セルオートマトンルール獲得のための進化計算設計に関する研究

セルオートマトンルール獲得のための進化計算設計に関する研究

亀島 力*，　遠藤 聡志**，　山田 孝治**

Research work for architectonics of Evolutional Computation in acquiring CA-Rules

Chikara KAMESHIMA*,　Satoshi ENDO**　and　Koji YAMADA**

Abstract

Evolving Cellular Automata (EvCA) is one of the methods for designing of CA-Rules by using evolutional computations. EvCA has two points of architectonic that must be considered for improve its performance. (1) The table of CA-Rules must be architected in conformity to the objective task.(2) Evolutional computations(EC) of EvCA must be architected for superior system of automatic-desining CA-Rules. In this paper,we propose following three new coding methods for EvCA. "Symmetric-Coding"and"Shifter-Coding"are techniques for solving CA-rules table problem, and "diploid model" is one of EC technique for EvCA. We analyze the results of our methods on computer experiments for Density Classification problem. Then, acquired CA-rules' performances are compared with GKL-rule which designed manually and has higher performablity.

Key Words: Cellular Automata, Evolving Cellular Automata,Evolutional Computations, Computational Task.

1. はじめに

複雑系現象を視覚的側面から擬似的に再現し，解析を行うツールに，セルオートマトン（Cellular Automata：CA）がある．CAを用いて現象を再現する場合，状態遷移則（CAルール）を解析対象に応じて設計しなければならない．解析対象についての完全な知識がない使用者にとって，この設計は容易ではない為，CAルール設計の自動化は必須である．

M.Mitchellは密度分類タスクを行うCAルールを，遺伝的アルゴリズム（Genetic Algorithms：GA）を用いて自動設計する実験を行ったが，heuristicルールであるGKLルールを越えるものは設計できなかった [Mitchell 93]．原因として，GAによる交叉オペレーションではタスク達成に有効なオートマトン出力が作成されにくい事，また破壊されやすい事が挙げられる．

本研究では，密度分類タスクに有効な処理が獲得可能なCAルール設計手法を提案する．また，ルール自動設計において，より優れたルールを探索できる進化計算として，CAの計算タスクの達成に有効な出力を多く保持したルールを生成する，2倍体モデルを用いた遺伝的操作手法を提案する．そして，これら提案手法と従来の進化計算手法との比較により，有効性を検証する．

受理：1999年6月7日
*大学院理工学研究科情報工学専攻
(Masters Course in Information Engineering, Graduate School of Science and Engineering)
**工学部情報工学科
(Dept. of Information Engineering, Fac. of Eng.)

2. セルオートマトン

全く同じ構造を持った有限オートマトンを空間内に格子状に配置し，近接するもの同士を相互に結合した系がCAである．それぞれのオートマトンを**セル**（Cell）と呼び、それらが同期的に動作することで空間全体の挙動が決定される．セルを直線上に並べて配置する場合を1次元、ある限られた平面状に敷き詰めて配置する場合を2次元のCAであるという．

セルが遷移する新しい状態は，それらが考慮する近傍に基づいて定義された**CAルール**（CA Rules）により決定する．セルの状態を a_t^i（iは位置座標，tはステップ）とすると，a_t^i は k 種類の内から1つの状態をとる．a_{t+1}^i は，隣接セルとして a_t^{i-r} から a_t^{i+r} までの $2r+1$ 個のセルを考慮し，隣のセルの状態を変数とする関数 $\mathcal{F}$ によって与えられる．

$$a_{t+1}^i = \mathcal{F}(a_t^{i-r}, a_t^{i-r+1}, \cdots, a_t^i, \cdots, a_t^{i+r-1}, a_t^{i+r}) \quad (1)$$

また，$\mathcal{F}$ をCAルールと呼び，$\mathcal{F}$ に与えられる a_t^{i-r} から a_t^{i+r} のセルを，a_t^i の近傍と呼ぶ．

