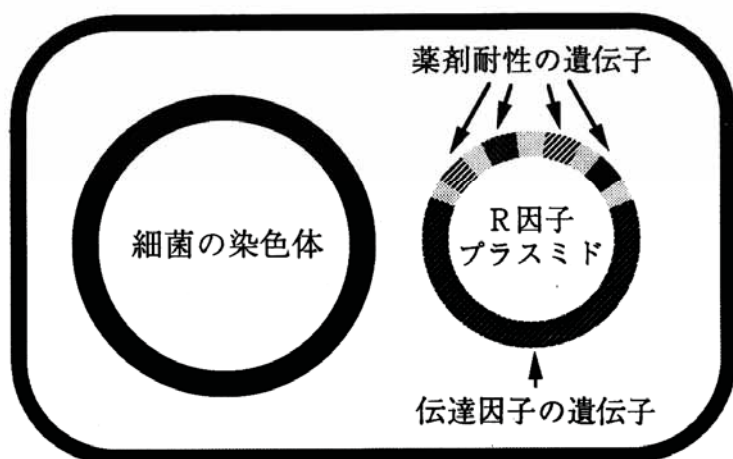


図1 細菌の薬剤耐性遺伝子

う不毛な議論はここでは避けておきましょう。どちらにしても、細菌の分裂サイクルの速さからすれば、その個体数は膨大であり、環境への適応能力の高さをみれば、進化の機構を考える上で十分に貴重な情報を与えてくれることだけは間違いありません。

多くの細菌は、自分自身の染色体の他に、プラスミドと呼ばれる遺伝因子を持っています。いくつかの異なる機能を持つプラスミドがありますが、それぞれのプラスミドによって細菌あたりのプラスミドの数は異なり、一個しかないものから五〇個ちかくあるものまであります。

プラスミドも染色体と同じように、複製によって子孫に受け継がれて行きます。抗生物質に対する耐性遺伝子を持つプラスミドをR因子プラスミドといいます。なかには、クロラムフェニコール、テトラサイクリン、サルファニルアミド、ストレプトマイシンなど複数の薬剤に対する耐性遺伝子を同時に持つR因子プラスミドもあり、これが細菌に多剤耐性を与えています(図1)。このよう

な多剤耐性が速やかに獲得される機構はどうなっているのでしょうか。

耐性を身につけた細菌は、分裂増殖することによって、染色体やプラスミドを複製して遺伝情報を伝達して行きますが、その複製時に生じるエラーや外的な突然変異誘導因子などによって変異が生じます。そして、新しい薬剤を細菌に加えると、たまたまその薬剤に対して耐性を持つ菌が、感受性を持つ菌を圧倒して増殖します。

しかし、細菌はこのような世代の時間軸にそった垂直方向の遺伝子伝達機構だけではなく、細菌どうしの水平方向の伝達機構も持ちます。このために猛烈な速度で多剤耐性が広まってしまうのです。細菌どうしが接合したときに、耐性遺伝子が含まれたプラスミドを持つ耐性細菌から、それを持たない感受性細菌にプラスミドがコピーされることによって、耐性を獲得した細菌が増えるのです。

そして、さらに驚いたことに、このプラスミドの上にある薬剤耐性遺伝子は、他の細菌の染色体やプラスミドの間を移動することができるのです。入り込む相手先の配列に好みはあるのですが、基本的には自分自身と相手先の配列の相同性とは無関係にあちこちと動き回り、文章のカット・アンド・ペーストのように元いたところから移動する場合と、コピー・アンド・ペーストのように元の配列をそのままにして新たに他の場所へ複写する場合があります。

このように動き回るDNAはトランスポゾンと呼ばれますが、自分自身の挿入を触媒するトランスポザアーゼやそれを抑制する酵素などをコードしています。これだけで動き回るトランスポゾンをとくに挿入配列と呼びます。R因子プラスミドに含まれるトランスポゾンには、その間に薬剤耐性遺伝子が乗っているのです。この機構のために、異なった薬剤耐性遺伝子が動き回っているうちに、一つのプラスミド上に集まって多剤耐性を獲得することが可能となっています。

また、トランスポゾンが遺伝子領域の間に挿入されたときは、その遺伝子が不活性化されることがあります。これとは逆に、トランスポゾンの中に含まれている自分の転写を促進するプロモーターが転位先の遺伝子を活性化することもあります。

このようにトランスポゾンが遺伝子発現を制御していることがあります。そして、同一DNA上に複写して転移する場合には、トランスポゾンが元と転位先とで重複します。この相同な二つの領域は結合しやすいので、この領域が結合すると2つのトランスポゾンの間のDNAがループ構造を作ります。そして、この結合部位での組換えが生じた場合、二つのトランスポゾンの向きが同じときにはループ領域の欠失、逆向きときはループ領域の逆位を引き起こします(図2)。

このループ領域に遺伝子が含まれていれば、遺伝子の欠失や逆位が生じます。また、

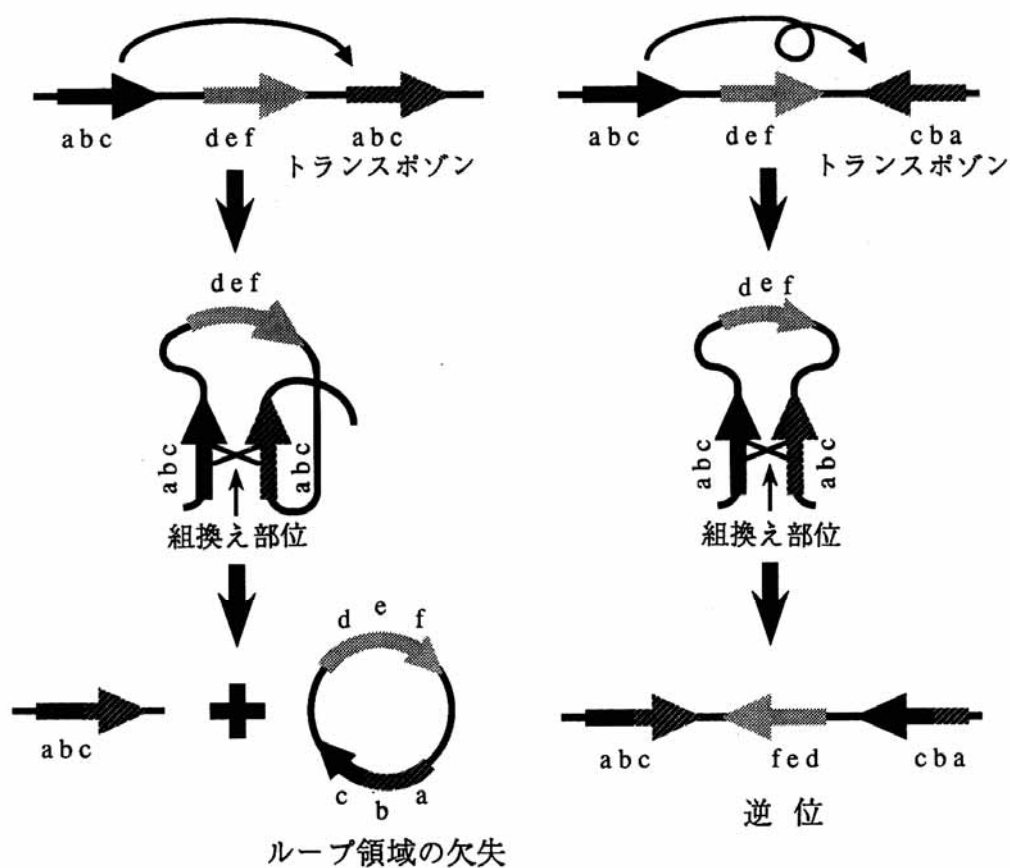


図2 トランスポゾンと組換え

後で説明する不等交叉の原因ともなります。

このように、トランスポゾンは細菌の遺伝子転位、遺伝子発現、遺伝子組換えなどで重要な機能をしているのですが、細菌だけにあるわけではありません。

トランスポゾンは最初、一九四〇年代にマクリントックによって、トウモロコシから発見されました。現在では、ほとんどすべての生物にあるものと考えられていて、進化の機構としても重要な要素と思われます。

この他にも、細菌を宿主とするファージというウイルスがあ

りますが、これが細菌の耐性遺伝子を媒介することもあります。したがって、ある病室で使われていた抗生物質に対して耐性を獲得した細菌と、別の病室で使われていた異なった抗生物質に対して耐性を獲得した細菌が、病人どうしや医療スタッフ、空調設備などを通じて接合したり、ファージによる感染などが起きると、それぞれの細菌がまだ出会ったことがないはずの薬剤耐性を身につけてしまう危険性があるのです。

細菌やウイルスの遺伝機構は両刃の剣のように、人類にとって脅威であると同時に、遺伝子工学によってうまくコントロールすれば測り知れない恩恵を蒙ることができます。また、人工的な進化システムを設計する際にも、さまざまな適応機構を示唆してくれま

3

突然変異とポーカーゲーム

ここで、ポーカーゲームを考えてみましょう。最初はカードを十分シャッフルしてから各プレイヤーに配られるので、それほど良い手がそろうことはありません。そこで、カードを交換することによって、より良い手がそろうことをめざします。しかし、最初のうちは交換によって良い手がそろう確率も高いのですが、何度か交換してかなり良い手がそろった段階では、さらに良い手が得られることを期待して、いちかばちかの交換をしても、大失敗してしまうことが多くなります。また、たいていのプレイヤーは、ある程度の手がそろってしまったら、不要なカードを交換するという安全策に切り替えるでしょう。たとえば四枚同じマークがそろったら、残りの一枚を交換し続けることによってフラッシュ(五枚とも同じ組)を狙うでしょう。このカードゲームは意外と、進化について多くのことを考えさせてくれます。

突然変異の不均一性

まず、ポーカーゲームでのカードの交換を突然変異にたとえて考えてみましょう。生物の突然変異は、まったくランダムに生ずるのでしょうか。ポーカーゲームではまったくランダムにカードを交換していたのでは、よほど幸運でないかぎり、いつまでたっても良い手はそろわないでしょう。先程のフラッシュを狙うように、すでにちゃんとそろったカードには手をつけず、そろっていないカードを交換するほうが得策でしょう。しかし、いつもこの戦略だけをとっていたのでは、より上位のロイヤル・フラッシュ(同じ組の10、J、Q、K、A)などを狙うことができません。このような戦略の切り替えは、周囲のプレイヤーの捨てたカードや表情などから判断するわけですが、その駆け引きの妙がこのゲームの醍醐味でもあるわけです。

生物でも、ただ一様に突然変異が生じているわけではありません。ベンザーはファージT4の π 変異体を使って、 π 領域内の約三五〇個の変異体を単離して、それぞれの突然変異可能部位をマップしました。その結果、ホットスポットと言って、他の部位に比べて、平均の二五倍近くも突然変異率の高い部位が存在することがわかりました。これと逆に極端に変異しない部位もあります。そして、ベンザーはこのような突然変異の不

均一性は、ある領域の特別な塩基配列に関係していると考えました。

また、細菌には、突然変異率を高める働きをするミューター遺伝子を持つものがあります。この遺伝子はDNAの誤りを修復する機能を低下させます。これらの機構が進化にどのように寄与しているのか、ということについてはまだ解明されてはいません。しかし、生存に不可欠な領域の変異率が高ければ生存率は下がるでしょうし、ウイルスの外皮の糖タンパク質のように宿主の免疫系によって攻撃されるような場合は、そのタンパク質をコードしている遺伝子の変異率は高いほうが、ウイルスにとっては有利に働くでしょう。また、薬剤耐性菌のように厳しい環境にさらされている場合には、環境に応じた突然変異率の制御が行なわれている可能性があります。

ここで、突然変異率と言ったときに少し微妙な問題が生じます。つまり、DNAが複製や修復されるときに突然変異と、サンプル個体の遺伝子配列に観察される突然変異は異なるということです。もし、卵子や精子のなかで、発生初期にかかわる遺伝子に変異して発生過程が進まなくなる場合には、生まれてこないのです、そのような変異は通常は観察されません。また、適応度を下げるような変異を持った個体は少なくなるので、仮にそのような変異が起きやすいとしても、観察される頻度は少なくなります。

したがって、仮にDNAの全領域にわたって突然変異率が一樣だとしても、重要な機

能を持つ遺伝子の変異率はきわめて小さいものと観察されます。このように自然な環境のなかから得られた遺伝子の変異率には、多かれ少なかれ淘汰が影響しています。したがって、ポーカーゲームのときのように、大事な手は取っておいて、残りだけを変えていくための積極的な機構が生物に備わっているのかどうかは、この淘汰の影響があるために観測が難しくなります。

突然変異率は、一般に体細胞ではなく生殖細胞に生ずる突然変異率のことをいいます。そして、この突然変異率は生物種や遺伝子によって異なりますが、たとえば、T4ファージでは一回の複製につき、 10^7 塩基当り一回の割合で生ずると言われています。また、ショウジョウバエではさらに低く、 10^{10} 塩基当り一回の割合だと言われています。驚くべき正確さですが、一塩基当りの割合が小さくても遺伝子長が長ければ、その遺伝子の間に変異が入る確率は高くなります。

ヒトのような真核生物では、細胞のなかで遺伝子が核によって区分されています。そして、真核生物の多くの遺伝子は、いくつかに分断されていて、タンパク質をコードしているエキソンの間に、非コード領域のイントロンがはさまった形態をとっています。

遺伝子が発現するときには、まずRNAポリメラーゼによってDNAからmRNA前駆体が転写されます。そして、核内でいくつかの修飾を受けて、このmRNAからイン

トロンが除去されて、エキソンどうしが連結されます。この過程をスプライシングとよびます。このようにして処理を施された成熟 mRNA が細胞核から外へ出て行きます。リボソームはこの mRNA の情報に従って、対応した tRNA を結合します。リボソームが mRNA 上を移動することによって、tRNA が運んできたアミノ酸を次々とペプチド結合して行きます。終止コドン(停止シグナル)に出会うと、この作業は完了し、タンパク質ができあがります。スプライシングは生物進化の上で重大な役割を持っているのですが、逆に精密さが要求されるために、有害な変異が生ずる原因ともなります。

ヒトの伴性劣性遺伝病の一つである、デュシェンヌ型進行性筋ジストロフィーは、その関連遺伝子が X 染色体の $Xp21$ という座位にあります。そして、その遺伝子は驚くことに、分子量四二・七万で三六八五個のアミノ酸からなる超巨大なジストロフィンというタンパク質をコードしています。そして、このタンパク質をコードしている DNA 領域は長さが二〇〇万塩基対以上もあり、ジストロフィンには七五個以上のエキソンに分断されています。したがって、ものすごく長いイントロンが間にいくつも挟まっていることになります。このジストロフィンがちゃんと作られるためには、このイントロンが正確にスプライシングされなければなりません。わずかな変異が生じて、スプライシング部位がずれたりすることによって、読み枠がずれ込んで早めに終止コドンが来るため、

それ以降のエキソンが読まれなくなることがあります。また、スプライシング部位を間違えて途中のエキソンが脱落する場合があります。このため、この遺伝病は高頻度で発生し、一世代当り、 $4 \sim 9 \times 10^{-5}$ もの確率で発症します。

DNAは遺伝情報を担う物質として安定なのか

そもそも、DNAは遺伝情報を担う物質として十分安定なのでしょう。実は予想外に、DNAは変異しやすい物質なのです。

DNAはデオキシリボースを糖成分とする核酸からできていて、その塩基成分はアデニン(A)、グアニン(G)、シトシン(C)、チミン(T)からなります。ワトソンとクリックによって、二本鎖DNAは二重らせん構造をしていて、それぞれのDNA鎖の塩基は相補的な対をなしていることが明らかにされました。AはTと、GはCとそれぞれ対をなします。RNAはリボースを糖成分とする核酸からできていて、塩基成分はチミンの代わりにウラシル(U)が使われます。驚いたことに、DNAの塩基であるシトシンは放っておくと脱アミノ化してウラシルとなります。ウラシルは複製のときにチミンのように振る舞ってアデニンと対をつくるので、GC塩基対がAT塩基対に変わってしまいます。このような変異は絶えず修復系によって校正されていますが、修復酵素に変異が生

じると複製エラーが蓄積して致死的な結果を招いてしまいます。

また、DNAの塩基配列のなかで、チミンが二つ続いた部位は紫外線を受けると、チミンどうしが結合してダイマー(二量体)ができやすくなります。このダイマーができるとDNAの複製の障害となります。このため、紫外線を多く浴びる皮膚細胞では、絶えず修復が行なわれています。この修復のための酵素が遺伝的に欠損していると色素性乾皮症になり、皮膚ガンが高頻度で発生します。

生物の遺伝子複製は半保存的であると言われます。オリジナルの遺伝情報を複製したものを子孫に伝達しているためです。生物が種として安定に存続するためには、遺伝情報を間違いなく忠実に伝えていくことが必要です。しかし、エラーなしの完全な複製というものとは機構的にも不可能であるし、また突然変異がまったくなければ進化もありえません。では、この拮抗関係をちょうどよくバランスして、集団の平均適応度を極大にするような最適な突然変異率というものは存在するのでしょうか。きわめて単純な系については、集団遺伝学を用いて計算することはできますが、遺伝子が複数あってそれらが複雑に相互作用しているような現実の系については解析することができません。このような突然変異率は、人工的な進化システムを設計するときには必須のパラメータ情報ですが、これについては、後でもう一度考えることにしましょう。

4

進化における交叉のはたらき

交叉という現象

生物では、相同な染色体の間に部分交換が生じる現象があります。これを交叉といいます。二つの遺伝子が同一染色体上にあるときは、二つの遺伝子間の相対距離が長くなるほど、その間で交叉の生じる確率が高くなります。つまり、相対距離が十分長いときや、異なった染色体上にあるときには、それぞれの遺伝子は独立に遺伝します。

逆に、二つの遺伝子が同一の染色体上にあつて、その相対距離が短いときには、その間に交叉の入る確率が小さくなるので、独立の法則から期待されるよりも高い頻度で結びついて遺伝します。これを連鎖といいます。

二倍体生物の生殖細胞の場合、交叉によって、父親由来の染色体と母親由来の染色体とを含む生殖細胞が減数分裂して一倍体の配偶子(卵子または精子)を作るとき、キアズ

マという部位を形成して、そこで二つの染色体が乗り換わることがあります。これによって父方と母方の遺伝子が組み合わさり、新たな遺伝子組成が生まれます。

遺伝子には神経成長因子や血液凝固因子などさまざまな機能を持ったものがあります。そのうちの多くは、生存に不可欠の汎用の遺伝子なので、個体による違いはほとんどなく、共通の塩基配列でできています。また、次の章で述べるヒストンのように必須のタンパク質をコードしている遺伝子は、種による違いもきわめて少ないのです。

そこで、残りの遺伝子、たとえば臓器移植のときに関連する主要組織適合性抗原遺伝子複合体などが個体によって異なるわけです。そして、これらの個性を主張するいくつかの遺伝子は、交叉によって、新しい組合せをつくるため、個体としての多様性が爆発的に増加することになるのです。

しかし、この交叉についても突然変異と同様の疑問が生じます。最適の交叉率はあるのでしょうか。いたるところで等しく交叉が起こりうるのでしょうか。

超遺伝子のかたまりになって遺伝する

この問いにも、生物学は完全な回答を持ち合わせてはいませんが、重要なヒントを与えてくれる例があります。シヨウジョウバエの唾腺の染色体では、複数の遺伝子を含む

4 進化における交叉のはたらき

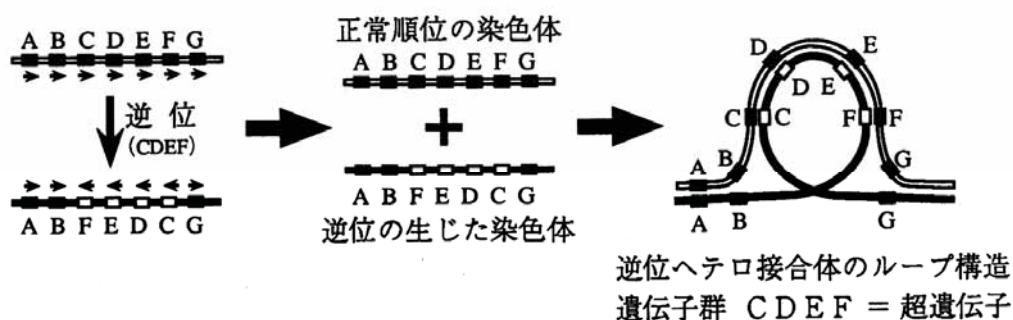


図3 超遺伝子

領域が、染色体のなかで向きを反転してしまうことがあります。このような領域を逆位といいます(図3)。

二本の相同染色体がヘテロ接合、つまり正常順位の染色体と逆位の染色体とが対合したときには、その領域が環状構造を形成します。そして、このループのなかに含まれた遺伝子群は、染色体の乗り換えが抑制されるために、交叉による組換えが生じにくくなります。したがって、これら一連の遺伝子群はあたかも大きな一つの遺伝子であるかのように遺伝します。これを超遺伝子と呼びます。

複数の遺伝子が、関連して一つの機能を実現しているような場合は、一度最適な遺伝子のセットがそろったら、その後はそれを一塊りの遺伝子として伝えたほうが、その機能を安定に遺伝させることができます。そのような超遺伝子を持つように変異した個体の適応度が増加するならば、その超遺伝子が集団中に固定する確率が高まります。

また、交叉が生じるときに形成されるキアズマは、どこにでも

できるわけではないようです。キアズマはGC%(四つの塩基のうち、GかCが使われている割合)が高い領域で形成される確率が高くなっています。

交叉は偶然のアクシデントではない

ヒトのゲノムは三〇億塩基対からなりますが、そのうちタンパク質をコードしている領域は約二%にすぎません。そのほかの領域は、非コード領域と呼ばれていて、その機能はよくわかってはいませんが、遺伝子発現や染色体のパッケージングに関与しているものと考えられています。この約二%の領域のなかに約一〇万個の遺伝子が含まれています。したがって、遺伝子は染色体のなかにポツリと浮かんだ島のように存在していて、その間に長い非コード領域があるわけです。そして、交叉はこの遺伝子のなかで、かつ相同性の高い領域で生じていることが、遺伝学によって明らかになりました。

