種分類を用いた共進化によるセルオートマトンの近傍則獲得

遠藤聡志[†] 山田孝治[†] 亀島力[††,☆]

セルオートマトン（Cellular Automata：CA）を用いて複雑系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適切に設計しなければならない．CA ルール設計のテスト問題として，1 次元 CA 密度分類タスクがある．このタスクは局所情報から大域的目的現象を再現しなければならないという CA ルール設計特有の困難さがある．Juillé らの示した共進化モデルは，GA の解探索に加えて，解集団の効率的な進化を促す問題空間の探索を利用することで探索性能の改善を図り，当タスクのベストレコードを示すルールの自動設計に成功している．本研究では，CA ルール群中の類似性を解集団（CA ルール群）から抽出し，それらを「種」という形で具体化したうえで，すべての「種」を包含するような解を求める．この遺伝的操作は，現象内に内包される局所的規則性（サブタスク）を基にして，サブタスク解間の CA ルール内での整合性を高めるものである．Juillé らの手法が問題依存型のクラスタリングと heuristic な適応度計算式の拡張を導入したのに対し，提案手法はモデル化する現象に応じたクラスタリングと，種統合のための適応度計算式の自動的な調整を行う点が特色である．

CA Rule Acquisition Using Speciation Based Co-evolution

Satoshi Endo,[†] Koji Yamada[†] and Chikara Kameshima[††,☆]

We describe an application of Evolutionary Computations to the design of Cellular Automata that can perform computations requiring global coordination. On recent works, Co-evolutionary Learning was used to acquire CA-Rules for the computational task that called “density classification”. In this task, ordinary ECs discovered rules that didn’t give rise to high performance and sophisticated strategy. Some reason presented this experiment seem to prevent continuous progress in evolutionary search. The major reason is that the dynamics of the search performed by the two co-evolving populations doesn’t drive individuals to the domain of the state space that contains most promising solutions because there is no “high-level” strategy to play that role. Our approach proposes a co-evolutionary framework in which those two issues are addressed as follows: (1) Classification of CA-Rules into some of “species” by the use of its state of affairs in CA task. (2) Allocating the perfect “Fitness-Function” for each species so as to aggregate own strategy each other. We analyze the results of our methods on computer experiments, and weigh against existing techniques in reference to general purpose.

1. はじめに

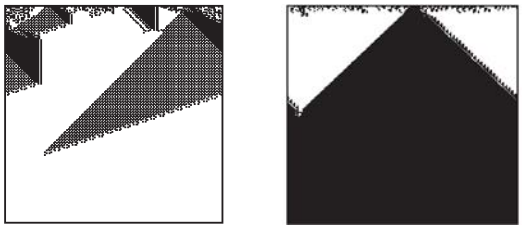
複雑系現象を視覚的側面から擬似的に再現し，解析を行うツールの 1 つに，セルオートマトン（Cellular Automata：CA）がある．CA は，1950 年代に von Neumann と Ulam によって考案された．70 年代から生物の形態形成モデルとして利用され，以降，DNA 配列や進化等の遺伝子生理学，森林火災や地震等の災害，高速道路の車の流れ，株価変動，景気の循環等，CA を利用した様様なモデルが提案された[1)]．CA を用いて現象を再現する，すなわち CA モデルを作成する場合，モデル化する現象を的確に再現する CA ルールを heuristic に設計する必要がある．しかし，現象がより複雑である場合，その複雑さを再現する CA ルールを設計するために，要素の挙動についての膨大な解析が必要となる．さらに CA モデルによるシミュレーションでは，いずれのセルも同一のルールに基づいて動作するという単純性を持つ反面，その大域的な挙動は CA ルール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 ルールを設計することは容易でない．このような背景のもと，CA ルールの進化計算による自動設計に関する研究が広くなされている．Mitchell らは，CA で再現する現象として密度分類タスクを取り上げ，遺伝的アルゴリズムを用いて自動設

† 琉球大学工学部
Faculty of Engineering, University of the Ryukyus
†† 琉球大学大学院理工学研究科
Graduate School of Science & Engineering, University of the Ryukyus
☆ 現在，株式会社沖縄電力
Presently with The Okinawa Electric Power Company

計を試みたが，高い性能を示すCAルールを設計することはできなかった[2),3)]．ほかに，GAをベースとした進化計算として，遺伝的プログラミング（Genetic Programing：GP）や共進化計算の手法による実験が行われ，高い性能のルール獲得に成功している[4),5)]．特に，Juillé らの共進化計算を用いたルール設計[6)]は，解集団が効率的に進化するための問題空間探索を取り入れた手法であり，膨大な問題空間を取り扱うCAルール設計に対して有効に働くことが，彼らの実験で示された．しかしながら，CAルール設計問題の1つである密度分類タスクに本手法を適用した結果，高い性能を示すルール設計を行うためには，タスクに関する先見知識を用いた拡張を行わなければならないという問題点も明らかになった[7),8)]．本研究では，CAの対象としている現象に内在する規則性を利用し，問題に依存しないCAルール自動設計を行う手法を提案する．この手法は，現象内に内包される局所的規則性（サブタスク）を元にして，これらサブタスク解のCAルール内での整合性を高めるものである．まず，**（1）各CAルールのタスク達成特徴を元に，集団をクラスタリングし**，**（2）各クラスタに，他のクラスタと統合できる方向に個体が進化する適応度計算式を割り当てる**という処理を行う．Juillé らの手法が問題依存型のクラスタリングと heuristic な適応度計算式の拡張を導入したのに対し，本提案手法はモデル化する現象に応じたクラスタリングと，種統合を行うための適応度計算式の自動的な調整を行い，あらゆる問題対象への適用可能性の向上を目指す．また，得られたCAルールの帰納学習アルゴリズムC4.5[15)]による解析により，ハンドコーディングルールであるGKLルール[10)]に比べて，きわめて複雑なルールが獲得されていることを示し，CAルールの獲得に対する機械学習アプローチの有効性を示す．