例えば，セルの状態数を2（0と1），近傍数を7とした場合，7つの近傍の状態の組み合わせは $2^7 = 128$ となる．

CAルールは，これら128個のパターンに対して，それぞれ'0'あるいは'1'の2種類の出力を行うので，その規則の種類は 2^{128} 通りあることになる．

CAモデルを構築するには，環境内の物理法則や要素の挙動についての解析結果を基にしたルール設計を，手動で行わなければならない．しかし，要素の挙動が複雑な現象では，その複雑さを導出するCAルールを設計するために，膨大な量の解析が必要となる．さらに，CAはルールに基づく法則性を持つ一方，その大域的な挙動は，その規則からは予測が困難な意外性を見せるため，目的の挙動を示すCAルールを設計する事は容易ではない．このCAルール設計問題を解決する手法として，大域的挙動を明確に規定し，その処理を行うことができるCAルールを進化的計算を用いて生成する**進化型セルオートマトン（Evolving CA：EvCA）**が注目されている [Mitchell 93]．

3. 進化型セルオートマトン:EvCA

CAルール設計実験の対象として，1次元CAを用いた計算処理のひとつである**密度分類タスク**がある．

密度分類タスク（Density-Classification）

状態数 $k=2$ （状態が'0'か'1'）のCAにおいて，初期状態群（Initial Configuration(s)：IC(s)）が，それぞれの状態平均値（密度）が臨界値 ρ_c（c：critical）以上のときには，すべてのセルが'1'に遷移し，ρ_c 未満のときには，'0'に遷移するオートマトン処理を，CAの**密度分類タスク**という．特に，臨界値 ρ_c が 1/2 のとき，これを，$\rho_c = 1/2$ タスクと呼ぶ．

このタスクは局所的な情報から大域的な現象を引き起こさなければならなく，的確なCAルールを作ることは非常に難しい．このタスクを解くCAルールとして，**GKLルール** [Gacs 78] がある．このルールはheulisticなルールであり，IC全体（$2^{\text{格子サイズ}}$）における 10^4 のサンプリングに対して82.7％の達成率を示した．

heuristicルールの他に，**EvCA**を適用してCAルールを自動設計した研究がある．M.MitchellらはEvCAの進化的計算に単純なGA（SGA）を用いて，密度分類タスクを達成するCAルールを設計する実験を行った [Mitchell 93]．この実験で設計したルール（SGAルール）の，10^4 のサンプリングに対して76.1％であり，GKLルールの達成率を超えるものが設計できなかった．本章では，SGAを用いたEvCAの追実験を行い，高い達成率を示すルールが設計できなかった原因を検証した．

3.1 実験1：CAルール自動設計実験

格子数 $N=149$ の1次元CAで，SGAを用いたEvCAの追実験を行った．実験の手順を以下に示す．

step.1 **P個のCAルールからなる集団を，CAルールの出力平均値 λ=[0.0 1.0] の範囲で均等に分布するように生成する．**

step.2 **各ルールに対し，状態平均値 ρ=[0.0 1.0] の範囲で均等に分布する I 個のICsを生成し，それらをオートマトン処理する．**

step.3 **集団内における各ルールの適応度 F_I を計算し，ルールのランク付けを行う．**

step.4 **適応度の上位 E 個をエリート群として残し，$P-E$ 個の新たなルールを，GAを用いて生成する．**

step.5 **試行回数が終了条件を満たさなければ，Step.2へ戻る．**

実験のパラメータは $I=100, P=100, E=20$ とした．また，適応度計算式は（2）とした．

$$F_I = \text{タスクを達成した回数}/I \qquad (2)$$

実験の結果，IC全体（2^N）から100個のサンプリングに対して約80％の達成度を示すルールが設計された．

3.2 実験2：達成率の比較実験

自動設計したルールの性能を調べるために，密度分類タスクで高い達成度を示すheuristicルールである**GKLルール** [Gacs 78] とのタスク達成率を比較した．$\rho_c = 1/2$ タスクでは，$\rho_0 \approx 1/2$ のICを正しく分類することが難しいとされている．状態平均値 ρ=[0.4 0.6] の，10000個のICにおけるタスク達成率は，自動設計ルール（SGAルール）が69.46％，GKLルールが81.41％であった．GKLルールよりも高い性能を示すルールを自動設計することはできなかったといえる．

3.3 問題点の検証

3.3.1 有効なタスク達成処理

Fig.1は，'0'，'1'の状態値の並びが左右対称なICに対して，3章の実験で得られたSGAルールを用いて処理した結果である．図1の2つのICは ρ_0 が同値であるので，いずれも同じ状態に収束しなければならないが，片方のICを正しい収束状態に導くことができていない．これが，タスク達成率を引き下げる原因のひとつとなっている．

