

進化システム構成法に関する研究

下原, 勝憲

<https://doi.org/10.11501/3178994>

出版情報 : 九州大学, 2000, 博士 (工学), 論文博士
バージョン :
権利関係 :

4. 4 セルオートマトン型人工脳実験装置 (CBM: CAM-Brain Machine)

ハードウェア進化モデルとしてのCAM-Brainをハードウェアとして具現化するとともに、巨大なニューラルネットとしての人工脳構築の可能性を検証するための実験装置がセルオートマトン型人工脳実験装置 (CBM) である [11].

開発した CBM の装置の概観を図 4-14 に、仕様を表 4-1 に示す. 3次元セルオートマトン(CA)空間にハードウェアとして実装できるニューラルネットは最高で 1,152 個のニューロンを持つことができる. そのニューラルネットを 1 個のニューラルモジュールとすると, 3次元 CA 空間を時分割使用することにより 1 秒間でニューラルモジュール 64,640 個を実現できる. 従って, ニューロン数が総計で 7,450 万ニューロンの人工脳 (巨大なニューラルネット) を構築することができる. 1 個のニューラルモジュールは, 外部インタフェースを通じて, 最高で 188 の他のモジュールからの軸索出力信号を受信でき, 最高で 64,640 個の他モジュールへ信号を出力できる.

4. 4. 1 CBM ニューラルモジュール

CBM は CAM-Brain CoDi モデルを実装したものである. 1 個のニューラルモジュールを進化させるためには, 30~100 個のモジュールからなる集団系を用いて, 200~600 世代に渡って遺伝的アルゴリズム (GA) を実行しなければならない. 従って, 最高 60,000 回のモジュールが生成・評価されることになる. $24 \times 24 \times 24$ の 13,824 セルからなる 3次元セルオートマトン (CA) 空間に, まず, 染色体にコード化された成長情報に基づいてニューラルネットを形成する. 次に, そのニューラルネット (モジュール) にスパイク状の信号系列を入力として与え, 同様にスパイク状の信号系列として出力される処理結果と目標の信号系列とを比較することによってモジュールの性能を評価する. そのためには 1 個のモジュール毎に CA 空間の全てのセルが最高で 1,000 回程度状態更新される必要がある. 従って, 1 個のニューラルモジュールを進化させるために $60,000 \times 13,824 \times 1000 =$ 約 8290 億回の CA セルの状態更新が必要となる.

ちなみに, セルオートマトン型の高速計算のために MIT で開発された専用マシン CAM-8 の場合, 8290 億回のセルの状態更新を実行するのに約 70 分かかる. CAM-8 のもうひとつの限界は実行モード時のスピードである. 何千ものモジュールが相互に接続されているような脳の構造を想定するとその全体シミュレーションは非現実的なものとなる. 例えば, 10,000 モジュールからなる人工脳を想定した場合, CAM-8 では全てのモジュールの更新が 1.4 回/秒毎にしかできない. しかし, ロボットの実時間制御のためには, モジュールあたりで 50~100 回/秒, 全体で 10~20 回/秒の更新速度が必要である.



図4-14 セルオートマトン型人工脳実験装置 (CBM)

表 4-1 CBM の仕様

セルオートマトンの更新速度	最高 1,310 億回/秒
実現可能なセルオートマトン数	最高 893 百万個
モジュールあたり実装可能なニューロン数	1,152 個
実現可能なニューラルモジュール数	最高 64,640 モジュール
実現可能なニューロン数	最高 74,465,280 個
ニューラルモジュールの染色体長	91,008 ビット
ニューロンレベルの情報転送速度	最高 1.86 G バイト/秒
モジュール間の情報転送速度	最高 64 M バイト/秒
FPGA 数	最高 72
FPGA の再構築可能な機能ユニット数	1,179,648
表現型・遺伝型メモリ容量	1.18 G バイト
消費電力	1.5 K ワット (5V, 300A)
相当する計算機パワー (推定値)	10,000 x Pentium III 500MHz

そこで、CAM-8 に比べ 500 倍の高速化を図り、人工脳構築へ向けたシミュレーションを現実のものとするためにもハードウェア進化が必要となる。

CoDi モデルは 3 次元 CA 空間にニューラルネット（ニューラルモジュール）を実現するモデルである。各 CA セルは立方体で構成され、6 面を介して近傍の CA セルと相互に接続される。各 CA セルは遺伝子型に応じてニューロン、軸索或いは樹状突起セルとして機能する。ニューロンセルは、複数の CA セル (2 x 2 x 3) で構成され、5 個の樹状突起から入力を受け、1 個の軸索から出力する。ニューロンセル用の遺伝子によって、ニューロンセルの中には 4 ビットのアキュムレータを再構成し、それを用いて入力信号を合計する。その合計値がしきい値を超えたら発火・出力する。樹状突起入力は抑制性或いは興奮性のいずれかが同様にニューロンセル用の遺伝子によって決められる。ここで興奮性とはアキュムレータに信号を加算し、抑制性とは減算することを意味する。ニューロンセルの出力方向は、軸索セルが立方体の 6 面のどこに位置するかによって決められる。

樹状突起セルは 5 面からの入力を受け、その 5 ビットの XOR (Exclusive OR) をとり、その結果を 1 面から出力する。軸索セルは、樹状突起セルとは反対に、1 入力 5 出力であり信号を近傍セルへ分配する

上記のようなニューラルモジュールの成長は、遺伝子型メモリに保存されている染色体に基づいて、4. 3. 2 節で述べた手順により行われる。ニューラルモジュールの成長フェーズが完了した後、形成されたニューラルネットの各セルの種類と空間的な構造は符号化され、表現型メモリに保存される。多くのニューラルモジュールで 3 次元 CA 空間を時分割使用する際、表現型メモリに保存された情報に基づいて、ニューラルモジュールが瞬時に再構築される。

4. 4. 2 CBM アーキテクチャ

CBM は大きくは以下の 6 つのブロックから構成される；

- 1) セルオートマトン部
- 2) 遺伝子型・表現型メモリ
- 3) 適応度評価ユニット
- 4) 遺伝的アルゴリズムユニット
- 5) モジュール間接続メモリ
- 6) 外部インタフェース

(1) セルオートマトン部

セルオートマトン (CA) 部は、CBM のハードウェアのコア部分になるものである。CA 部は、物理的にはハードウェア論理回路のアレイで構成され、24 x 24 x 24 のトータル 13,824 個の CA セルによる 3 次元構造を提供する。3 次元空間は東西、南北、

天地間が再帰的に接続され、全方向が再帰的なトラス空間である。この特徴により複雑な樹状突起と軸索の成長が可能になる。

CA 部は、シミュレーション動作時、複数のニューラルモジュールによって時分割的に使用される。但し、ある時点では 1 個のニューラルモジュールのみが実行される。FPGA のファームウェアは二重バッファ構造であるため、あるニューラルモジュールが実行中であっても、次のニューラルモジュールの構造形成を同時に行うことができる。即ち、複数ニューラルモジュールが時分割的に処理されるが、ニューラルモジュール実行間の空き時間なしに FPGA コアは連続的に動作することができる。

CA 部は、そのキューブ状表面を介して、他ニューラルモジュールとの信号授受のための外部インタフェースを有する。各表面は 64 信号分の行列を有し、それらは反対表面に折り返し接続されている。従って、合計 192 の異なる接続が利用可能となる。それらのうち 4 つの接続が出力点として使用される。

CA 部は Xilinx 社の FPGA 素子 XC6264 を用いて実装した。このデバイスは全体的にも部分的にも再構成可能であり、ユーザ入出力のほかデータバスおよびアドレスバスへのアクセス、さらに、データバスを通じて内部のフリップフロップへの書きこみ・読み出しも可能である。XC6264 は 16,384 個の論理機能セルを有し、それらの論理機能セルはフリップフロップやブール論理として 220MHz で動作する。論理機能セルはいくつかの階層レベルにおいて近傍同士が内部接続され、いかなる接続長に対しても同一の伝播遅延を保証している。この特徴は 3 次元の CA 空間を形成するのに非常に適している。

CA 部の実装に当り、同一の論理セルの 3 次元ブロックが各 FPGA 内に構築できるようにした。内部ルーティングや論理セルの容量を考慮すると、1 個の XC6264 に実装できる CA セルの最適な構成は 4 x 6 x 8、計 192CA セルとなる。この CA セルの基本ブロックを基に、仮想的な 3 次元空間の東西、南北、上下の 6 面で近傍の FPGA と内部接続して 3 次元構造を形成するためには 208 の外部接続が必要となる。結果的には、合計で 72 個の FPGA を用い、それらを 6 x 4 x 3 の構造に並べ、24 x 24 x 24 の CA セル空間を実現した。

CBM のセル更新のクロックレートは 8.25MHz、9.47MHz のなかから選択できる。このスピードで、全ての 13,824 個の CA セルは同時に更新され、結果的には 1 秒間に 1,140 億から 1,310 億のセルの状態更新が可能となる。このスピードは CAM-8 の 570 倍から 655 倍に相当する。

(2) 遺伝子型・表現型メモリ

FPGA を搭載する 72 枚のボードには各々 16M バイト、従って全体で約 1.18G バイトの EDO (Enhanced Data Out) DRAM を搭載し、遺伝子型やニューラルモジュールの表現型を保存する。

CBM の動作モードは進化モードと実行モードの 2 種類を設けた。進化モードは成

長フェーズと信号フェーズからなる。成長フェーズの間、メモリは染色体情報を遺伝子型として保存する。遺伝子型メモリには CA 空間内のニューロンの情報や各 CA セルの成長方向の情報を保存する。13,824 個の CA セルに対して成長方向を指定する 6 ビットの遺伝子を用意する。それとは別にニューロン用の遺伝子を用意する。1,152 個のニューロンの位置は固定であるが、ニューロンとどうかを選択できる。ニューロン用遺伝子は、ニューロンとどうかを指示する 1 ビットと樹状突起入力を興奮性・抑制性に指定する 6 ビットの計 7 ビットからなる。従って、ニューラルモジュールに対して、91K ビット (6 x 13,824 + 7 x 1,152) の遺伝子型メモリが必要となる。

実行モードの間、メモリは形成されたニューラルモジュールの表現型を保存する。CA 部が時分的に使用されるため、一旦成長した樹状突起と軸索の構造およびそれらに接続されるニューロンの情報などが表現型データとして保存される。その表現型データが CA 部に再びロードされることにより、ニューラルモジュールの表現型が復元される。

このように遺伝子型および表現型メモリは、FPGA ハードウェア CA 部上に形成されたニューラルモジュールの情報を保存し、かつ、そのニューラルモジュールをハードウェア CA 上に即座に再構築するために使用される。ニューラルモジュールの再構築は、各セルに備えられたデュアルパイプラインの遺伝子型・表現型レジスタによって、実行中のモジュールと並行して行うことができる。これにより、進化・実行の両モードにおいて再ロードのための割り込みをかける必要がなく、FPGA アレイの連続動作が保証される。フルスピードの場合、遺伝子型・表現型メモリは同時に 63,640 個のニューラルモジュールをサポートすることができる。付加メモリは、PCI バスを介して CBM に接続されるホストコンピュータ (Pentium-Pro 500MHz) のメインメモリにあり、データ転送速度は 132M バイト/秒である。

(3) 適応度評価ユニット

進化モードの信号フェーズにおいて、各ニューラルモジュールの表現型は目標タスクに応じた適応度によって評価される。モジュールからのスパイク状信号の出力系列とスパイク状信号の目標系列とを比較し、その結果をモジュールの性能を測る尺度、即ち、モジュールの適応度として進化プロセスへ反映する。

具体的には、適応度評価は入力スパイク状信号用および目標スパイク状信号用のバッファと適応度評価器からなるハードウェアユニットで行う。各クロックサイクルにおいて、適応度評価のための入力信号はそのスタックから読み出され、ニューラルモジュールの入力セルに与えられる。同時に、目標信号がそのバッファから読み出され、適応度評価器により現在のモジュール出力と比較される。この適応度評価器は、畳み込み演算フィルタにより 2 つのスパイク状信号の畳み込み演算を行い、それらの差分を計算する。

なお、CBM でのシグナリングは 1 ビットのスパイク状信号系列、即ち、生物的な神経回路と同様に 0 のインターバルで分離された 1 の系列によって行われる [12]。外部からの刺激などアナログ情報は Spike Interval 情報符号化 (SIIC) を用いてスパイク状信号にコード化される。スパイク状信号からアナログ信号への変換は、特別なマルチタップの線形フィルタによってスパイク状信号を畳み込み演算することにより行う。

(4) 遺伝的アルゴリズムユニット

上記の適応度評価をニューラルモジュールの集団系に対して行い、その中で評価値の高いニューラルモジュールの一群を次世代に残すものとして選択する。次に、評価値の高いモジュールを生成した遺伝子型 (染色体) に対して、交叉や突然変異などの遺伝的操作を施し、次世代のニューラルモジュールのための集団系を生成する。これら交叉や突然変異の遺伝的操作は、親となる 2 つのレジスタおよび子孫用のレジスタを実装することにより、CBM ハードウェアによって高速に実行される。このように次世代の染色体は直接ハードウェアによりナノ秒オーダーで生成されるが、選択アルゴリズムは PCI バスを介して CBM に接続されているホストが実行する。

(5) モジュール間接続メモリ

複数のニューラルモジュール、例えば、人工脳のシミュレーションを想定すると大量の数のニューラルモジュールをひとつのシステムとして機能させる必要がある。モジュール間接続メモリは、そのようなニューラルモジュール間の信号授受を行うために利用される。各ニューラルモジュールは最高 188 個の他のモジュールからの入力を受信できる。各モジュールが接続されるニューラルモジュールのリストはホストの CBM 相互参照メモリ (64M バイト) に保存される。このネットリストはモジュール内接続を規定するものとしてユーザが記述し、CBM ソフトウェアがそれを EDIF (Electronic Design Interchange Format) フォーマットのモジュール内接続ネットリストにコンパイルする。

モジュール内接続に要する時間は 96 クロックサイクルである。64,640 個の各々のモジュールに対する信号メモリは 4 つの 96 ビット長の出力スパイク状信号を保存する。

実行モード時、各ニューラルモジュールが CA 部に構築される (即ち、表現型がロードされる)。その際、相互参照メモリの相互参照リストに基づいて信号入力バッファもまた 188 のスパイク状信号とともにロードされる。スパイク状信号は直前の実行状態において保存された信号である。同時に、現在インストールされているニューラルモジュールの 4 つの出力信号も信号メモリから読み出される。こうして、ユーザが設定したニューラルモジュール群の内部接続構造に応じて、そのシステムを構成す

る全てのモジュール間での信号の繰り返しの保存と読み出しが可能となる。

最大 64,640 モジュールを持つニューラルシステムの場合、CBM の更新速度は 1 秒間で各セルが約 120 ビット長のスパイク状信号を伝播させることができる。120 ビット長のスパイク状信号は SIIC 符号化法を用いて換算すると約 5 バイト程度のオーダーとなる。各ニューロンは 5 個までのスパイク状信号を受信し、そのニューラルシステム全体では 3.72 億個 (5 x 1,152 x 64,640) のスパイク状信号がニューロンで処理される。従って、このニューラルシステムにおける全ニューロンによる情報処理は 1.86 G バイト (5 x 5 x 1,152 x 64,640) / 秒のオーダーとなる。

(6) 外部インタフェース

CBM アーキテクチャは、信号メモリからの信号受信および信号メモリへの信号送信と同様に、外部インタフェースを介して CBM 外部とも信号授受ができる。例えば、ロボット、音声処理システムなど外部システムに対しても、ニューラルモジュールは 188 までの入力信号を受信でき、4 本のスパイク状信号を出力できる。CBM と外部システムとの間の情報転送速度は 20M バイト/秒である。

4.5 まとめ

再構築可能なハードウェアを前提に、情報に依存してハードウェア構造を生成し、生成されたハードウェア構造を評価することによって元となった情報を選択し、変化を加え、その情報からハードウェア構造を生成するプロセスを繰り返すことによって、所望のハードウェア構造を得るハードウェア進化の考え方を提案した。具体的には、3次元セルオートマトン空間にニューラルネットをハードウェアとして発生・成長・進化させるセルオートマトン型人工脳モデルのシステム構成法を論じ、有効に動作することを示した。さらに、本モデルの高速シミュレーションのために開発したセルオートマトン型人工脳実験装置 (CBM) の構成について述べた。

CBM は、汎用の FPGA (Field Programmable Gate Array) を使って 3次元のセルオートマトン (CA) をハードウェアとして用意し、遺伝情報 (染色体) に基づいてニューラルネットを CA 空間に成長させ、遺伝的方法論によって進化させるための実験用マシンである。

CBM は本来ハードウェアであるため、規模を大きくすれば同程度の時間で 100 億個ニューロンのネットワークも成長させることができる。しかも、人間の情報処理が大体ミリ秒 (1/1000 秒) ~ 数百ミリ秒程度であるのに対して、電子デバイスの処理速度はその 100 万分の 1 であるため、10 億~100 億のニューロンが並列動作する実行時の状況を想像すると、その潜在的な情報処理能力は計りしれないものがある。

人間の脳は単に 140 億個の神経細胞からなるひとつの巨大なネットワークではなく、神経細胞やニューロン数が全てではない。脳は部位ごとに機能も構造も異なる複雑なシステムであるため、脳科学や神経科学など他の研究分野の知見を取り入れつつモデル構築を図っていかねばならない。その意味でいろいろな立場や方法論に基づく多様かつ多岐に渡る研究が今後も必要である。

しかし、道具の進歩が科学技術の進歩を加速してきたように、CBM のような研究・実験ツールが利用できるようになることで、脳機能の解明や工学的な応用研究の進展に何らかの貢献ができるものと期待している。CBM のような研究・実験ツールをさらに高度化する延長線上で我々が考えているもうひとつの研究展開は、量子デバイスの利用である。量子デバイスを使った量子セルオートマトンで CBM を構成できれば、集積度や動作速度といった点でさらなる高性能化が可能となるであろう。少なくともニューロン数では人間の脳をはるかに越えるものを電子のスピードで随時、発生・成長・進化させることも遠い夢ではない [1]。

第4章の参考文献

- [1] 下原勝憲：人工生命と進化するコンピュータ，183 頁，工業調査会，19981.
- [2] 下原勝憲：進化システムの研究と展望，画像電子学会誌，第 26 巻，第 5 号，pp. 517-523，1997.
- [3] 下原勝憲，邊見均：自己増殖型人工脳，Computer Today，No. 90，3 月号，pp. 4-9，1999.
- [4] de Garis, H.: An Artificial Brain: ATR's CAM-Brain Project Aims to Build/Evolve an Artificial Brain with a Million Neural Net Modules Inside a Trillion Cell Cellular Automata Machine, New Generation Computing, Vol. 12, No. 2, pp. 215-221, 1994.
- [5] 下原勝憲他 (ATR 進化システム研究室編)：人工生命と進化システム，223 頁，東京電機大学出版局，1998.
- [6] 下原勝憲：人工脳とは，Clinical Neuroscience (月刊臨床神経科学)，Vol. 16, No. 11, pp. 66-67, 1998.
- [7] 下原勝憲：生命論パラダイムに基づく情報処理，情報処理，第 36 巻，第 4 号，p. 289, 1995.
- [8] 下原勝憲：人工生命 (II) 情報処理への応用と展望，テレビジョン学会誌，Vol. 50, No. 11, p. 1746, 1996.
- [9] Gers, F., de Garis, H.: CAM-Brain: A New Model for ATR's Cellular Automata Based Artificial Brain Project, Int. Conf. on Evolvable Systems, Vol. 1259, pp. 437-452, 1997.