2. Cellular Automata モデル

均質の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，近接するものどうしを相互に結合した系が **Cellular Automata：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．各セルは他のセルと同期をとりながら独立に状態遷移を繰り返す．このような観点から，CAは並列動作型計算モデルとしてとらえることができる．すなわち，単純な計算をする同一のプロセッサが1次元や2次元に結合された並列計算機としてとらえるのである☆．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行わなければならない．つまり，**（1）CAに与えられるあらゆる初期状態（Initial Configuration：IC）に対する挙動（目標）を逐一見つけ出し**，**（2）それらすべての挙動を再現するCAルールを自動または手動で設計する**，という作業が必要となる．本研究では，CAルール自動設計問題として密度分類タスクを取り上げる．このタスクは，CAに与えられた初期状態（Initial Configuration：IC）に応じてCAの挙動が決定するという自己組織化現象の再現を目標とするものである．前述の（1）の操作で探索されるICの目標がすでに定義されている．本タスクの詳細を以下に示す．



$\rho_0 = 0.5$ の密度分類タクス．図の最上層がICを表し，下方向に時間ステップの経過にともなう各セル状態遷移を示している．左図は $\rho_0 = 0.44$ に対する達成例，右図は $\rho_0 = 0.58$ に対する達成例である．

図1 密度分類タスク
Fig. 1 Density-Classification Task.

密度分類タスク（Density-Classification Task）
状態数 $k = 2$（状態が‘0’か‘1’）の1次元CAにおいて，ICの状態平均値（密度）ρ_0 が臨界値 ρ_c（critical：c）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．ここで，状態平均値（密度）とは各セルの状態値の総和をセル数で割った値である．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．密度分類タスクの達成例を**図1**に示す．このタスクが取り扱うICの種類は膨大（$2^{\text{格子サイズ}}$）であり，すべてを的確な目標に導くCAルールを設計することは非常に難しい．密度分類タスクにおいて，すべてのICを正解に導くCAルールは存在しないことが証明されている[9)]．このタスクを解くCAルールとして，**GKLルール**[10)]がある．このルールはheulisticなルールであり，IC全体から 10^4 のサンプリングに

☆ このような並列計算機はシストリック・アレイ（systolic array）とよばれる．

対して 82.7%の達成率を示した．heuristic ルールのほかに，進化的計算を用いて CA ルールを生成する **EvCA** を適用した研究がある．

3. 進化セルオートマトン：EvCA

CA ルールを遺伝子型（genotype），そこから得られるダイナミクスを表現型（phenotype）と考え，CA のダイナミクスが高度な計算能力を持つように，CA ルールを進化計算を用いて自動設計する枠組みは，**進化セルオートマトン（Evolutional Cellular Automata：EvCA）**と呼ばれている．ここで，EvCA における 3 つの主要な研究について述べる．

3.1 SGA を導入した EvCA

GA において交叉は重要なオペレーションである．従来から GA における交叉の有効性はビルディングブロック仮説と暗黙の並列化によって説明されてきた．この説は最適解が並列に広がる有利なスキーマの組合せによって作られることを主張するもので，これまでの GA 研究は主にこれら 2 つの考えに沿って進められてきたといえる．Mitchell らは CA ルールを単純 GA（SGA：Simple GA）を用いて自動設計する研究を行った[2),3),5)]．この研究では，CA で再現する現象として密度分類タスクを取り上げた．実験ではタスク達成に十分な能力を持つ CA ルールを自動設計することができなかった．実験で用いた CA ルールと genotype の関係を**図 2** に示す．図 2 において，各近傍パターンが各々の遺伝子座に対応し，オートマトン出力が遺伝子となる．CA ルールが染色体として表現される．最も大きな原因となったのは，ヒッチハイク効果である．複数のサブタスクのもとでは，1 つのサブタスク A に有利に働く個体が選択的に伝達され，集団中の他の個体の対立遺伝子を上書きする．これによって，A 以外のサブタスクに有利なスキーマは集団から消失してしまい，暗黙の並列化が十分に働かなくなる．この実験で設計されたルール（SGA ルール）は，10^4 のサンプリングに対して 76.1%の達成率で，GKL ルールの性能を下回る値であった．