Fig. 1. 左右対称なIC（$\rho_0 = 0.73$）に対する処理

3.3.2 有効な出力の選択

実験1で用いたGAの交叉は，2つ個体の出力情報を組替える1倍体モデルであった．このモデルでは，CAルール同士の交叉を行うことで個体内の2つ以上の出力からなる戦略が切断される為，タスク達成に有効な処理が破壊され，次世代に残らない可能性が生じる（Fig.2）．

4. 提案手法

3.3節で取り上げた，CAルールの設計や，EvCAの進化的計算の問題点を解決する，CAルール設計と，GAの個体生成オペレーションに関する手法を提案する．

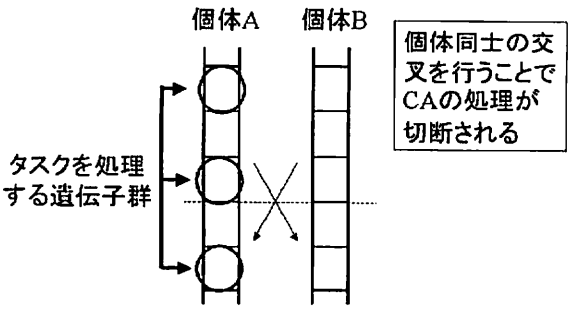

Fig. 2. 1倍体モデルの交叉で破壊されるCAルール

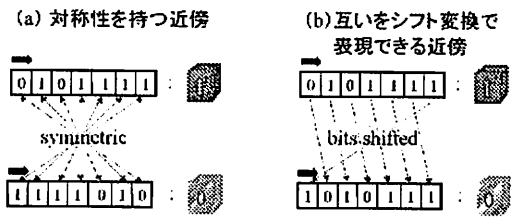

Fig. 3. 同系の近傍入力

4.1 ルール空間の圧縮

3.3.1節の問題を解決するために，一組の同系のIC[1]について，一方のICについての有効なタスク達成処理を，もう一方のICの処理に移植する手法を導入する．この手法により，対処可能なICを増加させることができ，タスク達成率を向上させることができると考えられる．有効なタスク達成処理を移植する機能を実現するために，CAルール空間内で同一のものと考えられる入力を統一し，ルール空間を圧縮する方策を提案する．

Symmetric-Code

任意の2つのセル近傍入力について，一方の近傍が他方の近傍の上位-下位桁の反転で表すことができる場合，これらの近傍は対称性を持つといえる（Fig.3(a)）．対称性を持つ2つの近傍に対する出力を統一すると，対称性を持った同系のICに関して，それらの処理が統一されるので，タスク達成率が向上すると考えられる．対称性を持つ近傍入力を統一したCAルール設計方法を，**Symmetric-Code**と呼ぶ．Symmetric-Codeを用いたCAルールのサイズは，単純2進符号を用いた場合に比べて56.25％に圧縮される．

Shifter-Code

Fig.3(b)の2つのセル近傍入力は互いに1bit-Shiftを行うことで表現できる関係である．1bit-Shiftで表現できる近傍に対する出力を統一すると，互いをShiftで表現できる同系のICに関して，それらの処理が統一されるので，タスク達成率が向上すると考えられる．Shiftで表現できる近傍入力を統一したCAルール設計方法を，**Shifter-Code**と呼ぶ．Shifter-Codeを用いたCAルールのサイズは，単純2進符号を用いた場合に比べて32.03％に圧縮される．

[1]密度分類タスクの性質上，0，1の状態値の並びが左右対称なICや，互いを1bit-Shiftで表せるICは，同一の処理でタスクを達成することが可能である．このようなICを，「同系のIC」と呼ぶ．

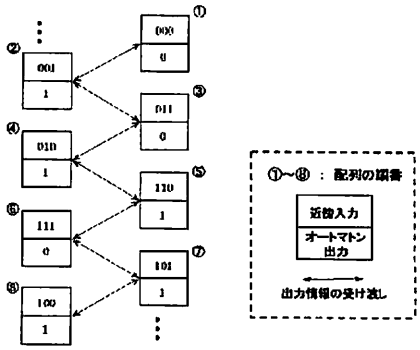

Fig. 4. 2倍体モデルを用いた個体生成

4.2 2倍体モデルを用いた個体生成

3.3.2節の問題を解決するために，個体内の各出力に対して，それがタスク達成に有効かどうかを判断し，選択するという機能を実装する手法を取りあげる．この手法により，タスク達成に必要な出力を個体内，集団内に広める効果が期待できる．この機能を実現する遺伝的操作として，CAルールの各出力の有効性を，過去に保持していた出力の履歴を元に評価する**2倍体モデル**を用いた．