遺伝子領域のなかで交叉が生じているということは、相同染色体が正確な位置で対合しているということです。なぜなら、組換え位置が一塩基でもずれたら、遺伝暗号であるコドンの読み枠がずれてしまい、その位置以降はまったく異なったアミノ酸に翻訳されてしまうからです。つまり、交叉とは相同的組換えのことであり、この組換えはDNAの切断もしくはギャップによって生じたDNAの一本鎖領域から始まります。

そして、この組換え機構は大腸菌で特に詳しく研究され、RecA, RecB, RecC, RecD 遺伝子が重要であることがわかっています。細菌が接合して雄の菌から導入されつつある DNA の末端に、RecBCD 酵素複合体が結合し、DNA 上を移動します。RecBCD 酵素複合体は ATP エネルギーを使って、二重らせんの DNA をほどこいては巻き直して、移動して行きます。ほどこく速度のほうが巻き直す速度より速いために、一本鎖にほどこけた部分が長くなり、DNA の両側に耳のような形のループを作って移動します。そして、この酵素が *Cis* と名付けられた八塩基からなる配列にさしかかると、ほどこけた一本鎖が切断されます。

大腸菌の染色体上には約一〇〇〇〇個の *Cis* 配列があります。この切断された一本鎖に RecA タンパク質がたくさん結合します。そして RecA タンパク質が結合した一本鎖 DNA が、もう一方の二本鎖 DNA と結合した後、DNA 上を走査して、相補的塩基配列部位が検出されると、そこで対合します。こうして別々の染色体由来の一本鎖 DNA とうしのハイブリッド DNA ができて組換えが始まります。

このように、交叉は決して偶然のアクシデントではなく、多くの酵素が関連した複雑で精密な機構なのです。そして、遺伝子組成の組換えや DNA 欠損部位の修復という、積極的な意味を持ちます。

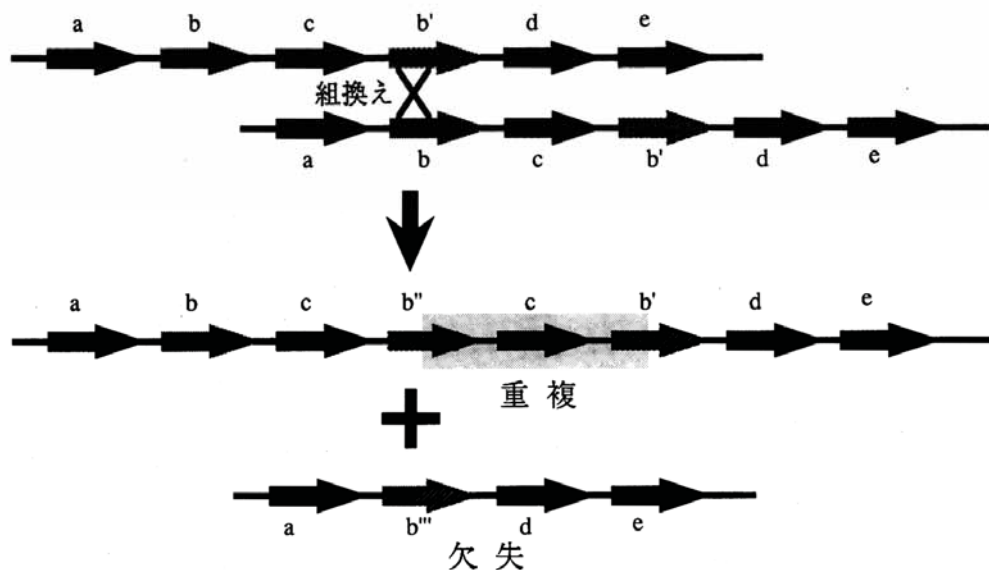


図4 不等交叉による遺伝子の重複と欠失

しかし、ときには交叉が生じる位置が相同な部位でないことがあります。染色体上にきわめて相同性の高い塩基配列を持った領域が二つ以上あると、本来の相同領域とは異なった座位にある偽の相同領域と対合することがあります。これを不等交叉と呼びますが、このために片方の染色体には遺伝子が重複して入り、他方には欠失が生じます(図4)。

また、すでにトランスポゾンのところで説明しましたが、同一染色体上の重複領域によって、遺伝子の欠失や逆位が生じることがあります。そして、このような遺伝子の重複や欠失が、生物の進化の過程で重要な役割を果たしたものと考えられています。このことも、人工的な進化システムを設計する際に、きわめて重要な機構となるので、あらためて説明することにします。

5 遺伝子の翻訳効率と方言

遺伝暗号コドン

遺伝情報を担う遺伝子の物質形態は生物によって異なり、二本鎖のDNAだったり、一本鎖のRNAだったりします。前にも述べたように、ヒトの場合は二本鎖のDNAです。DNAはデオキシリボースを糖成分とする核酸からできていて、その塩基成分はアデニン(A)、グアニン(G)、シトシン(C)、チミン(T)からなります。そして三個の塩基配列が単位となって、各アミノ酸に対応しています。この遺伝暗号の単位をコドンといいます。

したがって、六四通りのコドンがありますが、タンパク質に使われるアミノ酸は二〇種類しかないので、複数のコドンが同じアミノ酸をコードしているものがあります。たとえば、コドンの一番目と二番目の塩基がGCのときは、三番目が四塩基のうちいずれ

でも、アラニンというアミノ酸をコードしています。

そして、タンパク質の構造をコードしている構造遺伝子は、開始コドンに始まり、コドンの読み枠にしたがって、ある長さのアミノ酸の列をコードして、終止コドンで終わっています。このタンパク質が細胞を構築する要素となったり、酵素として働いたりします。

したがって、たとえばDNAが変異することによって、消化酵素の活性部位に変化をもたらすような場合には、その変異によって最終的に必須栄養素が摂取できなくなったリして適応度に影響を与えます。

ただ、DNAがタンパク質の構造をコードしているだけなのであれば、たとえば先ほど説明したように、タンパク質のなかにアミノ酸としてアラニンが含まれている場合、その遺伝暗号であるコドンの三番目の塩基がどう変異しても、アラニンというアミノ酸であり、適応度には何ら影響はしないはずで

分子進化の考え方

もし、変異が淘汰に対して有利でも不利でもないならば、その変異を中立突然変異と呼びます。そこで、ある長さの領域が中立突然変異を行なうと考えられるときには、進

化の初期に分岐した種どうしほど変異が蓄積して、その領域の塩基配列が異なっていることになります。たとえば、ヘモグロビンの α 鎖のアミノ酸配列をヒトとウマで比較すると、全部で一四一個のアミノ酸のうち一八個のアミノ酸座位で異なっています。一方、古生物学によるとヒトとウマはおよそ八〇〇〇万年前に分岐したものと推定されています。したがって、その間、この領域がずっと中立で置換速度が一定だと仮定すれば、およそ一三億年に一個の割合でアミノ酸が置換されてきたことになります。

この他にも、多くのタンパク質が多くの種の間で比較されましたが、その結果、これらのタンパク質のアミノ酸の置換速度はほぼ一定であるということがわかりました。ただし、置換速度はタンパク質によって異なります。あるタンパク質の置換速度が種によってあまり変わらず、ほぼ一定だということです。

機能的に重要な分子ほど、変異によって、その機能が破壊された場合に適応度が減少するので、そのようなタンパク質の置換速度は小さくなります。

このように遺伝子の分子構造の比較から生物の進化を眺めようというのが、分子進化の研究です。その計測手法を分子時計と言います。

この分子時計の一定性を逆に利用すると、あらかじめ置換速度を求めておけば、アミノ酸配列を比較することにより、異なった種どうしの分岐年代を推定することができま

す。分子進化では、ダーウィンの思い描いた進化のように、環境適応による形態や機能の進化を扱ってはいません。最初からミクロレベルの変異に限定して進化を考えています。

DNAはタンパク質をコードしているだけではない

しかし、生物の場合は遺伝情報がDNA（もしくはRNA）によってコードされているので、タンパク質のコード情報の他に物理化学的な制約が伴います。ウーラによって、睡眠病の病原虫であるトリパノゾーマのキネトプラストDNAのゲル中の移動速度が調べられました。キネトプラストとは、トリパノゾーマのなかにある特殊化したミトコンドリアのことで、自己増殖性を持ちます。測定の結果、キネトプラストDNAは異常に速くゲル中を移動することがわかりました。つまり、DNAが折りたたまれたりしてコンパクトになっているからだと推測されます。さらに、その塩基配列は興味深いことに、二重らせんのほぼ一巻に相当する約一〇塩基²⁰は、CAAAAATもしくはCAAAAATという配列が周期的に入っていました。

このようにDNAは、一般的にAやTが豊富なときは折れ曲がりやすく、GやCが豊富なときは直線的になる性質を持っています。このため塩基の並び方によってDNAの

幾何学的形状が異なります。

ヒトのような真核生物では、細胞のなかで遺伝子が細胞核によって区分されていて、DNAはヒストンというビーズのようなタンパク質の複合体に巻き付いています。DNAは一四六塩基対でヒストンを約二回転してヌクレオソームという反復単位を作っています。そしてこのヌクレオソームどうしをつないでいるDNAをリンカーDNAとよび、全体として数珠のような構造をしています。これがさらに階層的に折りたたまったものをクロマチンといいます。

この構造によって、三〇億塩基対、六センチメートルもあるDNAが、わずか一四〇マイクロメートルにパッキングされて細胞核の中に収まっています。

したがって、DNAはこのヒストンのカーブにそう構造をしていたほうが無理なく巻き付くことができます。仮にDNAがくねくねとヒストンとは異なった曲がり方をしていても、ヒストンは強い結合力でDNAを巻き付けてしまうので、歪みが生じてしまいます。したがって、その部位の歪みエネルギーが局所的に高くなり構造的に不安定になります。このために、この部位が他の領域に比べて複製のときに変異しやすくなることが考えられます。このような場合は、この部位の突然変異率は周囲の塩基の並びに影響を受けていることになります。

また、約一〇万個もある遺伝子のなかには、消化酵素のように絶えず頻繁に読み取られる遺伝子や、発生の初期にだけ読み取られる遺伝子など、利用頻度に差があります。一〇万冊の図書を図書館のように、ただアルファベット順に並べて利用することを考えましょう。このとき、利用したすべての本を手元に置きっぱなしにしていたのでは、混乱してかえって効率が悪くなるので、必要な本を机に持ってきて利用した後、もとの位置に返却するという作業を考えます。頻繁に利用する辞書などは、手元に置いておきカバーなどは外しておいた方が、効率が良くなります。

染色体でも、頻繁に利用される遺伝子と、一時的にしか利用されない遺伝子とに、何らかの違いを設けて読みだし効率を上げる工夫はあるのでしょうか。

染色体は、トリプシン・ギムザ染色などの染色法で染色すると、濃く染まる領域と、あまり染まらない領域に分かれ、濃淡の縞模様を作ります。この染色法で染まる濃い縞をGバンド、逆のバンドをRバンドと呼びます。この一つの縞の大きさにはギガベース¹⁰(塩基対)もある巨大な構造です。そして、池村と和田によってわかったことは、このギガベースもある巨大構造が、ミクロレベルのGC%と驚くほど対応がつくことです。

DNAの遺伝子データベースを用いてGC%の高いところと低いところとで塗り分けた模様と、染色によってできた縞模様とがきわめてよく一致するのです。つまり、Gバ

ンドは全般的にA T %が高く、一方、R バンドにはG C %の高い領域が多く含まれています。

前にも述べましたが、D N AはG C含量が高いと立体幾何学的な直線性が増すということを説明しました。先ほどの図書の利用効率とこのことから類推すると、非常に頻繁に利用される遺伝子のG C含量が高ければ、直線性が増して読みだし効率も高くなることが考えられます。逆に、一時期にしか使われない特殊な遺伝子のA T含量が高ければ、D N Aが折れ曲がりやすくなり、全体としてのパッキング効率が高くなります。

しかし、非コード領域だけではなく、コード領域にもこのG C %のマクロ構造は存在しています。遺伝子はタンパク質としてのコードを行なわなければなりません。そこで、前に説明したコードの多重暗号のことを思い出してください。

多くのコードンの三番目の塩基には冗長性がありました。したがって、タンパク質の構造を変えずにD N AのG C %が変わるためには、冗長なコードンの三番目の文字が変異すれば可能なことがわかります。そこで、遺伝子のコードンの三番目の塩基のみについてG C %を見てみると、やはりバンド構造ときれいな対応がつくことがわかりました。

驚くことには、高いものでは九七%というG C %を示し、低いものでは二七%のG C %を示すことがわかりました。

このように、DNAにはタンパク質としての情報の他に、パッキング効率や読み出し効率が遺伝情報として組み込まれている可能性が大きいわけです。

また、すでに交叉の説明のところで述べましたが、正確な組換えが起きるためには、対合する染色分体間にきわめて高い相同性が必要です。このような高い相同性は、保存性の高い汎用の必須遺伝子によく見られます。そして、汎用の必須遺伝子は利用頻度が高いために、コンパクトであるよりは直線性が高いほうが読み出し効率がよくなります。このような直線性の高い領域のGC%が高いということは、すでに述べました。したがって、これらのことをまとめると、交叉が生ずるときにできるキアズマという構造は、GC%の高い領域にできやすいということが推定されます。そして実際に、キアズマの密度とコドンの三番目の塩基のGC%との相関を調べてみると、きわめて高い相関を持つことが示されました。

遺伝暗号の方言はどのようにして生まれたのか

和田と池村らによって、ある生物種において、あるアミノ酸をコードするのに、どのコドンが好んで使用されているか、というコドン使用頻度を調べるために、最新の遺伝子データベースを用いて、知られているすべての遺伝子についての統計データが求めら

れています。

これによると、冗長なコドンの三番目の塩基の使用頻度が、生物種によってそれぞれ特徴的に異なっていることがわかります。これは、あたかも自然言語の方言とよく似ていて、犯罪捜査のときに、しゃべった言葉から、出身地が推定できるように、ある遺伝子の遺伝暗号の方言から種を推定することができます。

これをさらに推し進めて、方言の類似の度合いから、種の類似度を求めることができます。分子時計を用いた分子進化の系統樹と同じように、この類似度を尺度として、方言の類似の度合いからみた進化系統樹を書くことができます。この系統樹と古生物学の進化系統樹とがよく似ていることから、冗長なコドンの三番目の塩基には、進化の過程で、単なる偶然ではなく、何らかの選択圧が働いたことが推測されます。

リボソームにアミノ酸を運ぶ役目をする tRNA には、RNA と相補的なアンチコドンがコードされていて、ここで RNA と結合します。この tRNA の含量比は生物種によって異なっています。このことから、強い発現をする遺伝子では、多量にある tRNA を用いたほうが翻訳効率が高まるので、コドン間に選択が働いたと考えられています。

しかし、このコドンの使用頻度と tRNA の含量の因果関係は、ニワトリとたまごの

関係にも似て、どちらが先なのか疑問が残ります。そこで、このコドン選択が働いた別の機構の一つとして、上記のDNAの幾何学的な制約からきたパッキング効率および転写の際の読み出し効率による二次的な選択圧が考えられます。ここで、二次的と言ったのは、冗長コドンの使用頻度の変化はコドンの一番目や二番目の塩基の置換とは異なり、タンパク質に直接的な影響を与えるのではなく、その効率に影響を及ぼすということから、選択圧が小さいということを意味しています。

それぞれの生物種のDNAの幾何学的制約や転写効率からコドンの使用頻度を選択が働き、そしてコドンの使用頻度に応じて、tRNAの量が調節されていると考えることもできるのです。もし、これが真実だとしたら、生物では直接、間接を問わず、適応度にわずかな差でも生じる変異は、長い時間の間に淘汰によって蓄積する可能性があることとなります。

この辺で、生物進化の概略を一応終えて、進化機構を人工的なシステムに応用する話に移りたいと思います。そこで、進化論的にも最もナイーブなネオ・ダーウィニズムの思想に基づいた遺伝的アルゴリズムの説明から入りましょう。

6

進化システムの夜明け

制御システムの最適化

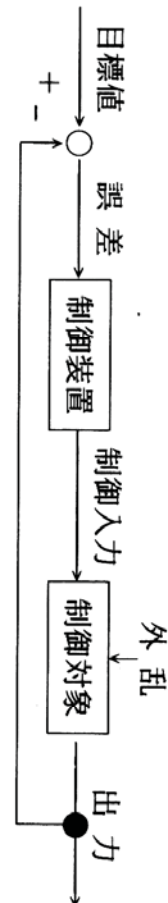
一九四八年のウィナーの『サイバネティクス——動物と機械における制御と通信』と題した著書がきっかけとなって、生物の制御機構や適応機構を模倣したシステムの研究が盛んになりました。一九五〇年代後半から、計算機の発達とともに人工知能や知的適応システムといった研究もなされるようになってきました。

現在のニューラルネットワークモデル(神経回路網モデル)は、このころのサイバネティクスから派生したバイオニクスにルーツを持っています。この一九六〇年代の楽観的とも言えるサイバネティクスブームの雰囲気なかで、ダーウィンの進化論にヒントを得た進化システムの研究が、フォーゲル、ワーブルトン、リードらによって行なわれました。

そして、ホランドが一九七五年に、基礎的な遺伝学的知識を背景に“Adaptation in Natural and Artificial Systems”を著して、進化する人工システムについて体系的にまとめあげました。最近になって再びブームとなった遺伝的アルゴリズム(Genetic Algorithms、以下GAと略記)とは、ここで記された確率的最適化手法のことを指します。特に組合せ最適化問題、制御システム、プログラム技術などへの応用が期待されています。

ここで、制御システムの最適化ということについて簡単に説明しておきましょう。まず、最適化したいシステムを考えます。そして、システムの設計パラメータのうちで調節可能な制御変数を決めておきます。この制御変数を何らかの方法で決定して、システムに入力します。システムはこれに回答して出力します。この出力が設計者の望んだ応答からずれている場合は、誤差を測ります。そして、制御規則を用いて誤差やコストなどに応じてシステムの制御変数を調節しなおします。この過程を繰り返すことによって、システムを設計者の望んだ状態に近づけます。これがシステムの制御です。

このとき制御対象と並べて制御装置を考え、制御対象の出力誤差を制御装置への入力として、また制御装置の出力を制御対象への制御入力としてとれば、制御対象と制御装置が閉じたループを構成します。このような制御方式をフィードバック制御といいます。



生物の神経系や内分泌系では、外的および内的な変化にさらされても、常に生理的狀態を一定に保とうとする働きがあり、これをホメオスタシスと言います。ウィナーは、ホメオスタシスは負のフィードバック制御によって維持されていると考え、これを定式化しました。しかし、実際には一七〇〇年代後半に、ワットが蒸気機関の調速器として遠心振り子をつけていたし、それ以前にもリーの風見車を使った風車の方向制御などがあり、これらはいずれもフィードバック制御の実例と言えます。

よくフィードバックの例に出されるのが、ひと昔前のコタツについていたサーモスタットです。よく知られているように、サーモスタットは二種類の金属を張り合わせたバイメタルです。そして、温度が高くなると二種類の金属の熱膨張率が異なるために曲がってきます。このサーモスタットの性質をコタツに利用すれば、温度が高くなり過ぎればスイッチが切れ、低くなれば再びスイッチが入り、ある温度範囲に収めることができます。

現代の最適制御では、多くの制御変数を扱うことができ、制御誤差と制御に要したコ

ストを同時に考えることによって、より少ないコストで制御誤差をより少なくするような最適パラメータを計算できるようになっています。また、システムの設計パラメータが劣化や環境変動によって変化する場合には、モデル規範適応制御が用いられます。これは制御対象の規範モデルをつくり、規範モデルの出力と制御対象の出力が近づくように制御パラメータを調節することによって制御が行なわれます。

このように最適制御はシステムの設計や制御を行なう際の不可欠の技術なのですが、適応領域に限界があります。まず、最適制御は制御対象をモデル化することが必要です。たとえば、電子部品で増幅器を設計する場合を考えると、基本的な回路の配線図はすでに何らかの基準で決まっているものとなります。そして、いくつかの抵抗やコンデンサーの値がまだ決まっていないものとしましょう。

この未決定の素子が多いときには、これらを可変抵抗や可変コンデンサーにして、実際に組み立てて目盛りを調整するには、あまりにも組合せが多すぎて現実的ではありません。そこで、トランジスタの特性をモデル化することにより、この回路全体のモデルを作ります。そして、この回路の方程式をもとにして、最適制御の手法を用いて、最小の電力で最大の増幅率を持つ回路のパラメータを決定します。

制御対象が完全にブラックボックスで入出力応答しかわからないときは、システム同

定という手法を使って、見掛け上入出力応答が同じになるようにモデルシステムの状態パラメータを調節します。いずれにしても、実際の制御対象とモデルとの違いが大きいと、制御入力がある値を越えたときに、思わぬ振動を引き起こしたりします。実際の制御対象は、多くの外的および内的因子が関係し、かつ非線形である場合が多いので、完全なモデル化は困難となります。

また、少なくとも区分的には入出力応答が連続的でなければなりません。つまり、区分の範囲内では、少し入力を変えたら、システムの状態や出力も少し変わることが必要です。制御はこの性質を利用しています。たとえば、ニューラルネットワークのモデル素子は、樹状突起から入力されるパルス頻度の総和が、しきい値を越えると神経細胞が発火して、軸索を通じてパルスを出力します。このときの入出力応答はモデルによって、S字カーブだったり、しきい値をこえると0から1にジャンプするステップ関数だったりします。ステップ関数のときでも、しきい値のところを除けば連続関数です。この性質によって、学習誤差が少なくなる方向に回路網を結んでいる線の強さを変更することによって、学習制御が行なわれます。

しかし、入出力関数がまったくびとびの値をとるシステムでは、制御のしようがありません。まして、後で述べるように、システムの状態が数値でなく記号などで表わさ

れる離散システムの場合には、根本的に発想を変えなければなりません。

ニューラルネットとアニーリング

ところが、興味深いことに、ニューラルネットワークのリバイバルブームの火付け役となったホップフィールドのモデルでは、このような離散的な組合せ最適化問題を例題として扱いました。多くの本で紹介されているので、詳しいことは省略しますが、その解法のミソは、物理学の初等的なエネルギー原理に基づいていました。