3.2 CA ルールサイズ圧縮による拡張

密度分類タスクでは，CA ルールに与えられる入力のうち，性質上同一と考えられるものが存在する．著者らは，CA ルール空間内で同一または類似性のきわめて高いと考えられる入力を統一し，ルール空間を圧縮する手法を提案した[11)]．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（**図 3** (a)）．また，図 3 (b) の 2 つのセル

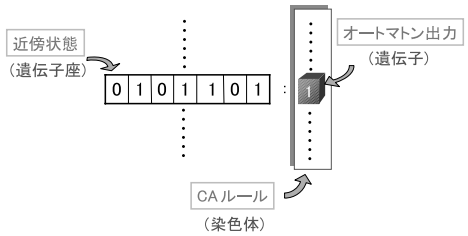


図 2 ルールと genotype の関係
Fig. 2 Relation between the rule and the genotype.

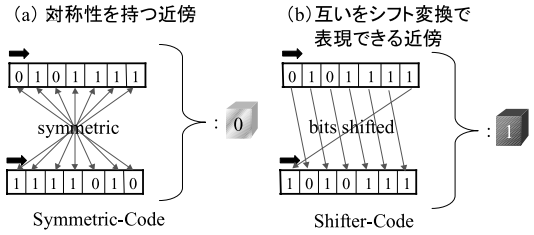


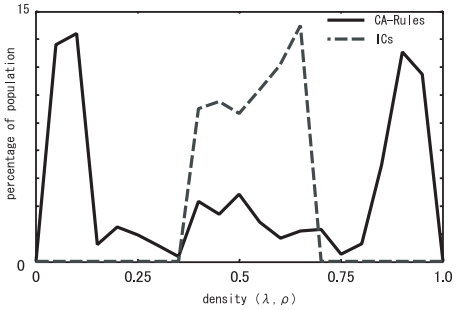
図 3 同系の近傍入力
Fig. 3 Isogenic neighborhood pattern.

近傍入力は，互いに 1 bit-Shift を行うことで表現できる関係である．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 1 bit-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とから，これらの入力を統一することでタスク達成可能な IC の数も向上すると考えた．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 コード，1 bit-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 コードとした．CA ルールのサイズは，非圧縮のルールに比べて Symmetric コードで 56.25%，Shifter コードで 32.03%に圧縮された．

実験の結果，Symmetric コード手法で 77.48%，Shifter コード手法で 74.67%のタスク達成率であった．Symmetric コード手法がわずかに SGA ルールを上回り，性能の向上が見られたものの，GKL ルールの性能に追随する性能を示すことができなかった．主な原因は，ルールサイズ圧縮のために，ヒッチハイク効果によるスキーマ消失の影響がより大きなものとなったことがあげられる．特に，Shifter コード手法は，$\rho_0 < 1/2$ である IC のサブタスク達成率が $\rho_0 > 1/2$ のそれと比べて極端に低く，サブタスク間に達成率の偏りが生じる結果となった．

3.3 共進化学習を導入した EvCA

Juillé らは，共進化計算を用いて CA ルールを獲得する研究を行った[6)~8)]．この実験における共進化計算は，CA ルールを「学習主体」，IC を制御対象である「環境」と見なした強化学習の実現手段の 1 つで



図中の λ は，CA ルールにおいて「静状態」を出力しない確率である．ここでは「静状態」を 0 と定め，λ を染色体内の 1 の割合の値と定める．CA ルール集団は，$\lambda \approx 0$ と $\lambda \approx 1$ に多くの個体が発生している．つまり，いかなる IC に対しても 0 もしくは 1 を出力するルールであるため，どの個体も（ランダムに IC を選択をした場合）50%程度の達成率しか見込めない．
一方 IC 集団は，本タスクで最も難解な $\rho_0 \approx 0.5$ に多くの個体が集中している．CA ルールが上記のとおり十分に進化していないにもかかわらず，IC が難解になったため，新たな戦略を持ったルール個体が発生できない．

図 4 競合共進化：世代終期の個体分布
Fig. 4 Co-evolution: distribution of rules and ICs.

ある．強化学習は，不確実な感覚入力と遅れをともなう報酬というきわめて弱い情報源を手がかりとして試行錯誤的に行動規則を改善してゆく機械学習である．CA タスクのように取り扱う環境が膨大にあるような強化学習の場面では，学習が効率良く行えるトレーニングケースをいかに環境から引き出し，学習主体に与えるかということが問題となる．Hillis は，環境に対して行動する学習主体とは別に，環境から必要な情報だけを抽出する学習主体を設計し，これら 2 主体のパフォーマンスを共進化計算により向上させる共進化学習を提案した[12]．Juillé らは，共進化学習を構成する 2 つの学習主体に CA ルール集団と IC 集団を割り当て，互いの集団が競合共進化で進化する設定で実験を試みた．実験の結果，獲得された CA ルールは 10^4 のサンプリングに対して 50%に満たない達成率であった．CA ルール集団が低い性能で収束を迎えた原因は，SGA を用いた実験同様ヒッチハイク効果によるものであった．特に，ヒッチハイク効果の影響で CA ルールの性能が伸び悩み，悪化する IC 集団に進化が追いつかないという弊害が観測された（**図 4**）．