CAルールは，2^7種類のルール要素を持つが，2倍体モデルではそれら各要素における，**出力優性度**というパラメータを新たに実装し，タスクに対する各出力の有効性を評価する機能を実現する．

あるルール要素に注目した時，そこの出力値は交叉や突然変異によって変動する．GAの選択あるいは淘汰により適応度の高い個体が得られると，その個体の優れた出力が継承されるようになる為，そのルール要素ではタスク達成に有効と考えられる値が出力されやすくなる．そこで，これら出力値の履歴を合計すると，そのルール要素においてどのような出力が有効であったかを示す出力優性度を得ることができる．

2倍体モデルで新たな個体を生成する場合，比較的近い内容の近傍入力（相同入力）をもつルール要素どうしで出力優性度を参照し合い，ルーレット選択で新たな出力値を決定する．こうして，有効出力を多く保持した個体を得ることが可能となる．

CAルールの各ルール要素が出力優性度の参照を行うためには，ルール配列内の隣り合うルール要素を，相同入力が隣接するように並べ替える必要がある．この問題を解決するCAルール設計手法として，Gray-Code [Gray 53]を用いた．以下に，2倍体モデルを用いた個体生成の手順を示す（Fig.4）．

step.1 CAルールのルール要素を，近傍入力がGray-Codeの順と一致するように配列する．

step.2 配列のすべての要素について，隣り合った要素

Table 1. 各提案手法とGKLルールとの比較

CA-rule	タスク達成率	ハミング距離
ϕ_{SGA}	69.46	60
$\phi_{Symmetric-Code}$	74.29	48
$\phi_{Shifter-Code}$	71.62	51
$\phi_{2倍体モデル}$	78.12	38

へ出力情報を与える操作を行う．各要素が情報を与える方向は，配列の昇順，降順のうちのいずれかとする．

step.3 新たな個体の出力を決定する．保有している出力情報と，受け取った出力情報が同一，すなわちホモであれば，その情報をその近傍における出力とする．同一でないヘテロならば，過去に保持していた出力の合計値で算出した出力優性度による，ルーレット選択で，新たな出力を決定する．

5. 計算機実験

各々の提案手法（Symmetric-Code，Shifter-Code，2倍体モデル）を実装した環境において，CAルール自動設計実験及び，達成率の比較実験を行った．

5.1 実験3：CAルール自動設計実験

提案手法を用いて，実験1と同様の実験を行った．実験1で設計したSGAルールと同様，3つの提案手法ルールは，IC全体（2^N）から100個のサンプリングに対して約80％の達成度を示すルールが設計された．

5.2 実験4：達成率の比較実験

実験3で設計したルールに対して，実験2と同様，ρ=[0.4 0.6]の，10000個のICsにおけるタスク達成率の比較を行った．実験の結果は，SGA，Symmetric-Code，Shifter-Code，2倍体モデルの順に，69.46％，74.29％，71.62％，78.12％であった．提案手法を用いた自動設計ルールは，いずれもSGAルールよりも高い達成率を得ることができた．Symmetric-Code，Shifter-Codeのルールが，同系のICについての問題を解決していると考えられる．また，2倍体モデルではタスク達成に有効な出力の選択が有効に作用し，高い達成率を得ることができている．

5.3 考察

実験において設計した各ルールとGKLルールを比較し，各ルールの性能を評価した．Table1は，各自動設計ルールとGKLルールとのハミング距離を計測した結果である．2倍体モデルの距離が最も小さいことから，GKLルールが持つタスク達成処理に関して，その他のルールよりも近似しているといえる．また，Symmetric-CodeとShifter-Codeについて，Symmetric-Codeのタスク達成率が高いにもかかわらず，ハミング距離に関して大きな差が見られない．これらの結果をさらに詳しく解析する為に，以下のような検証を行った．