たとえば、考えている問題が n 個の 0 もしくは 1 からなる離散状態で表現されている場合、それを 0 から 1 の連続量で補間します。最終的な解が 0 と 1 の中間になった場合は、0 と 1 の近さによって、0 か 1 の確からしさを表わすものと解釈します。そして、現在の状態が、与えられた問題の制約条件からずれていたら、その量をエネルギーとして計算します。

複数の制約条件がある場合には、それぞれの制約条件に重みを掛けて、それらを足し合わせることによって、総エネルギーを計算します。箱庭にボールを転がしたら、ポテンシャルエネルギーの低いところに向かって転がっていきます。同様に、ネットワークの結線をうまく構成することによって、解が自然と問題の制約条件からのずれを小さく

するところに落ち着きます。

現在、この研究の評価は研究者によって大きく異なりますが、少なくとも一大ブームを引き起こした原因は、組合せ最適化問題を連続的なシステムで解いたことと、エネルギーというアナロジーを使った説明がわかりやすかったことにあると考えられます。

ただし、数理計画法の分野では、制約条件からのずれを、ペナルティー関数として目的関数のなかに繰り込んで最適化問題を解く手法は古くから行なわれていました。

しかし、すぐに誰でも気づくことは、箱庭の山が一つではなくて、たくさんあったときにもうまくいくのか、ということです。当然ながら、エネルギー解法では初期値によっては、一番低い谷に辿り着けない場合が生じます。つまり、ほんとうの最小値ではなく、局所的な最小値、つまり極小値にひっかかってしまうのです。これを克服するにはどうしたらよいでしょうか。

雨どいにひっかかったボールやバドミントンの羽を下から棒でつついて落とした経験はありませんか。ここでも同じように、箱庭全体を下から揺すってみることにします。このように、振動を加えることによって、極小値にひっかかったボールがそこから脱出して、さらに低い谷に向かうことが期待できます。

しかし、振動が大きすぎれば、せっかく一番低い谷にあったボールもそこから飛び出

してしまい、めっちゃくちゃになってしまいます。逆に、振動が小さすぎれば、極小値の谷の壁を乗り越えることができません。そこで、最初は振動を大きくし、徐々に小さくして行きます。このような、方法をアニーリング(焼きなまし)法と言います。

物理学では、この振動が熱に相当し、情報論ではノイズに相当します。このアルゴリズムを多くの問題に適用できるようにカークパトリックらが一般化したものを、シミュレーテッド・アニーリング(Simulated Annealing、以下SAと略記)と呼びます。

アニーリングを行なうときの温度管理をアニーリング・スケジュールと言います。また、極小値がたくさんある中から真の最小値を探し出すことを、大域的探索と言います。アニーリング・スケジュールは、収束速度が大きくて、かつ大域的探索ができるように計画しなければなりません。最初の温度や温度を降下させるときの勾配などは、問題に応じて経験的に決めなければなりません。

7 遺伝的アルゴリズムとはなにか

遺伝的アルゴリズムのしくみ

遺伝的アルゴリズムGAのリバイバルブームも、前章のニューラルネットワークのブームの延長にあります。一般にGAは、離散システムなどが自然に扱えるのが特徴です。このため、もともと組合せ最適化問題などとの相性が良いGAとニューラルネットワーク、シミュレーテッド・アニーリングSAなどとの比較がいくつかの例題で試されました。その結果、GAがきわめて大域性にすぐれ、収束も速いことが示され、多くの研究者の注目を引くこととなりました。

ここでは、GAの仕組みを説明するために、組合せ問題としては最も単純で代表的なナップサック問題を例にとって考えてみましょう。ナップサック問題は組合せ最適化問題の一つで、もともと数理計画法の分野で扱われていた問題です。

価格と重さ(もしくは体積)の決まった品物がそれぞれ n 種類あるとします。なかには、値段が高くても軽いものもあれば、安くても重いものもあります。そして、重量制限の決められたナップサックに、この品物を選んで入れていきます。このとき、重量制限内でできるだけ価格の総和が大きくなるように品物をうまく選ぶ、というのがナップサック問題です。

重さと体積の制限を両方考えたり、同じ品物を複数個選べるようにしたり、問題を拡張することができますが、ここでは簡単に、制限は重さだけで、同じ品物は一つだけしか選べないものとします。したがって、品物が n 種類あったときには、それぞれの品物を選ぶか、選ばないかの二通りあるので、全部で二の n 乗の選び方があることになりました。全部の場合をしらみつぶしに順番に調べれば、一番良い解が得られるのは当然ですが、たとえば、品物が 100 種類あるときには、全部で二の 100 乗、つまり約 10^{30} 通りあることになります。一 100 種類あれば、約 10^{301} 乗通りにもなってしまう。

このように、品物の種類が多いときには、総当たり法ではあまりにも計算時間がかかり過ぎて、実質的に計算不可能となります。そこで、効率の良い解法が必要となります。遺伝的アルゴリズムGAはSAと同様に確率的な近似解法です。ただし、同時にいろいろな初期値から出発する点が特徴です。まず、ある一つの品物の選び方を 0 と 1 から

なる長さ n の文字列で表わします。品物を選ぶ場合を1、選ばない場合は0で表現します。最初は、とりあえず0と1をサイコロの偶奇によってランダムにならべて文字列を作ります。これを m 回繰り返し返して m 個の0-1の文字列を作ります。GAでは、このようにして作った文字列のそれぞれの文字を遺伝子、また文字列のことを染色体と呼びます。そして、その染色体を持ったデータを個体と呼びます。この場合、それぞれの遺伝子座には0と1の対立遺伝子があることになります。

もうここまでの説明でおわかりかと思いますが、それぞれの個体の染色体には、品物の選び方がコードされているわけです。そして、この選び方にしたがって品物を選んだときの重量と価格の総和がそれぞれ計算されて、その個体に付与されます。この個体の評価値(もしくは生存能力)は、この価格の総和で表わされるものとしましょう。ただし、ナップサックの重量制限を超過した場合は、ペナルティー分を価格の総和から差し引くか、価格をゼロにしてしまいます。ここでは、厳しくして超過のときは全部没収することにししょう。

人為淘汰のかけ方

m 匹のすべての個体について、この評価値を求めます。そして、この評価値に応じて

人為淘汰を行ないます。つまり育種と同じように、設計者の望んだ形質を持つ個体を繁殖させるわけです。選択の仕方によって、個体の適応度、この場合は単なる繁殖率が異なります。最も標準的な選択の仕方は、集団のサイズ、つまり総個体数を常に一定として、各個体の評価値によって、次世代の子孫数を比例配分して適応度を決めます。これをルーレット選択と呼びます。

ルーレットの目の総数は決まっています。そして、評価値の高い個体ほど多くの目を占領できて、ルーレットを回したときに駒の入る確率が高くなります。駒を m 回投げ入れて、各回で駒の入った個体が次世代に子孫を残します。

したがって、評価値の高い個体ほど、適応度が高く、したがって次世代の子孫数の期待値は大きくなります。しかし、確率事象であるため、必ずしも適応度の高い個体が多くの子孫を残すとは限りません。集団のサイズが十分大きいときは、大数の法則にしたがって、実現値と期待値はほとんど等しくなります。

しかし、集団のサイズが小さいときには、確率的な揺らぎによって、適応度の高い個体が選択されなくて、適応度の低い個体を選択される場合があります。計算機でシミュレーションするような場合、それほど大きな個体数が取れないときには、この揺らぎの効果が大きくなります。集団遺伝学では、この確率的な揺らぎによって、集団中の遺伝

子頻度が偶然的に変動することを遺伝的浮動と呼びます。

この方法では、世代が進んですべての個体の評価値が揃ってくると、淘汰による効果が薄れてきます。たとえば、真の最大評価値が100だとすると、進化の初期では、評価値が7の個体や35の個体がまちまちに存在します。したがって、適応度は個体によって大きく異なり子孫数に大きな差が生じます。しかし、進化の後期では、評価値がどれも100に近くなっていくので、適応度に差がなくなり、どの個体を選ばれるかは確率の揺らぎの効果が大きくなり、偶然によって決まってしまう。このため、集団の多様性がそれほど急速に減少することはありませんが、最後の微調整がうまく行きません。

もう一つの選択法として、相対評価値を用いた選択が考えられます。その世代での最低の評価値を探し、各個体の評価値から最低評価値を減じて、これを改めてその個体の評価値とします。このように、常に相対値によって評価することにすれば、進化が進んで各個体の評価値に差が少なくなっても選択は働きます。このため、最後まで微調整はうまく行きますが、集団の多様性は急速に減少します。

これらの方法を両方表現できるようにしたものが次の式です。まず、 i 番目の個体の評価値を p_i とします。そして各世代における集団の中での最小の評価値を $\min$ と表わします。これらを用いて、各個体の評価値を新たに決め直します。これを v 値と呼ぶことに

しましょう。i 番目の個体の v 値は、

$$v_i = \min + S \times (p_i - \min)$$

で計算されます。ここで、S は選択の強さを表わすパラメータです。そして、i 番目の個体の繁殖率は、この v 値に比例するとすれば、次世代での子孫数の期待値、つまり適応度 f_i は、

$$f_i = m \times \frac{v_i}{\sum_{j=1}^m v_j}$$

と表わされます。ここで、m は集団の総個体数です。右の二式より、選択の強さ S がゼロならば、すべての個体の v 値は同じになるので、適応度も等しくなります。つまり、評価値による選択は行なわれないことになります。選択の強さ S が 1 のときは、 v_i が p_i に等しくなるので、先ほど述べた絶対評価値で選択する場合に相当します。また、選択の強さ S が十分大きいときは、 v_i が $p_i - \min$ にほぼ比例するので、相対評価値で選択する場合に相当します。

淘汰 + 突然変異 + 交叉 = 単純 GA

しかし、いずれにしても淘汰のみでは、最初ランダムに生成した個体のなかから良い

ものを選ぶだけなので、多様性は失われて行きます。そこで、多様性を生み出す機構として点突然変異を導入しましょう。ここでは、淘汰を行なった後、選ばれてきた個体の子孫を生成するときに点突然変異が入るものとしましょう。

各個体は長さ n の 0-1 文字列からなります。そこで、あらかじめ突然変異率を決めておいて、各遺伝子座について突然変異が起きるかどうかを確率的に試みます。つまり、0 から 1 までの実数の一様乱数を生成して、この値が突然変異率の値より小さければ、その遺伝子座の遺伝子に突然変異が生じると決めておきます。このとき、その遺伝子座の文字が 0 ならば 1 に、1 ならば 0 に反転します。この操作によって、前の世代にはなかった染色体を持った個体が次世代に生まれることが可能となります。そして、このようにして m 個の子孫を生成して次世代の集団とします。

このサイクルを繰り返すのですが、次の淘汰を行なうときに、前の世代の集団と次世代の集団との間に重なりがなく、次世代集団のみを用いるモデルを離散世代構造モデル、新旧の世代集団をいっしょにして淘汰をかけて、そのなかから次世代の集団を選択するモデルを世代の重なる離散世代構造モデル、もしくは重複離散世代構造モデルと呼ぶことにします。

世代の重なる世代構造モデルでは、場合によっては不死の個体ができますが、年齢構造

を導入して寿命を考えることもできます。また、これとは別に、前の世代で一番評価値の高かった個体を次世代に確実に生き残らせる戦略を考えることもできます。これをエリート保存戦略と呼びます。

一般に、離散世代構造モデルでは、突然変異によって前の集団の優れた個体の遺伝情報が破壊されて、次世代に遺伝しなくなる危険性があります。このため離散世代構造モデルでは、エリート保存戦略が用いられるのが普通です。また、重複離散世代構造モデルでも、集団のサイズが小さいときには、エリートとはいえ確実に選択されるわけではないので、これを保証する必要があるときにはエリート保存戦略を併用します。しかし、このエリートが局所的な極大評価値に集団全体を誘導してしまう場合があるので、絶対的な有効性があるわけではありません。無性生殖の場合は、基本的には上記のように、淘汰と突然変異を用いて集団の最適化を行ないます。

有性生殖の場合は、さらに交叉が追加されます。ただし、交叉を操作に加えるときには、遺伝子コーディングを行なう際に注意が必要となります。コーディングの仕方によっては、交叉を行なうことによって問題の制約条件をみたさない致死遺伝子が生じやすくなってしまうからです。したがって、遺伝子のコード体系を決める段階から交叉のことを考慮して設計をすすめる必要があります。

現在では、まだほとんどの進化システムは単細胞システムです。単細胞生物でもゾウリムシなどでは、生殖細胞系列と体細胞系列のような区別がありますが、ここでは簡単のために、このような区別は考えないことにします。また染色体も通常は一本しかありません。また、性染色体や大腸菌のF因子のように、性を決定する要素は通常ないので、雌雄は便宜上のものに過ぎません。一倍体と二倍体とで交叉、もしくは相同な遺伝子間での組換えの入り方は少し異なります。どちらの場合でも、淘汰と突然変異の操作はすでに完了しているものとします。

一倍体モデルでは、任意の二個体を選んできて、各遺伝子座間であらかじめ決められた交叉率にしたがって、交叉が生じるかどうかを確率的に決定します。確率の決め方は突然変異率のときと同様です。交叉が生じることになったら、その部位で二個体の染色体の組換えを行ないます。このようにして各遺伝子座間で順々に決めていきますが、次の交叉が生じるまでは、染色体の乗り換えは持続するものとします。したがって、もし一箇所のみで交叉が生じたときには、その部位でX字型に染色体が乗り換えた状態になります。

二倍体モデルでは、すでに交叉の説明のところで述べたように、交叉は二個体の間ではなく、一個体の(生殖細胞の)なかで生じます。この細胞が減数分裂して一倍体の配偶

子を作るときに、相同染色体間で生じます。また、ある遺伝子座の遺伝子型がヘテロになったときは、対立遺伝子間の優性・劣性によって、その遺伝子座の表現型を決めます。たとえば、ナップサック問題の場合、 i 番目の品物を父方の染色体では1（選択する）、母方の染色体では0（選択しない）であったとき、もし1が0に対して優性であれば、表現型としては1です。また、この優性・劣性の関係もコントロールしたい場合は、対立遺伝子をもう一つ増やします。それを2と表わすことにして、遺伝子0と内容としては同じでも、1に対する優性の関係が逆転しているものとします。

たとえば、この関係を $0 \wedge 1 \wedge 2$ と仮定すれば、遺伝子型が、00、22、02、12のときの表現型は0、遺伝子型が11、01のときの表現型は1となります。

遺伝的アルゴリズムでは、突然変異と淘汰の他に、この交叉を重要なオペレータであると考えます。淘汰によって選びぬかれた個体には、それなりの適応度を持つ遺伝情報が含まれているので、それらを組み換えることにより、さらに良い遺伝子組成を持った子孫が生まれることを期待するわけです。

人工的なシステムでは、各個体はまったくランダムな遺伝子から出発するので、進化初期の段階では、交叉によって両親とまったく異なった遺伝子を持った子孫が生まれます。しかし、進化の過程が進んで集団の平均適応度が極大値に近づくと、集団の遺伝的

多様性は減少して行きます。より優れた個体が繁殖することによって、その遺伝情報が集団全体に拡散していくからです。このため、各個体間の遺伝子の違いが少なくなつて、交叉が生じてても両親とよく似た子孫が生まれるようになります。ただし、ナップサック問題の場合は、事情が少し異なります。それぞれの個体が重量制限いっぱいまで荷物を積んでいるので、よく似ていても少しだけ異なった選び方をした二個体が交叉を行なうと、その子孫が重量制限を超過する危険が高まってしまいます。

突然変異が世代によらず一定の効果を及ぼすのに対して、一般的に交叉は世代とともにそのシャッフル効果が自然と減少していきます。交叉はこのように副作用が少ないために好んで人工的な進化システムの操作に用いられます。

これで、淘汰＋突然変異＋交叉の操作からなるGAの準備ができました。この最もシンプルなGAを単純GAと呼びます。また、前にも述べたように、通常は単細胞で染色体は一本、雌雄の区別はありません。システムによっては、単純GAに逆位や挿入、欠失、転位などさまざまな遺伝子操作を必要に応じて加えていきます。

8 遺伝的アルゴリズムの長所と短所

動的計画法と分枝限定法との比較

では、具体例として、単純遺伝的アルゴリズム(単純GA)を使ってナップサック問題を解いてみましょう。そのための視覚化ツールとして、図5に示したような解析ツールを作りました。すでに述べたように、それぞれの操作は至極単純なので、少し慣れたプログラマーならば簡単に自分でプログラムすることが出来ます。ただし、乱数を多く使うので、あらかじめ乱数の性質は調べておく必要があります。場合によっては、文献などを調べて自分で乱数ルーチンをプログラムしたほうが良いことがあります。

図の中心に示してあるのが、メイン・コントロールパネルで、突然変異率、交叉率、集団の大きさ、制約条件、淘汰強度の変動周期、エリート保存戦略の有無などをオンラインでコントロールできるようになっています。中央のグラフには、最大評価値と平均

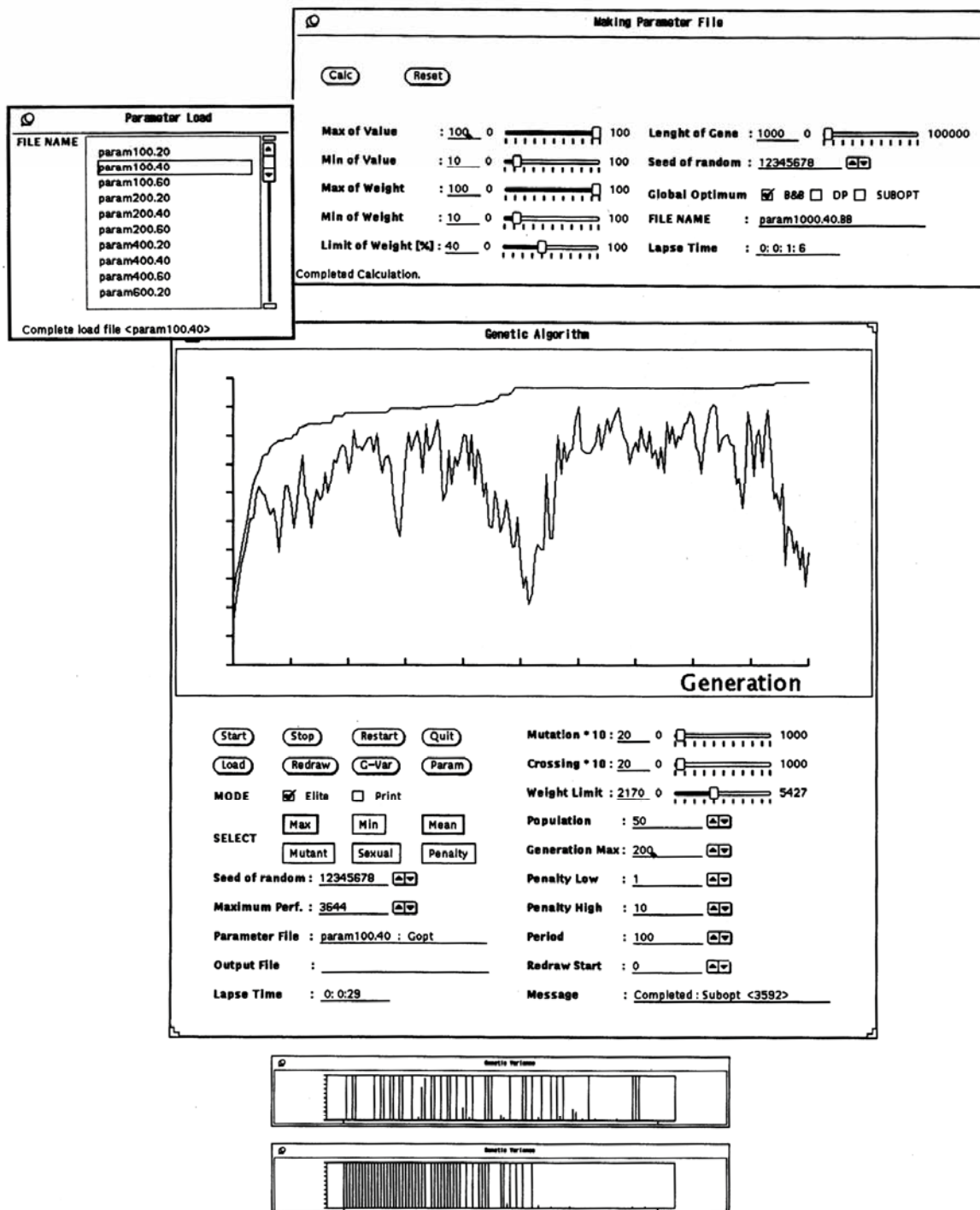


図5 GAの解析ツール

評価値のみを示してあります。

また、メイン・パネルからのコントロールにより、遺伝子プール内の各遺伝子頻度がリアルタイムでアニメーションできるようになっていて、遺伝的多様性の時間変化を把握することができます。図5の下の方の二つの遺伝子頻度グラフでは左からナップサックに入れる品物をコスト比(価格/重さ)の降順に並べてあり、対立遺伝子1の遺伝子プール内での遺伝子頻度を棒グラフによって示しています。上の遺伝子頻度グラフは進化初期のもの、下は進化後期のものです。

このグラフを観察することにより、集団から急速に遺伝的多様性が失われていくことが観察されます。進化の早い段階で、左側の効率の高い遺伝子は集団中に固定し、右側の効率の低い遺伝子は集団中から消失します。中央に位置する遺伝子群が徐々に調整されて次第に平均適応度が上昇して行きます。

そこで、GAの収束特性を調べるために従来手法との比較を行ってみましょう。従来手法としては、いずれも数理計画法でよく使われる動的計画法と分枝限定法を用いました。これらの方法についてきちんと説明しようとする、少し込み入った説明が必要になるので省略しますが、きわめてオーソドックスな解法だと考えてください。

簡単に要点だけを説明すると、動的計画法では、仮の重量制限を1から順に本来の重

量制限値まで増やしたときに得られる各問題（ナップサックサイズの異なる問題）を考えます。そして、品物の部分集合についての最適解を求め、この部分集合に次の品物を一つ加えたときの新たな最適解を表として順次記憶して行きます。つまり漸化式を解くことによって最適解を求めています。このために、欠点として、品物の数や重量制限が増えると、余分の計算容量と計算時間を必要とします。