彼らはこの問題に対して，CA ルール集団の進化と IC 集団の進化を同期化する手法を提案し，高い性能を示す CA ルール獲得に成功した．この手法は，IC 集団をタスクの難易度に基づいて複数のグループに分割し，各グループに CA ルールの進化に応じて適応度を与える heuristic な関数 E () を，IC 集団の適応度計算式に導入するものであった．E () によって，IC が極端に難しくなり CA ルールの進化が停滞するという問題が解決され，10^4 のサンプリングに対して 90%以上の達成率を示す CA ルールを設計したとされている．

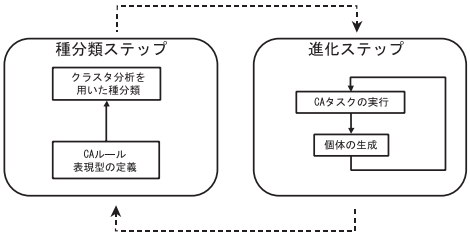


図 5 進化計算サイクル
Fig. 5 Evolution cycle.

4. 提案手法

Juillé らの手法は，以下にあげる点で EvCA の汎用性を著しく低下させるものであった．

- **IC をグループ化する機能が，密度分類タスクの特性に応じた先見的な知識を用いたものであった点．**
- **IC 集団の適応度計算式に導入された関数 E () は heuristic な定義によるものであり，他のタスクにおいても有効に機能する事が保障されない点．**

より複雑な現象を再現する CA ルールを自動設計する場合，問題領域に対する設計者の知識を必要としない処理が必須であると筆者らは考える．本章では，このような処理系を満足する進化計算オペレーションの設計について論じる．

4.1 進化モデル設計

密度分類タスクは，処理対象となる IC の種類が膨大（$2^{\text{格子サイズ}}$）であり，機械学習を行うには何らかの方法で IC をサンプリングしなければならない．また，先行研究において GA の選択・淘汰がヒッチハイク効果という弊害を生むことが確認されている．EvCA を設計するにあたり，次の機能を実現する必要がある．

1. **膨大な量となる IC を，CA ルールの能力に応じて最適化する機械学習の枠組み．**
2. **サブタスクの処理メカニズムが進化の単位として明示化され，それらが並列に評価される遺伝的プロセス．**

1. については，Juillé ら同様，共進化学習を用いて実現する．一方，2. については，(a) CA ルール集団の各個体が持つタスク処理能力に応じて個体をグループ化，(b) CA ルール集団の各個体に対して，各グループにおける達成率の比較等の処理によって評価を行い，GA によって解探索を行う，という 2 段階のプロセスで実現する．(a) を**種分類ステップ**，(b) を**進化ステップ**と呼ぶ（**図 5**）．

4.2 種分類ステップ

筆者らが提案する進化モデルでは，種という概念を導入する．種は，相互に交配する個体の集団で，他の交配集団と生殖に関して隔離されたものと定義される．本モデルにおける種は，表現型の特徴によって分類される個体群であり，GA によって遺伝子型を変化させる対象である．さらに，これら種に属する各個体（種個体）は，以降で説明する遺伝的プロセスを用いることによって，種どうしが互いに共存するように進化する．ここでは，CA ルール集団内に種を発現して各個体を分類（種分類）する方法について述べる．

4.2.1 CA ルールの表現型の定義

CA ルールを種分類するためには，集団内の各個体の持つタスク達成状況を用いる必要がある．提案法では，タスク達成状況の表現形式として特徴ベクトルを採用する．CA ルールの特徴ベクトルは，IC 集団の全メンバに対するタスク達成の成否（成功=‘1’，失敗=‘0’）を表したビット列とする．この特徴ベクトルは，CA ルールの内容を遺伝子型とした場合の表現型に相当する．本研究の計算機実験では先行研究同様，IC 集団が GA の各世代で新たに生成されるため，これにともなって表現型も再計算される．

4.2.2 クラスタ分析を用いた分類操作

CA ルールを表現型を用いて種分類する手段として，クラスタ分析を用いる．クラスタ分析における距離は，CA ルールの表現型から計算される．2 つのベクトル a と b の間の距離 d_{ab} は以下のように計算される．

$$d_{ab} = \left[\sum_{j}^{m} (x_{aj} - x_{bj})^2 \right]^{1/2} \tag{1}$$

式 (1) は，ユークリッド距離である．以下に，クラスタ分析の一般的な手順を示す．

Step1. すべての個体間の距離を計算する．
Step2. 最も近い 2 つの個体を併合して 1 つのクラスタとする．
Step3. 作成したクラスタを新たな個体とし，すべての個体間の距離を計算する☆．
Step4. 以下同様にクラスタへの併合を進め，すべてが 1 つのクラスタに含まれるまで行う．

クラスタとクラスタ，またはクラスタと個体との間の距離を計算する方法がいくつか提案されている．本研究では，種分類のアプローチとして以下に示す 3 種類の距離法の適用を試みた．