- [10] Cho, S-B., Song, G. B., Lee, J. H. : Evolving CAM-Brain to control a mobile robot, Proc. of third Int. Symposium on Artificial Life and Robotics, pp. 271-274, 1998.
- [11] Korkin, M., de Garis, H., Gers, F., Hemmi, H. : CBM (CAM-Brain Machine) : A Hardware Tool which Evolves a Neural Net Module in a Fraction of a Second and Runs a Million Neuron Artificial Brain in Real Time, Genetic Programming Conference (GP' 97), pp. 498-503, 1997.
- [12] Korkin, M., Nawa, N. E., de Garis, H. : A 'Spike Interval Information Coding' Representation for ATR's CAM-Brain Machine (CBM), Second Int. Conf. on Evolvable Systems: from Biology to Hardware (ICES' 98), pp. 256-267, 1998.
- [13] Buller, A., Chodakowski, T., Hemmi, H., Shimohara, K. : CoDi Technique: Cellular Automata as a Large-Scale Neural Network, ATR Technical Report, TR-H-277, 1999.

第5章 ハードウェア記述言語を利用したハードウェア進化

5.1 はじめに

前章では、セルオートマトン (CA) を利用して、ニューラルネットをハードウェアとして発生・成長させ、さらに、進化させるというハードウェア進化について述べた。回路としての CA そのものは不変であるが、それらを敷き詰めた空間上にニューラルネットの構造を動的に創り出し、進化的方法論を用いてそれを繰り返し進化させることでニューラルネットとしての機能を実現するものであった。

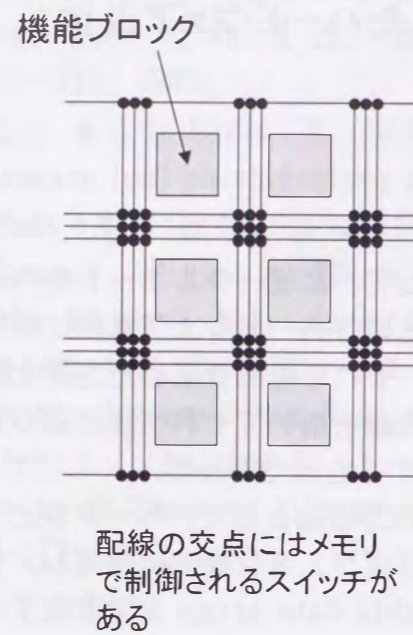
本章では、ハードウェア進化のもうひとつの方法論として、ハードウェア記述言語 (HDL : Hardware Description Language) プログラムを自動的に生成し、それに対応する論理回路の進化を FPGA (Field Programmable Gate Array) 上で実現する方法論について論ずる。ハードウェアの動作を HDL で記述できることに着目し、ハードウェアをプログラムとして扱い、HDL プログラムを進化させることによりハードウェアの進化を可能とするものである。容易にハードウェアに変換できる HDL プログラムはハードウェアと同一視できる。従って、ソフトウェアと同等の柔軟性を持ち、かつ、高級言語を用いることによる記述性と了解性の高いハードウェア進化システムを実現することができる。

5.2 基本コンセプト

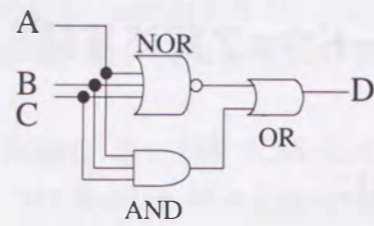
5.2.1 FPGA とハードウェア記述言語

代表的な再構成可能なハードウェア・デバイスのひとつが FPGA である。つまり、ハードウェア進化の基本的な枠組み (図 4-1) において、ハードウェアを成長させるための畑とすることができる。例えば、図 5-1 (b) の回路において、メモリ A が 0 の時は NOR (B, C) が、また A が 1 の時は AND (B, C) が 0 に出力される。この種の回路を 1 つのブロックとし、多数のブロックを格子状の配線のなかに配置・接続し、さらに、配線の交点に同じようにメモリで制御されるスイッチを設けることで FPGA と呼ばれる回路ができる。この回路は各ブロックやスイッチのメモリに適当な値 (アーキテクチャ・ビット) を与えることによって、任意の論理機能をプログラムすることができる。

論理回路の機能は、ハードウェア記述言語 (HDL) と呼ばれる C 言語に似たプログラム言語で記述することができる。HDL プログラムはソフトウェアと同等の柔軟性を持ち、またコンパイルすることで FPGA のコンフィグレーション・データ (アーキ



(a) FPGAの概念図



A	B	C	D
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Aが0の時は
BとCのAND

Aが1の時は
BとCのNOR

(b) 機能ブロックの例

図5-1 FPGAのしくみ

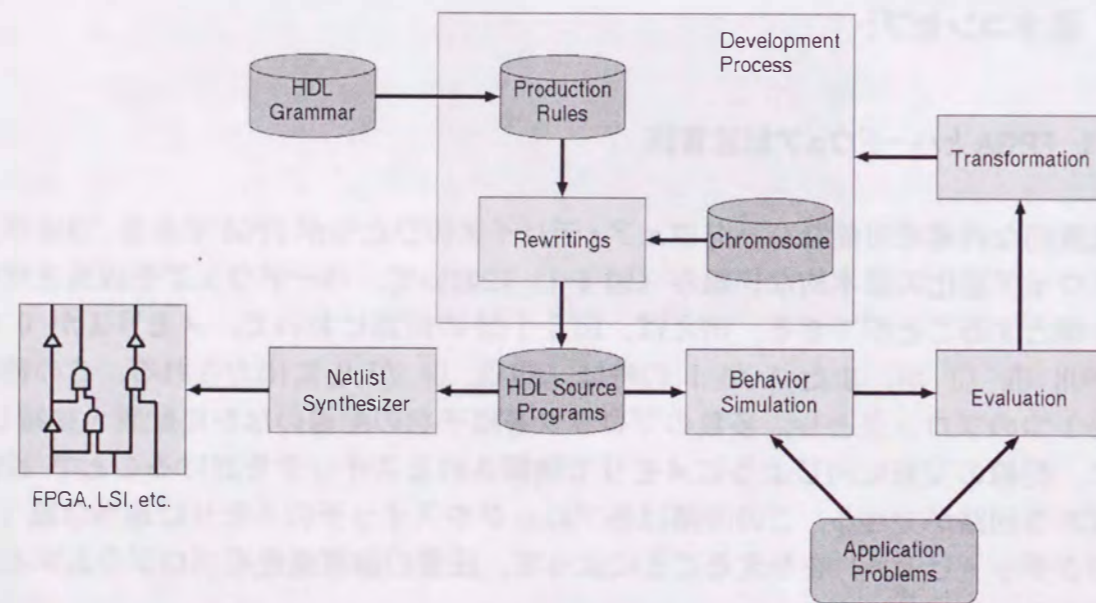


図5-2 ハードウェア行動の進化システム

テクチャ・ビットの列)へ自動的に変換することができる。

現在 HDL を用いて回路動作を設計 (プログラム) するのは設計者であるが、我々は、進化的な方法論を用いて、回路動作のプログラムを自動的に生成し、淘汰しながら、進化によって所望の回路動作を創り出すことを考えた (図 5-2) [1]。同様にハードウェア進化とはいえ、畑がセルオートマトンではなく FPGA であるという違いだけでなく、前章の CAM-Brain とは大きく異なる点がある。

CAM-Brain では、ニューラルネットを発生・成長・進化させる環境として CA というハードウェアを直接利用した [2]。同様に、FPGA のコンフィグレーション・データを遺伝情報として直接ハードウェアを FPGA 上に進化させるアプローチもある [3]。

しかし、本研究では、回路動作を記述するプログラムをソフトウェアとして進化させる。動作のシミュレーションをソフトウェアとして行う。直接 FPGA 上に実現して実際のハードウェア動作としてシミュレーションすることもできる。CAM-Brain がハード内進化のモデルであるのに対して、このアプローチはハード外進化のモデルである点が大きな違いである。プログラム進化型のハードウェアの自動設計法と捉え、CAD システム [4] とを組み合わせ、総合的なハードウェア進化システムを構築することも考えられる。

5. 2. 2 発生を模擬したプログラムの自動生成

ソフトウェア進化や CAM-Brain においては進化的な操作を容易化するため、遺伝的な情報に特別な情報表現を用いた。ティエラでは DNA・RNA など生体分子と同様に冗長な命令コードとテンプレート・マッチング型のアドレス方式を特徴とする機械語系を用いた。CAM-Brain ではニューラルネットの形成情報表現として数値情報を用いた。特に、CAM-Brain では、進化の方法論としては遺伝的アルゴリズム (GA: Genetic Algorithm) [5] を用い、基本的には一連の数値情報を変換・解釈して表現型を生成した。従って、進化によって最終的に良い結果が得られたとしても、その遺伝情報は数値情報の塊である。問題が単純であれば、その数値情報を解読して実際の意味を理解することも可能である。しかし、問題が複雑になると、その意味解釈は極めて難しくなる。つまり、了解性や記述性が乏しいことが難点である。

一方、LISP 言語で書かれたプログラムそのものを遺伝的な情報とするのが遺伝的プログラミング (GP: Genetic Programming) である [6]。本来望ましくない変化 (エラー) を生ずる可能性も否定できない進化的な操作に対して、木構造でプログラムを表現できるという LISP 言語の特徴をうまく利用したものである。即ち、枝に相当する部分でプログラム同士を交叉させ、或いは突然変異を与えても、文法上の制約を保持できる利点がある。当初 GP は、LISP 言語に限定されていたが、最近ソフトウェアの分野で広く利用されている C や Fortran といったプログラミング言語も扱えるように拡張された。遺伝的アルゴリズム (GA) に比べ、遺伝情報の記述性と了解性が高いことが大きな利点である。

同様に、本研究においても、ハードウェア記述言語というプログラム言語をいかに進化的な操作に耐えうるものとするかがひとつの課題であった。我々は、HDL 文法を書き換えシステム (Rewriting System) として表現し、プログラムの解析木 (生成ルールの構造体) を遺伝情報とすることにより、プログラムへの進化的な操作を可能とした [7]。

このようなアプローチの大きな利点はプログラムの自動生成も可能となったことである。一種の発生のプロセスを模擬して、1 個の受精卵がルール書き換え則を使って分裂・増殖していく。その結果、構文 (シンタックス) 的に正しいプログラムが自動的に生成される。従って、ハードウェアの動作を記述する HDL プログラムを自動生成し、かつ、進化的な手法によってそのプログラムを進化させることが可能となる。

5.3 ハードウェア記述言語 (HDL) の文法とプログラムの発生

ハードウェア記述言語 (HDL) を用いた進化システムでは、まずそれぞれの個体となる HDL プログラムを生成する必要がある。この節では HDL プログラムを自動的に生成する方法について述べる。なお、以下の説明では一般的な HDL ではなく、本研究で我々が採用した、HDL のひとつである SFL (Structured Function description Language) [8] に即して述べることにする。通常の HDL と異なり、SFL の場合はコンパイルせずに SFL 記述そのものを用いて動作シミュレーションを行うことができる。そのため高速な動作シミュレーションが可能となり、多くの個体を検証しなければならない進化システム向けの言語ともいえる。

SFL のプログラムは表面的には ASCII 文字の列である。しかし、逆に任意の ASCII 文字列が SFL のプログラムとして正当なものではない。また SFL は C 言語と同じく自由フォーマットの言語であるため、プログラム中にコメントを書くこともできる。従って、二つの ASCII 文字列が見かけ上多少異なっても、SFL プログラムとしては全く同一とみなされることもある。

ランダムに生成された ASCII 文字列が正当な SFL プログラムになる確率は非常に低い。例えば、ASCII 文字列そのものを遺伝的アルゴリズムの染色体とするアイデアが考えられるが、文法的に正しい遺伝子型が生まれることはほとんどないと考えられる。つまり、高い確率で文法的に正当な SFL プログラムを生成するメカニズムが必要となる。

さて、C、C++ 等多くのプログラム言語はその文法を厳密に定義するため BNF (Bacrus-Naur Form) と呼ばれる形式を使用している。SFL の文法も BNF で定義されている。図 5-3 にその定義の一部を示す。

本来の意味からいうと BNF は定義の列と見なすことができる。つまり、開始記号 (SFL_desc) が順番に定義されていき、定義文が全てそれ以上定義される必要のない記号 (終端記号) に到達したときに終了し、言語の仕様が完全に確定する。ここで、

```
(r0.0) sfl_desc -> seq0_mod_def

(r1.0) seq0_mod_def -> empty
(r1.1) seq0_mod_def -> seq0_mod_def mod_def

(r2.0) mod_def -> K_MODULE mod_name K_LBRACE seq0_submod_type_OR_fc_type_def \
  seq0_mod_component_def seq0_mod_ctrl_pin_arg_def seq0_stage_and_task_def \
  seq0_mod_act_with_ctrl_pin_def seq0_stage_act_process_def \
  K_RBRACE

(r3.0) mod_name -> nm_mod

(r4.0) nm_mod -> NM_MOD
      :
      :
      :
(r15.0) formula -> monomial | formula /* or */
(r15.1) formula -> monomial @ formula /* eor */
(r15.2) formula -> monomial & formula /* and */
(r15.3) formula -> monomial || formula /* concatenation */
(r15.4) formula -> monomial + formula /* addition */
      :
      :
```

図5-3 SFLの文法定義

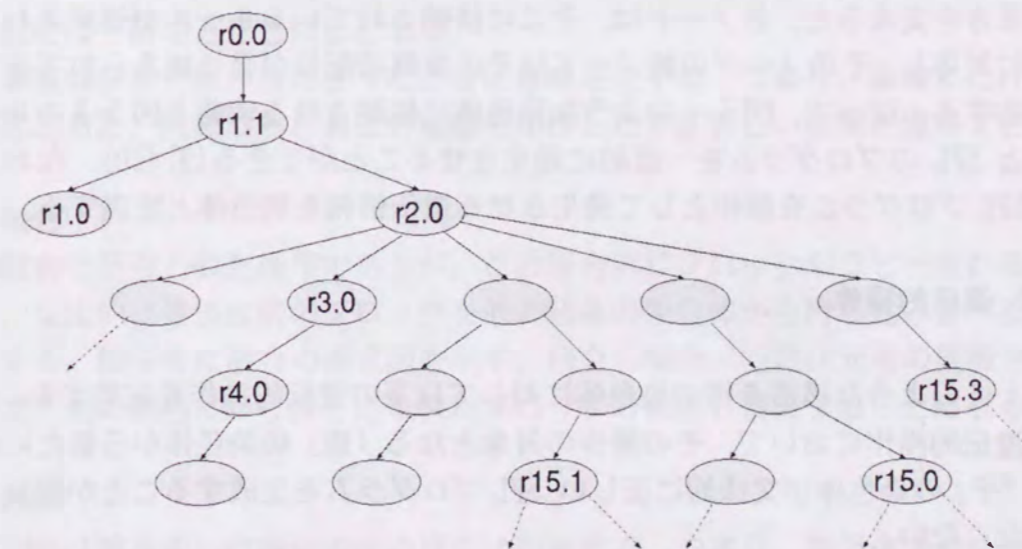


図5-4 染色体表現

非終端記号とは例えば SFL のキーワード (“input” “output” 等), 変数の名前, 数字などである。

この SFL 文法の BNF 表記は, 文法に則った SFL プログラムを発生させる機構に利用することができる。即ち, まず 1 個の開始記号を用意し, それを図 5-3 のルールにしたがって書き換えていく。全ての記号が終端記号に置き換わったところでひとつのプログラムができ上がることになる。ひとつの非終端記号に適用できるルールは複数ありうる。

SFL 文法の BNF 表記は, 「書き換えルール」(rewriting rule) であり, 「プロダクションルール」(production rule) である。このようにプログラムの自動生成の観点からみると, BNF はプログラムをどのように発生させるかの方法を提供しているともいえる。

5. 4 染色体表現と遺伝的操作

5. 4. 1 染色体表現

SFL プログラムの発生において, 現れた非終端記号に適用可能なプロダクションルールが複数ある場合にどのルールを選択するかが問題となる。ここで提案する進化システムではルールの選択情報そのものを染色体のデータとする。即ち, 図 5-4 のような木構造の染色体を考え, 各ノードの位置にどのルールを選択するかの情報を格納する。見方を変えると, 各ノードは, そこに格納されているルールが適用される非終端記号に対応し, そのノードの枝ノードはその非終端記号が書き換えられて生ずる記号に対応する。従って, 図 5-4 のような染色体に格納される情報と図 5-3 のルールを用いると SFL のプログラムを一意的に発生させることができる [9] [10]。なお, ここでは, SFL プログラムを個体として発生させる遺伝情報を染色体と定義する。

5. 4. 2 遺伝的操作

図 5-4 のような構造を持つ染色体に対して以下の遺伝的操作を定義する。以下の全ての遺伝的操作において, その操作の対象となる「親」の染色体から新たに創り出される「子」の染色体が文法的に正しい SFL プログラムを生成することが保証されなければならない。

- 交叉

交叉は二つの染色体を部分的に組み合わせて別の染色体をつくる操作である。図 5-4 のような木構造の染色体において, それぞれの副木 (sub tree) はプログラム中のある文法的にまとまったブロックを示している。副木の根の位置に対応する

非終端記号につけられた名前がこのブロックを表している。同じ非終端記号で表されるブロックは相互に入れ換え可能である。つまり, ある染色体の副木をその副木と同じ非終端記号で表される別のブロックと置き換えても文法的には正しい SFL プログラムがつけられる。従って, 交叉とは, 親同士 of 染色体の副木を他方の等価な副木と書き換える操作とする。図 5-5 に交叉の模式図を示す。

- 変異

同じ非終端記号に着目する交叉の考えに基づき, 変異では元になる染色体の一部の副木を文法的に等価となるように新たにつくり直す。具体的には, その副木のルートのあるノードのルールを, 同じ非終端記号に適用される別のルールに置き換える操作とする。複数の選択肢がある場合はランダムに選ぶこととする。図 5-6 に変異操作の模式図を示す。

- 重複

重複は再帰的なルールに関係する操作である。図 5-3 のルール r 5.1 ではこのルールが適用される左辺の記号が, 書き換えの結果を示す右辺の記号列の中に再び現れている。このようなルールを使うと染色体中にリストと同様の構造をつくることができる。重複はこのようなルールを用いて染色体の部分ブロックをコピーして並べる操作である。図 5-7 に重複の模式図を示す。

重複は SFL プログラムの機能にとっては中立な操作である。即ち, 重複操作の前後ではプログラムの機能は変化しない。全く同じことをする部分が 2 個あっても機能的には 1 個のときと同じとなる。