一方、分枝限定法では、問題の性質をうまく利用して部分問題に分解し、効率的に部分問題のテストを行ないます。まず、連続ナップサック問題を線形計画法によって解いた解を用いて条件式を作り、これに基づいて、変数の部分的な固定を行なって、自由変数の数を減らす前処理を行ないます。残った自由変数をコスト比の良い順に並べ直して、この順に従って、できるだけ品物を選んだときの漸近解を初期の近似最適解として求めます。そして、この解に対して分枝操作を行なって部分問題に分解し、それぞれの部分問題を限定操作によってテストして、効率良く探索領域を狭めていきます。簡単に言ってしまうと、順に品物を選ぶかどうか決めるときに、あらかじめまったく無駄とわかっている探索を早めに省いてしまう手法です。また動的計画法と異なり計算容量もきわめて少なくすむ、という長所を持ちます。

動的計画法と分枝限定法をGAと同一条件で収束速度を比較できるように、システム

に組み込んでおいて試してみました。ナップサック問題での実計算時間を各手法について比較してみました。

遺伝的アルゴリズムはあくまでも確率的近似解法にすぎない

GAは分枝限定法や動的計画法とは異なり、確率的近似解法であるため、大域最適解に到達するまでの時間は、各パラメータや乱数の振り直しによって異なり、そのばらつきはきわめて大きくなります。一般的な傾向として、大域最適解の九〇%くらいまではきわめて速く到達しますが、大域最適解に到達するまでには長い時間を必要とします。先ほど述べたホップフィールド型ニューラルネットワークやシミュレーテッド・アニーリングなどよりは、確かに収束時間は格段に短く、大域的探索性能に優れています。しかし、品物の数が二〇〇個を越えるときわめて遅くなり、真の大域最適解を求めるという目的には用いることができないことがわかりました。ちなみに、GAでは品物の数が五〇個で、重量制限が総重量の四〇%のときには、ワークステーションで五秒から一五秒を要します。品物の数が一〇〇個のときは、エリート保存戦略を入れた重複離散世代構造モデルのGAでも数分から数十分もかかります。

ところが、動的計画法では、品物の種類数が一〇〇個程度なら一秒もかかりません。

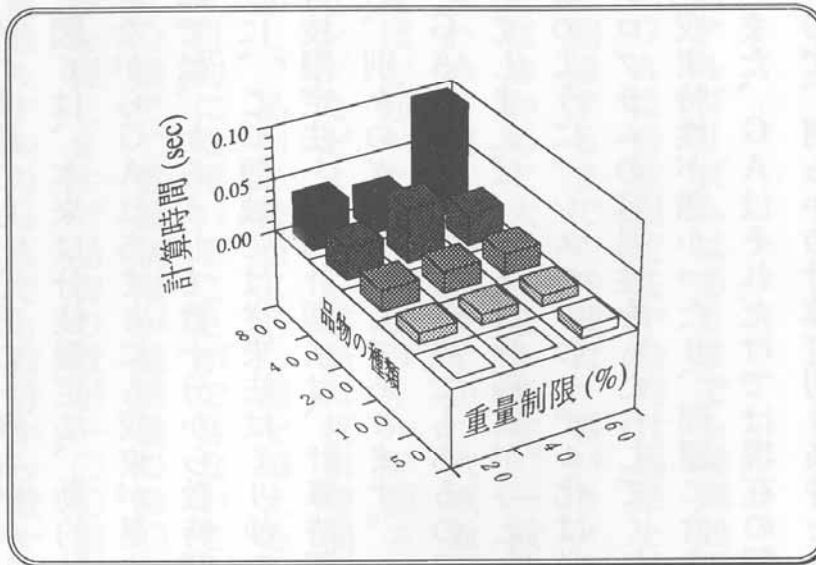
一〇〇〇〇個の問題でも一分以内で解くことができます。一方、分枝限定法では、三〇〇〇個の問題でも一〇分の一秒もかからずに厳密解を求めることができます。

図6は、本来は分枝限定法、動的計画法、GAを比較するためのものなのですが、残念ながらGAはあまりにも収束が遅すぎて計測ができませんでした。二〇〇〇個程度の問題では、かろうじて数十分から数時間で大域解に到達しますが、図6を見てもわかるように、この領域では従来法はミリ秒のオーダーで確実に計算が終了しています。また、分枝限定法と動的計画法は、計算時間のオーダーや重量制限値に対する特性が異なるため、別々のグラフにしています。

GAの良い点でも欠点でもあるのですが、問題のコントロール変数を遺伝子にコードしてしまえば、あとは評価値が与えられさえすれば、GAのループは回ります。最適制御のように、システムのモデル化は必要ありません。この記述の自由度が適用の広さやプログラムの容易性をもたらしてくれるのですが、逆にコーディングの工夫が足りないという収束特性が悪かったり、問題に含まれている情報を生かすきれなかったりします。

また、GAはそれだけでは現在の解が真の大域最適解かどうか判定することができないので、何らかの計算打ち切り条件を設けなければなりません。したがって、基本的には準最適解を得るための確率的近似解法に過ぎません。たとえば、ナップサック問題の場

分枝限定法



動的計画法

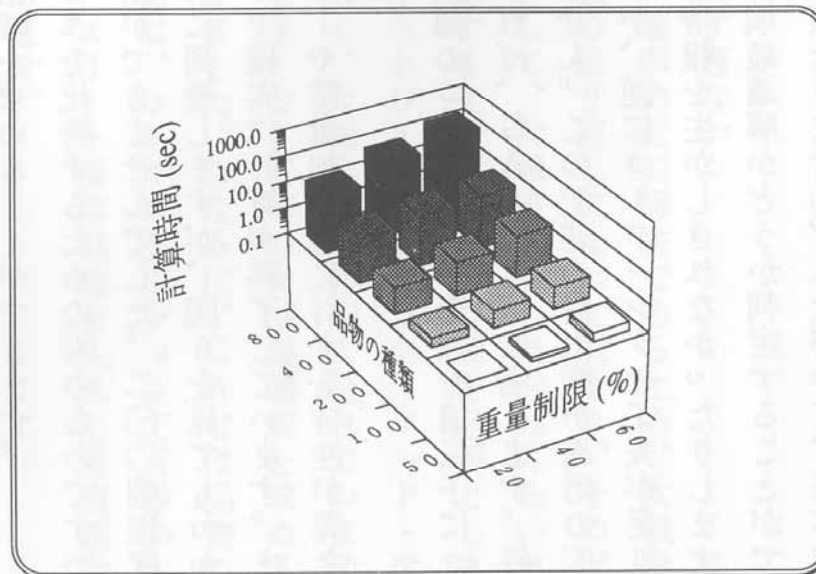


図6 分枝限定法と動的計画法の計算時間

合であれば、分枝限定法や動的計画法を持ち出すまでもなく、品物をコスト比（価格／重さ）の降順に選んで、重量制限いっぱいまで詰め込むだけで、これでかなり良い選び方になっています。この単純な方法だと、準最適解ではあっても、かなり質の良い解が得られます。このように品物の情報さえ利用すれば、いくら荷物の数が増えても、近似解を瞬時に計算できます。これでは、GAを使う意味がありません。

組合せ最適化問題では、あらかじめ問題の設定枠やパラメータが決まっていて、そのパラメータセットのなかから、最適な組合せを見つけることが課題となります。個々の問題については、ケースバイケースになりますが、一般的にこのような領域、つまり問題の各パラメータが、環境変動や時間的変化をしない静的な最適化問題の領域で、GAが真の意味で従来手法を凌駕するような例を見いだすのは困難なことと思われます。

従来、複雑すぎて工学的課題としては敬遠されてきたような現実的な問題領域にこそ、GAが真価を発揮できる問題があるのではないかと思います。そのような領域には、たとえば、未知の変動環境下での適応システムの構築や、自らの仕様記述を進化させていくシステムなど、手つかずの興味深い問題が多く残されているのです。

9

遺伝子複製の不均衡仮説

リーディング鎖とラギング鎖

何度か説明しましたが、G Aでは通常、遺伝子は一倍体で一本の染色体からなります。しかし、遺伝子の遺伝情報の方に注目していたので、その遺伝子の構造について触れることはありませんでした。生物では、すでに述べたように、遺伝子はDNAの場合も、RNAの場合もあります。また、その構造は二本鎖と一本鎖の場合があります。

ヒトの場合、よく知られた二重らせんという言葉からもわかるように、遺伝子は二本鎖DNAでできていて、らせん構造をしています。この二本鎖と先ほど説明した二倍体とを混同しないように気をつけてください。ヒトの場合は、二本鎖のDNAで染色体ができていて、それぞれの相同染色体が父方と母方で一本ずつ、合計二本あるので二倍体というのです。つまり、一对の染色体は四本のDNAからなります。

9 遺伝子複製の不均衡仮説

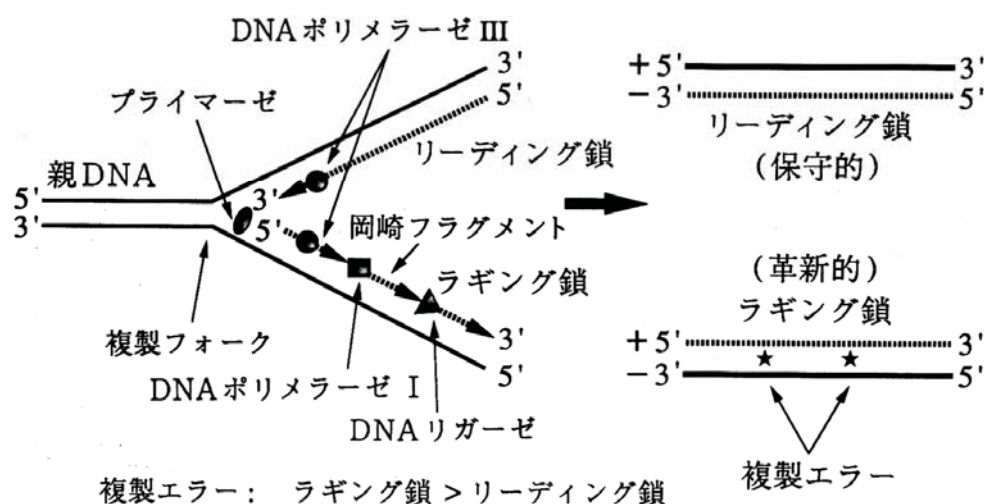


図7 遺伝子複製の不均衡性

この二本鎖DNAが複製されるときには、複製されるDNAはいずれも決まった方向(図7で5'から3'方向)に向かって複製されます。親DNAが開く方向と同じ方向に合成されるDNA鎖をリーディング鎖、逆方向に合成されるDNA鎖をラギング鎖と呼びます。図7に簡単に示したように、リーディング鎖とラギング鎖とは複製機構が大きく異なります。リーディング鎖ではDNAが連続的に合成されるのに対して、ラギング鎖ではいくつもの酵素が関与して、岡崎フラグメントと呼ばれる断片を繋ぐことによって複製されます。

このことから、古沢と土居は遺伝子複製機構に注目して新しい進化の考え方を示しました。まず、図7のような複製機構の違いから、リーディング鎖とラギング鎖との間で複製エラーの割合に差が生じている、つまり複製機構の複雑なラギング鎖のほうに、

より多くの突然変異が蓄積するだろう、と考えるのです。このように考えると、世代を重ねるにしたがって、複製フォーク(図7参照)の開始起点の片側に突然変異の集中した遺伝子領域ができたり、逆に何世代にもわたって突然変異を受けない遺伝子領域が存在したりすることになります。この変異の不均衡性が生物の進化にとっても重要な機構として機能しているものと考えて、彼らは不均衡進化論を提唱しました(Mitsuru Furusawa and Hirofumi Doi, *J. Theor. Biol.* (1992) Vol. 157, pp. 127-133)。

これを検証するために、私たちは単純な人工的な進化システムに、この遺伝子複製の不均衡性を導入することによって、進化システムの最適化問題に対するパフォーマンスがどのように変化するかを調べました(Ken-nosuke Wada, Hirofumi Doi, Shin-ichi Tanaka, Yosiko Wada, and Mitsuru Furusawa, *Proc. Natl. Acad. Sci. USA* (1993) Vol. 90, pp. 11934-11938)。集団遺伝学では、塩基配列ではなく遺伝子を最小単位と捉えているために、遺伝子の適応度や変異率を、あらかじめ仮定してから議論を始めなければなりません。そこで、GA的手法を導入することによって、この塩基配列と適応度との溝を埋めることが可能となります。

まず、最初の例題として、ナップサック問題を扱いました。前章でナップサック問題を解くには、GAより従来手法の方が優れているため、工学的な意味はないと述べたばかりです。しかし、なぜこの問題を選んだかというと、単純な割には生物学的に捕え直

したときに、興味深い対応が浮かび上がってくるからです。

この問題では、進化初期の段階で突然変異や交叉による遺伝子組換えが生じることによって、広域の探索に有効に機能します。このため、淘汰に有利となる変異が生ずる確率は高いものと考えられます。そして、進化が進むにつれて、重量制限の限界まで品物を選んだ個体が残ります。したがって、わずかの突然変異が生じても重量制限を超過する危険が高まり、致死突然変異が急激に増加して行きます。

さらに、それぞれの遺伝子は単位重さ当りの価格という情報が付随していて、適応度に寄与する度合いが異なっています。したがって、淘汰によって選択された個体の遺伝子を見ると、変異の起きやすい領域と起きにくい領域とに分かれます。これらは、すべて以前の章で、生物の進化を説明するときに述べたことです。

確かに、このような単純な問題から、生物の進化を論ずることはきわめて危険です。しかし、集団遺伝学とは異なり、遺伝子型と適応度との間を最適化問題によって関連づけることができるため、新しい視点が生まれます。

均衡モデルと不均衡モデル

そこで、これらのことを念頭においてシミュレーション実験を行なってみることにし

ました。ここで、遺伝子複製の際にリーディング鎖とラギング鎖との間の複製エラーに不均衡が生ずると仮定したときのモデルを不均衡モデル、両者とも同じと仮定したときのモデルを均衡モデルと呼ぶことにします。そして、初期世代に両者のモデルに基づいた個体を半数ずつ生成して、生存競争を行なわせてみました。このとき、それぞれのモデルの両DNAに生ずる突然変異率の和が等しくなるように設定しました。つまり、不均衡モデルのリーディング鎖とラギング鎖の突然変異率を足し合わせて半分にした率で、均衡モデルの両DNA鎖に突然変異が入るものとしています。

さまざまな遺伝機構やパラメータについて調べた結果、不均衡モデルは均衡モデルに比べて、圧倒的に優位であることが示されました。特に次のような条件のもとで、その優位性は顕著となります。

- (1) 突然変異率が高い
- (2) 淘汰の強度が大きい
- (3) 集団のサイズが小さい
- (4) 有性生殖で二倍体である

また、初期世代での不均衡モデルの個体数を均衡モデルの一〇分の一に設定しても、不均衡モデルの優位性は変わりませんでした。

不均衡モデルでは、親と同じ形質を持つ個体と、親とはかなり違った形質を持つハイリスクな個体とを両方とも生成できるので、遺伝子空間を安全に遠方探索できます。保

険をかけながら賭けをしているようなものです。したがって、環境変動に対する適応能力も高く、突然変異率が高いときにも安定なシステムを構成できるため、進化スピードを速めることができます。

しかし、染色体上に複製開始起点が一つしかない場合は、均衡モデルでも重複のある世代構造にすることによって、同じような効果が疑似的に得られます。リーディング鎖由来のDNAを持った個体が前の世代の個体に相当し、ラギング鎖由来のDNAを持った個体が次世代の個体に相当します。

しかし、染色体上に複製開始起点が複数存在する場合には、本質的な差が生じます。重複離散世代構造の均衡モデルでは、突然変異がすべての個体のすべての遺伝子に様に影響するのに対し、不均衡モデルにおいては、各遺伝子が複製開始起点のどちら側に存在するかによって、保存性の高い領域と可変領域とに分れることになります。

図8では、このことを模式的に表わすために変異が決定論的に描いてあります。一回の複製でラギング鎖に一つエラーが入り、リーディング鎖には入らないとしたときに、世代とともにそれぞれのDNAにどのようにエラーが蓄積していくかを示してあります。一番右下のDNAをみると、遺伝子aと遺伝子cは、同じように突然変異が蓄積し、もしここに変異がたくさん入っている場合には、遺伝子bと遺伝子dには変異がまったく

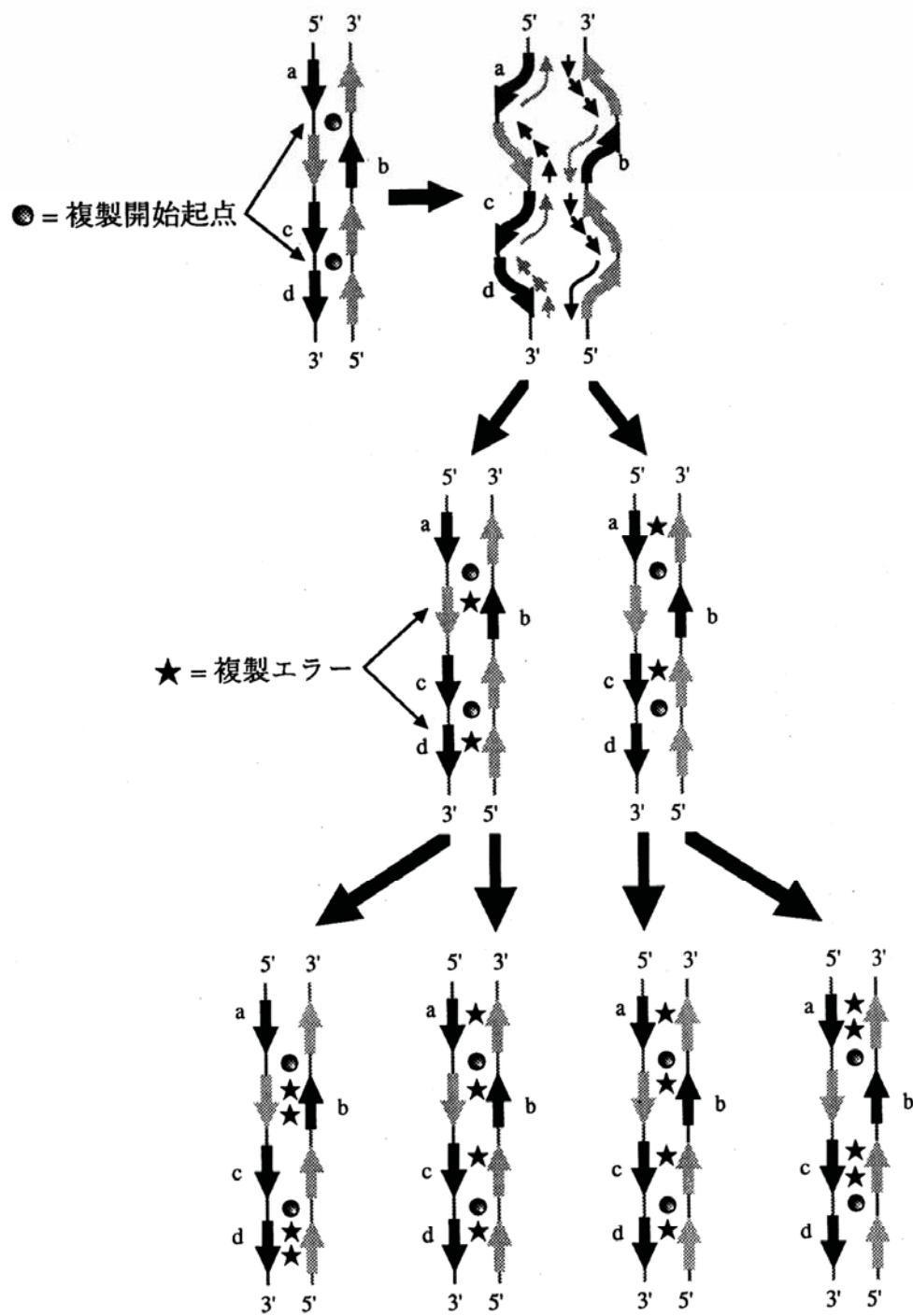


図8 不均衡エラーの蓄積

入っていないことになります。

もし、遺伝子bと遺伝子cを含む領域が、その遺伝子座の持っている意味ごと逆位を起こせば、こんどは遺伝子aと遺伝子bが組になって同じ変異の仕方をすることになります。ここで、「遺伝子座の持っている意味ごと」と表現したのは、遺伝子座の意味をそのままにして、遺伝子データだけを逆順に並べたのではないことを意味しています。たとえば、ある個体の染色体が、二番目の遺伝子座が1で三番目の遺伝子座が0だとすると、この二つの遺伝子を含む領域が逆位を生じたとします。データだけ逆順に並べるときは、入れ替わった後、二番目の遺伝子座に0が、三番目の遺伝子座に1が入ります。したがって、これは表現型では、二番目の品物を選ばず、三番目の品物を選ぶ、ということの意味し、逆位前とは異なった選択をすることになります。そうではなくて、遺伝子データと遺伝子座の意味がいっしょに入れ替われば、最初から三番目の遺伝子座に二番目の品物についての遺伝子1が、二番目の遺伝子座に三番目の品物についての遺伝子0が入ります。これは、逆位前と同様、二番目の品物を選び、三番目の品物を選ばない、ことを意味します。変わったのはそれぞれの変異のしやすさです。

このことにより、遺伝子の転座や逆位を取り入れることにより、淘汰によって各遺伝子が環境に応じて異なった変異率を獲得することが可能となります。

10

適応能力イコール適応度ではない

環境変動を生きぬく個体

さらに、環境変動下で不均衡モデルがどのように機能するのか調べるために、二次元の多峰性の最大値探索問題に地殻変動を加えてみます。各個体に山脈の全貌が見えていないときに、たくさんの山脈のなかから一番高い山頂を探し出す問題です。ただし、各山の位置や高さはゆっくりと地殻変動をして、絶えず全体の地形が変わって行きます。未知の環境変動にリアルタイムで追従する適応システムについては、従来複雑すぎて扱うことが困難でしたが、この不均衡モデルを導入した進化システムによって新しい進展が期待できるかどうか試してみよう。