☆ 実際には，新しく作られたクラスタと残りの個体（およびクラスタ）との距離を再計算するだけでよい．

最短距離法 それぞれのクラスタに属する個体間の距離のうち，最小値をクラスタ間の距離とする．
最長距離法 それぞれのクラスタに属する個体間の距離のうち，最大値をクラスタ間の距離とする．
重心法 それぞれのクラスタの重心間の距離をクラスタ間の距離とする．

上記のクラスタ分析の技法を用いて CA ルールをクラスタリングした後，作成された全クラスタに対して以下の式を用いて，クラスタ類似値を計測する．クラスタ類似値とは，クラスタに属する個体の類似度を表す指標である．

$$S_{Cluster\_n} = 1 - d_{ab}/d_{max} \tag{2}$$

ただし，

d_{ab} ：任意のクラスタ n を二分するサブクラスタ a，b の距離
d_{max} ：全個体で，最も離れた 2 つの距離

さらに，クラスタを「種」として認める最小のクラスタ類似値 $S_{species}$ を定め，各個体の所属「種」を，$S_{species}$ より大きい値で最小となる類似値を持つクラスタとする．この操作により，各個体は，属するクラスタのうちで最大のクラスタを「種」とする「種個体」として再定義される．

4.3 進化ステップ

CA ルール集団と IC 集団の共進化学習を用いて，解探索を行う．先行研究同様，CA ルール集団の進化がタスク達成率の総和が最大となるように進み，IC 集団の進化がそれを妨げるように進む競合共進化を考える．

CA ルール

CA ルールは IC 集団のすべてを処理し，タスク達成率に基づいた評価が行われる．CA ルールは，達成率が低い IC をより多く成功に導くほど，高い評価が与えられることとする．適応度計算式として式 (3) を与える．ここで，$covered(Rule_i, IC_j)$ は，$Rule_i$ によって，IC_j が達成されることを示す．

$$f(Rule_i) = \sum_{j=1}^{n_{IC}} Weight\_IC_j \times covered(Rule_i, IC_j) \tag{3}$$

ただし，

$$Weight\_IC_j = \frac{1}{\sum_{k=1}^{n_{Rule}} covered(Rule_k, IC_j)}$$

ここで，n_{Rule} とは集団中のルールの数，n_{IC} とは集団中の IC の数を意味する．

IC

IC は CA ルール群のすべてにオートマトン処理さ

れ，タスク達成率に基づいた評価が行われる．IC の評価には，前述の種分類ステップの結果が反映される．

ある CA ルールに注目したとき，自身が達成できない IC を処理する部分解（スキーマ）を GA の交叉等のオペレーションで取り込むことができれば，個体の性能は向上する．一方，IC はより難しい（達成率の低い）個体を生成するように進化するが，その難解な IC を処理するスキーマが CA ルール集団内に存在しないならば，その IC に高い評価が与えられるべきではない．なぜなら，そのような IC を解決する解は，スキーマの取り込みで解探索する交叉オペレーションで生み出されることはありえないからであり，唯一可能性がある突然変異で解探索される見込みは，一般に交叉で解探索されるそれに比べてはるかに低い．CA ルールが進化するうえで注目すべきは，自身が達成できておらず，他の CA ルールが達成できている（つまりスキーマが存在する）IC である．CA ルールのこのような IC の注目を**目標**，IC の目標とされた回数を**履修欲求値**と呼ぶ．

以上のことから，達成率が高い CA ルールでさえ成功しにくく，かつ履修欲求値が高い IC に高い評価が与えられることとする．適応度計算式は式 (4) のとおりである．

$$f(IC_j) = \sum_{i=1}^{n_{Rule}} Weight\_Rule_i \times F(Rule_i, IC_j) \times \overline{covered(Rule_i, IC_j)} \tag{4}$$

ただし，

$$Weight\_Rule_i = \frac{1}{\sum_{k=1}^{n_{IC}} F(Rule_i, IC_k) \times \overline{covered(Rule_i, IC_k)}} \tag{5}$$

式 (4) における $\overline{covered(Rule_i, IC_j)}$ はルール i が ICj を達成できないときに 1 を返し，達成できるときには 0 を返す関数である．また，式 (5) における $\overline{covered(Rule_i, IC_k)}$ はルール i が ICk を達成できないときに 1 を返し，達成できるときには 0 を返す関数である．

履修欲求値関数： F ()

$$F(Rule_i, IC_j) = covered(Species(Rule_i), IC_j) \times covered(Species(\overline{Rule_i}), IC_j)$$

CA ルール・IC ともに，淘汰率 G で個体が破棄され，突然変異および交叉によって新たな個体を生成することとする．また両集団の進化世代は 1：1 に対応する．

表 1 パラメータ（IC 数：10^4）
Table 1 Parameter (IC#：10^4).