重複は変異と組み合わせるときに意味を生ずる。つまり, 重複した片方に変異が起こると, 回路はもともとの機能を保持したまま新しい機能を獲得できる。

- 融合

融合は重複と似た操作であるが, この場合同じブロックがコピーされるのではなく, 文法的に等価な別のブロックを他の個体の染色体から持ち込み並べることに相当する。図 5-8 に融合の模式図を示す。融合の場合, 回路は元来の機能を保存したまま, 他の染色体から持ち込まれたブロックの機能も獲得することになる。

- 削除

削除は重複或いは融合の逆の操作に相当する。つまり, 染色体中のリスト構造をした部分から 1 つのブロックを消去する操作である。図 5-9 に削除の模式図を示す。文法的に正しい形を保持したまま, 削除によりひとかたまりの機能が SFL プログラムから消去される。

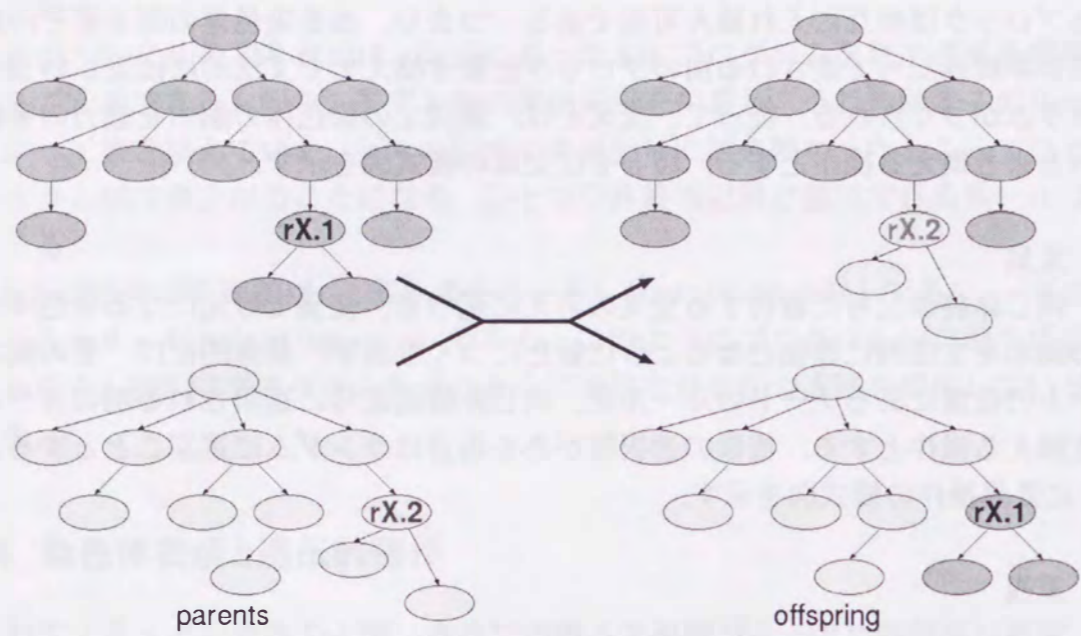


図5-5 交叉

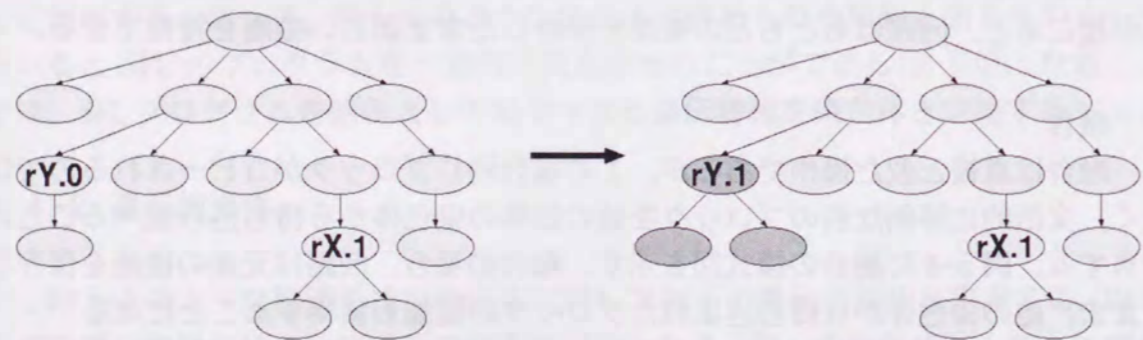


図5-6 変異

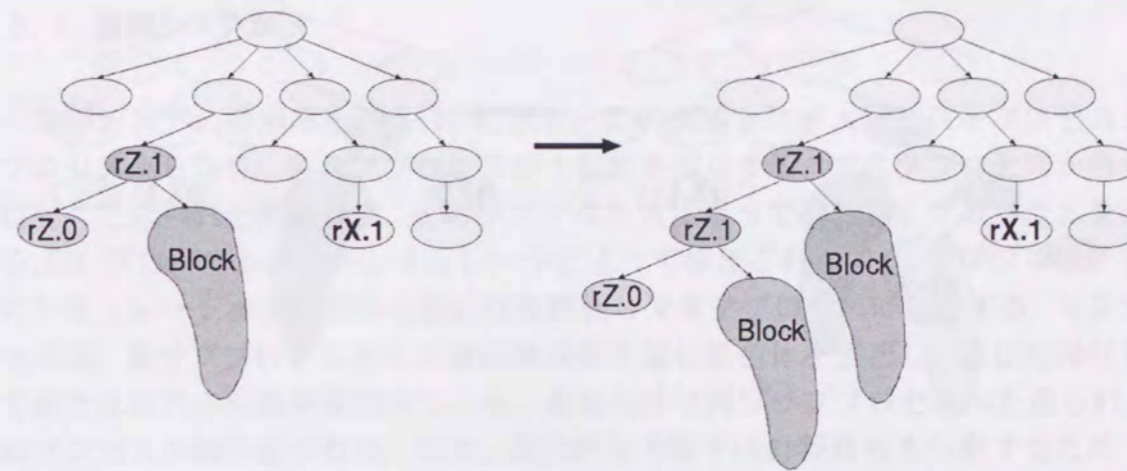


図5-7 重複

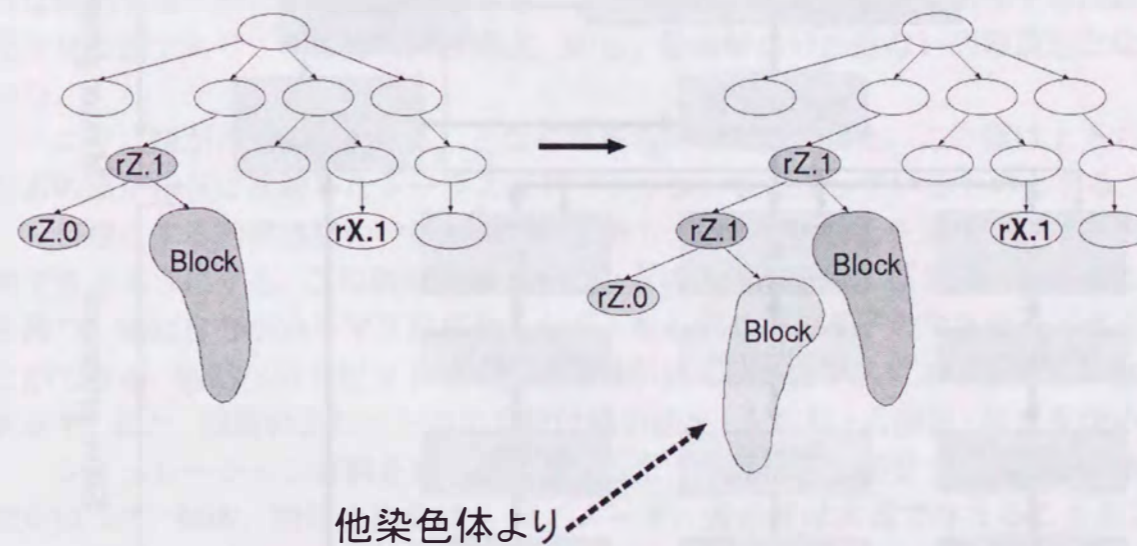


図5-8 融合

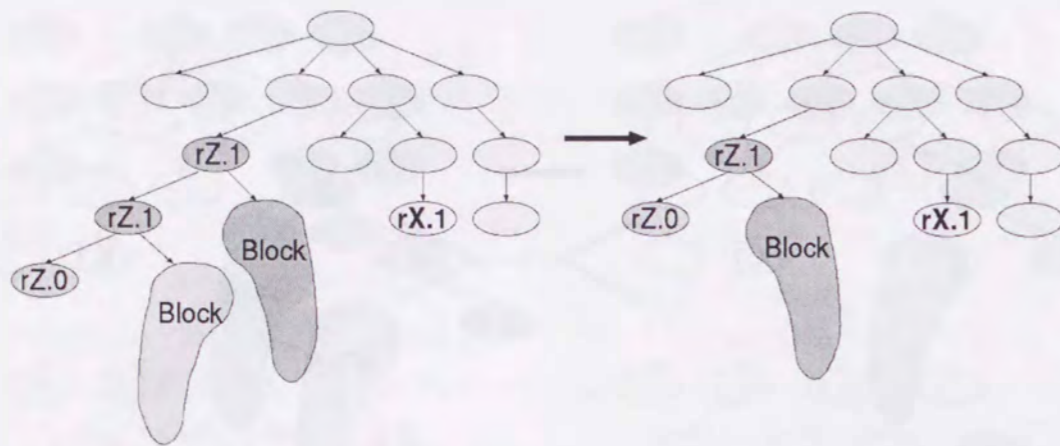


図5-9 削除

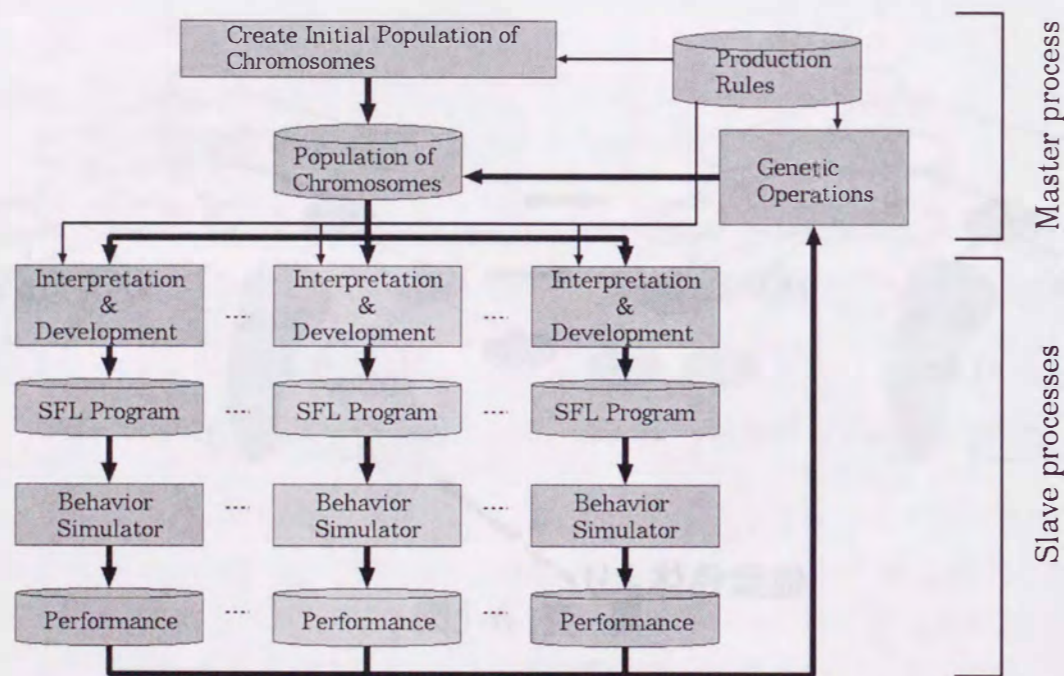


図5-10 実験システムの構成

5.5 シミュレーション実験

5.5.1 実験システム

実験システムの構成を図5-10に示す。この実験システムはマスタプロセスとサブプロセスからなり、各サブプロセスが1個体を担当する。マスタプロセスからサブプロセスに送られた染色体は、そのサブプロセスによって個体HDLプログラムを発生する。HDLプログラムは動作シミュレータによって検証される[11]。サブプロセスは、そのシミュレーション結果から適応度を評価しマスタプロセスに返送する。マスタプロセスは、各サブプロセスからの適応度評価を基に染色体を選択し、遺伝的操作を施して新たな世代の染色体集団をつくる。各染色体は再びサブプロセスへと送られ、同様のプロセスが繰り返される。なお、具体的な実装では計算負荷を分散するためマスタプロセスおよび各々のサブプロセスがLAN上の別々のコンピュータで実行できるようにした。

5.5.2 人工蟻シミュレーション

図5-11はJohn Muir Trailと呼ばれる問題を示す。図において黒く塗りつぶしたマス目にエサが置かれているとし、このエサを蟻がたどるといふ問題である。白いマス目および灰色のマス目にはエサはない。灰色のマス目は黒のマス目をつなげるような経路を便宜的に示したものである。この問題は、未知の環境で動作する回路を進化させる例であり、理想的な回路構成、即ち、最適解が分からない問題設定となっている。

エサは蟻がたどる(食べる)となくなるものとする。また、この面は上下および左右の面が相互に連続したトーラス(ドーナツ)状になっているものとする。

目標とする回路は蟻の行動制御回路であり、蟻がマス目を適切にかつ迅速に移動できるようにする。この制御回路は5ビットの入力端子および2ビットの出力端子を持つ。蟻は自分がいるマス目の前・左右・左右斜め前の5子のマス目を「見る」ことができる。回路への5ビットの入力値は蟻が見ている各マスにエサがあるかないかを示す。また、回路の2ビットの出力値は蟻の前出・左回転・右回転・停止を決める。

シミュレーション実験を次の条件で行った。1世代の集団サイズは200個体、交叉率は50%/個体、突然変異率は0.5%/ノード。適応度は次式で与えることとした。

$$\text{Performance} = \text{score} + (\text{time_limit} - \text{time_steps}) \quad (5.1)$$

但し、score=食べたエサの数(最大89)+1, time_limit = 350
time_steps: 実際に要したステップ数である。

即ち、蟻は350ステップの間だけ行動することができ、その間に食べたエサが多いほど適応度が高くなる。また、350ステップを要せずして全てのエサを食べる場合はより短いステップ数でエサを食べたものほど適応度が高くなる。シミュレーションの結果を図5-12に示す。

123世代目に293点をマークする個体が現れた。これは147ステップで全てのエサを食べることを意味する。これは図5-11に描かれた軌跡を通過するための最短のステップである。

1ビット入力2ビット出力とした場合における人工蟻のSFLプログラムの進化結果を図5-13に、それに対応する回路を図5-14に示す。

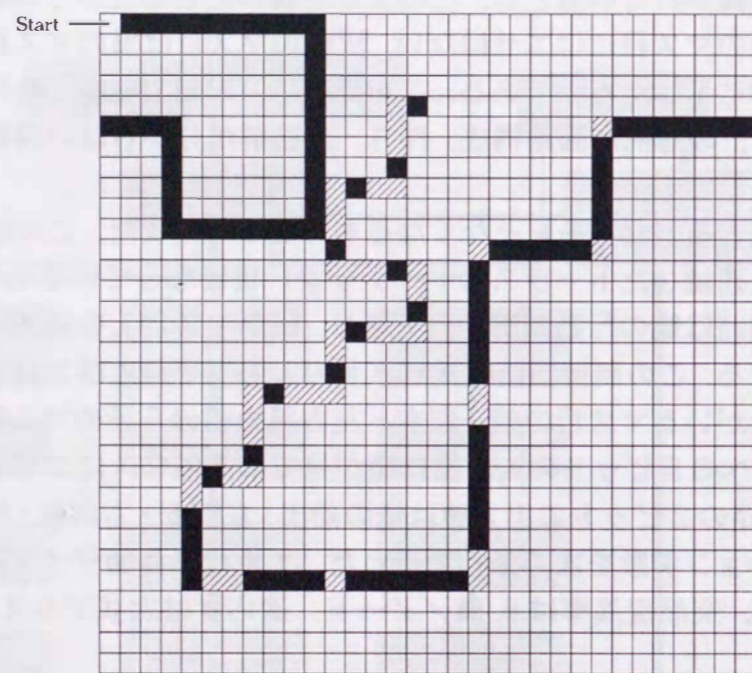


図5-11 人工蟻の実験環境(John Muir trail)