各個体はx座標とy座標のデータ、および両DNA鎖の変異率をコードした染色体を持ちます。それぞれの遺伝子は32ビットの0-1文字列で表わされています。この文字

表1 2進数とグレイコード

10進数	2進数	グレイコード
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 0	0 1 1
3	0 1 1	0 1 0
4	1 0 0	1 1 0
5	1 0 1	1 1 1
6	1 1 0	1 0 1
7	1 1 1	1 0 0

列を数値に置き換えるときに、二種類のコード体系を用いました。一つはよく知られた2進法です。もう一つは情報科学の分野で使われるグレイコードと呼ばれるコードです。2進数からグレイコードを得るためには、上桁から下桁に順にみて、もし1があつたら、次の下位桁のビットを反転させることによって変換します。ただし、上書きしながら変換するのではなくて、もとのコードはそのままにして、変換したコードを別に書き出します。具体的に0から7までの数をそれぞれのコードで書いてみると、表1のようになります。

グレイコードは隣り合った数が1ビットの反転で得られていることに注意して下さい。このため、突然変異率が低いときには、0-1文字列からなる遺伝子のコードと、その文字列を数値としてみたときの値との間に、ある程度の連続性が生まれます。

2進数では、たとえば10進数で3から4へ変化するときのように、新たに上位桁に桁上がりするときに、すべてのビットが反転しなければなりません。このため、低突然変異率のときは、ここがギャップとなります。しかし、グレ

イコードも1ビット反転で連続しているとはいっても、ランダムに1ビット反転したのでは、不連続となります。また、突然変異率が大きくなって、2ビット以上反転するときにも、不連続となります。したがって、必ずしもグレイコードの方が有利とは限りません。

図9-1にシミュレーションの結果が示してあります。それぞれの遺伝機構をもった個体の分布と各種の個体数の変動のようすを、世代1、10、30、80について表示しました。ここで、Dは不均衡モデル、Pは均衡モデル、Bは二進法コード、Gはグレイコードを表わしています。四つの組合せを、それぞれのアルファベットの組で表現しています。

ある程度親の形質を保った子孫のほうが、次の世代に親と近くの場所で生き残れる確率が高まりますが、これでは地殻変動によって、さらに高い山が遠くにできても、その山に飛び移るというようなことはできません。そのためには、有性生殖による個体間の染色体の交叉を行ったり、突然変異率を大きく取らなければなりません。しかし、あまり大きくすると、せっかく親がよい場所を占めていても、まったく違った場所に子孫が生まれ、リスクが大きくなります。

不均衡モデルでは、二本鎖の遺伝子複製の際に、それぞれ違ったエラー率で複製を行

なうことによって、親とほとんど同じ遺伝子コードを持った保守的な子孫と、かなり違った遺伝子コードを持った革新的な子孫を、共通の親から生じさせることによって、このジレンマを解消することができます。このため、前の章の問題と同様に、総突然変異率を同じにして、不均衡モデルと均衡モデルを競合させると、圧倒的に不均衡モデルの個体が繁殖します。

超保守集団の戦略

ところが、いろいろなパラメータで試しているうちに、おもしろいことに気がつきました。繰り返しになりますが、不均衡モデルのラギング鎖のエラー率は高く、リーディング鎖のエラー率は低い値をとります。均衡モデルでは、ラギング鎖もリーディング鎖も同じエラー率を持ちます。そして、両モデルの総突然変異率が一定のときは、不均衡モデルの両DNA鎖のエラー率の和を半分にしたエラー率で、均衡モデルの両方のDNA鎖に等しく突然変異が入ります。

ここで、両モデルの総突然変異率が一定、という設定をやめてみましょう。均衡モデルの両方のDNA鎖のエラー率を、不均衡モデルのリーディング鎖のエラー率と同じにしてみましょう。この場合は、均衡モデルの総突然変異率は不均衡モデルのそれに比べ

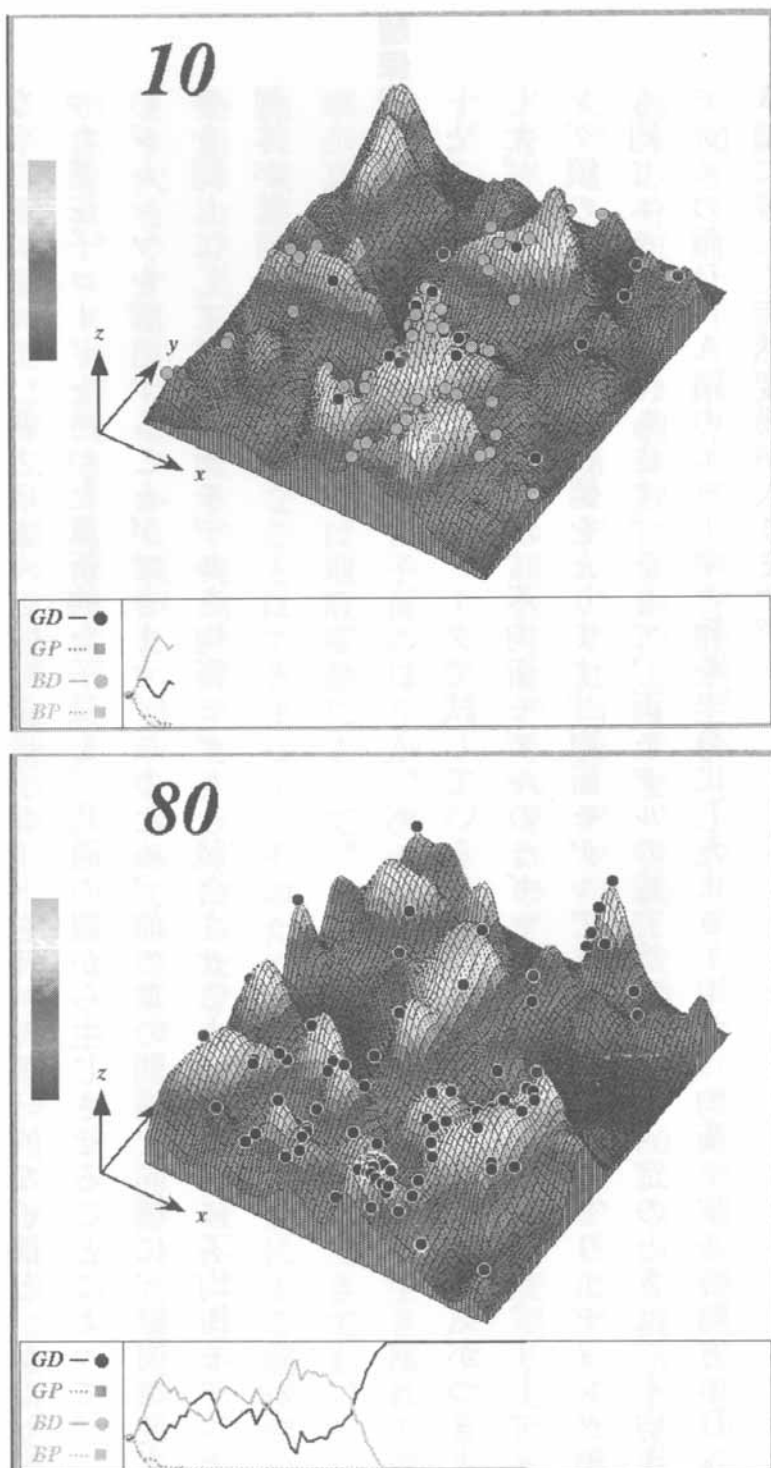
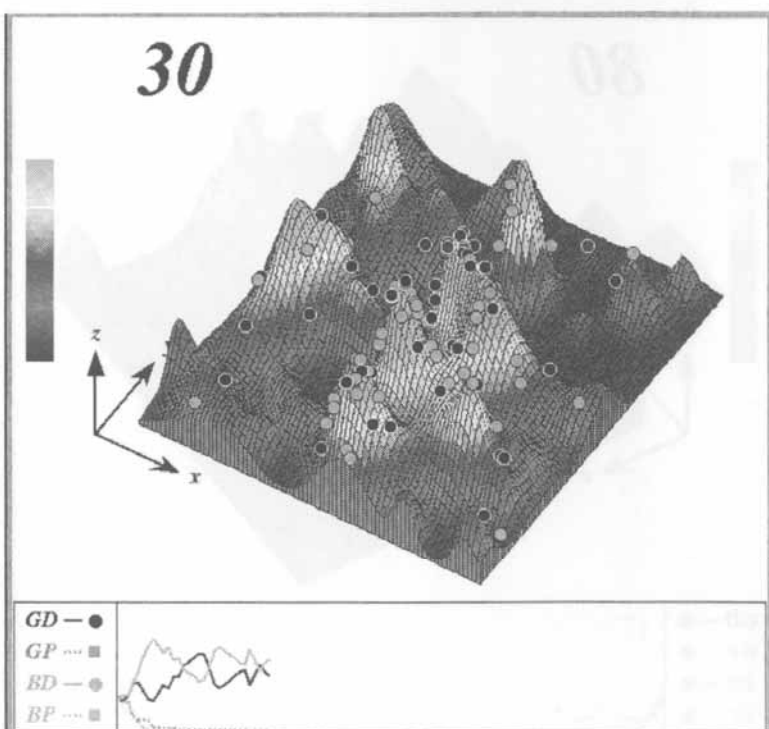
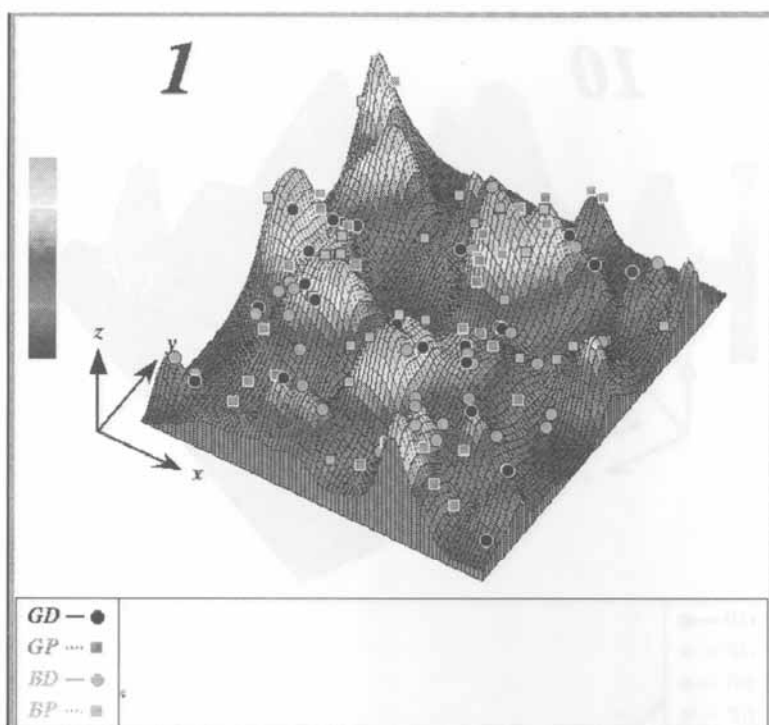


図 9-1 不均衡進化論(不均衡モデル vs 均衡モデル)



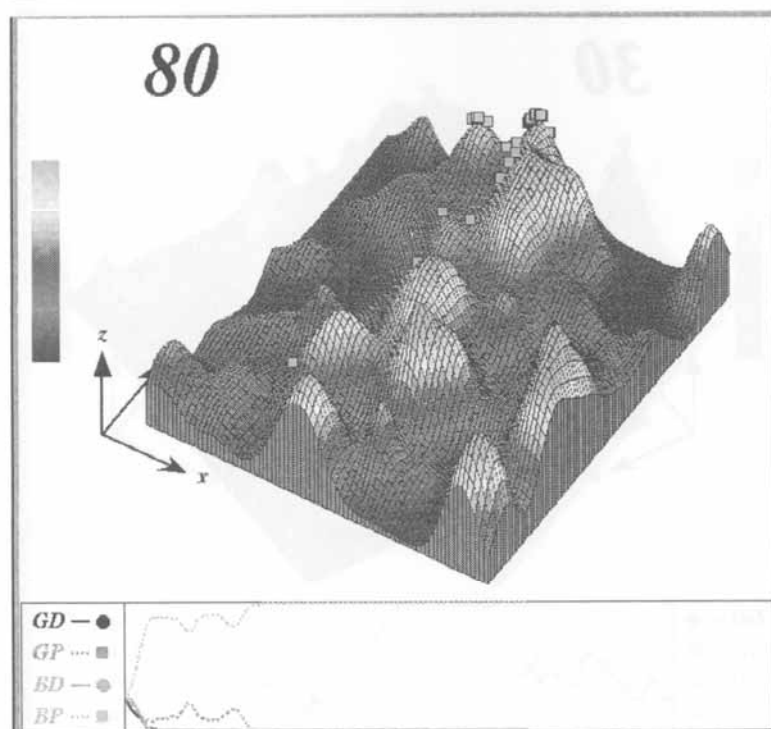
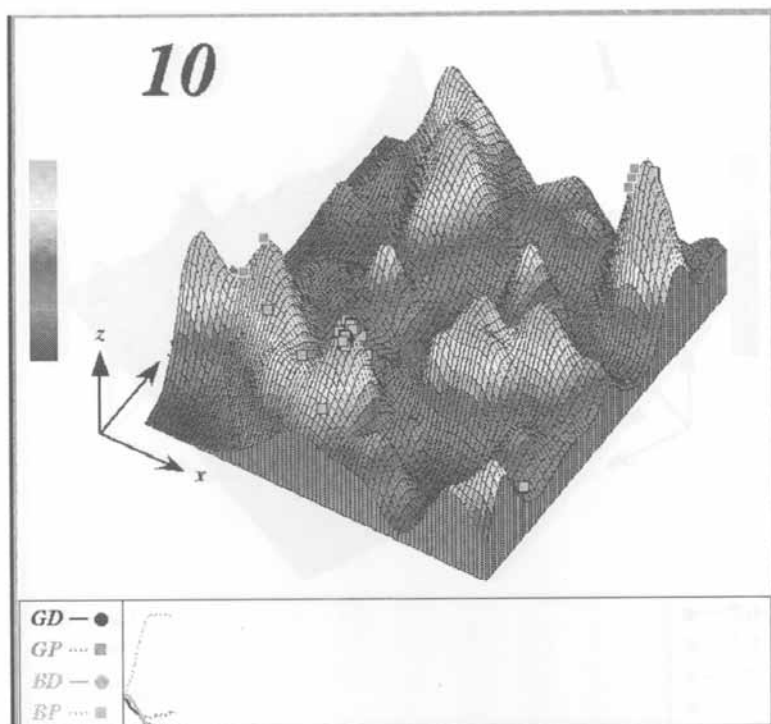
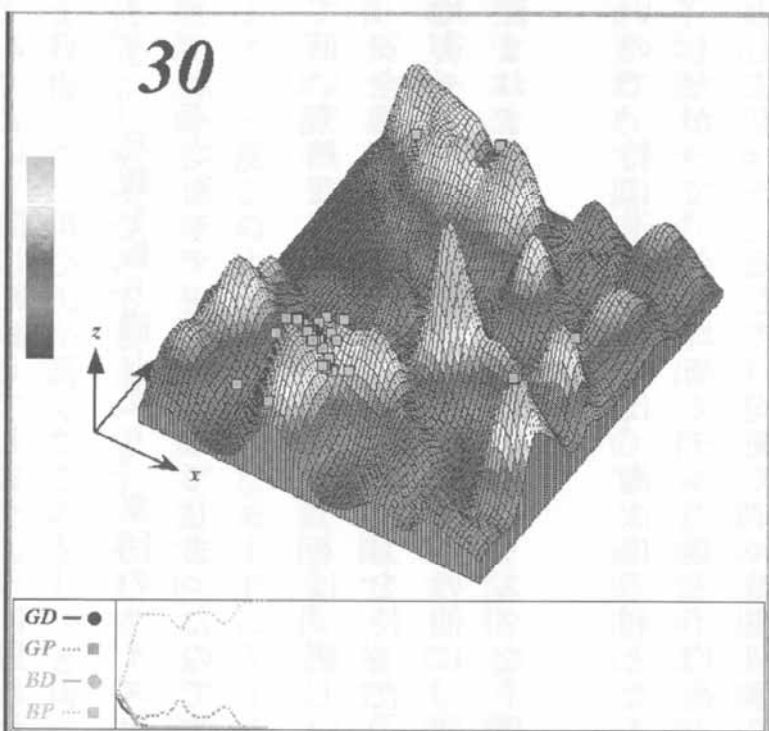
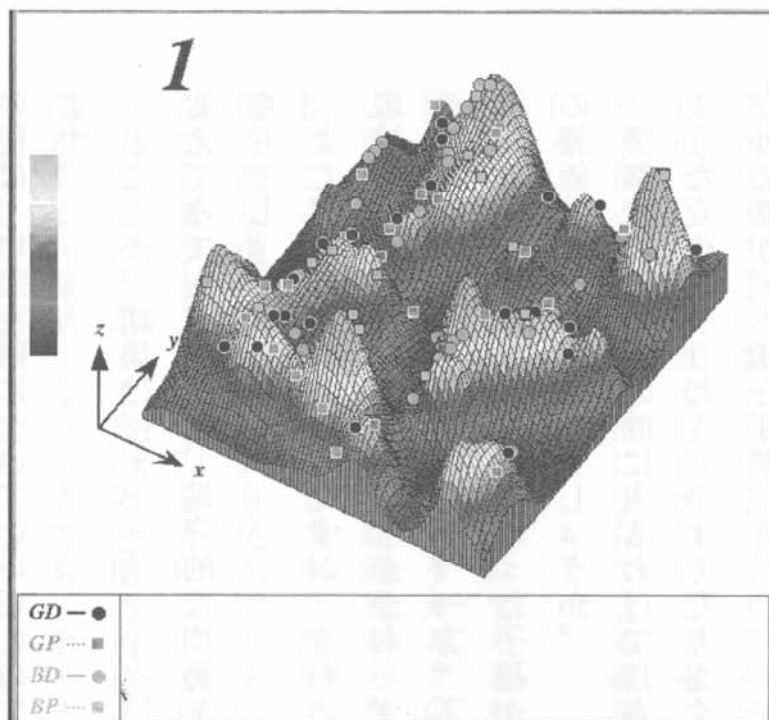


図 9-2

不均衡進化論(不均衡モデル vs 低突然変異率モデル)



て、著しく低い値になります。したがって、ものすごく保守的な集団を構成します。しかし、地殻変動のパラメータはそのままにしておきます。

さて、この二つの種を競合させたらどうなるでしょうか。当然、保守的な均衡モデルの種は、地殻変動についていけないので、あっという間に絶滅してしまうように思われます。

ところが、結果は逆だったのです(図9-2)。乱数を振り直したり、集団のサイズを変えてみても、やはり、保守的な均衡モデルが不均衡モデルを圧倒してしまったのです。なぜでしょうか。

よく考えればわかるのですが、きわめて低い突然変異率を持つ均衡モデルは、新しい環境への適応能力が低いにもかかわらず、安全確実に親と同程度の適応能力を持った子孫のみを産むことができます。一方、不均衡モデルでは、親とよく似た子孫の他に、変異によってかなり形質の異なった子孫が産まれます。はたして、このハイリスクな子孫の評価値は高くなれるでしょうか。

突然変異は合目的に入るわけではないので、特に進化後期では、淘汰に不利となるような変化を引き起こしやすくなります。したがって、仮に均衡モデルの親と不均衡モデルの親が同じ遺伝子組成を持っていたとしましょう。そして、均衡モデルの親個体の、

次世代での子孫数の期待値、つまり適応度が二匹だとします。すると、親の遺伝子組成は同じ、つまり親の評価値は同じでも、不均衡モデルの親個体は、自分と形質のよく似た子孫の他に、次世代に生存できる確率がきわめて低い子孫も産むので、適応度は半分になってしまいます。

この不均衡モデルから産まれてくるハイリスクな子孫こそ、新しい場所を開拓し、集団全体の進化速度を高めるパイオニアになりうるのですが、その成功率はあまりにも低いのです。このため、低突然変異率の均衡モデルの種は、進化の初期の段階で、犠牲を払うことなく安全に繁殖します。次第に、不均衡モデルの種を圧倒し、集団全体として保守的な進化モードに入り込んでしまいます。一度このような保守的なモードに入り込んでしまうと、情けないことに、いくら地殻変動してもっと高い山ができていても、その山がよほど近くでない限り前の世代の位置にしがみついてしまいます。そして、その山がある程度低くなってきたても、そのまま我慢して、再び山が高くなるのをじっと待つようになってしまいます。この集団からは、遠くの山を見つげるために危険をおかすような子孫は産まれないので、周りより少し良ければ、それで十分繁殖できるのです。ああ、情けない！

進化の途中で、このいまましい保守的な均衡モデルの種を作為的に絶滅させると、

不均衡モデルのめざましいまでの活躍が観察できます。前の世代の位置を保ちつつ、多くの犠牲を払いながら、造山活動で新しくできた高い山に果敢にチャレンジします。したがって、不均衡モデルの種は、かなり地殻変動を激しくしても追従することができません。小さな峰が一〇〇余りもある複雑な地形が激しく変動しているにもかかわらず、ほとんど常に最高峰に個体が登頂しています。

以上のことは、個体の適応能力と適応度とは違うということを、改めて痛感させてくれます。適応度とは、誤解を恐れずに、あえて簡単に言ってしまうえば、子孫の繁殖成功度のことです。低突然変異率の均衡モデルは、環境変動への適応能力には欠けますが、適応度が大きいので繁殖に成功したのです。逆に、不均衡モデルは環境変動への適応能力が高いにもかかわらず、適応度が小さいために、繁殖することができなかったのです。進化システムの工学的応用の見地からは、適応度より適応能力に関心があるわけですから、不均衡モデルの方が望ましいわけです。前にも述べたように、染色体上に複数の複製開始起点があるときには、特に有益となります。

実際の生物はどうしているのか

では、メタファーであるにもかかわらず、あえて疑問を投げかけてみましょう。生物

は、いったいどっちなんでしょうか。

現時点で、この疑問に答えることが無謀なのは、十分承知の上で言うならば、進化の初期や環境の激変時においては、複製機構の違いによる不均衡が積極的に作用して、進化速度を速める働きをしたのではないかと考えられます。また、進化が進んだ後でも、細菌などのように厳しい環境下にさらされている生物では、もし複製装置や修復装置が変異して不均衡性が強まれば、新しい環境への適応能力が増して、その変異が有利に働くものと考えられます。