集団サイズ	P_{Rule}	500
集団サイズ	P_{IC}	5000
淘汰率	G_{Rule}	0.8
淘汰率	G_{IC}	0.05
突然変異率	m_{Rule}	0.04
突然変異率	m_{IC}	0
最小クラスタ類似値	$S_{Species}$	0.2

4.4 計算機実験

7 近傍の 1 次元 CA において提案手法の進化計算によるルール探索の実験を行う．近傍数は GKL ルール設計時の設定に準ずるように，自身を含めて左右 3 セルずつの 7 セルとした．また，機械学習に用いる CA の格子サイズ $N = 149$ とした．CA ルールの初期集団を，パラメータ $\lambda^{☆} \in [0.0, 1.0]$ の範囲に，均等に分布するようにランダム生成した P_{Rule} 個の個体によって構成する．近傍数 7 の CA において，CA ルールのビットサイズは $2^7 = 128$ であり，ルールの種類（CA ルールの探索範囲）は，$2^{2^7} = 2^{128}$ である．

一方，IC の初期集団は，$\rho_0 \in [0.0, 1.0]$ の範囲に均等に分布するようにランダムに生成した．格子サイズ $N = 149$ の CA において，タスクが取り扱う IC の種類（IC のサンプリング範囲）は $2^N = 2^{149}$ である．

設定パラメータを**表 1** とし，提案した進化モデル設計で約 1200 世代までの集団を進化させた．各世代に遺伝的操作を適用し，新たな世代の集団を生成する手順を以下に示す．

step.1 IC・CA ルールの初期集団を生成する．
step.2 すべての CA ルールに対し，すべての IC の処理実行を行う．
step.3 CA ルールの IC に対するタスク達成状況の情報（bit 列）を元に，CA ルールを分類（種分化）する．
step.4 両集団に，適応度計算式に従って評価値を与える．
step.5 両集団の個体に対して，評価値の下位 S の割合数を削除する．
step.6 ランダムな 1 点の Cutting Place による交叉オペレーションを行い，G の割合数の新たな個体を生成し，集団に加える．
step.7 突然変異率 m で，個体のビット列にミスコピーを起こす．
step.8 世代数が終了条件を満たさなければ，step.2 へ

☆ λ は，CA ルールにおいて「静状態」を出力しない確率である．ここでは「静状態」を 0 と定め，λ を染色体内の 1 の割合の値と定める[13),14)].

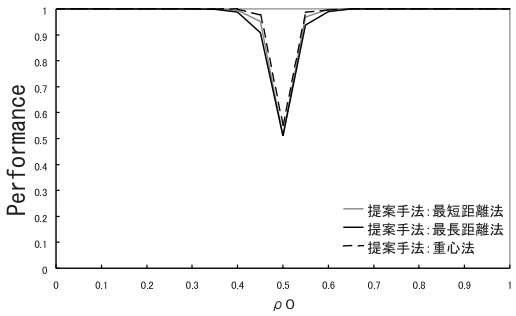
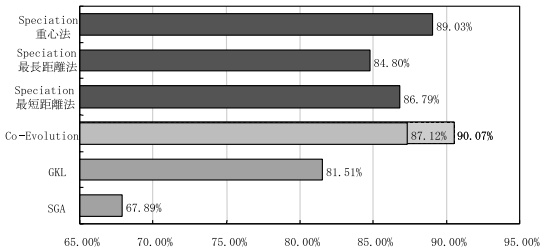


図 6 タスク達成状況
Fig. 6 Task achievement pattern.



Juillé らの提案手法については追実験を行い，追実験結果と Juillé らの論文に示された結果を併記した（各々達成率は，87.12%，90.07%）．追実験の CA ルール（Co-Evolution ルール）の達成率を筆者らの手法と比較した結果，重心法を用いた CA ルールが若干上回った．しかし，Juillé の論文に示される達成率には届かなかった．

図 7 タスク達成率（IC 数：10^4）
Fig. 7 Correct classification rate (number of IC: 10^4).

戻る．

4.5 実験結果

図 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，提案手法ルールのタスク達成率を計測した結果である．$\rho_0 > 1/2$，$\rho_0 < 1/2$ のいずれの IC に対しても偏りなく均等にタスクを達成する一方，$\rho_0 \approx 1/2$ で達成率が急激に落ち込んでいる．本タスクは IC の密度 ρ_0 が 1/2 に近いほど，タスクを達成することが難しくなるという特徴を持つ．

計算機実験によって得られた CA ルールの，ランダムに抽出した 10^4 個の IC に対するタスク達成率を **図 7** に示す．Juillé らの手法に従って追実験を行い，筆者らの提案手法および SGA ルール，GKL ルールと比較した．

共進化計算を取り入れた手法（Juillé 手法・最短距離法・最長距離法・重心法）は，いずれも SGA，GKL よりも高い達成率を示すルールの獲得に成功した．このことから共進化計算が，EvCA において非常に有効な手法であるといえる．特に，重心法を導入した手法は本実験で最も高い性能を示している．

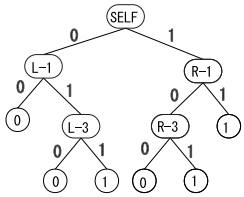


図 8 GKL ルールの決定木
Fig. 8 Decision tree of GKL rule.