CAルールの$2^7 = 128$個のルール要素は，セルが参照

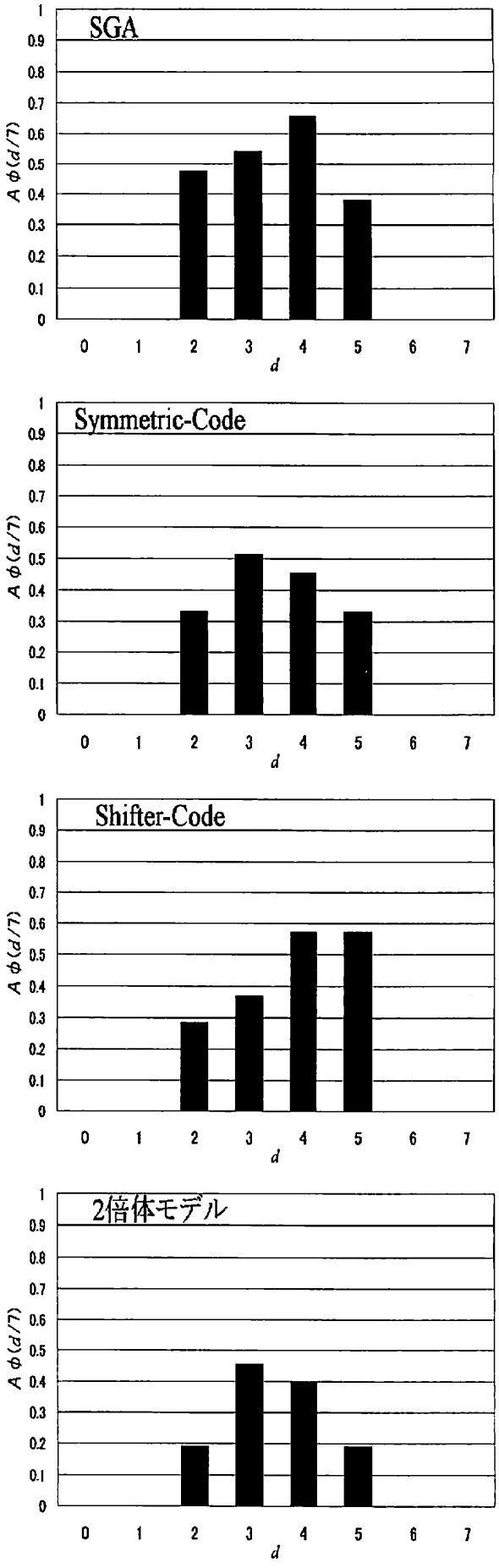

Fig. 5. GKLルールとの誤差$A\phi(d/7)$

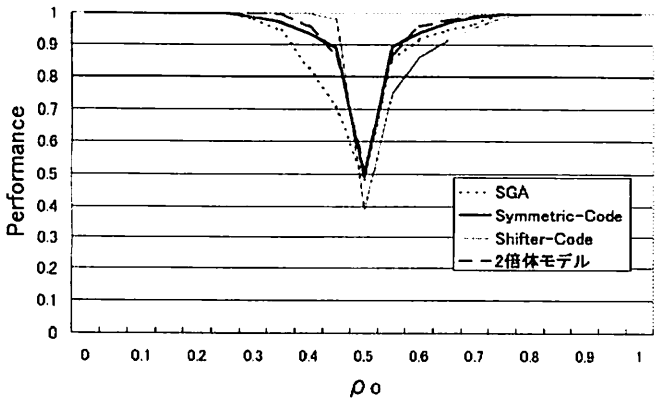

Fig. 6. 初期状態平均値に対するタスク達成率

する近傍に含まれる'1'の数dによって分類できる．近傍数が7であるCAでは$d = \{0,1,2,3,4,5,6,7\}$の8種類である．$d = x$の時の，CAルールϕとGKLルールの誤差率$A\phi(x/7)$を計測した（Fig.5）．

$$A(x/7) = \gamma(x/7)/M_x \qquad (3)$$

$\gamma(x/7)$	$d = x$の時の，GKLとのハミング距離．
M_x	$d = x$であるルール要素の数．

いずれの自動設計ルールも，$A\phi(0/7)$〜$A\phi(1/7)$，$A\phi(6/7)$〜$A\phi(7/7)$において違いが見られないことから，$d = \{0,1,6,7\}$のオートマトン出力について，密度分類タスクを達成するための同一の処理を獲得しているといえる．$A\phi(3/7)$，$A\phi(4/7)$では，Shifter-Code，Symmetric-Code，2倍体モデルの誤差率が，SGAと比較して小さい．このことから，$d = \{3,4\}$に関して，3つの提案手法ルールは既存の手法に比べて，GKLルールに追随したタスク処理を獲得しているといえる．さらに，$A\phi(2/7)$，$A\phi(5/7)$に関しては，SGA，Shiftert-Codeの誤差率が，タスク達成率の高いSymmetric-Code，2倍体モデルよりも大きいことから，高いタスク達成率を示すCAルールは，$d = \{2,5\}$において，タスク達成に有効な処理が獲得できているといえる．ハミング距離に関してSymmetric-Codeとほぼ等しいShifter-Codeが同等なタスク達成率を示さないのは，$d = \{2,5\}$における有効な処理が獲得できていない為であると考えられる．