細菌などでは、一つのコロニーは同一のクローンからなる場合が多いので、遺伝子からみれば、どの細菌の遺伝子もよく似ているために、どれが生き残っても大差はありません。もし、クローン全体が保守的な均衡モードになったとすると、新しい薬剤に対する耐性を獲得する能力などが下がることになり、場合によってはクローン全体の絶滅を引き起こしてしまいます。しかし、進化が進み、複雑で精緻な代謝機構や環境適応機構を獲得した生物では、ほとんどの変異が淘汰に対して不利か、よくて中立となります。したがって、このような生物は、精度のよい複製機構や修復機構も同時に獲得して、低突然変異率の均衡モードに移り、適応度が増して安定に繁殖することができたのではないかと考えられます。

では、このような複雑な代謝機構などは、どのようにして獲得されてきたのでしょうか。とてもダーウィンの言うような、淘汰にとって有利な突然変異の蓄積だけで進化してきたとは思えません。

人工的な進化システムを設計するときにも、複数遺伝子からなる多機能システムを進化させるためには、もっと安定でかつ飛躍的な進化機構が必要となります。しかし、安定でかつ飛躍的といった相矛盾する性質を持った進化機構など存在するのでしょうか？ 次の章で、この進化機構の一つの可能性として、遺伝子重複と中立突然変異を考えてみましょう。

11

遺伝子重複と中立突然変異

免疫と遺伝子重複

すでに述べたように、染色体上にはトランスポゾンや不等交叉によって重複が生ずることがあります。そして、多くの遺伝子を解析することから、生物は遺伝子重複機構を利用して、次第に多機能化することによって進化してきたと考えられるようになりました。

たとえば、生体に異物が侵入してきたときに、これを防御するために免疫機構が働きますが、このときB細胞は抗原(侵入者の部品)に抗体という飛び道具を投げつけて応戦します。この抗体が抗原をちゃんと識別できるように、抗体と抗原は鍵と鍵穴の関係になっています。未知の鍵穴にうまくはまる鍵を作るために、免疫細胞は実に巧妙な仕掛けを持っています。抗体には、体内に侵入したウイルスや毒物を中和する「鍵」と呼ば

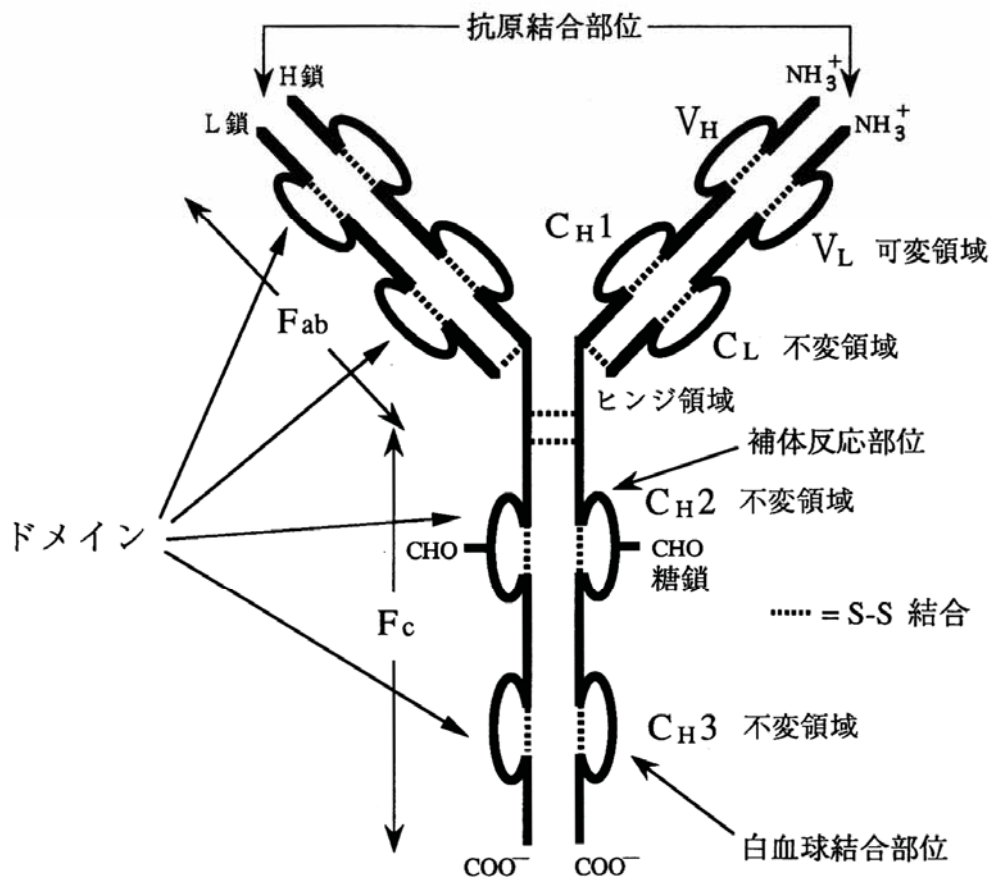


図 10 IgG1 の基本構造

れる抗体や、母乳や腸や気管支の粘液に含まれていて、外界と接している器官で侵入を阻止するIgAと呼ばれる抗体などがあり、五つのクラスに分れます。たとえば、IgGの抗体は図10のようにY字型をしていて、二本の同一のH鎖と、二本の同一のL鎖の複合体からできています。

図のように各H鎖は四つの同じような領域からできていて、これをドメインといいます。各L鎖は二つのドメインからなります。各抗体はクラスごとに共通の構造と、抗原

11 遺伝子重複と中立突然変異

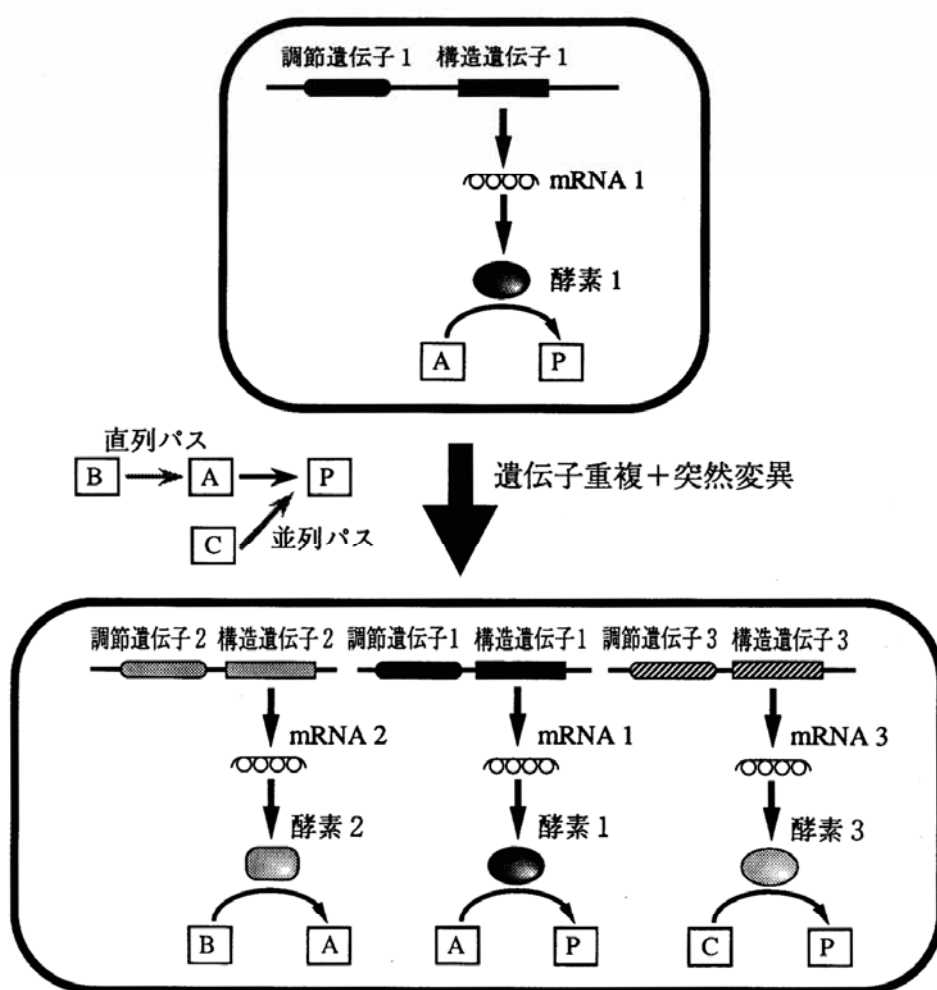


図11 遺伝子重複による代謝経路の進化

にぴったり対応した固有の構造の両方からできています。このドメインは別々のエクソンにコードされています。そして、この遺伝子を解析することで、すべてのエクソンは単一のエクソンを祖先に持ち、重複を重ねることによって、次第に防御能力の高いタンパク質に進化してきたものと考えられています。この他にも、アルブミンやコラーゲンなど多くのタ

ンパク質が、遺伝子重複によって進化してきたものと考えられています。

また、ホロウィッツが提唱したように、遺伝子重複によるタンパク質の構造変化によって、代謝機能が進化することがあります。図11に示したように、遺伝子が重複することにより、どれかの遺伝子が以前の機能を保持することができれば、他の重複遺伝子が代謝系に悪影響を与えないかぎり変異することが可能となります。たいていの場合は、これらの遺伝子は偽遺伝子化するでしょうが、ごくまれには新しい代謝反応を生成して、従来の代謝系に付加する形で新たな代謝経路を獲得することができるかもしれません。このようなときには、機能的に大きな進化が生じるものと期待されます。

中立な突然変異

中立説は分子の進化速度(塩基やアミノ酸配列が突然変異によって変わる速さ)の研究から生まれた分子進化の理論です。中立説では、分子進化を引き起こす突然変異の大部分が、淘汰に対して有利でも不利でもない中立な変異で、これは主に淘汰ではなくて、偶然的浮動(確率的な揺らぎ)によって突然変異遺伝子が集団中に固定する、と考えます。しかし、これは決して、中立説がネオ・ダーウィニズムと相反する概念であることを意味しているわけではありません。中立説を提唱した木村は次のように述べています。

『ここで注意しておきたい点が二つある。第一は、中立説がすべての突然変異が淘汰に対し中立であることを主張しているわけでないことである。…(中略)…第二は、ダーウィン流の淘汰に有利な突然変異についてで、このようなものの存在を決して否定するわけでないが、中立説では、有利な突然変異はごくまれにしか起こらないので、通常の分子進化速度を扱うには無視してもさしつかえないと仮定する。したがって、中立説では突然変異を淘汰に中立な(すなわち、有利でも不利でもない)突然変異と、淘汰に不利な(有害な)突然変異の二種類に分けて考える。』『生物進化を考える』木村資生、岩波新書、一九八八年、二二八頁

かといって、中立説が大進化の説明に対して無力かというと、そうではない、と主張しています。再び同書から引用してみましよう。

『ところで、大進化に直接寄与する要因は何であろうか。…(中略)…過去の進化の道すじのうちでとくに目ざましい進化の例を二つ上げると、一つは前カンブリア紀における多細胞生物(とくに動物)の出現と、もう一つは新生代とともに始まった哺乳類の適応放散である。前者では多細胞生物という新しい生命形態への道が開かれ、海は高等生物進化のための一大実験場と化した。後者では恐竜の絶滅により、それらによって占められていた各種の生態的住所が空いてしまい、そこへ生き残った哺

乳類が爆発的に放散していった。要するに、新しい生活環境の出現である。…(中略)…したがって、大きな進化的変化を起こすためには、まずそれから解放されることが前提条件である。これを筆者は「淘汰的制約からの解放」と呼びたい。』(二五六頁)

そして、進化の素材となる遺伝的変異の出現機構について、その章の最後で次のように締めくくっています。

『…(前略)…新しい機能をもつ分子の出現がどのような仕組で起こるかの説明は今後の分子進化学や集団遺伝学にとっての重要な研究課題である。この場合も、遺伝子重複につづく中立進化だけでなく、ダーウィン流の自然淘汰が働くことは確実であるが、問題はどんな働き方をするかである。今後の研究の発展を期待したい。』

(二五九頁)

つまり、淘汰的制約から解放された時期に、各個体は遺伝子重複と中立突然変異によって遺伝的多様性を獲得し、淘汰の圧力が強くなった時期に、その世代の環境に最も適した個体を選別されると考えるのです。

12

紙とエンピツでできた機械

チューリング機械

前章で述べた遺伝子重複と中立突然変異の機構を機械システムの進化機構に取り入れることによって、前の世代の機能を継承しつつ、進化する機械を構築してみよう。その前に紙とエンピツでできた機械の設計の仕方を説明しておきましょう。

まず、ビデオテープデッキのようなものを思い浮かべてください。機械は、入出力用のテープを読み書きするためのヘッドと情報処理機構を持っています。説明の便宜上、この機械はビデオデッキとは異なり、テープは真っ直ぐ両方向に伸びていて、機械の方が動きながらテープに読み書きを行なうものと考えて、機械はテープ上を両方向に移動できるものとしめます。テープの残りを気にしなくてもすむように、両方向に十分長いものとしましょう。もし、書き込む必要がないときは、話を簡単にするために、読み取っ

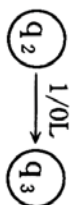
た記号と同じ記号を書き込むものとしましょう。テープはあらかじめフォーマットされているものとします。つまり、テープ上にデータを読み書きするときの升目がちゃんと引いてあり、とくにことわりがないときは、すべての升目にゼロが書き込んであるとしましょう。そして、機械は最初、あらかじめ決めておいたテープ上の読み取り開始位置にヘッドをつけているものとします。このような機械をチューリング機械と呼びます。

そして、これはシステム一般に言えることなのですが、システムは外界から入力を受け取って、外界に出力を出して応答するとき、同じ入力に対して常に同じ出力を出すとは限りません。たとえば、ソロバンや電卓ならば、足し算するときの各桁の結果は、下桁からの繰り上がりがあるかどうかによって異なります。この場合、桁上がりの有無によって、状態を二つに分けることができます。人間の場合なら、同じ食べ物を見ても、空腹時と満腹時とは感じ方が異なります。このようにシステムの状態は、過去の履歴によって異なります。したがって、システムを考えるとときには、外界との入出力の他に、システムの取りうる状態もいっしょにして考えます。

システムの最初の状態を初期状態といいます。そして、機械は終状態と呼ばれる状態に達したら停止するものとします。機械はテープ上の記号に応じて、状態を変えます。この状態が遷移するとき、テープに記号を書き込んで、左右のどちらかの升目に一つ

だけ移動します。次のようにして、チューリング機械のプログラムを行ないます。

このプログラムの書き方はいろいろありますが、状態遷移図と呼ばれる書き方がもっとも直感的でわかりやすいかと思います。状態を丸で表わして、それぞれを区別するために丸の中に適当な記号を書き入れます。ここでは、初期状態を q_0 、終状態を q_F と書くことにしましょう。それ以外の途中の状態は、 q_1 、 q_2 、 q_3 、 $\vdots$ と表わすことにします。最初は、機械をあらかじめ決められたテープの読み取り開始位置に置いて、状態を初期状態にセットしておきます。そして、テープ上の記号を読み取って、入力された記号と現在の状態に応じて、出力記号をテープに書き込んで、左右どちらかに移動した後、次の状態に遷移する、という動作を停止するまで繰り返します。たとえば、現在の状態が q_2 でテープ上の記号 1 を入力したとします。次の動作で、その記号を 0 に書き換えて、テープ上を一升左に移動した後、状態 q_3 になるものとしましょう。このときは、真ん中に q_2 と書いた丸から q_3 と書いた丸に矢印を引きます。そして、矢印の横に、1/0L と書き添えておきます (L は左方向への移動を表わします)。



これは状態 q_2 から入力 1 を読んで状態 q_3 に移るときに、記号 0 を書き込んで、左に移

動しろ、という指示を意味しています。右移動の場合はRです。このような、状態遷移の矢印を、各状態間にすべて書き込めば、プログラム完了です。

ここで、この機械に慣れただくために、紙と鉛筆だけを使って、ちょっとプログラムしてみましょう。簡単な例題を出しますので、答え(図13)を覗く前にチャレンジしてみてください。まず、テープは先ほど説明したように、読み取り開始位置の両方向に無限に続いているものとしてください。そして、テープは一升ずつ区切ってあって、あらかじめすべての升目にゼロが書き込んであるものとしましょう。

さて、ここで問題です。読み取り開始位置の左右のどこかの升目に記号1を一つだけ書き込みます。左右どちらなのか、またどれくらいの距離かはわかりません。機械をうまくプログラムして、この1を見つけてください。ただし、ちゃんと有限回の動作で終わるようにプログラムして下さい。つまり、機械は初期状態から出発して、有限回の動作の後に、1を見つけたら終状態に遷移してちゃんと停止して下さい。

したがって、たとえば、図12に示したような、いちかばちかのようなプログラムではだめです。このプログラムを解読してみると、 q_0 と書いた初期状態から出発して、0を読んだら、そのまま0を書き込んで左へ移動して、またもとの状態 q_0 に戻っています。そして、ひたすら左に移動していったら、偶然1が見つかったら、1を書き込んで左

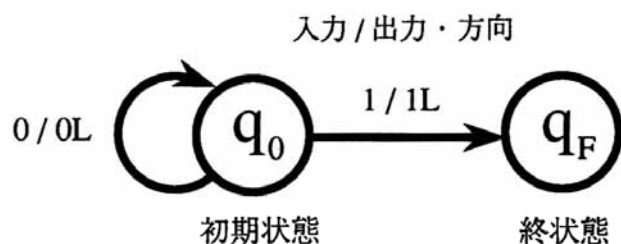


図 12 ひたすら左の 1 を探すプログラム

へ移動して、終状態に移動して停止します。もし運わるく、1 が読み取り開始位置の右にあれば、いつまでも 1 を探して左に移動し続けるので、停止することはありません。どうも、バグ（プログラムミス）のようです。状態は必要に応じていくらでも増やして構いません。また、記号の種類もいくらでも使って構いません。しかし、できるだけ少ない状態数と記号で書かれたプログラムほどエレガントです。次の頁をめくる前に、もう

ちょっと考えてみてください。さあ、できましたか？ もし、答え

（図 13）を見てもよくわからないときは、具体的に動作を一つ一つ追ってみてください。左右にジグザグに行ったり来たりしながら、徐々にその探索範囲を拡げていることがわかります。また、すでに調べたところをマークするために、新たな記号 2 を用いています。

もし、簡単すぎて物足りない方は、任意の数の掛け算のプログラムに挑んでみてください。ただし、数は単進法で表わすことにします。つまり桁上がり概念は使いません。幼児のように、たとえば 14 という数字は、1 を一四個升目に並べるものとします。やはり、あらかじめテープはフォーマット済みで、すべての升目に 0 が書き込んであるとします。問題として、テープ上に二つの数が与えられているとしま

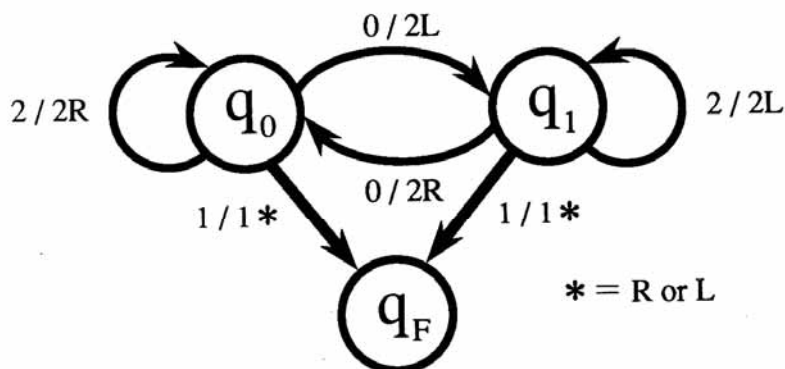


図 13 1を見つける正解プログラム

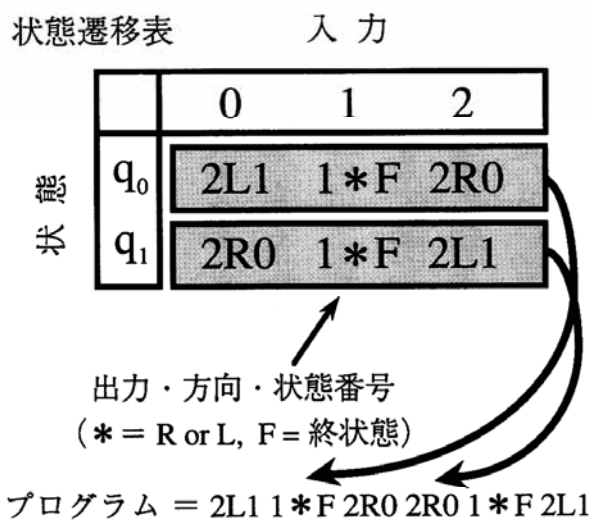


図 14 1を見つけるプログラム(状態遷移表)

す。たとえば4と7の場合、1が四つ並んで書いてあって、一つ0をはさんで、その右に1が七つ並んでいます。読み取り開始位置は、この真ん中の0にしましょう。そして、プログラムが終了したら、4と7を表わす二つの1の並びの右に一つ0をはさんで、二八個の1が並べば正解です。そして、この4とか7をいくつにしても、正解が得られるように

プログラムしてみてください。

今考えたチューリング機械は、データはテープ上であって、プログラム(状態遷移図)は別に存在していました。データとプログラムを別物として扱っています。しかし、この状態遷移図は、図14のように表にすることもできます。それぞれの対応に気をつけて下さい。遷移表では、遷移直前の状態と入力、出力と移動方向と遷移先の状態番号を並べて書き入れます。この対応をあらかじめ約束しておけば、表の枠組みをとり除いて、中の記号列だけでプログラムを表現することができます。たとえば、一行目の記号列の次に二行目の記号列を繋ぐというふうにすれば、このプログラムもデータのテープ上に書き込んでおくことができます。