また，各距離法において共進化学習が進行する段階では，10〜100 までのクラスタ数が観測された．しかし，その増減は進化の停滞直前まで続く．そして，最終的に 1 クラスタあるいは 1 クラスタと例外的な 1〜2 個の単体の固体となり進化停滞となる．各距離法の特徴としては，最短距離法は，サンプルは最も近い距離にあるクラスタに配属されるため，クラスタが 1 つずつ個体を増やしていく傾向がある．この影響で種個体の数に大きな偏りが生じ，目標とする IC を正しく絞り込めない場合があったと推察する．一方，最長距離方では，クラスタの中で最も離れた個体が，サンプルの配属に影響を与える．その結果，他の個体から離れた個体をひとたび配属したクラスタには，それ以降新たなサンプルが配属されないという不具合が生じることがあった．重心法では，いくつかのクラスタができ，最後にそれらがまとまっていく傾向があるため，種個体数が偏らず，進化が有効に働いたと考えられる．

5. CA ルールの決定木比較

自動設計した 3 つの CA ルール（最短距離法・最長距離法・重心法）と，heuristic に設計された GKL ルールを，C4.5 [15] を用いて分析し，得られた決定木の比較を行った．C4.5 は，事例を表す属性値とその事例が属するクラスの対からなる多数のデータから，各々のデータを正しいクラスに分類する決定木を生成するための帰納学習アルゴリズムである．

5.1 heuristic ルール

C4.5 の概念獲得に関する性能を評価するにあたり，まず heuristic ルールである GKL ルールを解析した．GKL ルールを 128 個の入出力対として表現したルックアップテーブルを C4.5 の入力ファイルに変換し，作成した決定木を **図 8** に示す．根ノードおよび中間ノードは属性を表す．ここで，属性は近傍セルの位置である．たとえば，SELF は着目セル，L-1 は左隣のセルを示す．また，各エッジに付された値は属性のとる値である．葉ノードはクラスを示しており，ルートから葉ノードへのパスは，ある近傍状態に対する着目セルの遷移規則を表現する．この決定木は，完全に元の論

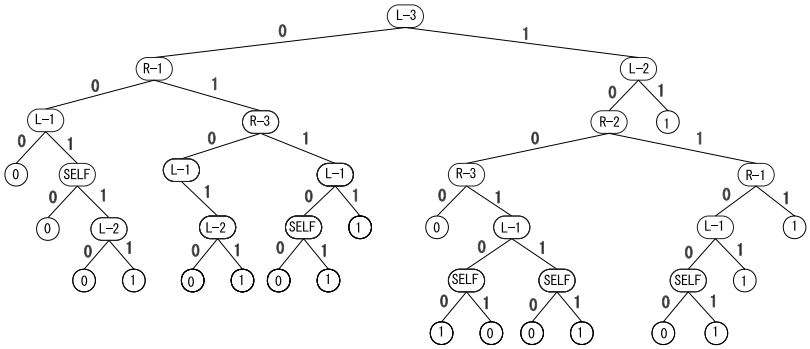


図 9 C4.5 で生成した Speciation（最短距離法）ルールの決定木
Fig. 9 Decision tree of Speciation rule.

理式と同値である．したがって，GKL ルールは C4.5 によって完全に表現可能であるといえる．

GKL の決定木の特徴は非常にコンパクトな構造で表現されているということである．C4.5 は，セルが参照する入力に対して，出力を決定するのに役立たないと思われるものを取り除くことによって，CA ルールを圧縮している．学習前のルックアップテーブルが 128 個の出力クラスを持っていたのに対し，決定木が持つ出力クラスの数は 6 個であり，約 4.7%にまで圧縮された．

5.2 自動設計ルール

自動設計ルールから生成した決定木は，非常に複雑なものとなる．**図 9** は，最短距離法を用いた種分類による自動設計ルールである．heuristic ルールに比べてサイズが大きく，複雑な構造となっている．この決定木の出力クラスは 19 個（圧縮率 14.8%）であり，GKL ルールに比べて約 3 倍程度のサイズとなっている．また，いずれの自動設計ルールも，一部のルール要素が C4.5 によりノイズとして取り扱われ，決定木に収納できず，結果として誤りを含んだ決定木となる．完全な決定木にするためには，誤った部分を手動で訂正しなければならず，構造はより複雑なものとなる．

表 2 は，GKL ルールと自動設計ルール（最短距離法・最長距離法・重心法）の，決定木比較である．生成した決定木において，heuristic な設計による GKL ルールが最もサイズが小さく，なおかつエラー率のない完全な決定木として表現される．ここで，エラー率とは，(正しく分類されない事例数)/(全事例数) として与えられる．一方，自動設計ルールの決定木サイズは，いずれも GKL ルールより大きく，また約 10〜20%程度のエラーを含んだ決定木である．タスク達成率とあわせて両者を比較すると，(a) heuristic に設計した CA ルールは論理的に理解しやすいが達成率が低い，(b) 自動設計による CA ルールは論理的に理解することが難しいが達成率が高い，ということになる．つまり，目的の挙動を示す CA ルールを試行錯誤的に設計する手法は，その性能に限界があり，高い精度が要求される場合には自動設計手法に大きな優位性があるといえる．

表 2 決定木の比較（GKL ルール・Speciation ルール）
Table 2 Comparison of Decision tree (GKL & Speciation).