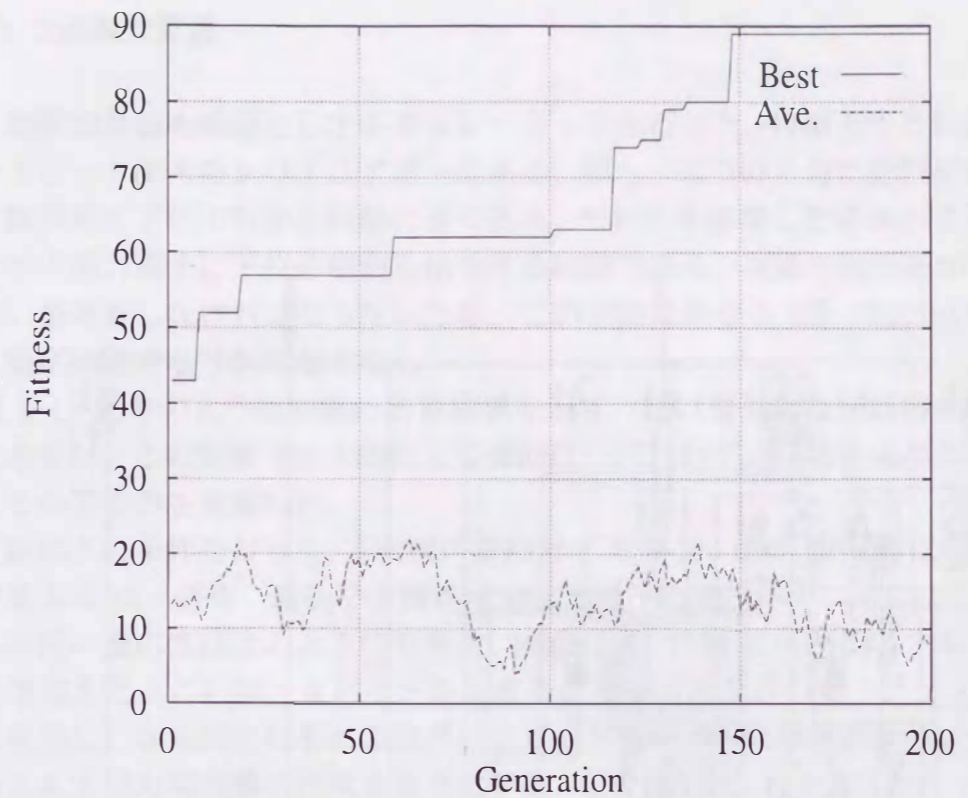


図5-12 人工蟻の進化

```

module p_module (
input p_in1;
output p_out1,p_out2;
instrin p_instin1;
instrout p_instout1;
reg_ws p_reg1;
instruct_arg p_instout1 (p_out1 ,p_out2);
stage_name p_stage1 {
task p_task1 (p_reg1);
}
instruct p_instin1 generate p_stage1 .p_task1 (0b1);
stage p_stage1 {
state_name p_state1;
state_name p_state2;
state_name p_state3;
state_name p_state4;
state_name p_state5;
first_state p_state1;
state p_state1 alt{
^p_reg1 par{
p_reg1 :=^p_in1 ;
p_instout1 (^p_in1 ,^p_reg1 );
p_instout1 (^0b1 ,^p_reg1 );
goto p_state2 ;
}
^p_reg1 par{
p_reg1 :=^p_in1 ;
p_instout1 (^p_in1 ,^p_reg1 );
p_instout1 (^0b1 ,^p_reg1 );
goto p_state2 ;
}
else par{
p_reg1 :=^p_reg1 ;
goto p_state5 ;
p_instout1 (^0b1 ,^p_reg1 );
goto p_state1 ;
}
}
state p_state2 alt{
^0b0 par{
goto p_state5 ;
p_instout1 (^p_in1 ,^p_reg1 );
p_instout1 (^0b1 ,^p_in1 );
goto p_state2 ;
}
else par{
p_instout1 (^0b0 ,^p_reg1 );
p_instout1 (^p_reg1 ,^p_in1 );
p_instout1 (^0b1 ,^p_in1 );
goto p_state4 ;
}
}
state p_state3 alt{
^p_in1 par{
p_reg1 :=^p_reg1 ;
p_reg1 :=^0b1 ;
p_instout1 (^0b1 ,^p_reg1 );
goto p_state2 ;
}
}
state p_state4 alt{
^p_in1 par{
p_reg1 :=^p_reg1 ;
p_reg1 :=^0b0 ;
p_instout1 (^0b1 ,^p_reg1 );
goto p_state2 ;
}
}
state p_state5 alt{
^p_in1 par{
goto p_state3 ;
p_instout1 (^p_in1 ,^p_reg1 );
p_instout1 (^0b1 ,^p_in1 );
goto p_state1 ;
}
}
state p_state5 alt{
^p_in1 par{
goto p_state4 ;
goto p_state2 ;
p_instout1 (^p_in1 ,^p_reg1 );
goto p_state5 ;
}
else par{
p_instout1 (0b1,p_reg1 );
goto p_state1 ;
p_instout1 (^p_in1 ,^p_reg1 );
goto p_state3 ;
}
}
}
}

```

図5-13 進化した人工蟻のSFLプログラム

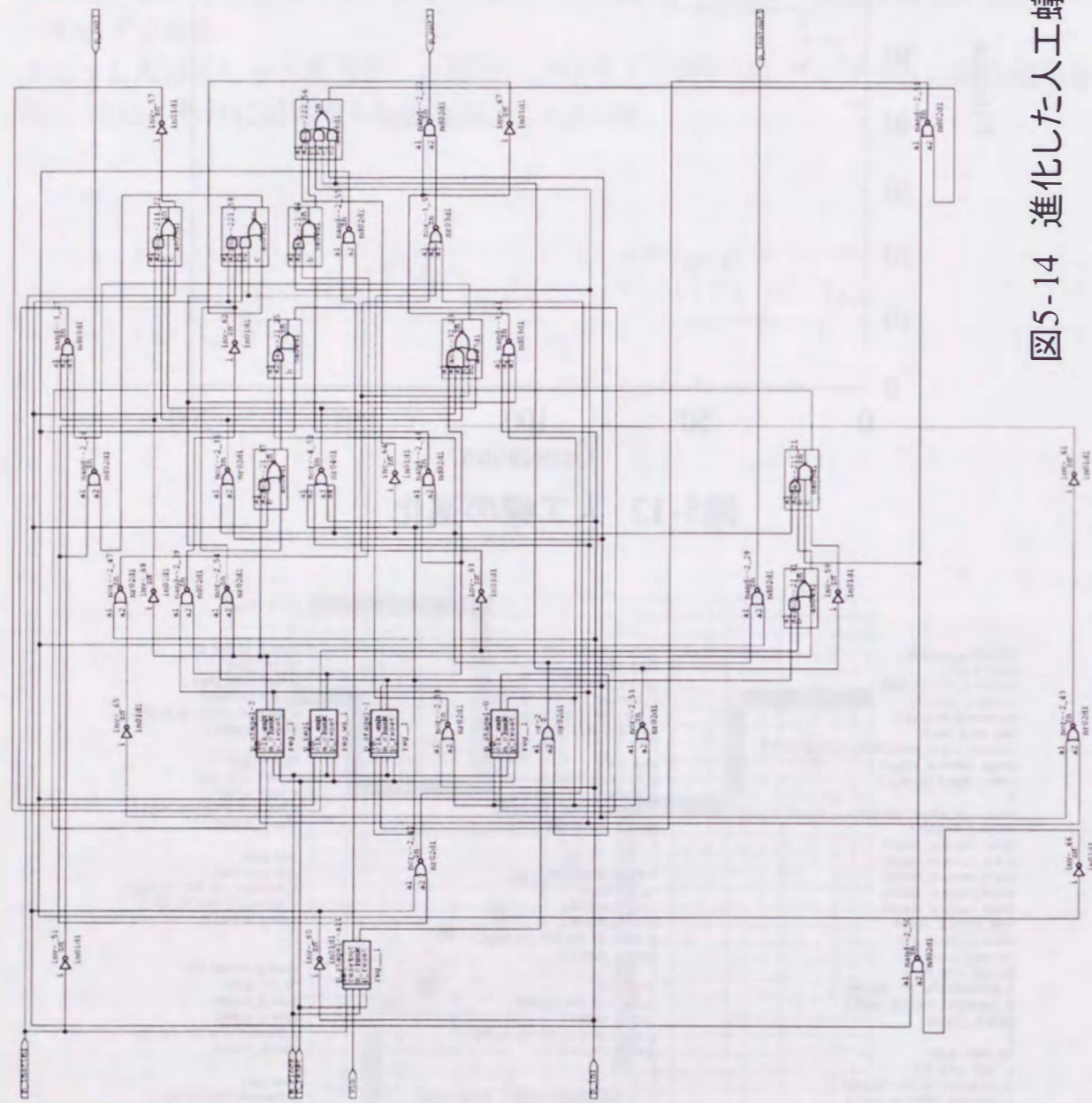


図5-14 進化した人工蟻の回路

5.5.3 2進数加算器

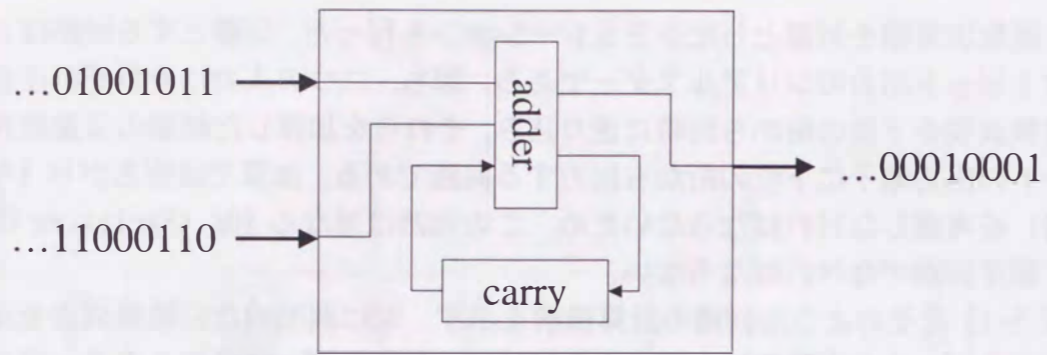
2進数加算器を例題としたシミュレーションも行った。目標とする回路は2ビット入力1ビット出力のシリアルアダーである。即ち、二つの入力にそれぞれ任意の数の2進数表現を下位の桁から同時に送り込み、それらを加算した結果の2進数表現を1ビットの出力端子に下位の桁から出力する回路である。加算では桁あがり（キャリー伝播）を考慮しなければならないため、この回路は単なるXOR（Exclusive OR）ではなく順序回路でなければならない。

図5-15にそのような回路の計算機構を示す。図は典型的な回路構成法を示したものであるが、この実験では人間による補助は一切行わず、回路をスクラッチから構成できるかどうかを実験した。

実験は次の条件で行った。1世代の集団サイズは100個体、交叉率は50%/個体、突然変異率は1%/個体、遺伝子欠損率は1%/個体とした。

各回路の適応度は次のように与えた。回路には、1536ビット長の2本1組のテスト入力を与えた。これは、4ビットの2進のあらゆる組み合わせ（ $2^4 \times 2^4 = 256$ 通り）を用意し、さらにそれぞれの境界に2ビット長の00のパディング（キャリー伝播をおさえて出力期待値の作成を容易化する）を付加した。つまり $(4+2) \times 256 = 1,536$ となる。回路には最初に持ち点が1,536点あり、出力が期待値と合えば1点加え、期待値と異なれば1点減点する。従って、テストを完全にクリアすれば3,072点、全部間違えれば0点となる。

実験結果を図5-16に示す。251世代にテストを完全にクリアする回路が得られた。ところで、適応度で満点を取るだけでは4ビットまでの大きさの数の加算しか保証されない。目標は任意の大きさの数の加算ができる加算器であった。そこで、満点を取った回路のHDL記述を人手によりチェックした。その結果、人間の設計者の典型的な記述より冗長にはなっているものの論理的には同一の記述であるが確認できた（図5-17、図5-18）。



集団サイズ : 100
 交叉率 : 50% per 個体
 突然変異率 : 1% per ノード
 重複率 : 1% per 個体
 削除率 : 1% per 個体

図5-15 シリアルアダー

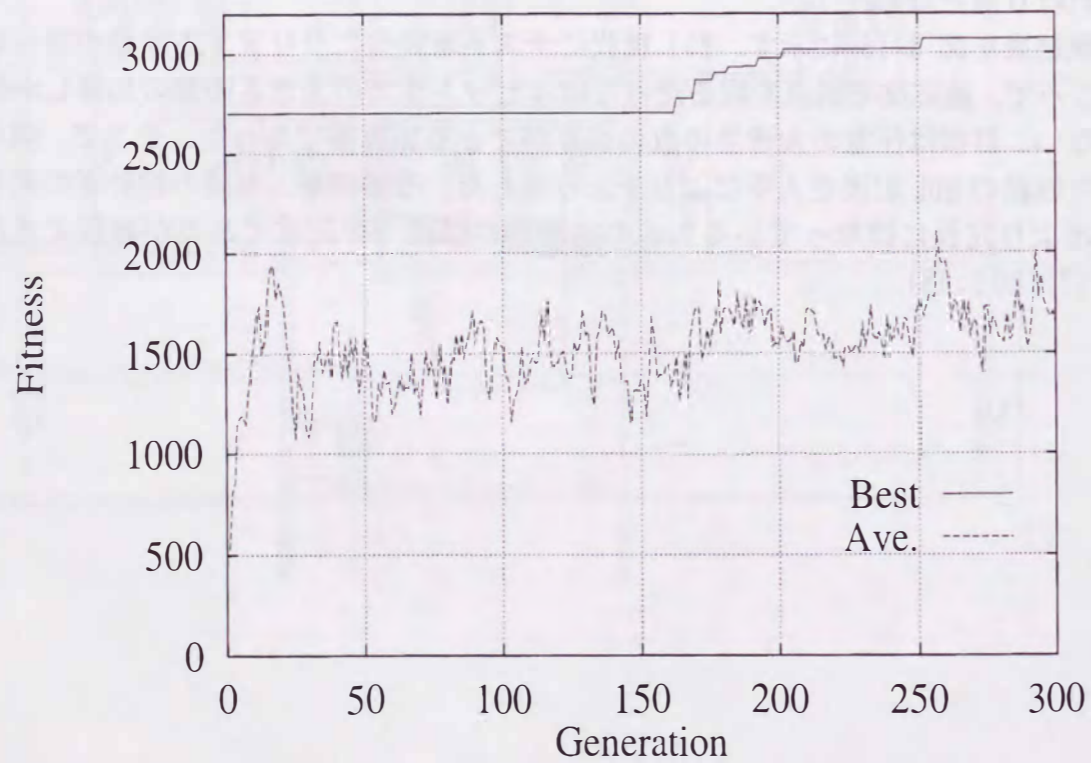


図5-16 シリアルアダー回路の進化

```

module p_module (
  input p_in1, p_in2;
  output p_out1;
  instrin p_instin1;
  instrout p_instout1;
  reg_ws p_reg1, p_reg2, p_reg3;
  instruct_arg p_instout1 ( p_out1 );
  stage p_stage1 {
    state_name p_state1;
    state_name p_state2;
    first_state p_state1;
    state p_state1 alt {
      ^p_in1 & ^p_in2: par {
        p_instout1 ( 0b0 );
        goto p_state1;
      }
      ^p_in1 & p_in2: par {
        p_instout1 ( 0b1 );
        goto p_state1;
      }
      p_in1 & ^p_in2: par {
        p_instout1 ( 0b0 );
        goto p_state2;
      }
      p_in1 & p_in2: par {
        p_instout1 ( 0b1 );
        goto p_state2;
      }
    }
    state p_state2 alt {
      ^p_in1 & ^p_in2: par {
        p_instout1 ( 0b0 );
        goto p_state2;
      }
      ^p_in1 & p_in2: par {
        p_instout1 ( 0b1 );
        goto p_state2;
      }
      p_in1 & ^p_in2: par {
        p_instout1 ( 0b0 );
        goto p_state2;
      }
      p_in1 & p_in2: par {
        p_instout1 ( 0b1 );
        goto p_state2;
      }
    }
  }
}
  
```

図5-17 進化したシリアルアダーのSFLプログラム

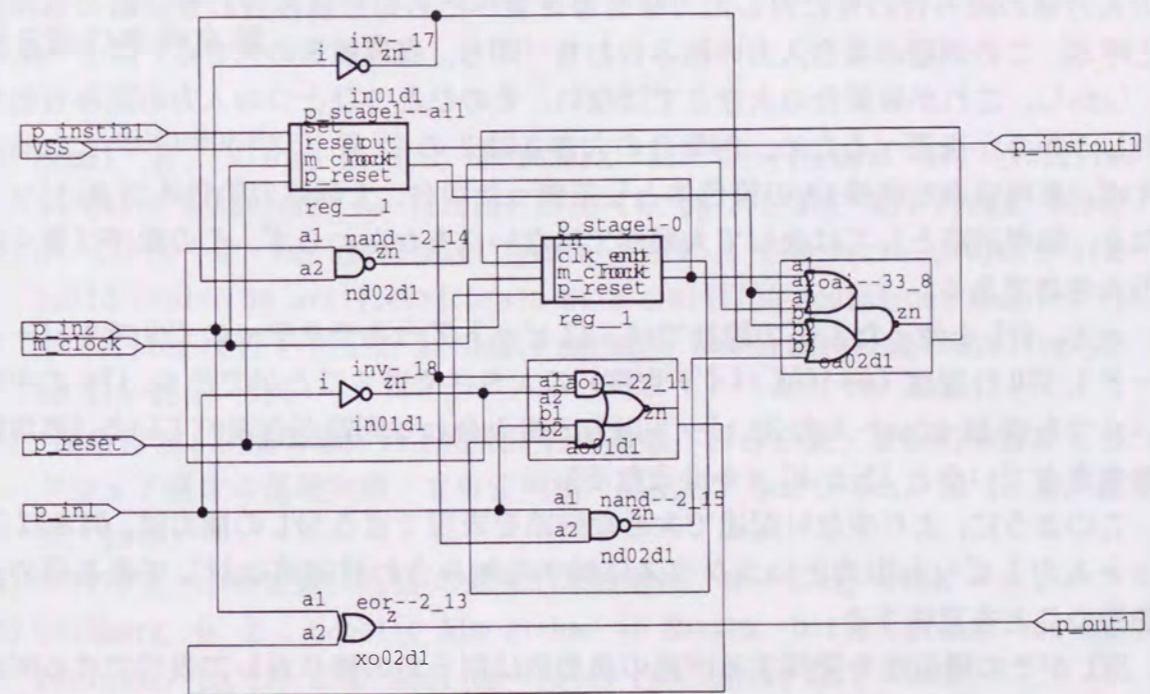


図5-18 進化したシリアルアダー回路

5.6 まとめ

ハードウェア記述言語 (HDL) に基づくプログラムの自動生成と進化的方法論を組み合わせたハードウェア進化システムの構成法について論じた。ハードウェアの動作を記述する HDL プログラムを自動生成する方法を提案し、さらに、それらを進化させるシステム構成法を明らかにして、シミュレーション実験により本システムの機能・動作を確認した。

本研究の着眼点は、ハードウェア記述言語 (HDL) という一種のプログラム言語を用いてハードウェアの動作を記述することにより、ハードウェアをプログラムとして扱う点にある。換言すると、HDL プログラムは容易にハードウェアに変換できるため、これをハードウェアと同一視するという考え方である。

本来「固い」はずのハードウェアに進化の機構を持ちこむため、ハードウェアを可変にする手段として HDL を使い、HDL プログラムを自動生成する方法と組み合わせでハードウェア進化を実現した。

以下では、本システムのスケラビリティについて考察する [9]。即ち、本方法はより大規模な回路に対しても有効か否かを考えてみる。一般的には、問題の規模が大規模になるということは解空間が大きくなるということである。そこで、32 ビット入力 1 ビット出力の組み合わせ回路について、解集合の大きさを計算してみる。組み合わせ回路の論理機能は入力値に対する出力値との関係で完全に決定される。それぞれの入力値の組み合わせに対し出力値を書き並べたものを組み合わせ回路の真理値表と呼ぶ。この例題の場合入力の組み合わせ (即ち、真理値表の大きさ) は $2^{32}=4G$ ある。しかし、これが解集合の大きさではない。そのひとつひとつの入力の組み合わせに対し 1 か 0 かを選べるため、解集合の大きさは 2 の 2^{32} 乗 ($2^{4 \times 1024 \times 1024 \times 1024}$) となる。例えば、真理値表を直接 GA の染色体として使った場合、1 個体の染色体が 4G ビットとなる。論理回路としては決して大規模ではないにもかかわらず、この数字は驚くほどの大きさである。

一方、SFL を使った LSI の設計では、32 ビットアーキテクチャの CPU でもソースコード 1,500 行程度 (約 100K バイト程度) の大きさで書くことができる。CPU の中にはいくつもの 32 ビット入力 32 ビット出力の組み合わせ回路が含まれている (真理値表の大きさでいうと $32 \times 4G = 64G$ となる)。

このように、より少ない記述で大きな回路を表現できる SFL の能力は、例えば 32 ビット入力 1 ビット出力というクラスに属する組み合わせ回路に対してある種の適性を持つことを意味する。

SFL がその優位性を発揮する回路の典型例は何らかの繰り返しで表現できる回路である。SFL も多くのプログラム言語と同様に繰り返しに対していくつかの種類のサポート機能がある。繰り返しはよりメタなレベルでも有効である。即ち、人間の設計者が書く SFL のプログラムにも、アイデアが良く整理されている場合、何らかの繰り返し