プログラムの停止問題

実は、万能チューリング機械というものがあって、これを使うとテープ上に書かれた、任意のデータとプログラムを実行することができます。紙と鉛筆だけで計算する場合は、スピードこそ現在の計算機とは比べ物にならないほど遅いのですが、計算できる関数の種類という意味では同じパワーを持つのです。六〇年近くも前に、このような計算機のモデルを考え出したチューリングはやはり天才だったのでしょう。

しかし、現代の計算機でも同じことですが、プログラムがちゃんと停止するかどうか、という問題が残ります。これも、計算機に判断させようと考えると、ちょっとややこしくなりますが、任意のプログラムとデータが与えられたときに、それがちゃんと停止するかどうかを判定するプログラムは存在しない、ということが証明されています。実際のコンピュータでは、暴走したな、と思ったらそのプロセスを強制的に止めるか、最悪の場合はリセットすることになります。

能力は高くても、いつ計算が終了するかわからないモデルでは、これを少しずつ進化させて賢くしようと思っても、それぞれの個体の計算が終了していないときには、いつ打ち切ってしまうのかわかりません。適当な打ち切り時間を設定して、そこで終了していないのんびり屋の個体は脱落させてもよいのですが、ここでは、もう一つ別の方法を取ることにします。

13

言語を識別する機械

オートマトンという機械

能力は落ちますが、チューリング機械に制限を加えて、もう少し単純な機械を考えてみましょう。そこで、ヘッドの動きを片方向のみに限定することにしましょう。たとえば、この方向を右方向とすれば、読み取り開始位置から順に一つずつ読み取って、状態を遷移させて右に一升移動します。先ほどのチューリング機械では、状態遷移しているうちに終状態になったら停止しました。しかし、この制限した機械では、入力文字列の最後に停止記号#が書き込んであるものと考えます。このようにすると、入力記号列を一度読み終わった後、#を読んで必ず停止します。そして、停止したときの状態が、終状態であるかないかで、入力文字列を二つに分けることができます。終状態で停止したときには、その文字列を受理した、といえます。終状態は必要に応じて複数個設けても

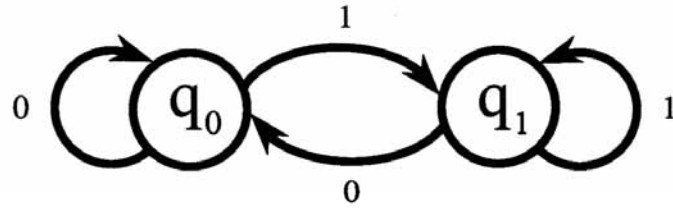


図 15 記号列を識別するオートマトン

構いません。このように制限した機械をオートマトンと呼びます。

オートマトンは、テープに文字列が与えられるたびに、初期状態から出発して、それを読んだ後、受理したかどうかを返事します。つまり、このオートマトンにとっては、受理する文字列の集合と、受理しない文字列の集合があつて、それを識別していることになります。このような文字列を単語、単語の集合を言語と呼ぶことにすれば、オートマトンは、ある言語に属している単語のみを受理する、ということができます。

たとえば、図 15 のような状態遷移図を持ったオートマトンを考えましょう。初期状態 q_0 と終状態 q_1 しかない 2 状態のオートマトンです。このオートマトンに 011# という単語を入力してみましよう。読み取り開始位置は、左端の 0 です。最初のステップで、初期状態 q_0 で入力 0 を読むと状態 q_0 に戻ってきます。この状態 q_0 で次の入力 1 を読むと状態 q_1 に遷移します。この状態 q_1 で次の入力 1 を読むと状態 q_1 に戻ります。最後に停止記号 # を読んで停止します。このときの状態は、終状態 q_1 なので、この単語を受理した、という返事を返します。今度は違う単語で試してみましよう。同様に一つずつステップを追っ

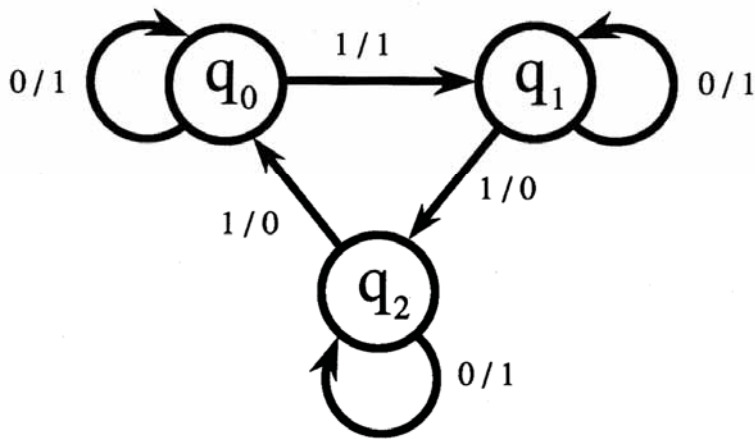


図 16 次に入力される文字を予測するオートマトン

てみると、10110# という単語を同じオートマトンに入力したときには、停止したときの状態が初期状態 q_0 であることがわかります。したがって、この入力に対しては、受理しない、という出力をだします。実は、このオートマトンは、2 進数の奇数を受理して、偶数を受理しないようにできています。一般に状態数を増やすことにより、より複雑な言語を受理できるようになります。

上記のオートマトンをもう少し変更してみましょう。そして、特に終状態は考えないことにしましょう。図 16 を見てください。初期状態 q_0 から始めて、今度はテープ上の記号列を左から順に読んでいくものとします。もう状態遷移図の見方は、おわかりかと思いますが、たとえば状態 q_1 にいたときに入力 1 を読み取ると、テープに 0 を出力して一升右に移動したあと状態 q_2 に遷移します。今ここで、01101 という長さ 5 の記号列を繰り返す単位とした文字列を入力テープに書き込んでおきます。

表2 記号列 01101 を繰り返し単位として入力した
ときの状態遷移

前の状態	q ₀	q ₀	q ₁	q ₂	q ₂	q ₀	q ₀	q ₁	q ₂	q ₂	...
入 力	0	1	1	0	1	0	1	1	0	1	...
出 力	1	1	0	1	0	1	1	0	1	0	...
後の状態	q ₀	q ₁	q ₂	q ₂	q ₀	q ₀	q ₁	q ₂	q ₂	q ₀	...

突然変異をおこすオートマトン

これをこのオートマトンに入力したら、出力はどうなるでしょうか？すでに読者は、この程度のことであれば簡単に答えられるでしょうが、一応答えを表2に書いておきます。一段目が状態遷移する前の状態、2段目が入力記号、3段目が出力記号、4段目が状態遷移した後の状態を表わします。

ここで、入力と出力をよく見比べると、ある時点での出力が、まだ読み取っていないはずの、次の入力記号になっていることに気がつきます。つまりこの機械は、この記号列に対しては、一つ先を正確に予測しているのです。もちろん、他の文字列に対してはでたらめな出力になってしまいます。

そこで、逆問題を考えてみましょう。まず最初に適当な文字列を入力テープに書き込んでおきます。そして、左から順に読んで、次の升目に書いてある入力記号を予測して出力します。これを繰り返して右に移動していくものとしましょう。このときの予測率

が最大となるようなオートマトンをできるだけ少ない状態数で実現する問題です。ちょっとやってみると、意外と難しいことがわかります。フォーゲルらは、一九六七年にこの問題を人工的な進化システムで解いてみることにしました。乱数を振ってオートマトンを作り、成績がよいものを選んで突然変異を起こさせる、というプロセスを繰り返すのです。突然変異の種類は、遷移先の変更、状態の追加、状態の削除、などからなっていました。また、これらの操作はまったくランダムに行なわれたため、追加や削除によって、オートマトンの構造が大きく変わってしまうものでした。

また、オートマトン理論を用いた、オートマトンどうしの合成という操作もありましたが、オートマトンの状態遷移表を染色体上にコードするという概念や、相同染色体上の交叉といった操作はありませんでした。残念ながら、進化論や遺伝学的な知見を、このようなシステムに導入する段階には至っていませんでした。また、予測すべき文字列も、左から数えて素数番号に1が書いてあるものなど、本来オートマトンでは予測不可能な例題なども扱われていました。しかし、この機械のモデルを進化させるというアイデアは、実に興味深く思われます。そこで、もう一度問題を設定しなおして、今まで述べてきた生物学的知見を取り入れてみることにします。

14

進化する機械を設計する

進化システムとしてのオートマトン

ここでは、機械のモデルとして、前章で述べた言語識別用のオートマトンを用いることにします。そして、すでに説明したように、あるオートマトンがあったときに、その状態遷移図から、状態遷移表を書き起こします。そして図17のように、状態遷移表の枠を取り除いて、中の遷移データだけを連結して一本の記号列を作ります。この操作をオートマトンのコーディングと呼び、その記号列をオートマトンの染色体と呼ぶことにします。逆にオートマトンの染色体のデータから、オートマトンの状態遷移図を作成することを、デコーディングと呼ぶことにします。

前章では、状態を表わすのに q という文字と状態番号の下添え字を使っていました。ここでは、状態番号だけにします。そして、初期状態を状態1、終状態を状態0、

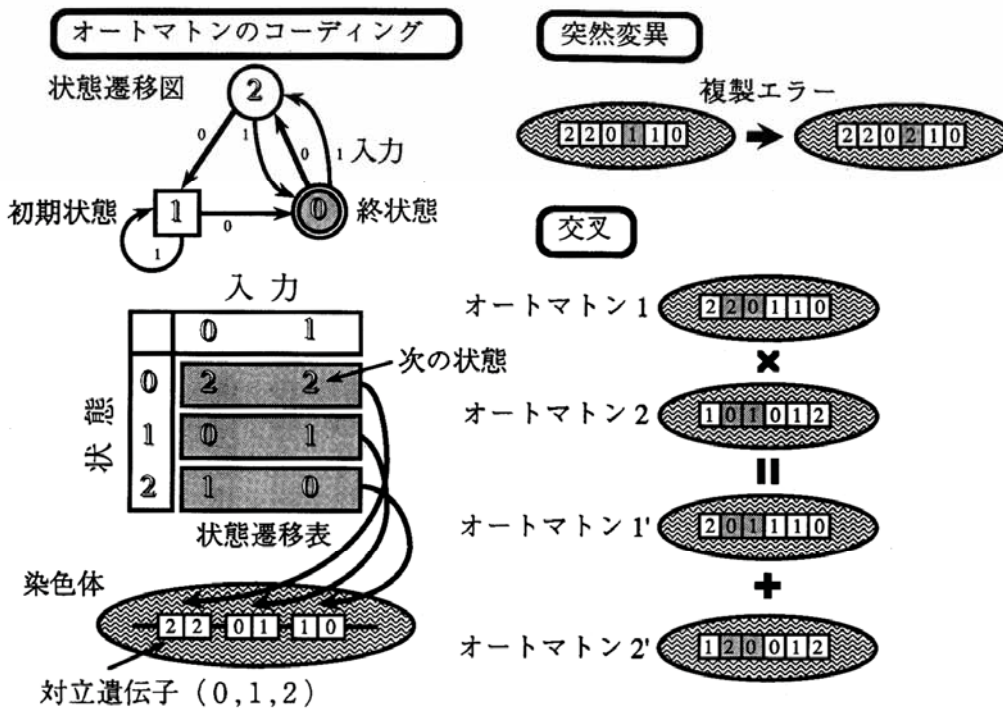


図 17 オートマトンのコーディングと遺伝子操作

それ以外の状態を、2、3、4、…で表わすものと約束しておきます。したがって、図17のような3状態のオートマトンの染色体の長さは6となります。各染色体の遺伝子座には、それぞれの状態遷移先を意味する遺伝子が乗っています。3状態なので、各遺伝子座での対立遺伝子には0、1、2の3種類があります。一倍体では、遺伝子型がそのまま表現型となります。また、一本の染色体にコードしたので、ゲノムはこの染色体一本のみです。

突然変異は、図17に示したように、各遺伝子座で乱数を振って、あらかじめ決められた突然変異率に従って、その座位が突然変異するかどうかを決定

します。変異する場合には、もう一度乱数を使って変異先を決めます。

図17に示したように、交叉も7章で述べた方法に従って行ないます。交叉は特にオートマトンの場合は、構造が大きく変わる可能性が高くなります。

これで、ほぼ準備は整ったので、問題設定に移りましょう。欲を言えば、工学的にも意味があつて、かつ今まで生物学的に解説してきた内容と関連があり、さらにオートマトンの進化のようすが客観的に判断しやすければ言うことはありません。そんなに都合よくいくかどうかわかりませんが、次のように考えたらどうでしょうか。

大腸菌などの細菌は培地からグルコースなどの糖分を吸収して、分裂増殖します。また、グルコースはラクトースを代謝することでも得られます。したがって、ラクトースのみの環境に切り替えても、大腸菌はラクトースを細胞内に取り込んで、これをβ-ガラクトシダーゼという酵素で加水分解することによって、グルコースを得ることができます。ラクトースの取り込みを促進するための酵素や、加水分解する酵素は、大腸菌の遺伝子にコードされていて、その遺伝子の発現は、正負両方のフィードバック機構によって制御されています。ラクトースがないときには、無駄に酵素を合成しないようになっています。また、これらの遺伝子に突然変異が入ると、ラクトースを正常に分解代謝できなくなるので、ラクトースのみの環境では増殖することができません。

また、細菌はある抗生物質に対して耐性を獲得すると、それとよく似た化学構造と作用機構を持った抗生物質に対しても耐性を獲得することがあります。これを交差耐性と言います。そして、すでに説明したように、プラスミドやトランスポゾンを用いて、複数の薬剤に対して同時に耐性を獲得することもあります。

これらのことを何となくイメージして、前章の言語を認識するオートマトンをもう一度眺めてみることにしましょう。たとえば、ある文字列を一つの栄養素に見立てましょう。オートマトンはこれを内部に取り込んで処理したあと、この文字列が栄養素として利用できるかどうかを試すものとします。利用できるかどうかは、その文字列を読み終わったときに、状態が終状態で終わったかどうかで判定するものとしましょう。

すると、いくつかの栄養素を含む培地に、このオートマトンを入れると、オートマトンはそれぞれの栄養素を利用できるかどうかを試します。したがってオートマトンの構造が異なれば、利用率が異なることになります。より利用率が高いオートマトンの評価値が高くなります。薬剤耐性の場合、逆のイメージになります。抗生物質によって、内部の代謝系を破壊されないように、耐性を獲得するわけですから、培地に与えられた文字列を受理したら、評価値が下がるものとしましょう。

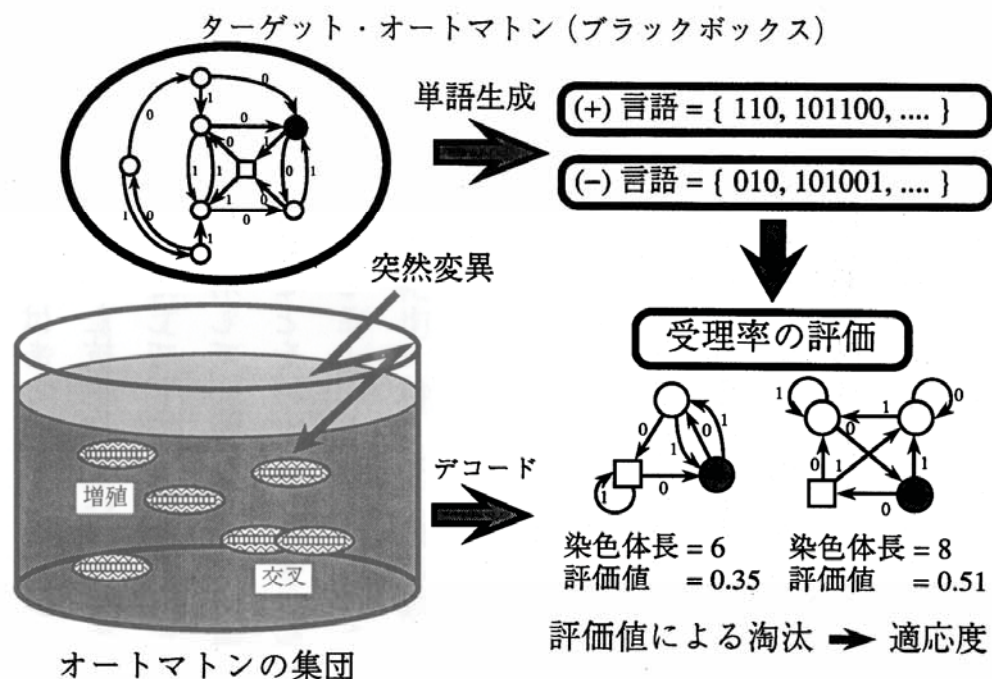


図 18 オートマトンの進化

環境に適応するオートマトン

次は文字列の集まりである培地、つまり環境の設定です。適当な文字列にしてしまっても良いのですが、もうひと工夫してみましょう。図18を見て下さい。培地の文字列は図18(左上)のようなオートマトンから作られているものと考えてみましょう。つまり、このオートマトンが受理する言語の中から適当な文字列を選んで、培地の文字列とします。

このためには、言語の認識に用いたオートマトンを逆に動作させます。最初は初期状態から始めます。各状態からは二本の矢印が出ていて、それぞれ0と1のラベルが付いています。前とは異なり、どちらの矢印を選ぶかは、乱数を振って

確率二分の一で決めます。このようにして選んだら、矢印に付いているラベルを出力して、矢印の先の状態に遷移します。つまり、入力は必要なくて、確率的に自律的動作をしながら、出力を順次出していきます。これではいつまでも遷移し続けるので、一つの文字列を生成する前に、毎回乱数を振って、その文字列の長さを決めておきましょう。この長さを n とすると、このオートマトンは、 n 回だけ確率的な状態遷移を行なって、長さ n の文字列を出力した後、どこかの状態で停止します。この状態が終状態に含まれていたら、それを正の単語と呼ぶことにします。逆に停止したときの状態が終状態に含まれていないときには、その文字列を負の単語と呼びます。

この出力用オートマトンを前のように、言語の認識用のオートマトンだと考えれば、同じものを逆に見ただけですから、上記の正の単語を入力すれば、終状態で停止して、これを受理するし、負の単語を入力すれば受理しません。つまり、出力用オートマトンは、受理する言語と受理しない言語の中から、それぞれ正の単語と負の単語をサンプルしていることになります。そして、正の単語と負の単語がほぼ同数になるように、サンプルしたものを培地に与えておくことにしましょう。

集団の大きさは、遺伝的アルゴリズムGAのときと同じように各世代ごとに一定とし、まず、状態数は乱数を生成します。ランダムに生成します。まず、状態数は乱数を

振って決めます。これを1とすると、0から100までの乱数を振って、染色体の遺伝子として順番に書き込んでいきます。これでオートマトンの染色体が一つできました。前に説明したように、この染色体をデコードすると、オートマトンができあがります。このようにして、さまざまな長さの染色体を持ったオートマトンが生成されます。状態数の異なるオートマトンどうしは、染色体長が異なるので、交配することができません。互いに交配可能な集団をメンデル集団といいます。オートマトン集団は状態数の違いによって、異なったメンデル集団を構成します。同じ染色体長を持ったオートマトンどうしの競争を種内競争、異なった染色体どうしの競争を種間競争と呼ぶことにします。それぞれのオートマトンは、培地の単語のサンプルを一つずつ拾ってきて入力します。正の単語のときに受理した割合と、負の単語のときに受理しなかった割合とをそれぞれ計算します。そして、そのオートマトンはその割合を掛け合わせたものを評価値として与えられるものとします。この評価値によってGAのところで説明したような方法で淘汰を行います。

一つの世代ではすべてのオートマトンが共通のサンプルの集合を入力するものとし、次の世代では、出力用オートマトンは、乱数を振り直して新たなサンプルの集合を生成するものとします。しかし出力用オートマトンの構造は同じですから、サンプルの

集合は異なっている、同じ生成規則に従っています。集団の中のオートマトンたちには、この出力オートマトンの状態数や遷移構造は目隠しされていて、サンプルだけが与えられます。このブラックボックスのオートマトンをターゲット・オートマトンと呼ぶことにしましょう。上で述べたように、ターゲット・オートマトンを言語の認識装置と考えれば、文字列サンプルを入力、受理したかどうかという応答を出力、と考えることができます。このような見方をすると、オートマトンたちは、ターゲット・オートマトンの入出力応答だけから、その状態数と遷移構造を推定しているように考えることができます。これをオートマトンのシステム同定と呼びます。

状態数がわかっているときには、染色体の長さはすべて一定です。このときは組合せ最適化問題と同じように、文字列の並び方だけしか問題にならないので、組合せの数は有限です。この中からより良いものを選び出すだけです。GAのところで説明したように、このような静的な問題設定はあまりおもしろくありません。

しかし、このターゲット・オートマトンは状態数がわからないので、探索すべき領域は無限です。2状態できてきているかも知れないし、100状態できてきているかも知れません。したがって、あらかじめランダムに状態数を決めておいても、それでは足りないかも知れません。これを解決するために、すでに述べた遺伝子重複の考え方を導入します。遺

伝子重複で大事なことは、遺伝子が重複しても、基本的には前の機能を変えないことにあります。そこで、オートマトンの場合は、図19(上)のように挿入操作を行なって、遺伝子重複を生じさせます。乱数を振って、どれかの状態を選んだら、その状態を重複させてコピーします。重複前の構造が変わらないように、図19(上)の右図のように状態の番号を振り直します。状態遷移図を見てもわかるように、新たに追加された状態3は、状態2のコピーであることがわかります。この時点では、状態3はどのような入力を受けても、初期状態1からたどり着けないので、このオートマトンはあってもなくても同じことで、オートマトンの状態としては機能していないことになります。このため、遺伝子重複後もまったく同じ動作をします。

このように初期状態から到達不可能な状態を、非機能状態、初期状態から到達可能な状態を、機能状態と呼ぶことにします。そして、染色体上で機能状態をコードしている遺伝子の集合を、コード領域、非機能状態をコードしている遺伝子の集合を、非コード領域と呼ぶことにします。

すでに説明したように、突然変異の操作が非コード領域に生じてても、動作は変わりません。しかし、コード領域に変異が入ると動作を変えるため、一般に性能が変わります。このとき、機能状態の遷移先が非機能状態に変異すると、その時点から非機能状態は、