	サイズ	圧縮率 (%)	エラー率 (%)	達成率 (%)
GKL	6	4.68	0	81.51
最短	19	14.84	10.2	86.79
最長	14	10.93	21.1	84.80
重心	21	16.40	11.7	89.03

6. おわりに

共進化学習を用いた CA ルール自動設計環境において，CA ルールを特徴づけるパラメータである表現型に着目して解集団をクラスタリングし，これらに適切な評価を割り当てる手法を提案した．この手法は，問題対象に依存して設計系を調整する従来手法に比べて汎用性を高めるものであり，より複雑・高度な現象を取り扱う際に有効な自動設計モデルとなることが期待できる．提案手法を密度分類タスクに適応し計算機実験を試みた結果，heuristic な設計によるものと同等あるいは場合によっては高い性能を示す CA ルールの獲得に成功した．また，獲得されたルールを決定木により分析した結果，高いタスク達成度を示すルールが非常に複雑な構造を持つことが示された．ここ結果からも，CA ルール獲得の自動設計アプローチが重要であると結論される．今後の課題として，時系列の状態値に対する評価が必要となる CA ルール作成のテスト問題である「一斉射撃問題」への本手法の適応があげられる．また，CA 応用の代表例である交通モデルにおいて，道路構造とドライバールールの相互作用による

モデル設計への本手法の適用がある．

謝辞 本研究の一部は，文部科学省科学研究費（13780296）の補助を受けて行った．

参 考 文 献

1) 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).
2) Mitchell, M., Hraber, P.T. and Crutchfield, J.P.: Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations, *Complex Systems 7*, pp.89–130 (1993).
3) Mitchell, M., Crutchfield, J.P. and Hraber, P.T.: Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments, *Physica D*, Vol.75, pp.361–391 (1994).
4) Koza, J.R., Bennett III, F.H., Andre, D. and Keane, M.A.: Discovery of Cellular Automata Rules, *Genetic Programming III*, pp.985–1017 (1999).
5) Werfel, J., Mitchell, M. and Crutchfield, J.P.: Resource Sharing and Coevolution in Evolving Cellular Automata, *IEEE Trans. Evolutionary Computation*, pp.388–393 (2000).
6) Juilé, H. and Pollack, J.B.: Coevolving the "Ideal" Trainer: Application to the Discovery of Cellular Automata Rules, *Proc. 3rd Annual Genetic Programming Conf.* (*GP-98*), pp.519–527 (1998).
7) Juillé, H. and Pollack, J.B.: Coevolutionary Learning: a Case Study, *Proc. 15th International Conference on Machine Learning* (1998).
8) Juillé, H. and Pollack, J.B.: Coevolutionary Learning and the Design of Complex Systems, *Advanced In Complex Systems*, Vol.2, No.4, pp.371–393 (1999).
9) Land, M. and Belew, R.K.: No perfect two-state cellular automata for density classification exists, *Physical Review Letters*, Vol.74, No.25, pp.1548–1550 (1995).
10) Gacs, P., Kurdyumov, G.L. and Levin, L.A.: One-dimensional uniform arrays that wash out finite islands, *Probl Peredachi. Inform*, Vol.14, pp.92–98 (1978).
11) 亀島　力，遠藤聡志，山田孝治：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，pp.417–422 (2000).
12) Hills, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D*, Vol.42, pp.228–234 (1990).
13) Langton, C.G.: Studying artificial life with cellular automata, *Physica D*, Vol.22, pp.120–149 (1986).
14) Langton, C.G.: Computation at the edge of chaos: phase transition and emergent computation, *Physica D*, Vol.42, pp.12–37 (1990).
15) Quinlan, J.R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc. (1993).

(平成 15 年 4 月 9 日受付)
(平成 15 年 6 月 3 日再受付)
(平成 15 年 6 月 17 日採録)

遠藤　聡志（正会員）

昭和 39 年生．平成 2 年北海道大学大学院工学研究科電気工学専攻修士課程修了．博士（工学）．同年北海道大学工学部助手．システム工学に関する研究に従事．平成 5 年より琉球大学工学部講師．平成 6 年より同助教授．進化計算，マルチエージェント，セルオートマトンに関する研究に従事．人工知能学会，計測自動制御学会各会員．

山田　孝治（正会員）

昭和 40 年生．平成 5 年北海道大学大学院工学研究科情報工学専攻修了．博士（工学）．同年琉球大学工学部情報工学科助手，平成 6 年より同講師，平成 9 年より同助教授．マルチエージェント，知能ロボットに関する研究に従事．人工知能学会，ロボット学会，機械学会，計測制御自動制御学会各会員．

亀島　　力

昭和 51 年生．平成 11 年琉球大学工学部情報工学科卒業．平成 14 年琉球大学大学院理工学研究科情報工学専攻博士課程前期修了．CA ルールの自動設計に関する研究に従事．同年（株）沖縄電力入社．