返しパターンを認めることができる。つまり、繰り返しは大規模なもの、複雑なものに対して幅広いレベルで有効性を持つ手段といえる。

本章で提案したハードウェア進化システムも、繰り返しパターンを生み出しやすい「重複」という遺伝的操作を備えている。その効果を暗示する結果もある。前節で示した人工蟻の実験 (入力 5 ビット) では、入力が 1 ビットのときよりも進化が早かった。入力が多い方が回路の規模は大きくなるが、記述には繰り返しが多くなる傾向があるためと考えられる。

プログラムを染色体とする進化的方法論として LISP 言語プログラムを進化の対象とする遺伝的プログラミングがある [12]。ここでは、HDL 言語プログラムを進化の対象とした。しかし、ここで提案した方法論は LISP に限らず原理的には言語仕様が BNF で定義される任意の言語に対しても適用可能な方法である。それは C や C++ といった広く普及している言語は勿論、現在使われているほとんど全てのプログラム言語を包含する。従って、この技術は一般的なソフトウェア進化の研究に対しても大きなインパクトを与えるものと考えられる。

さらに、もうひとつの研究展開の可能性として、書き換えシステムからなる文法系そのものを進化させることも考えられる。ある種の目的に特化した文法やプログラム言語を進化的に創り出すことも可能かもしれない。

第5章の参考文献

- [1] Hemmi, H., Mizoguchi J., Shimohara, K.: Development and evolution of hardware behaviors, Artificial Life IV, pp.371-376, MIT Press, 1994.
- [2] de Garis, H: An artificial brain — ATR's CAM-brain project aims to build/evolve an artificial brain with a million neural net modules inside a trillion cell cellular automata machine, New Generation Computing, No. 12, pp. 215-221, 1994.
- [3] 樋口哲也, 伊庭斉志, 丹羽竜哉, 田中敏雄, 古谷立美: 遺伝的学習によるハードウェア進化の基礎実験, 北野宏明編, 遺伝的アルゴリズム, 第 10 章, 産業図書, 1993.
- [4] 中村行広, 小野定康: ULSI の効果的な設計法, オーム社, 1994.
- [5] Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, pp.217-276, Addison-Wesley Publishing Co., 1989.
- [6] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.
- [7] Mizoguchi, J., Hemmi, H., Shimohara, K.: Production genetic algorithms for

- automated hardware design through an evolutionary process, Proceedings of the First IEEE Conf. on Evolutionary Computation, Vol. 2, pp. 661-664, 1994.
- [8] NTT データ通信株式会社: PARATHENON User's Manual, 1990.
- [9] 邊見均, 溝口潤一, 下原勝憲: 行動型ハードウェアの進化, 遺伝的アルゴリズム 2, pp. 207-249, 産業図書, 1995.
- [10] Hemmi, H., Mizoguchi, J., Shimohara, K.: Development and Evolution of Hardware Behaviors, Towards Evolvable Hardware, pp. 250-265, Springer-Verlag, 1996.
- [11] Hemmi, H., Shimohara, K.: Hardware Evolution A Real "Life on the Silicon," Int. J. Artificial Life and Robotics, Vol. 1, pp. 267-270, 1998.
- [12] 伊庭齊志: 遺伝的アルゴリズムの構造的表現への応用, 北野宏明編, 遺伝的アルゴリズム, 第4章, 産業図書, 1993.

第6章 今後の研究展開

6.1 はじめに

よくコンピュータは脳に例えられる。我々も、コミュニケーションの中核である脳の機能モデルとして、自律性と創造性を有する、新しいタイプのコンピュータの実現を目指している。自律性と創造性を有し、進化するコンピュータ、そのひとつの“かたち”が人工脳である。

これまで論じたように、我々は、システムの内部に発生と変化の機構を持ち多種多様な要素からなる社会のようなモデルを考えている。そして、進化システムというコンセプトのもと、ソフトウェアだけでなくハードウェアの構造をも自律的に創り変えてゆく進化システムの構成法を論じてきた。機能的には自律性と創造性を有するシステム、方法論的にはソフトウェア進化とハードウェア進化を統合したもの、それが進化システムとしての人工脳である。

我々は、生体の脳をそっくりそのまま人工的に創り出そうとしているわけではなく、むしろ生体の脳を超えるものを目指している。

本章では、そのような人工脳の構築に向けた方法論としての進化システムの今後の研究展開に関し、その可能性と意義について論ずる。

6.2 ソフトウェアとハードウェアの融合に向けて

ハードウェア進化の基本的枠組みを改めて図 6-1 に示す。情報群から各情報に依存した構造を持つハードウェア群を形成し、実現される機能や性能に応じてハードウェア群を評価・淘汰する。淘汰したハードウェアに対応する情報に進化的な操作を加えて次世代の情報群をつくり、またその情報群からハードウェア群を発生・成長・進化させる。

ハードウェア進化とは情報に依存してハードウェア構造を創り変えようということである。情報を種、ハードウェア構造を再構築できるハードウェア素子群を畑に例えると、種に応じたハードウェア構造を畑に創り出し、目的に合うようにそれを繰り返しながら徐々に品種改良していくことに相当する。従って、この例えを踏襲してその特徴をまとめると以下ようになる。

- 種は同じでも畑が違えば、そこに成長するハードウェア構造が異なる。
- そのことは、ある意味で、畑はハードウェア素子群として均一性を要求されない。畑ごとに多少の違い（デバイスの構造欠陥やエラー）があってもよい、畑

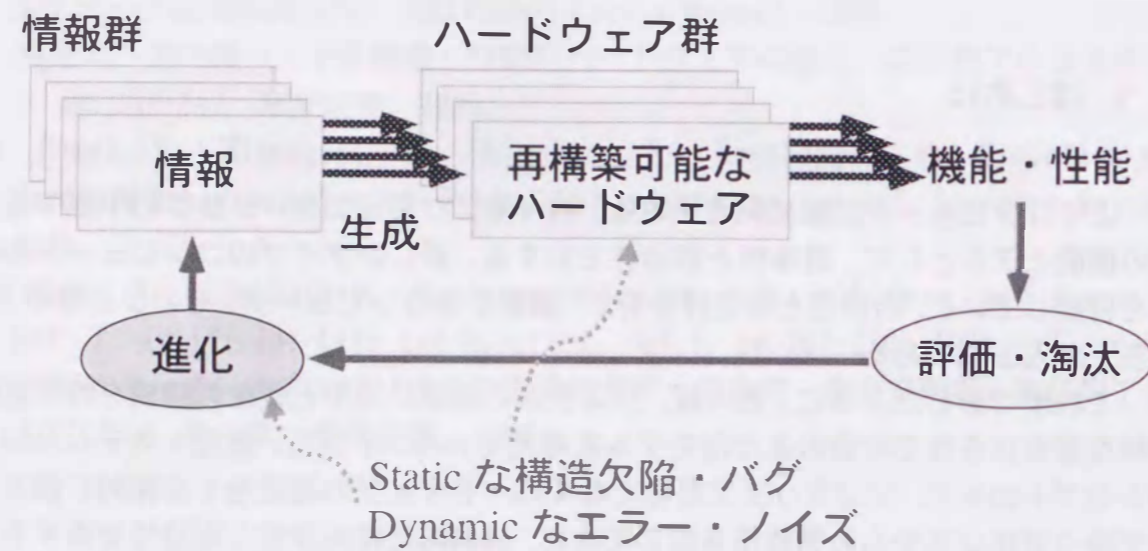


図 6-1 ハードウェア進化の基本的枠組み

に多少穴ボコがあってもよいということになる。

- 進化のプロセスを利用して、畑に合うように種を進化させればよく、畑の構造欠陥やエラーは種（情報）の変化や改良によって吸収すればよい。
- 通常的人工システムでは排除すべき対象である動的なノイズやエラーも、進化の枠組みの中では、それらが情報系に変化を与えるものとして利用できる。

種である情報はメモリが許す限りいくらでも準備することができる。一方、畑であるハードウェア素子群についても冗長性の高い集団系を前提とするため、大量のハードウェア資源を必要とする。そこで、ハードウェア進化のフィージビリティについて以下に論ずる。

現在、技術はハードウェア（物質）系とソフトウェア（情報）系とに明確に区分される。特にハード側からソフト側に対しては、品質と性能を保証するという厳しい暗黙の約束事がある。つまり、LSI 提供側は均一性が高く動作を保証できるデバイスや部品を提供しなければ使ってはもらえない。畑には穴ボコがあっては駄目なのである。ソフト開発側は、ハード部品は同じ性能を出せる共通的な枠組みであることを前提に、プログラムによって異なる機能やサービスを具現化する。

一方、ハード側の製造技術の進展は著しく、今や $10^{10} \sim 10^{11}$ の素子を 1 チップに集積することが可能となりつつある。しかし、素子数が膨大となるにつれて、テスト

ビリティ、即ち、それらを限られた時間内で試験することが今や非現実的になりつつある。事実、現状技術と同程度の歩留りで構造欠陥やエラーのないものを提供することも極めて難しい状況にある。半導体技術のブレークスルーを目指す量子デバイスなどのナノ・エレクトロニクスやナノ・テクノロジーにおいても、このテストビリティは大きな問題である。

しかし、このテストビリティの問題は、ハード側からソフト側に対して品質と性能を保証するという大前提があるから生ずるとも考えられる。ここで発想を転換し、ハードウェアの構造欠陥やエラーを排除するという従来の考え方を改め、ハードウェアの構造欠陥やエラーを許容できるような新たな枠組みへ方向転換することにより道は開けるものと考えられる。

ハードウェア進化の考え方は、これまでのハード（物質）系とソフト（情報）系との関係に対して、そのような発想の転換を現実のものとするができる。つまり、ハードウェア進化の枠組みでは、ハード側は構造欠陥やエラーを排除することに躍起になることはない。その代わりに、たとえ精度は多少悪くとも大量のハードウェア資源をソフト側に提供すればよい。ソフト側は、大量のハードウェア資源を利用して、必要とするハードウェア構造を情報に依存して成長・進化させる。ハードウェア進化がもたらす新たな発想に立てば、ハード・ソフト双方に大きなブレークスルーをもたらすことができると考える。

冗長な集団系を用いてどのような情報処理が新たに可能となるのか、に答えるもののひとつがこのハードウェア進化である。ハードウェアの構造欠陥や動的なエラーを許容し、さらに、進化に活用することさえできる、ハードウェア進化の考え方は情報系と物質系を融合化・統合化する新しい方法論へと展開できるものと考えられる。

6.3 人工脳の構築に向けて

生体の脳では、生誕前後から幼少期にかけて神経細胞が大量に発生する。不思議なことに、それらの大半が予めプログラムされていたかのように死んでしまう。そのようなプログラムされた細胞死はアポトーシスと呼ばれる。大半が死んでしまうとはいえ、それでも 140 億ともいわれる神経細胞が生き残る。生き残った神経細胞は、他の神経細胞からの信号を受け取るための樹状突起と、他の細胞に信号を出力するための軸索を成長させる。軸索と樹状突起との結合部がシナプスとなり、軸索がシナプスを介して樹状突起に接続されることでネットワークが形成される。

幼児期の脳が柔軟性や創造性を有するのは、そのように脳のハードウェア構造が成長過程にあり柔らかいからといえるのかもしれない。軸索と樹状突起とを結ぶシナプスの感度は可変である。シナプスの感度を調整することによって、脳は学習することができる。つまり、シナプスの感度は、学習を繰り返す、経験を積むことによって影響を受け、少しずつ変化してネットワークの構造が固定化されていく。

しかし、そのような発生と成長の後、脳の神経細胞は再生されることなく減り続ける。このことは脳自身がかかなりの冗長性を有していることの例証でもある。

さて、脳は生体とともに死に、脳（つまり人）がいくら学習や経験を積んでも子孫には直接的には遺伝子しか残せない。そのような発生・成長・学習を、遺伝を通して何万世代に渡って繰り返しながら、即ち、何万世代にも及ぶ生と死を繰り返しながら、脳は進化してきた（図6-2）。

仮に人工脳というものが実現できた場合はどうであろう。人工脳にはいろんな利点が考えられる。ひとつは、柔軟性が要求される時には、全体でも部分的にでも何度でも幼児期に戻し、神経細胞を増やしてネットワークを張り替えることができる。生体の脳はそれを身体で支えるために生体に見合った大きさに限定されるが、人工脳は限定されない。

そして、最も大きな利点は、人工脳は死ななくてもよいことである。生物学的な脳は、後天的なものは引き継げず、長い年月に渡る膨大な数の世代交代を経て人間の脳まで進化してきた。しかし人工脳は死ななくてもいい。当初から人間の脳のように複雑で高度な人工脳を目指さなくとも、最初は魚の脳から始めればいい。魚の脳をうまく学習・成長させて、その外側に爬虫類の脳、その外側に哺乳類、さらに大脳新皮質というように、いうなれば重層的に人工的な脳を構築していくことができる。脳の部位は視覚、聴覚、運動、言語といった機能ごとに分割されるが、或いはそのような部

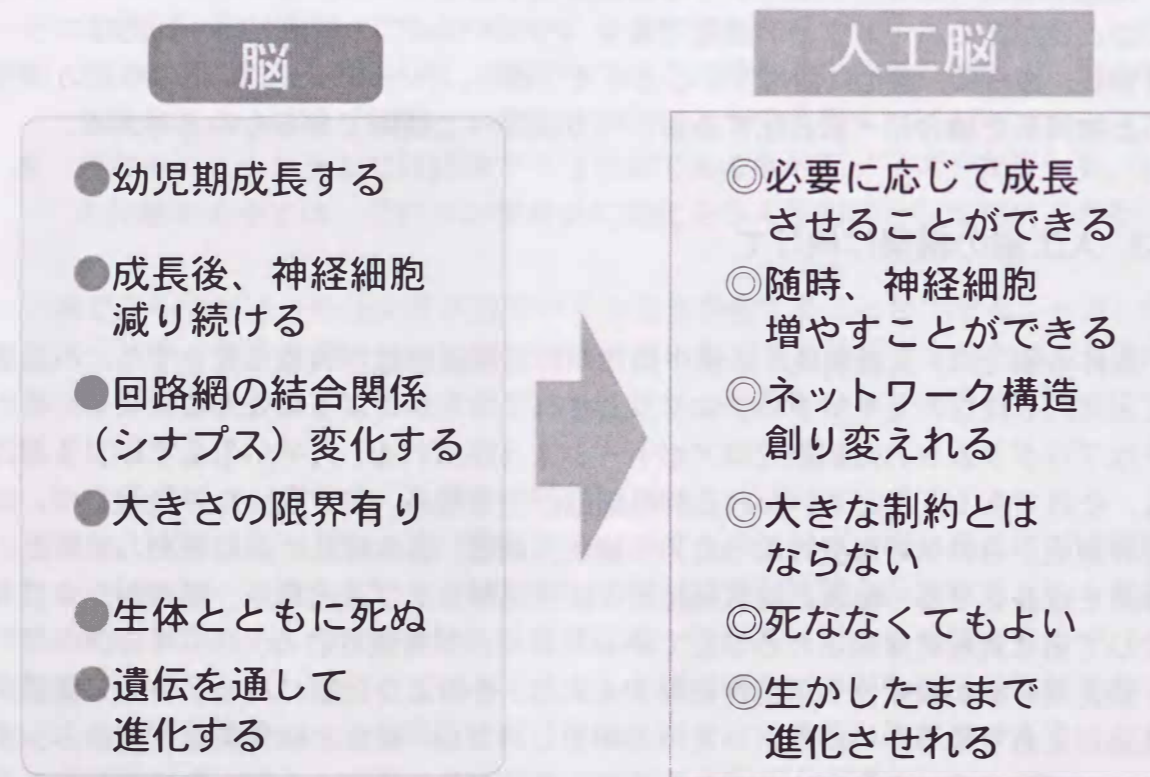


図6-2 生体脳と人工脳

位ごとに人工脳を発生・成長させ、それらを統合しながら発達させることもできるであろう。

脳の発生、成長、発達、機能分化、進化をモデル化し、多種多様な情報処理構造や処理様式を電子のスピードで、かつ、異なる条件で並列にシミュレーションすることができる。人工脳は、生体の脳が生体であるがために課されている限界を打ち破ることができるかもしれない[1]。

本論文で論じたセルオートマトン型人工脳実験装置（CBM）がそのようなシミュレーションのための有力な道具になるであろう。そのためにも、CBMの拡張・展開の延長上で、CBMを基にした人工脳のプロトタイプづくりが今後の課題となる。

人工脳は一種類というわけではない。人工的な脳の発生、成長、発達、機能分化、進化を、電子のスピードで、かつ、異なる条件で並列にシミュレーションできるため、多種多様な人工脳を増殖、進化させることができる。どのような機能を、どのように進化させ、どのように統合化するか、或いはまた、どのような処理をそれに行わせ、人がそれとどのように付き合うかによって、成長・進化した人工脳は異なるものとなるはずである。ネットワーク・ティアにおいて、デジタル生物の多様性を支えるためには進化環境の多様性が必要であった。そのためには地域性が大事であったように、多様かつ個性的な人工脳を成長・進化させるためにはローカリティ（地域性）を活かすことが重要である。

多種多様な人工脳を情報インフラによって相互にネットワーク化することによって人工脳の社会をつくることも可能となるであろう。ネットワーク化された複数の人工脳によって、ひとつの社会脳が構築できるかもしれない。これらは未だアイデアに過ぎないが、人工脳はそのスケールや速度のみならず生体の脳を超える大きな可能性を秘めている。

6.4 感性技術への展開

現在のコンピュータは、ある意味で、合理性や論理性を尊ぶ左脳的な側面を強化・拡張した、いわば理性の化身といえることができるかもしれない。或いはまた、最適化や効率化を重視し物質的豊かさを追及してきた、これまでの科学技術の申し子であるといってもよいかもしれない。科学技術は確かに物質的豊かさを我々にもたらした。しかし、物質的豊かさに比べて、心の豊かさについてはとても誇れる状況にはないように感じる。

自律性・創造性を有するコンピュータ、“生き物”みたいなコンピュータは、感性的な情報表現やクリエイションの分野でも有力な方法論となり得ると考える。人の想像性や創造性を喚起することは人の理性よりは感性に訴えることでもあるからだ。ここで、感性とは、心や情動といった論理では説明できない人のより生物学的な、より

本能的・感覚的なものを想定している。コンピュータが自律性と創造性を持ち、人がコミュニケーションできる、つまり、人の想像性や創造性を喚起し内部世界を広げてくれる“生き物”のような存在にコンピュータがなり得たとき、コンピュータは、理性の化身としてではなく、人が感性を通わすことができる存在へと変身できるのではないかと考える。

本研究のひとつの展開として、コンピュータの創り出す自律性や創造性と人がごく日常的な関わりを持つことによって可能となるコミュニケーションについて論じてみたい。具体的には、人と進化型コンピュータとの相互的な関わり（インタラクション）を利用したコンピュータ・グラフィックス（CG）の自動生成である。