挿入

((例)) 挿入位置 = 2

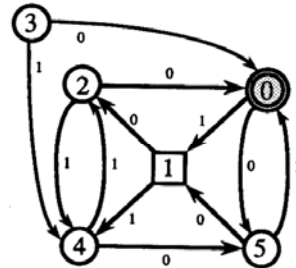
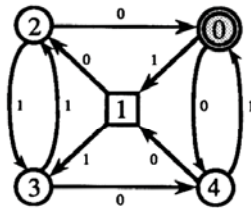
入力		
	0	1
状態 0	4	1
1	2	3
2	0	3
3	4	2
4	1	0

フレーム拡張
+
下方向コピー

入力		
	0	1
状態 0	4	1
1	2	3
2	0	3
3	0	3
4	4	2
5	1	0

if 状態 > 2 then
状態 ← 状態 + 1

入力		
	0	1
状態 0	5	1
1	2	4
2	0	4
3	0	4
4	5	2
5	1	0



欠失

((例)) 欠失位置 = 3

入力		
	0	1
状態 0	5	1
1	2	4
2	0	4
3	2	4
4	5	2
5	3	0

上方向コピー
+
フレーム圧縮

入力		
	0	1
状態 0	5	1
1	2	4
2	0	4
3	5	2
4	3	0

if 状態 >= 3 then
状態 ← 状態 - 1

入力		
	0	1
状態 0	4	1
1	2	3
2	0	3
3	4	2
4	2	0

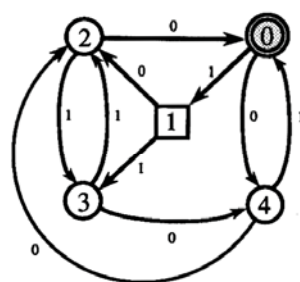
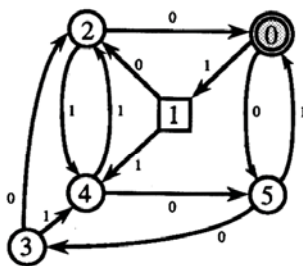


図 19 オートマトン遺伝子の挿入と欠失

初期状態から到達可能になるので、機能状態に取り込まれます。たとえば、図19(上)の右図の状態4は0を入力すると状態5へ遷移しますが、この遷移先が状態3に変異すると、この状態3は初期状態1から到達可能となり、機能しはじめます。

進化の段階が進んでくると、コード領域の変異は、オートマトンの構造を大きく変えて、変異前の性能を下げる有害変異となる可能性が大きくなります。このため、コード領域は、次第に非常に変化しにくい保存領域となります。また、非コード領域の変異は評価値を変えないために、淘汰に対して中立な変異となり、突然変異率に比例して変異の蓄積しやすい可変領域となります。

遺伝子重複とは逆に遺伝子の欠失の操作も考えましょう。図19(下)のように二つの状態を圧縮する操作です。コード領域に欠失操作が入れば、一般に動作が変わります。しかし、コード領域に欠失が生じて、状態数の少ないオートマトンになっても、まったく動作が変わらないこともあります。任意の入力に対して動作が同じ場合、この二つのオートマトンは等価であるといえます。したがって、欠失操作を導入することにより、ターゲット・オートマトンよりコンパクトで等価な個体が現われる可能性があります。また、オートマトンの染色体の長さが長いほど突然変異が生じやすいために、同じ性能である場合にはコード領域の長さが短いほど安定に存続できます。このように遺伝子長に

関する淘汰が働くことにより、より状態数の小さいオートマトンが淘汰によって生き残る場合があります。このような変異の繰り返しにより、オートマトンはゲノムサイズを変えながら、構造的に複雑化し、機能的にも適応進化して、ターゲット・オートマトンに近づいて行きます。また、このシステムでは、中立説の説明で述べたように、遺伝的多様性を維持するため、淘汰強度の強い時期と弱い時期を周期的に変動させています。ここで、淘汰の強さとは、GAの説明箇所です。

オートマトンの集団が環境(ターゲット・オートマトンの出力文字列)に100%適応することができた場合は、さらにターゲット・オートマトンの状態数を増やして、環境条件を次第に複雑で厳しいものにしていきます。

劇的な進化をとげるオートマトン

実際のシミュレーション結果が図20と図21に示してあります。各状態数のオートマトン集団の最大評価値と集団サイズの世代変化がグラフに示されています。集団の大きさは200匹です。初期世代から200世代までは、5状態のオートマトンを、それ以降は3状態追加した8状態オートマトンをターゲット・オートマトンとして用いています。これを見ると、種から種へのきわめて急激なシフトが何度か生じていることがわかり

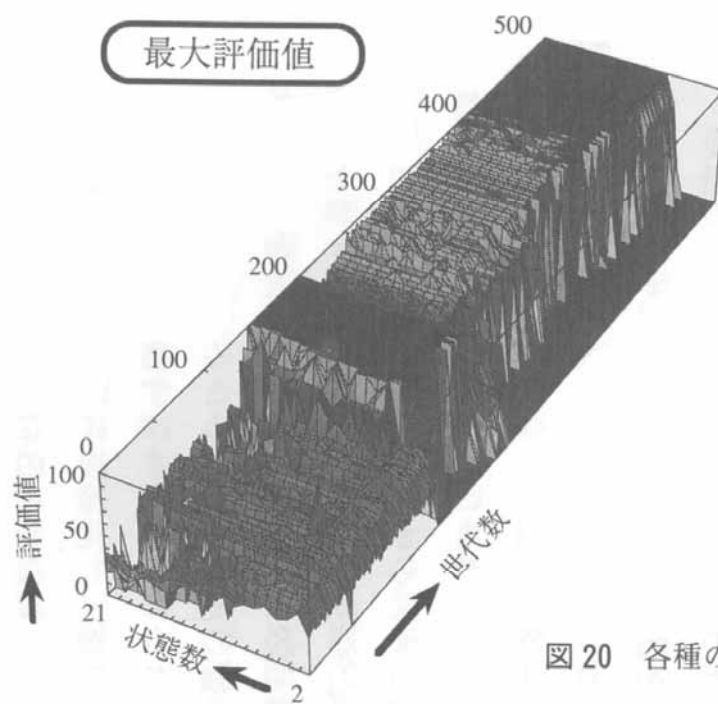


図 20 各種の最大評価値

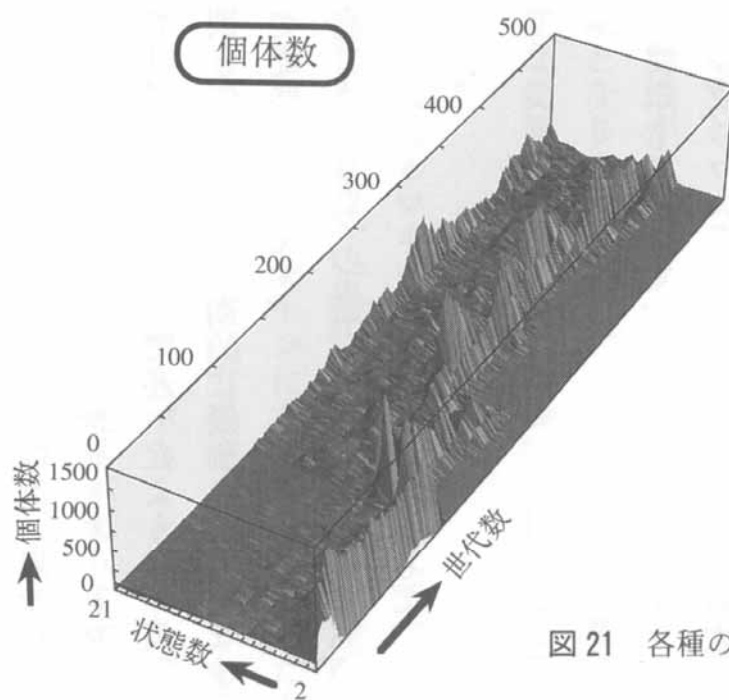


図 21 各種の個体数

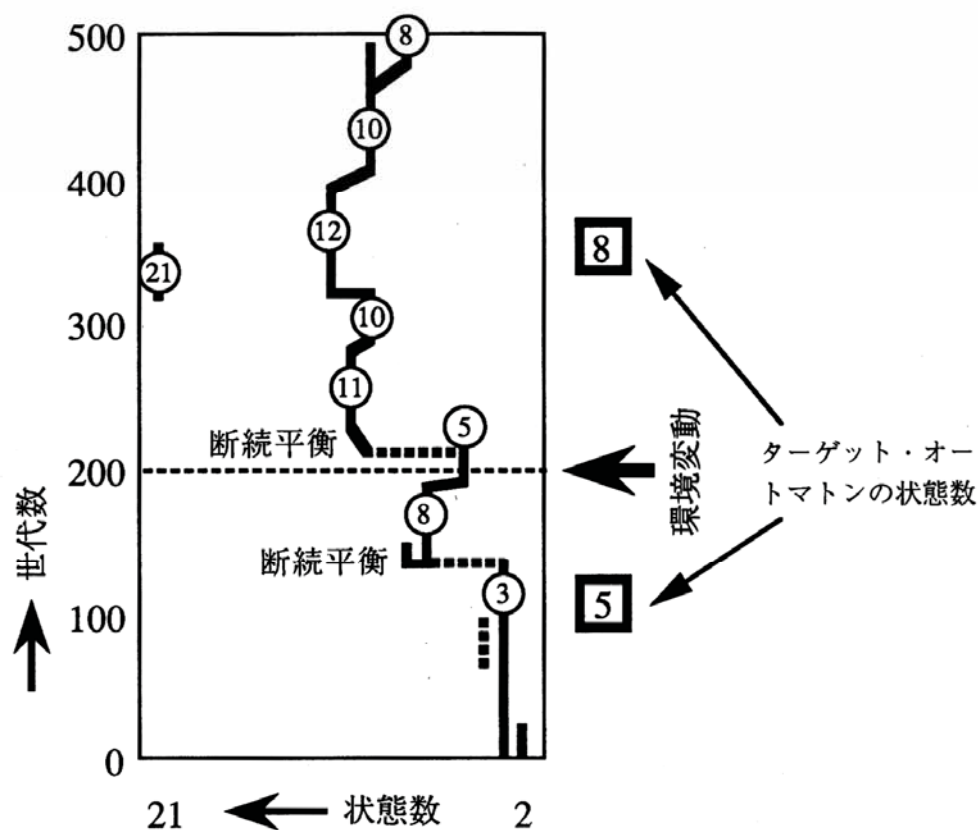


図 22 オートマトンの進化系統樹

ます。生物進化の足跡を検証するために、化石を採取して形態学的特徴を比較する方法があります。同様に、オートマトン集団から各世代のオートマトンをランダムにサンプルしたら、図22のような断続平衡的な進化系統樹が得られました。この図の線は、図21を上からみたときに、山のピークを結んだ線と一致します。

この図22とオートマトンの遺伝子を見比べてみると、徐々に遺伝子重複していった個体の中から、世代130において状態数8の評価値一〇〇%の個体が現

われ、この種が圧倒的に優位となったことがわかります。その後、遺伝子欠失などにより、世代180において評価値一〇〇%でかつ最もコンパクトな5状態オートマトンが出現して、この種が圧倒的に増殖した結果、集団がこの種にシフトしています。

世代200において、ターゲット・オートマトンの状態数を増やし複雑化したために、評価値の劇的な減少が起きていますが、再び急激なシフトを生じて11状態の種に移行しています。その後、種間競争を繰り返し、世代360において状態数が11と12の種に評価値一〇〇%の個体が現われています。そして、世代500では最もコンパクトな8状態オートマトンが優位となっています。

生物とは異なり、機械のシミュレーション進化では、遺伝情報と構造、機能などがどのように進化してきたのかという過程を詳細に追跡することができます。また、乱数を振り直すことによって、違った進化プロセスを観察することもできます。このようにして、急激なシフトが生じている世代の前後のオートマトン集団の遺伝子を解析することによって、遺伝子重複や中立変異、そして淘汰がどのように関係して、シフトが生じたのかを理解することができます。

遺伝子重複は、本来の遺伝子の機能を損なわずに、新たな機能遺伝子を試す場を提供しています。そして、非コード領域に生じた中立的な突然変異が、遺伝子プールに蓄積

していることがわかります(図23)。このような遺伝的多様性があるからこそ、新しい環境の下で非機能状態が機能状態に取り込まれるような変異が生じた個体のなかから、淘汰によってより適応能力の高い個体が選択される可能性が生まれてくるのです。このように、少なくとも人工的な進化システムにおいては、遺伝子重複、中立突然変異、淘汰といった進化機構が互いに補完する形で機能していることがわかります。

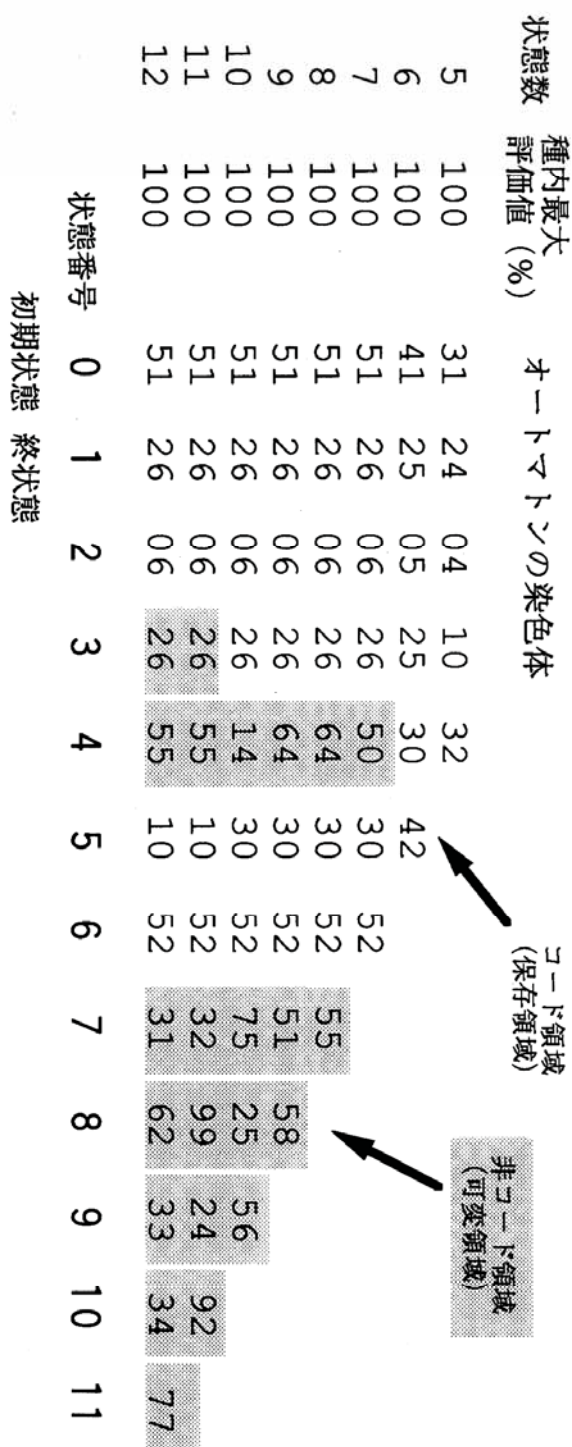


図23 オートマトンのコード領域と非コード領域(世代数 179)

15

エピローグ

生物学と情報科学は、ここ二〇年で急速に進歩を遂げてきました。両者ともに共通した特徴は、その応用の広さとビジネスへの密着性です。そして、誰もがあと一〇年後の研究テーマを正確には想像できないくらい、飛躍的なスピードで進展しています。

生物学は分子生物学によってその姿を大きく変えてきました。現在、分子生物学にとっては遺伝子データベースは必須の要素となっています。研究者が遺伝子をシーケンスしたら、必ずと言ってよいほど、遺伝子データベースを用いてホモロジー検索(塩基やアミノ酸の配列から見ても、類似性の高いものどうしを探す手法)を行ないます。これによって生物種をこえて相同性の高い領域が発見されることがあります。幸運にもこのような遺伝子が見つかった場合には、生物に共通の重要な機構が解明される可能性が高くなります。また、すでに述べたように、分子進化学は積極的にこの相同性を利用して

生物種間の分岐年代を推定します。これまでも、このようにしていくつかの重要な遺伝子が発見されてきました。

そして、あと数年で二一世紀を迎える今、生物学では、生物学始まって以来の大規模プロジェクトであるヒトゲノム解析計画が各国の協力体制のもとで進められています。このプロジェクトでは、三〇億塩基もあるヒトの全遺伝子と関連生物の遺伝子とを解読し、その制御機構や機能を解明することを目指しています。そのためには、遺伝子の読み取り技術のさらなる開発や、巨大データベースを扱うための生物情報学の研究などが不可欠となります。現在の生物情報学は、情報科学や統計解析学から生物学にとって必要な情報処理技術を利用するという立場に立っています。しかし、約一〇万個もある遺伝子の全貌を把握するためには、それとは異質の科学が必要になるものと思われれます。

コンピュータは、指数関数的にスピード、量ともに発展してきました。しかし、最近になってきて、そのスピードに陰りが見えてきました。そのため、必然的に並列コンピュータ、さらに超並列コンピュータへと向かおうとしています。情報科学もそれにとまって、新しい領域を模索しています。最近ブームになった、ニューラルネットワーク、ファジィ、遺伝的アルゴリズム、などはいずれも三〇年近く前に行なわれていた研究が再燃したものです。これらの研究は、適応性や柔軟かさなど、生物の持つ特長を利用

することを目指しています。しかし、人工生命も含めて、いずれも情報科学の分野に両足を置いたまま、生物学からちょっと美味しい所をつまみ食いする、といった姿勢が目立ちます。約六〇兆個の細胞からできている私たちの身体は、いうまでもなく超並列システムです。そして、その細胞のなかには約三〇億文字からなるプログラムが書き込まれていて、DNAや細胞など、さまざまなレベルで自己修復を行ないながら安定に機能しています。この驚異的なシステムを記述し工学的に応用するためには、片方の足を常にもう一方の領域に置いておくくらいの覚悟がなくては無理だと思っています。

古い学問体質のなかで育つと、なんとなく両端の近くが住みごごちが良いので安住してしまいがちになります。しかし、これからはプリズムのように、その両端の間を埋める幅広いスペクトルを持った人材が益々必要になっていくように思います。これからの若い世代の研究者たちが、このスペクトルを埋めることによって、まったく新しいパラダイムを持った分野が生まれるのではないかと期待しています。

あとがき

7章と8章では遺伝的アルゴリズムGAの特徴について述べましたが、GAはニューラルネットワークや数理計画法などと相反する択一的な手法ではありません。重要なことは、それぞれが得手不得手とする問題領域をはっきりと認識することです。

より包括的な進化システムでは遺伝機構の他に、発生、免疫、学習などの生物の適応機構を取り入れ、かつ工学的な最適化手法も必要に応じて組み入れることによって、互いの長所を生かしながら、より実用的なハイブリッドシステムを構築して行きます。

現在、私たちは生物のスプライシング機構を調べるために、学習と進化の機構を両方備えた人工的なシステムを用いて、数百個体に全遺伝子データベースを読み取らせて、より識別能力の高いシステムを生成する研究を行っています。紙数の都合や入門書という性格上、このような大規模なシステムや、発生や免疫についての話をする余裕がありませんでした。

そもそも、この本を書くことになったのは、一年半ほど前に『デジタル生命の進化―遺

伝・発生・免疫のアルゴリズム』という仮題で執筆依頼を受けたのがきっかけです。この「デジタル生命」という言葉に私は、多少羞恥心を感じてしまうので、「進化するシステムを設計する」という題にしたかったのですが、残念ながら聞き入れてもらえませんでした。

私も、コア・ウォーズというコンピュータゲームが流行っていた一九八四年頃には、コンピュータの中の生命などという夢に惹かれたことがありました。一九八六年に開かれた第一回国際コア・ウォーズトーナメントで優勝した、ウェンデル作のMICEという名前の自己増殖プログラムは実にうまくできていて、効率良く自分自身のコピーを作ってメモリ空間上を増殖していきます。その後も、多くのハッカーの興味を引き、国際コア・ウォーズ協会なるものができるほどの盛況ぶりでした。しかし、彼らが直接の原因ではないにしても、これが後にコンピュータ・ウイルスに変貌し、ビジネスでのコンピュータ利用の拡大とともに、深刻な社会問題となったことはご存知かと思います。

私には、デジタル生命の進化という点、どうしてもこの暗いイメージがつきまわってしまいます。最近では、エイズウイルスのように、自らのコードを変えながら伝染していくウイルスも出現したと聞きます。コンピュータウイルスのワクチンの開発は、すでにビジネスとしても成立していて、是非とも必要なセキュリティ技術です。しかし、これはセキュリティ技術ですから、少なくとも私ごとときにはわからないくらい高度なものであって欲しいと

期待しています。

UNIXのOSやネットワーク技術を完全に把握しているハッカーが、進化や免疫などの機構をウイルスに順次付け加えて、ほんのちょっとした出来心でそれを放そうものなら、本人も想像できないほどの厄介な状況を引き起こしてしまうでしょう。いや、そんなものとは違う、あの奇麗なグラフィックスの美しさがわからないのか！とおっしゃる方もいらっしゃるでしょう。一般に人工生命と言った場合には、広い意味ではコンピュータ・グラフィックスへの応用研究も含まれているようです。昔からあるLシステム（記号列を書換え規則によって書き換えることによって、樹木のように成長するパターンを生成するシステム）やフラクタルなどをもとにして、テクスチャマップやモルフィングなどさまざまな3D画像処理技術を応用したもので、その生成パターンをアニメーションしてみせることによって、生物っぽい動きを表現しています。バイオ・アートの美しさには魅力を感じます。

しかし、本物の生物の仕組みを知れば知るほど、そのギャップの大きさに気づきます。これが人工生命だ！と人が言っているのを聞いていても、ちょっと恥ずかしくなってしまう。遺伝的アルゴリズムという訳語も同様な意味で使っています。この「的」という字を外すには、まだ多くの難問が残っているように思います。この一字が外れたときには、生物科学も情報科学も今からは想像もできないほどに変貌を遂げるような気がしています。

最後になりましたが、この本の企画をしてくださり、一年半も辛抱強く待ち続けてくださった岩波書店の吉田宇一氏に感謝いたします。

一九九四年二月

和田健之介