Fig.6は，$\rho_0 \approx 0$から$\rho_0 \approx 1$のICを0.05刻みにそれぞれランダムに100生成し，提案手法ルールとSGAルールのタスク達成率を計測した結果である．どのルールも，$\rho_0 \approx 1/2$で達成率が低くなっている．SGAルールが$\rho_0 > 1/2$で，Shifter-Codeルールが$\rho_0 < 1/2$でタスク達成率が偏って高くなっている．これに対して，Symmetric-Code，2倍体モデルルールはIC全体に対するタスク達成率の偏りが少なく，かつ高い達成率を示した．この結果より，複数のICをそれぞれ固有の目標に近づけるタスクにおいて高い達成率を示すCAルールは，任意の目標に対する達成率の偏りが少ないという性質を持つということがわかる．

6. まとめ

本研究では，CAルール獲得実験の対象として密度分類タスクを取りあげ，Symmetric-Code，Shifter-Codeを用いたCAルールの設計手法を提案した．また，CAルールを自動生成するGAの個体生成オペレーションについて，2倍体モデルを実装した手法を提案した．提案手法を，タスク達成率の評価実験に適用した結果，いずれの手法も，従来の進化計算手法を用いた場合よりも高い達成率を示すルールが設計された．特に，Shifter-Codeを用いた手法と2倍体モデルを実装した手法は，IC全体に対するタスク達成率の偏りが少ない，高い達成率を示すルールが得られた．任意のCAタスクにおいて有効な，CAルール設計及び進化計算設計を自動化する機能を開発し，より一般性の高いCAルール自動獲得モデルへの改善を図る必要がある．

謝辞

本研究は，文部省科学研究費（課題番号10780240）の補助を受けて行った．

参考文献

[Gray 53] F.Gray. "Pulse Code Communication",*U.S.Patent*2 632 058,March 17. 1953.

[Gacs 78] P.Gacs, G.L.Kurdyumov, L.A.Levin. One-dimensional uniform arrays that wash out finite islands. *Probl Peredachi. Inform.*, 14:92-98. 1978.

[Langton 86] C.G.Langton. Studying artificial life with cellular automata. *Physica D*, 22:120-149. 1986.

[Packard 88] N.H.Packard. Adaptation Toward the Edge of Chaos. In J.A.S.Kelso, A.J.Mandell, M.F.Shlesinger, eds., *Dynaamic patterns in Complex Systems*, 293-301. Singapole:World Scientific. 1988.

[Langton 90] C.G.Langton. Computation at the edge of chaos: phase transition and emergent computation. *Physica D*, 42:12-37. 1990.

[Mitchell 93] M.Mitchell, P.T.Hraber, J.P.Crutchfield. Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations. *Complex Systems*,In press. 1993.

[Das 94] Rajarshi Das, M.Mitchell, J.P.Crutchfield. A Genetic Algorithm Discovers Particle-Based Computation in Cellular Automata. Y.Davidor, H.P.Schwefel, and R.Manner (editors),*Parallel Problem Solvingx in Nature* **PPSN III**, Lecture Notes in Computer Science, 344-353, Springer-Verlag, Berlin, 1994.

[Mitchell 96] M.Mitchell, J.P.Crutchfield, Rajarshi Das. Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96)* . Moscow, Russia: Russian Acadamy of Sciences, 1996.

[Moreles 96] F.Jimenez Morales, J.P.Crutchfield, M.Mitchell. Evolving Two-Dimensional Cellular Automata to Perform DensityClassification: A Report on Work in Progress.

[Eckart 97] J.D.Eckart. Cellang: Language Reference Manual. "http://www.cs.runet.edu/ dana/". 1997.

[加藤 98] 加藤恭義,光成友孝,築山洋. セルオートマトン法, 森北出版. 1998.

[亀島 00] 亀島力,山田孝治,遠藤聡志. 進化型セルオートマトン：ルール自動設計システムに関する考察. 第12回自律分散システム・シンポジウム資料, 417-422, 2000.