6.4.1 理性的のみならず感性的な存在としてのコンピュータ

これまでの科学技術がある意味で人の関与を極力減らす方向に進歩してきたのに対して、これから望まれる技術とは人の自発的な関与を喚起し促す技術といえることができるであろう（図6-3）。

人の自発的な関与をいかに喚起し促すか…。ひとつは人の想像力（イマジネーション）や創造性を刺激すること。もうひとつは、人の自発的行為としての創造（クリエイション）を可能にし、かつ、容易にすることだと考える。

そこで、ひとつの視点として、人の自己表現を支援し、誰もが簡単に自己表現できる仕組みを考えてみたい。人は、“自己表現したい”という欲求を持っている[4]。そして、自己表現のひとつとして何かを創造することに喜びを感じる。またそれを認められることにも喜びを感じる。特別の技術もスキルもないごく普通の人々が、ごく日常的なインタフェースを介して、自らの内部状態に依存したコンピュータ映像や音楽の表現ができる。進化型コンピュータが創り出す自律性や創造性と人がごく日常的な関わりを持つことによって、そのような創造的コミュニケーションが可能となるものと考えられる。

6.4.2 自己表現や感性表現の支援に向けて

そのような技術開発に向けた研究コンセプトが、感性の主体としての人と、コンピュータが創り出す自律的な創造性との即興的なインタラクションである（図6-4）。そのような人とコンピュータとの即興的なインタラクションによって映像を自動的に生成する仕組みを考えることができる[3]。

いわば、カラオケの映像版である。我々は声を出して歌うことで楽しくなれる。ストレスを解消することができる。自分の声以外、つまり、作詞、作曲、演奏は全てプロによるものであるが、それでも我々は自分の声で歌うことで十分に自己表現できる。

コンピュータを使って絵を描き表現するには、プログラムを組み、ツールを使い

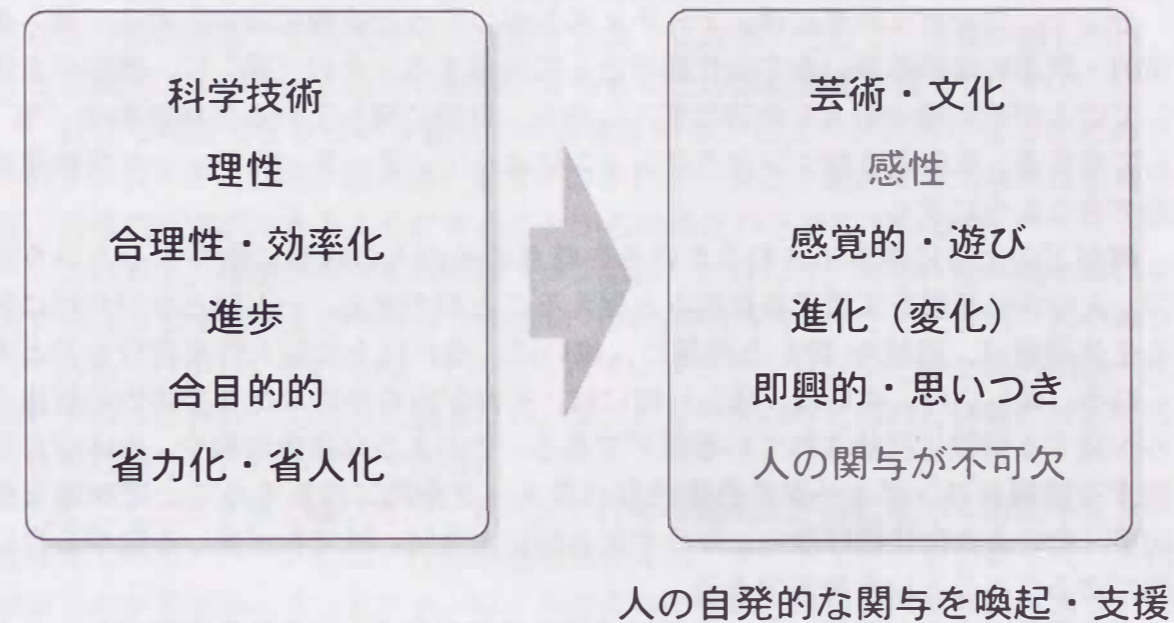


図6-3 コンピュータの果たす役割の変化

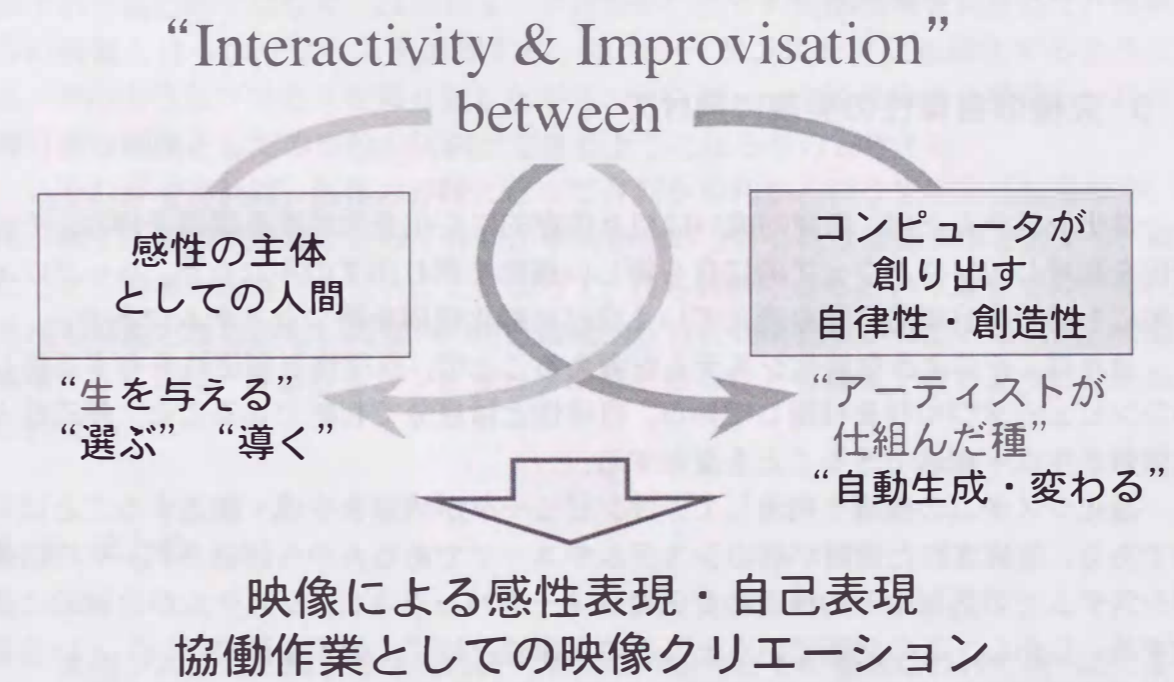


図6-4 創発的映像生成のコンセプト

こなすといった特別なスキルや技術が要求される。従って、ごく普通の人にはなかなか映像や音楽による自己表現は難しいのが現状である。

そこで、コンピュータには、アーティストがつくった映像生成のための“種”を自律的・創造的な振る舞いをする仕掛けとして用意する。その“種”に、感性の主体としての人がある時その人しか持たず、しかも、自然に発している生体情報を“生”として与える。そのようなインタラクションによって、アーティストックな映像を創造できるようにする。

何がどのように創り出されるかはその時点のその人の状態に依存するという意味で、人の自己表現を支援する仕組みと捉えることができる。一人ひとりが自然に発する生体情報は、指紋や DNA と同様に、或いは、それ以上に個人性を含むものと考えられる。楽しい時、或いは、悲しい時には、それが何らかのかたちで体内の情報や体外へ発する情報に反映されているはずである。そのような体内情報や、生体が自然に発する情報をコンピュータの自律的なパラメータ空間に投影することで映像を創り出す。そのような仕掛けなら、カラオケと同じように、誰でもが楽しみながら自己表現できるだろうという発想である。

ごく普通の人々が気軽にかつ気楽に映像や音楽を創造し、それらを身近な人々との間で交換する、そのようなコミュニティに根ざしたコミュニケーションの世界が広がることを期待できる。自律性、創造性を有し、生き物のような進化型のコンピュータは、それなりに満足できる質の高い映像や音楽の生成を通してその人独自の自己表現を支援することによって、コミュニティでのマルチメディア・コミュニケーションの活性化に大きく貢献できるものと考ええる。

6.5 究極の自律性の実現に向けて

進化システムとは、自発的或いは相互依存的に変化を生成する機構を持ち、その変化を利用してソフトウェア的に自ら新しい機能を創り出すのみならず、ハードウェア的にも構造を自律的に創り変えていく自己組織化機構を持つシステムである。

我々は、そのような進化システムを考えることで、自律性と創造性を有する新しいコンピュータづくりを目指している。自律性とは自分で判断できること、創造性とは情報を生成・創造できることを意味する。

進化システムの機構を利用して、コンピュータが情報を生成・創造することは可能である。生成された情報が他のシステムやユーザである人から評価され、その結果がシステムでの処理やその構造の変化にフィードバックされ、システムが自律的に進化する。しかし、この文脈でいうところの自律性とは“人の手を借りない”という意味の自律性である。以下では、“自分で判断できる”という意味での自律性をコンピュータに持たせることの可能性を考えてみる。

産まれたばかりの赤ん坊は、当初わけもなく手足をばたつかせる。そして、少し

ずつ手足をうまく操ることができるようになる。つまり、自分が制御できるものとして手足を認識するようになる。自ら生成した行動に対する母親なりの反応を見ながら、自らの行動の影響や意味を徐々に理解するようになる。そして、母親が話しかけ、教える言葉（シンボル）と自分の制御できるものとの対応関係を獲得していく。

自分で判断できるという意味での自律性をコンピュータに持たせるためのひとつのアプローチとして、我々は、まず、コンピュータに上記のような身体性を獲得させ、自他の区別ができるようにすることから始めたいと考えている [4]。

これまで、外部から与える情報（シンボル）に対して、ある行動や振る舞いを生成するコンピュータ上のキャラクタはいろいろ存在する。“たまごっち”もそのひとつである。しかし、それらは、そのようなインタラクションやそのための刺激・反応系を予め人が設計しているに過ぎない。コンピュータは与えられた情報に対応する処理を単に実行しているだけである。

そこで、第5章で論じたハードウェア行動の進化モデルに基づく仮想キャラクタを考えてみる。ハードウェア行動の進化モデルは、オートマトンを自動生成し進化させることができる。オートマトンは、入力された情報と自らの内部状態に基づいて出力情報を発生し、次々に内部状態を変えていく。従って、行動の進化モデルを用いて、いろんなオートマトンを自動生成する。例えば、“右手を上げる”という人が与える情報に対して正しい行動を生成するようオートマトンを淘汰し、進化させる。そのようなことを繰り返すうちに、人が与えるシンボルに対する行動（即ち、一連の動きを組織化したもの）をコンピュータ内部に自己組織化できるのではないかと考える。人が予め仕組むのではなく、コンピュータ自身が、自らの内部機構を利用して、外界からの情報と自らのリソースを制御する一連のシーケンスを自己組織化するようにする。そのようなプロセスを繰り返しながら、コンピュータは身体性を獲得し、自ら制御可能な範囲としての自他の区別ができるようになるものと考ええる。

それができれば、個体の存続にとって有利か不利かという生物的価値基準や、個体が属する集団や組織にとって有利か不利かという社会的な価値基準を設定し、可変にできる機構と組み合わせることによって、自ら判断できるという意味での自律性の実現も可能となるかもしれない。自己組織的な自己判断機構が可能となれば、情動や感情、さらには、自意識をコンピュータに持たせることも挑戦できるようになるかもしれない。

6.6 まとめ

進化システムに関する今後の研究展開の可能性とその意義について述べた。まず、情報に依存してハードウェア構造を可変にし、進化的な方法論によりハードウェアを進化させるハードウェア進化の考え方は、現在明確に区分されている情報系と物質系を融合・統合化する新しい方法論へ展開できることを述べた。人工脳の構築に向けて

は、セルオートマトン型人工脳実験装置を用いていかに研究を展開するかについて述べ、人工脳が、生物的脳が生体であるがために課されている限界を打ち破る可能性を有することを述べた。また、進化システムの新たな応用可能性として、コンピュータが創り出す自律性・創造性と人との即興的インタラクションをコンセプトとする感性技術への応用展開について述べた。さらに、システム進化の機構を利用してコンピュータが自己組織的に自己判断機構や身体性を獲得し、情動や自意識を持つことの可能性について考察した。

第6章の参考文献

- [1] 下原勝憲:「人工生命と進化するコンピュータ」, pp. 1-183, 工業調査会, 1998.
- [2] 下原勝憲, 邊見均: 自己増殖型人工脳, Computer Today, No. 90, 3月号, pp. 4-9, 1999.
- [3] 下原勝憲:「感性・人間・コンピュータ」, pp. 9-46, 富士通ブックス, 1995.
- [4] 下原勝憲: 生物型情報技術とバリアフリーなサイバー社会, 日本社会情報学会誌 第11号, pp. 48-56, 1999.

第7章 結論

本論文では、自律性と創造性を有する新たな情報処理系の創出を目指した進化システムの構成法について論じた。コミュニケーションの意義をコミュニケーションするもの同士が想像性や創造性を喚起し合いお互いの内部世界を拓げることと捉え、人とコンピュータとのそのようなコミュニケーションを可能とすることが狙いである。即ち、人とのインタラクションを通じて自ら変化・成長し、自ら情報を生成・創造し、かつ、自ら判断できる、いわば、生き物のようなコンピュータの実現が究極の目的である。

本研究では、情報を生成・創造する機能（創造性）や自ら判断する機能（自律性）を実現するための枠組みとして、生き物のように変化・成長・進化するシステムを提案した。即ち、進化システムとは、自発的・相互依存的に生成される変化を利用して、ソフトウェアが新しい機能を自ら創造するとともに、ハードウェアの構造をも自律的に創り変えていくことができるシステムである。このような研究分野は野心的・挑戦的である反面、そこには未だ確立された方法論はなく、多様なアプローチから可能性を探りつつ研究を展開せざるを得ない。

本論文では、進化システムを実現するための基本的な考え方を明らかにして、ソフトウェア進化システムおよびハードウェア進化システムのシステム構成法を提案し、有効に機能・動作することを示した。合わせて、このような方法論の可能性と意義について考察した。各章で報告した研究内容は以下の通りである。

第1章では、研究の背景として、人と人とのコミュニケーションと同様な人とコンピュータとのコミュニケーションを実現する上でシステムが具備すべき自律性と創造性の意義について考察し、自律性と創造性を有する情報処理系を創出するための方法論としての本研究の目的について述べた。

第2章では、システム進化機構を実現するための基本的な考え方について考察し、生命体的な発展性を持ち、かつ、社会的モデルとしても普遍化できる考え方を提案した。具体的には、同様な考え方を採る人工生命を新しい情報処理パラダイムとして捉え直し、進化を情報操作のプロセスとしてモデル化した進化的方法論に基づき、進化システムを構成するための基本思想について論じた。即ち、要素集団群とそれらの相互作用に基づくシステム構成を前提に、(1) 要素集団群とそれらの相互作用が変化する機構を有し、(2) それらの変化に応じてシステム構造を動的に生成し、(3) さらに、システム構造生成のプロセスが要素集団群とそれらの相互作用の変化機構に影響を与えることにより、システムを発展・進化させることを基本思想とした。その基本思想を“進化と創発”、“ミクローマクロの情報循環”の2つの研究コンセプトとして整理し、進化システム構築の方法論の枠組みについて論じた。

第3章では、進化システム構成法の核技術としてソフトウェア進化システムの構成法について論じた。一様なメモリ空間を環境とし、自然淘汰と突然変異により単細胞生物型（シングルスレッド型）のプログラム進化を実証したデジタル生態系ティアラを発展させて、多様かつ複雑なメモリ空間を提供できるコンピュータ・ネットワークを環境とし、多細胞型（マルチスレッド型）のプログラム進化を実現するシステム構成法を明らかにした。また、シミュレーション実験の結果からソフトウェア進化の可能性を確認し、本アプローチの意義と今後の展開について述べた。

第4章では、進化システム構成法のもうひとつの核技術であるハードウェア進化システムの構成法について論じた。再構築可能なハードウェアを前提に、情報に依存してハードウェア構造を生成し、生成されたハードウェア構造を評価することによって元となった情報を選択し、変化を加え、その情報からハードウェア構造を生成するプロセスを繰り返すことによって、所望のハードウェア構造を得るハードウェア進化の考え方を提案した。具体的には、3次元セルオートマトン空間にニューラルネットをハードウェアとして発生・成長・進化させるセルオートマトン型人工脳モデルのシステム構成法を論じ、有効に動作することを示した。さらに、本モデルの高速シミュレーションのために開発したセルオートマトン型人工脳実験装置の構成について述べた。

第5章では、ハードウェア記述言語（HDL）に基づくプログラムの自動生成と進化的方法論を組み合わせたハードウェア進化システムの構成法について論じた。ハードウェアに容易に変換できるHDLプログラムをハードウェアと同一視する観点から、ハードウェアの動作を記述するHDLプログラムを自動生成する方法を提案した。さらに、HDLプログラムを進化させるシステム構成法を明らかにして、シミュレーション実験により本システムの機能・動作を確認した。

第6章では、人工脳の構築に向けた方法論としての進化システムの今後の研究展開に関して、その可能性と意義について論じた。人工脳の構築に向けた研究展開のみならず、ソフトウェアとハードウェアの融合化に向けた研究展開、新たな方向性としての感性技術の研究、さらには、究極の自律性の実現に向けた研究展開などについて考察した。

以上、本論文では、進化システムを構成する上での考え方や課題を明らかにして、ソフトウェア進化およびハードウェア進化を実現するシステム構成法を示した。

謝辞

本論文の執筆をお勧めいただくとともに、本研究をまとめるにあたり、種々の暖かいご指導をいただいた九州大学大学院システム情報科学研究所長 牛島和夫 教授に心から深謝いたします。

本論文の内容に関し、ご意見、ご助言をいただいた九州大学大学院システム情報科学研究所 荒木啓二郎 教授ならびに 長谷川 勉 教授に心から感謝いたします。

本論文の研究内容は、ATR 人間情報通信研究所において行ったものです。このような研究を遂行する機会をいただき、また、日頃からご指導いただく 葉原耕平 元人間情報通信研究所会長（現在 NTT アドバンステクノロジー顧問）、酒井保良 人間情報通信研究所会長、東倉洋一 元人間情報通信研究所社長（現在 NTT 先端技術総合研究所所長）並びに一ノ瀬 裕 人間情報通信研究所社長に心より感謝いたします。

本研究は多くの方々との共同研究や討論を経たものです。Thomas S. Ray 氏（米国オクラホマ大学）、Hugo de Garis 氏（ベルギー・スターラボ）、邊見 均氏、Sung-Bae Cho 氏（韓国ヨンセン大学）、Andrzej Buller 氏、木目沢 司氏、村上恵子さんをはじめとする人間情報通信研究所進化システム研究室の多くの研究員の方々に心から感謝します。

索引

あ

アポトーシス..... 103

い

EDIF フォーマット 77
移住..... 41
遺伝形質..... 29
遺伝子..... 8, 30, 41, 60
遺伝子型..... 9, 14, 74, 77
遺伝子型メモリ..... 76
遺伝子重複..... 47
遺伝子バンク..... 27
遺伝子変異..... 14
遺伝情報..... 8
遺伝的アルゴリズム14, 60, 65, 72, 77, 83
遺伝的プログラミング.... 45, 83, 99
インターネット..... 36

え

A-Life..... 11
エキスパート・システム..... 13
SFL..... 84
HDL..... 8, 81
HDL プログラム..... 81
HDL 文法..... 84
FPGA..... 54, 75, 81

お

OS..... 27
オートマタ..... 54
オートマトン..... 109

か

開始記号..... 86

解析木..... 84
外部インタフェース..... 78
書き換えシステム..... 84
書き換えルール..... 86
仮想キャラクタ..... 109
CAM-8..... 72
CAM-Brain..... 55, 59
感性..... 105, 106
感性技術..... 105
カンブリア爆発..... 33

き

木構造..... 83
寄生種..... 30
共生種..... 30

く

空間トポロジー..... 35
群衆..... 47

け

Getipp 命令..... 37
ゲノム..... 9, 41
ゲノム情報..... 9, 27, 37, 45

こ

交叉..... 9, 27, 53, 59, 86
興奮性..... 74
CoDi..... 60
CoDi モデル..... 59
子細胞..... 27, 45
固定..... 14
コネクショニズム..... 13
コレクショニズム..... 11
コンフィグレーション・データ.... 81

さ

再構成可能なハードウェア..... 53
 最適設計..... 13
 サイバー世界..... 48
 細胞型..... 41
 細胞死..... 103
 細胞組織..... 41
 細胞分裂..... 44
 Xilinx..... 75
 詐欺種..... 30
 削除..... 87
 サブプロセス..... 91

し

CBM..... 72, 105
 軸索..... 55, 56
 軸索成長信号..... 58, 61
 軸索セル..... 60
 自己組織化..... 16
 自己表現..... 106, 108
 自己複製..... 14, 25
 自然淘汰..... 9, 14, 26
 シナプス..... 55, 56
 死に神..... 29
 社会的な種..... 30
 社会的モデル..... 11, 17
 社会脳..... 105
 終端記号..... 86
 集团的冗長設計..... 13
 集中制御..... 13
 樹状突起..... 55, 56
 樹状突起成長信号..... 58, 61
 樹状突起セル..... 60
 出力セル..... 63
 状態更新則..... 57, 58
 John Muir Trail..... 91
 自律性..... 6, 7, 108
 進化..... 16

進化システム..... 4, 7, 15
 進化的方法論..... 7, 11
 進化と創発..... 7, 11, 16
 神経回路網..... 55
 神経細胞..... 54
 人工蟻..... 91
 人工進化..... 26
 人工生命..... 7, 11
 人工脳..... 55, 72, 101, 104, 105
 身体性..... 109
 真理値表..... 98

す

スープ..... 37
 スケーラビリティ..... 98
 助っ人エージェント..... 48
 SIIC..... 77
 Spike Interval 情報符号化..... 77
 スパイク状信号..... 64
 スレッド..... 25, 41

せ

成長..... 14
 成長信号..... 58
 成長フェーズ..... 61
 生物集団的な情報処理..... 13
 セル..... 56
 セルオートマトン..... 8, 53
 セルオートマトン型人工脳..... 8
 セルオートマトン型人工脳実験装置72, 105
 セルオートマトン空間..... 56
 セルタイプ..... 60
 セルラーオートマトン..... 54
 センサ細胞..... 41
 センサプロセス..... 44, 46
 染色体..... 9, 60, 86

そ

相互参照メモリ..... 77
 創造性..... 6, 7, 108
 創発..... 11, 16
 組織型..... 50
 祖先種生物..... 29
 ソフトウェア進化..... 7, 19, 25

た

第2の自己..... 6
 第3者のな存在..... 6
 代謝..... 27
 大変動..... 36
 大量絶滅..... 36
 多細胞生物..... 32
 多細胞デジタル生物..... 33
 多産性..... 46
 多様性..... 13, 33
 単細胞生物..... 32

ち

重複..... 87, 99

て

TPing データ..... 37, 44
 DNA..... 8
 ティエラ..... 7, 25
 ティエラ・サーバ..... 36
 ティエラ言語..... 27
 デジタル生態系..... 25
 デジタル生物..... 26
 TCP/IP..... 37
 適応戦略..... 14
 適応度..... 65, 76, 91
 適応度評価..... 74, 76
 テスタビリティ..... 103
 伝言ゲーム..... 5
 電子生物..... 25

テンプレート・マッチング..... 27

と

淘汰..... 5, 9
 淘汰圧..... 9, 47
 突然変異..... 25, 26, 29, 53, 59
 トップダウン..... 13

な

ナノ・エレクトロニクス..... 103
 ナノ・テクノロジー..... 103

に

ニューラルネット..... 14, 53
 ニューラルモジュール..... 72, 74
 入力セル..... 63
 ニューロン..... 55
 ニューロンセル..... 60

ね

ネットワーク・ティエラ... 8, 25, 32

の

脳内情報処理..... 17

は

ハードウェア記述言語..... 8, 81
 ハードウェア進化..... 7, 20, 53
 ハード外進化..... 83
 ハード内進化..... 83
 発生..... 14
 発生機構..... 11
 発達..... 14
 ハミング距離..... 65

ひ

Beagle..... 36
 ビーグル..... 36
 BNF..... 84
 非終端記号..... 86

ビット反転.....	29
表現型.....	9, 14, 53, 65, 74
表現型メモリ.....	76
ふ	
VRML.....	37
副本.....	86
複雑系.....	11
複雑適応系.....	11
複製細胞.....	41
ブランクセル.....	60
プロダクションルール.....	86
分化.....	41
へ	
並列分散制御.....	13
変異.....	9, 87
ほ	
ホストコンピュータ.....	76
ボトムアップ.....	11
ま	
マスタプロセス.....	91
マップファイル.....	37
マルチスレッド.....	25, 41

み	
ミクローマクロの情報循環.....	7, 11, 16
め	
免疫種.....	30
も	
モジュール間接続.....	77
ゆ	
融合.....	87
優先度.....	37
よ	
抑制性.....	74
り	
リーパー.....	29
流動.....	14
量子セルオートマトン.....	79
量子デバイス.....	103
わ	
World Wide Web.....	36
渡り鳥.....	33

発表論文

<学術論文>

1. SHIMOHARA, K. TOKUNAGA, Y., MATSUI, H. : An Information Terminal for A Local Community, IEEE Communication Magazine, Vol. 26, No. 5, pp. 52-58, 1988.
2. SHIMOHARA, K., UCHIYAMA, T., TOKUNAGA, Y. : Subconnection Neural Network for Event-Driven Temporal Sequence Processing, Neural Networks, Vol. 6, No. 5, pp. 709-718, 1993.
3. 秋山健二, 大塚作一, 佐藤裕一, 下原勝憲, 沢田立夫: デジタルスケッチホン, 研究実用化報告第33巻, 第8号, pp. 1933-1948, 1984.
4. BULLER, A., SHIMOHARA, K. : Decision Making as A Debate in The Society of Memes in A Neural Working Memory, J. Three Dimensional Images, Vol. 13, No. 3, pp. 77-82, 1999.
5. CHO, S. B., SHIMOHARA, K. : Evolutionary Learning of Modular Neural Networks with Genetic Programming, Applied Intelligence, Vol. 9, No. 3, 191-200, 1998.
6. HEMMI, H., SHIMOHARA, K. : Hardware Evolution A Real "Life on the Silicon," Int. J. Artificial Life and Robotics, Vol. 1, No. 4, pp. 191-195, 1998.
7. HEMMI, H., HIKAGE, T., SHIMOHARA, K. : On Mixed-level Hardware Evolutionary System, Int. J. Artificial Life and Robotics, Vol. 3, No. 4, pp. 209-212, 2000.
8. HEMMI, H., SHINOZAWA, K., HIKAGE, T., SHIMOHARA, K. : Learning and Relocation Capability of CAM-Brain Machine, Int. J. Artificial Life and Robotics, Vol. 3, No. 4, pp. 213-216, 2000.
9. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : Progressive Evolution Model using a Hardware Evolution System, Int. J. Artificial Life and Robotics, Vol. 2, No. 4, pp. 157-161, 1998.
10. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : Progressive Evolution Model and its Dynamics, J. Three Dimensional Image, Vol. 13, No. 3, pp. 83-88, 1999.

11. HIRAIWA, A., SHIMOHARA, K., TOKUNAGA, Y.: EEG Topography Recognition by Neural Networks, IEEE Engineering in Medicine and Biology, Vol. 9, No. 3, pp. 39-42, 1990.
12. 平岩明, 内田典佳, 下原勝憲, 曾根原登: 筋電操作ハンドの制御のための皮膚表面筋電信号のニューラルネットによる認識, 計測自動制御学会論文誌, Vol. 30, No. 2, pp. 216-224, 1994.
13. 平岩明, 内田典佳, 下原勝憲, 曾根原登: 随意運動発生直前の頭皮電位分布パターンのニューラルネットによる認識, 電子情報通信学会論文誌 A, Vol. J79-A, No. 2, pp. 408-415, 1996.
14. HIRAIWA, A., UCHIDA, N., SHIMOHARA, K., SONEHARA, N: Neural Network Recognition of Electro-Encephalogram Patterns Preceding Voluntary Movements, Electronics and Communications in Japan, Part 3, Vol. 80, No. 7, pp. 65-73, 1997
15. 磯俊樹, 渡部保日児, 下原勝憲: 自動監視システムのための顔分類方法—不審人物検出アルゴリズムの検討, 画像電子学会論文誌, 第25巻, 第4号, pp. 328-338, 1996.
16. KANAYA, I., SHIMADA, T., HEMMI, H., SHIMOHARA, K.: Evolutionary Design of Analog Electrical Circuits using Genetic Programming, J. Three Dimensional Images, Vol. 13, No. 3, PP. 89-94, 1999.
17. 川嶋功, 中野慎夫, 永井直文, 下原勝憲: 複合通信機能をもつデジタルボタン電話, 研究実用化報告, 第30巻, 第7号, pp. 1889-1902, 1984.
18. MATSUO, N., SHIMOHARA, K., MATSUI, H., TOKUNAGA, Y.: Personal Telephone Services Using IC-Cards, IEEE Communication Magazine, Vol. 27, No. 7, pp. 41-48, 1989.
19. MIZUTANI, S., SANNO, T., SHIMOHARA, K.: Enhanced Resonance by Coupling and Summing in Sinusoidally Driven Chaotic Neural Networks, IEICE Fundamental, Vol. 82-A, No. 4, pp. 648-657, 1999.
20. 森本正志, 末永康仁, 下原勝憲: 座標変換パラメータの並列推定に基づく複数距離情報の統合, 電子情報通信学会誌 D-II, Vol. J80, No. 2, pp. 548-557, 1997.

21. 篠沢一彦, 下原勝憲, 曾根原登, 徳永幸生: ヘシアン行列を用いた忘却抑制法, 電子情報通信学会論文誌 D-II, Vol. J82-D-II, No. 7, pp. 1190-1198, 1999.
22. TAKADAMA, K., HAJIRI, K., NOMURA, T., SHIMOHARA, K., NAKASUKA, S.: Organizational Learning Model for Adaptive Collective Behaviors in Multiple Robots, Advanced Robotics, Vol. 12, No. 3, pp. 243-269, 1998.
23. TAKADAMA, K., HAJIRI, K., NOMURA, T., OKADA, M., NAKASUKA, S., SHIMOHARA, K.: Learning Model for Adaptive Behaviors as an Organized Group of Swarm Robots, Int. J. Artificial Life and Robotics, Vol. 2, No. 3, pp. 123-128, 1999.
24. 高玉圭樹, 寺野隆雄, 下原勝憲, 堀浩一, 中須賀真一: マルチエージェント学習における知識の再利用と電気回路設計への応用, コンピュータソフトウェア, Vol. 16, No. 5, pp. 37-49, 1999.
25. TAKADAMA, K., TERANO, T., SHIMOHARA, K., Hori, K., NAKASUKA, S.: Making Organizational Learning Operational: Implication from Learning Classifier System, Computational and Mathematical Organization Theory, Vol. 5, No. 3, pp. 229-252, 1999.
26. 高玉圭樹, 寺野隆雄, 下原勝憲, 堀浩一, 中須賀真一: 組織学習に基づく分散分類システムを用いた創発的問題解決への接近, 計測自動制御学会論文誌, Vol. 35, No. 11, pp. 1486-1495, 1999.
27. 立居場光生, 下原勝憲: 幾何光学による到達点変動の解析における問題点, 電子通信学会論文誌, Vol. 60-B, No. 12, pp. 982-984, 1977.
28. TATEIBA, M., SHIMOHARA, K.: Spot Dancing in Inhomogeneous Random Media, Transactions of the IECE of Japan, Vol. E61, pp. 38-39, 1978.
29. 立居場光生, 下原勝憲: 不均質ランダム媒質におけるスポットダンシング, 電子通信学会論文誌, Vol. 61-B, No. 1, pp. 64-71, 1978.
30. 筒口拳, 末永康仁, 渡部保日児, 下原勝憲: 3次元シーン内の人物像歩行動作生成システム, 情報処理学会論文誌, Vol. 38, No. 4, pp. 783-796, 1997.
31. 内山匡, 下原勝憲: リカレントニューラルネットワークのための実時間学習アル

ゴリズム, 電子情報通信学会論文誌 D-II, Vol. J73-D-II, No. 8, pp. 1158-1163, 1990.

32. 山田辰美, 下原勝憲, 橋本秋彦, 安達文夫, 徳永幸生: 遺伝的アルゴリズムを用いたシームレステクスチャ生成方法, 電子情報通信学会論文誌 D-II, Vol. J82-D-II, No. 11, pp. 2017-2025, 1999.
 33. YASUURA, K., MIYAMOTO, N., SHIMOHARA, K.: Numerical Analysis of a Thin-film Waveguide by Mode-Matching Method, J. of the Optical Society of America, Vol. 70, No. 2, pp. 183-191, 1980.
- <国際会議>
34. SHIMOHARA, K., YUITO, M., TOKUNAGA, Y.: Modified Voice Telecommunications and Their Protocol Models, IEEE Int. Conf. on Communications, pp. 1274-1279, 1986.
 35. SHIMOHARA, K., UCHIYAMA, T., TOKUNAGA, Y.: Back-Propagation Networks for Temporal Sequence Processing, IEEE Int. Conf. on Neural Networks '88, Vol. I, pp. 665-672, 1988.
 36. SHIMOHARA, K., TOKUNAGA, Y., UCHIYAMA, T., KIMURA, Y.: A Neural Network System with an Automatic Generation Mechanism for Distorted Patterns, International Neural Network Society (INNS'88), p. 52, 1988.
 37. SHIMOHARA, K.: Evolutionary Systems for Brain Communications-Towards an Artificial Brain, Artificial Life IV, Proc. 4th Int. Workshop on the Synthesis and Simulation of Living Systems, pp. 3-7, 1994.
 38. SHIMOHARA, K.: Evolutionary Systems for Brain Communications, Proc. Notebook for Int. Symposium on Low-Power and High-Speed Chips (COOL Chips II) pp. 37-50, 1999.
 39. SHIMOHARA, K., LIU, J. Q.: DNA Computing -A Novel Paradigm of Artificial Life in vitro, Proc. 5th Int. Symposium on Artificial Life and Robotics. (AROB 5th '00), Vol. 2, pp. 777-780, 2000.
 40. BULLER, A., CHODAKOWSKI, T., SHIMOHARA, K., HEMMI, H.: CoDi Technique:

Cellular Automata as Neural Networks, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 1, pp. 192-195, 2000.

41. BULLER, A., SHIMOHARA, K.: Does The Butterfly Effect Take Place in Human Working Memory?, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 1, pp. 204-207, 2000.
42. CHO, S. B., SHIMOHARA, K.: Modular Neural Networks Evolved by Genetic Programming, Proc. 1996 IEEE Intl. Conf. Evolutionary Computation (ICEC'96), pp. 681-684, 1996.
43. CHO, S. B., SHIMOHARA, K.: Development of Modular Neural Networks by Evolutionary Algorithm, Proc. Int. Conf. Neural Networks (ICNN'96), pp. 149-152, 1996.
44. CHO, S. B., SHIMOHARA, K.: Emergence of Structure and Function in Evolutionary Modular Neural Networks, Proc. Fourth European Conf. on Artificial Life (ECAL'97), pp. 197-204, 1997.
45. CHO, S. B., SHIMOHARA, K.: Cooperative Behavior in Evolved Modular Neural Networks, Methodologies for the Conception, Design and Application of Soft Computing, Proc. 5th Int. Conf. on Soft Computing and Information /Intelligent Systems (IIZUKA '98), Vol. 2, pp. 606-609, 1998.
46. CHO, S. B., SHIMOHARA, K.: Applying L-System to Development of Evolutionary Modular Neural Networks, Proc. Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL 98), Vol. 2, Session 22, Evolving Neural Networks, 1998.
47. CHO, S. B., SHIMOHARA, K.: Grammatical Development of Evolutionary Modular Neural Networks, Lecture Notes in Artificial Intelligence, 1585, Proc. Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL 98), pp. 413-420, 1999.
48. HEMMI, H., MIZOGUCHI, J., SHIMOHARA, K.: Hardware Evolution An HDL Approach, Proc. 1994 Japan-U. S. A. Symposium on Flexible Automation, pp. 1285-1288, 1994.
49. HEMMI, H., MIZOGUCHI, J., SHIMOHARA, K.: Development and Evolution of

Hardware Behaviors, Artificial Life IV, Proc. 4th Int. Workshop on the Synthesis and Simulation of Living Systems, pp.371-376, 1994.

50. HEMMI, H., SHIMOHARA, K. : Hardware Evolution A Real "Life on the Silicon," Proc. 1st Int. Symposium on Artificial Life and Robotics (AROB'96), pp.191-194, 1996.
51. HEMMI, H., MIZOGUCHI, J., SHIMOHARA, K. : Evolving Large Scale Digital Circuits, Artificial Life V, Proc. 5th Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, pp.213-218, The MIT Press, 1997.
52. HEMMI, H., SHINOZAWA, K., HIKAGE, T., SHIMOHARA, K. : Learning and Relocation Capability of CAM-Brain Machine, Proc. 3rd Int. Symposium on Artificial Life and Robotics (AROB III '98), Vol.1, pp.267-270, 1998.
53. HEMMI, H., HIKAGE, T., SHIMOHARA, K. : An Interactive mechanism for CAM-Brain, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol.1, pp.196-199, 2000.
54. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : Progressive Evolution Model using a Hardware Evolution System, Proc. Int. Symposium on Artificial Life and Robotics (AROB'97), pp.18-21, 1997.
55. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : Hardware Evolution System Introducing Dominant and Recessive Heredity, Lecture Notes in Computer Science, Evolvable Systems: From Biology to Hardware, Vol. 1259, First International Conf. on Evolvable Systems from Biology to Hardware (ICES '96), pp.423-436, 1997.
56. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : On Mixed-level Hardware Evolutionary System, Artificial Life and Robotics, Proc. 3rd Int. Symposium on Artificial Life and Robotics (AROB III '98), Vol.1, pp.196-199, 1998.
57. HIKAGE, T., HEMMI, H., SHIMOHARA, K. : Comparison of Evolutionary Methods for Smoothing Evolution, Lecture Notes in Computer Science, 1478, Evolvable Systems: From Biology to Hardware, Second Int. Conf. on Evolvable Systems (ICES'98), pp.115-124, 1998.
58. HIRAIWA, A., SHIMOHARA, K., TOKUNAGA, Y. : EMG Pattern Analysis and

Classification by Neural Networks, IEEE Int. Conf. on Systems, Man and Cybernetics, pp.1113-1115, 1989.

59. HIRAIWA, A., SHIMOHARA, K., TOKUNAGA, Y. : EEG Topography Recognition by Neural Networks, IEEE Int. Conf. on Systems, Man and Cybernetics, pp.1116-1117, 1989.
60. HIRAIWA, A., UCHIDA, N., SHIMOHARA, K. : EMG/EEG pattern recognition by neural networks, Proc. the Eleventh European Meeting on Cybernetics and Systems Research, pp.1383-1390, 1992.
61. HIRAIWA, A., UCHIDA, N., SHIMOHARA, K. : EMG pattern recognition by neural networks for Prosthetic Fingers Control, Proc. 1992 IFAC/IFIP/IMACS Int. Symposium on Artificial Intelligence in Real-Time Control, pp.413-419, 1992.
62. INOUE, H., TAKADAMA, K., SHIMOHARA, K., OKADA, M., KATAI, O. : Agent Architecture Based on Self-Reflection Learning Classifier System, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol.2, pp.454-457, 2000.
63. ISO, T., WATANABE, Y., SHIMOHARA, K. : Human Face Classification for Security System, Proc. IEEE Int. Conf. on Image Processing (ICIP'96), pp.479-482, 1996.
64. KASAHARA, H., TAKADAMA, K., NAKASUKA, S., SHIMOHARA, K. : A Troubleshooting Mechanism based on an Organizational Learning Model for Multiple Robots, Proc. 1998 Int. Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC '98), Vol. II, pp.1023-1026, 1998.
65. KASAHARA, H., TAKADAMA, K., NAKASUKA, S., SHIMOHARA, K. : Fault Tolerance in Organizational based on an Organizational Learning Model, Proc. 1998 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC '98), pp.2261-2266, 1998.
66. KAWASHIMA, I., NAKANO, S., NAGAI, N., SHIMOHARA, K. : Terminal Control Equipment for Integrated Telecommunications Service, 38th NEC Symposium, pp.181-186, 1982.
67. LIU, J. Q., SHIMOHARA, K. : DNA Computing by Genomic Dynamics II: -A Simulation

- Wetware Prototype of Dynamical DNA Computation, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 2, pp. 785-788, 2000.
68. LIU, J. Q., SHIMOHARA, K. : DNA Computing by Genomic Dynamics I: -Evolutionary Modeling of Emergence and Context-Sensible Grammar Representation, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 2, pp. 781-784, 2000
69. MAEKAWA, T., UENO, O., NISHINA, E., KAWAI, N., SHIMOHARA, K., OOHASHI, T. : Evolutionary Advantage of Self-Decomposition Mechanism, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 1, p. 40-44, 2000.
70. MAEKAWA, T., UENO, O., NISHINA, E., KAWAI, N., SHIMOHARA, K., OOHASHI, T. : Diffusible Immortal ALife Rarely Exterminate Diffusible Mortal ALife in One Finite, Heterogeneous Ecosystem, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB5th '00), Vol. 1, p. 34-39, 2000.
71. MIYATAKE, T., SHIMOHARA, K. : Autonomous Graphics Generation System for Supporting Human Self-Expression, 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, pp. 473-478, 1998.
72. MIZOGUCHI, J., HEMMI, H., SHIMOHARA, K. : Production Genetic Algorithms for Automated Hardware Design through an Evolutionary Process, Proc. 1st IEEE Conf. on Evolutionary Computation (ICEC'94), Vol. II, pp. 661-664, 1994.
73. MIZUTANI, S., SANO, T., UCHIYAMA, T., SONEHARA, N., SHIMOHARA, K. : Numerical Study of Resonance in a Sinusoidally Driven Chaotic Neuron Model and Its Network, Proc. Int. Symposium on Artificial Life and Robotics (AROB'97), pp. 131-135, 1997.
74. MIZUTANI, S., SANO, T., UCHIYAMA, T., SONEHARA, N., SHIMOHARA, K. : Numerical Studies of Resonance in a Sinusoidally Driven Chaotic Neuron Model and Its Network, Proc. Int. Conf. on Neural Networks (ICNN97), pp. 663-668, 1997.
75. MIZUTANI, S., SANO, T., SHIMOHARA, K. : Resonance in a Global Coupling Chaotic Neural Network Driven by a Weak Sinusoid, Int. Conf. on Neural Information Processing (ICONIP'97), Vol. 1, pp. 178-181, 1997.
76. MIZUTANI, S., SANO, T., SHIMOHARA, K. : Resonance in Chaotic Neural Networks Driven by a Weak Sinusoid, Int. Joint Conf. on Neural Network '98, pp. 2551-2556, 1998.
77. MIZUTANI, S., SHIMOHARA, K. : Ordering in the Mean Field of a Chaotic Neural Network with Quenched Disorder, Proc. 1998 Int. Symposium on Nonlinear Theory and its Applications (NOLTA '98), pp. 707-710, 1998.
78. MIZUTANI, S., SHIMOHARA, K. : Heterogeneity Enhanced Order in a Chaotic Neural Network, European Symposium on Artificial Neural Network '99 (ESANN99), pp. 147-152, 1999.
79. NAWA, N. E., TAKADAMA, K., SHIMOHARA, K., KATAI, O. : On The Efficiency of a Decentralized Continuous Double Auction Market of Locally Aware Agents - Bridging The Gap Between Economics and Interactive Self-Reflection, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 2, pp. 446-449, 2000.
80. NOMURA, T., SHIMOHARA, K. : A Description of Quasi-Autopoietic Systems based on the Framework of (M, R) Systems, Proc. 3rd Int. Symposium on Artificial Life and Robotics (AROB III '98), Vol. 2, pp. 658-661, 1998.
81. OHTANI, S., ISHIDO, M., SHIMOHARA, K. : Multi-Agent Based Neural Network as A Dynamical Brain Model, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 1, pp. 200-203, 2000.
82. OKADA, N., TAKADAMA, K., OKADA, M., KATAI, O. : Interactive Self-Reflection Model for Communications, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 2, pp. 438-441, 2000.
83. ONITSUKA, A., VAARIO, J., SHIMOHARA, K., UEDA, K. : Behavior Generation based on Enhanced DLA Model on Neural Network, Proc. First Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL'96), S10-4, pp. 454-461, 1996.
84. OOHASHI, T., MAEKAWA, T., UENO, O., NISHINA, E., KAWAI, N., SHIMOHARA, K. : Proposal of a General Simulator "SIVA" for Heterogeneous Environment and Self-Decomposable ALife, Proc. 5th Int. Symposium on Artificial Life and

Robotics (AROB 5th '00), Vol. 1, pp. 45-49, 2000.

85. SAGATA, Y. S., SHIMOHARA, K., TONOMURA, Y.: A Study of Computer Agents to Help Younger Children, 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, pp. 59-64, 1998.
86. SHINOZAWA, K., UCHIYAMA, T., SHIMOHARA, K.: An Approach for Solving Dynamic TSPs Using Neural Networks, IEEE/INNS Int. Joint. Conf. on Neural Networks '91, pp. 2450-2454, 1991.
87. SHINOZAWA, K., SHIMOHARA, K.: A Method for Reducing Forgetfulness of Incremental Learning, Int. Conf. on Neural Information Processing (ICONIP '97), Vol. 1, pp. 296-299, 1997.
88. SHINOZAWA, K., SHIMOHARA, K.: An improved method for reducing the forgetfulness in incremental learning, Proc. 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'99), Vol. IV, pp. 1068-1073, 1999.
89. TAKADAMA, K., HAJIRI, K., NOMURA, T., OKADA, M., SHIMOHARA, K.: Learning Model for Adaptive Behaviors as an Organized Group of Swarm Robots, Proc. Int. Symposium on Artificial Life and Robotics (AROB'97), pp. 103-106, 1997.
90. TAKADAMA, K., HAJIRI, K., NOMURA, T., OKADA, M., SHIMOHARA, K., NAKASUKA, S.: A Computational Group Dialogue Model with Organizational Learning, Proc. IEEE Int. Conf. on Intelligent Processing Systems, pp. 174-179, 1997.
91. TAKADAMA, K., HAJIRI, K., NOMURA, T., NAKASUKA, S., SHIMOHARA, K.: Organizational Knowledge on Formation in Multiple Robots Learning, Proc. 3rd Int. Symposium on Artificial Life and Robotics (AROB III '98), Vol. 2, pp. 397-401, 1998.
92. TAKADAMA, K., HAJIRI, K., NOMURA, T., NAKASUKA, S., SHIMOHARA, K.: Reinforcement Learning for Multiple Robots with Organizational Learning, Proc. 3rd Int. Symposium on Artificial Life and Robotics (AROB III '98), Vol. 1, pp. 392-396, 1998.
93. TAKADAMA, K., HAJIRI, K., NOMURA, T., NAKASUKA, S., SHIMOHARA, K.: Design for Sequence Behavior Rules in Multiple Robots Reinforcement Learning, Intelligent Autonomous Systems (IAS-5), pp. 327-334, 1998.

94. TAKADAMA, K., HAJIRI, K., NOMURA, T., NAKASUKA, S., SHIMOHARA, K.: Adaptation to Multiple Robots Organization with Organizational Knowledge on Formation, Proc. Fifth Int. Conf. on the Society of Adaptive Behavior (SAB '98), From Animals to Animates 5, pp. 483-488, 1998.
95. TAKADAMA, K., TERANO, T., SHIMOHARA, K., HORI, K., NAKASUKA, S.: Toward Emergent Intelligence in Multiagent Learning, Proc. 4th Int. Symposium on Artificial Life and Robotics (AROB'99), pp. 464-467, 1999.
96. TAKADAMA, K., TERANO, T., SHIMOHARA, K.: How to Design Good Learning Agents in Organization?, Proc. Genetic and Evolutionary Computation Conf. 99 (GECCO'99), Vol. 2, pp. 1398-1405, 1999.
97. TAKADAMA, K., KASAHARA, H., Huang, L., WATABE, M., II, H., SHIMOHARA, K., NAKASUKA, S.: Organizational Learning Agents for Task Scheduling in Space Crew and Robot Operations, Proc. 5th Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'99), pp. 561-568, 1999.
98. TAKADAMA, K., TERANO, T., SHIMOHARA, K., HORI, K., NAKASUKA, S.: Can Multiagents Learn in Organization? -Analyzing Organizational Learning Oriented Classifier System-, Workshop on Agents Learning About, From and With Other Agents, 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99), 1999.
99. TAKADAMA, K., TERANO, T., SHIMOHARA, K.: Agent-Based Model Toward Organizational Computing: From Organizational Learning of Genetics-Based Machine Learning, Proc. 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'99), Vol. II, pp. 604-609, 1999.
100. TAKADAMA, T., WATABE, M., SHIMOHARA, K., NAKASUKA, S.: How to Design Good Rules for Multiple Learning Agents in Scheduling Problems?, Lecture Notes in Artificial Intelligence, 1733, 2nd Pacific Rim Int. Workshop on Multi-Agents (PRIMA'99), pp. 126-140, Springer, 1999.
101. TAKADAMA, K., HAJIRI, K., SHIMOHARA, K.: Interactive Self-Reflection Architecture Using Cellular Automata, Proc. 5th Int. Symposium on Artificial

Life and Robotics (AROB 5th '00), Vol. 2, pp. 458-461, 2000.

102. TAKADAMA, K., SHIMOHARA, K.: Interactive Self-Reflection Model: Aim and Overview, Proc. 5th Int. Symposium on Artificial Life and Robotics (AROB 5th '00), Vol. 2, pp. 430-433, 2000.
103. UCHIYAMA, T., SHIMOHARA, K., TOKUNAGA, Y.: Boltzmann Machines In Which Weights Converge To Global Optimal Solution, IEEE Int. Conf. on Neural Networks '88, Vol. I, 1988.
104. UCHIYAMA, T., SHIMOHARA, T., TOKUNAGA, Y.: A Modified Leaky Integrator Network for Temporal Pattern Processing, Int. Joint Conf. on Neural Networks, Vol. I, pp. 469-475, 1989.
105. VAARIO, J., SHIMOHARA, K.: On Formation of Structures, Proc. 3rd European Conf. on Artificial Life (ECAL'95), pp. 421-435, 1995.
106. VAARIO, J., SHIMOHARA, K.: Modeling Environment Sensitive L-Systems, Kyoto Conf. on Mathematical Biology '96, (KCMB'96) P-128, 1996.
107. VAARIO, J., OGATA, N., SHIMOHARA, K.: Synthesis of Environment Directed and Genetic Growth, Artificial Life V, Proc. 5th Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, pp. 244-251, The MIT Press, 1997.
108. VAARIO, J., ONITSUKA, A., SHIMOHARA, K.: Formation of Neural Structures, Proc. Fourth European Conf. on Artificial Life (ECAL'97), pp. 214-223, 1997.
109. VAARIO, J., SHIMOHARA, K.: Synthesis of Developmental and Evolutionary Modeling of Adaptive Autonomous Agents, Proc. 7th Int. Conf. on Artificial Neural Networks, pp. 721-726, 1997.
110. YAMADA, T., HASHIMOTO, A., ADACHI, F., SHIMOHARA, K.: Generating Seamless Texture using Genetic Information, Signal and Image Processing '98 (SIP'98), pp. 22-26, 1998.
111. YUITO, M., SHIMOHARA, K., TOKUNAGA, Y.: A New Telecommunications System with Modified Voice, Int. Symposium on Subscriber Loops and Services (ISSLS'86),

Vol. II-2, pp. 117-122, 1986.

<テクニカルレポート>

112. BULLER, A., CHODAKOWSKI, T., HEMMI, H., SHIMOHARA, K.: CoDi Technique: Cellular Automata as a Large-Scale Neural Network, ATR Technical Report, TR-H-277, 1999.
113. DE GARIS, H., KORKIN, M., SHIMOHARA, K.: ATR's "CAM-Brain Machine" (CBM) and Artificial Brains, ATR Technical Reports, TR-H-295, 2000.

<解説論文および書籍>

114. 下原勝憲: 人工生命, bit 別冊バーチャルリアリティ「仮想現実学への序曲」, pp. 124-131, 共立出版, 1994.
115. 下原勝憲: 脳コミュニケーションの進化/自律生成機構, 日本の科学と技術, Vol. 35, No. 272, pp. 57-61, 1994.
116. 下原勝憲: ロボット集団における生物学的コミュニケーションと協調動作—人工生命の視点から—, 日本ロボット学会誌, Vol. 12, No. 6, pp. 808-812, 1994.
117. 下原勝憲: 人工生態系ティエラ, 画像ラボ, Vol. 5, No. 10, pp. 55-58, 1994.
118. 下原勝憲: ニューラルネットとコンピュータ「ニューラルネットワーク応用」, pp. 15-26, オーム社, 1995.
119. 下原勝憲: 生命論パラダイムに基づく情報処理, 情報処理, Vol. 36, No. 4, pp. 289-296, 1995.
120. 下原勝憲: 人工生命パラダイム・人工生命の歴史・人工生命の現状・発生/発達モデル・脳コミュニケーションのための進化システムの研究, 「人工生命の方法」, pp. 11-39, 55-66, 153-162, 263-288, 工業調査会, 1995.
121. 下原勝憲: 人工生命 (II) —情報処理への応用と展望—, テレビジョン学会誌, Vol. 50, No. 11, pp. 1746-1754, 1996.
122. 下原勝憲: 新しい情報処理パラダイムとしての人工生命, 電子情報通信学会誌, Vol. 80, No. 2, pp. 195-196, 1997.

123. 下原勝憲: 進化システムの研究と展望, 画像電子学会誌, 第 26 卷, 第 5 号, pp. 517-523, 1997.
124. 下原勝憲, 邊見均, RAY, T., VAARIO, J., 和田健之介, 溝口潤一, DE GARIS, H., 今田彬: 脳コミュニケーションのための進化システム・進化システムとしての人工脳をめざして・今後の展望「人工生命と進化システム/ATR 進化システム研究室編」東京電機大学出版局, pp. 1-223, 1998.
125. 下原勝憲, 大橋力: デジタル生物と死のプログラム, 学術新報・Academia, 173号, pp. 56-62, 1998.
126. 下原勝憲, 米澤保雄: 人工生命の基礎と応用, 映像情報メディア学会編, 先端技術の手ほどきシリーズ「複雑系の理論と応用」, pp. 180-194, オーム社, 1998.
127. 下原勝憲: 生物型情報技術とバリアフリーなサイバー社会, 日本社会情報学会誌 第11号, pp. 48-56, 1999.
128. 邊見均, 溝口潤一, 下原勝憲: 行動型ハードウェアの進化, 遺伝的アルゴリズム 2, pp. 207-249, 産業図書, 1995.
129. HEMMI, H., MIZOGUCHI, J., SHIMOHARA, K.: Development and Evolution of Hardware Behaviors, Towards Evolvable Hardware, pp. 250-265, Springer-Verlag, 1996.
130. TAKADAMA, K., HAJIRI, K., NOMURA, T., SHIMOHARA, K., NAKASUKA, S.: Grammatical Learning Model for Adaptive Collective Behaviors in Multiple Robots, Grammatical Model of Multi Agent Systems, Topics in Computer Mathematics: 8, pp. 343-355, Gordon and Breach Science Publishers, 1999.
131. VAARIO, J., 溝口潤一, 下原勝憲: 創発と進化, シミュレーション, Vol. 13, No. 4, pp. 302-310, 1994.

受賞

著者	対象(論文名)	受賞内容	受賞年月日
平岩 明 内田典佳 下原勝憲	EMG/EEG pattern recognition by neural networks, Proc. of the Eleventh European Meeting on Cybernetics and Systems Research, 1992 (Vienna, Austria)	Best Paper Award (Program Committee)	平成4年4月
ATR 人間情報通信 研究所進化システム 研究室 代表: 下原勝憲	進化システム・人工生命の研究により, 学術, 技術, 芸術の3つの世界を一つに融合し, 21世紀に先駆する業績を世に送り出している文化創造集団	日本文化デザイン大賞 (日本文化デザインフォーラム)	平成7年10月
Sung-Bae Cho 下原勝憲	Cooperative Behavior in Evolved Modular Neural Networks, Proc. 5th Int. Conf. on Soft Computing and Information/Intelligent Systems (IIZUKA '98), 1998 (Iizuka, Japan)	Best Paper Award (プログラム実行委員会)	平成10年10月

