対話型進化計算による形態構成規則の獲得モデル

A MODEL TO ACQUIRE RULES IN DRAWING BY USING INTERACTIVE EVOLUTIONARY COMPUTATION

松下大輔^{*1}, 宗本順三^{*2}

Daisuke MATSUSHITA and Junzo MUNEMOTO

In this IEC(Interactive Evolutionary Computation) model, designer's implicit preference in making window pattern was acquired as explicit rules of figure pattern, and the possibility of emergent 2-D figure design was shown.

In the model, "design world" was made of "primitive" and "design operator", and this "design world" is translated to "critical language" made of first-order logic by using tree structure.

The rules of rhythm, proportion and symmetry were composed as dominant pattern in the tree structure, and under this pattern, diverse individuals were acquired.

Keywords : *Design Process, Emergence, Genetic Programming, Interactive Evolutionary Computation, Knowledge Acquisition, Knowledge Representation*

設計過程, 創発, 遺伝的プログラミング, 対話型進化計算, 知識獲得, 知識表現

1. はじめに

建築設計は、帰納的または発見的思考に大きく依存する点で特殊であり、多くの場合悪構造問題^{文献1)}となる。経験豊富な設計者は設計過程において、多様な候補から優れたものを経験的知識により選択して案を収束させて行くが、そこに用いられている非明示的な知識を明示的に記述することは重要である。

Akin^{文献1)}は建築設計の特殊性を指摘した上で、十分な時間とふさわしい道具が用いられれば設計に対する意識的なアプローチが可能となると述べた。記述モデル(descriptive model)の提案に際して建築設計は問題解決行為の一つであることを示し、一般的な問題解決の手法を建築設計に適用すること、計算機の利用の重要性について言及した。

建築設計を問題解決行為の一つと捉えた場合の特殊性は、明確に定義された問題を解くだけでなく、定まった問いも答えも不明な状況において、設計者自らが設計案について可能性を見出し、かつ批評するような試行を行い、更に自分の持つ他の経験的知識とも関連させて行く点にある。

Schon^{文献2)}は事例分析により、設計者は設計過程の個々の局面における状況を熟考し反映しながら設計を進めるということを対話(conversation)というアナロジーを用いて説明した。

Mitchell^{文献3)}はまた、設計とは形態的可能性の探求と批評的推論とを同時に展開することで、それはある複雑なゲームと捉えることがで

きると述べた。

最終的な解答は不詳であるが設計過程のある時点において実在する解の評価ならば可能という場合、各時点における解答案の生成、評価を通して逐次的に解の探索がなされていくと言える。このような問題解決行為は発生・検証のプロセス(generate-and-test process)として知られるが、Mitchell^{文献3)}はさらに、計算機を用いてこのプロセスを実行する場合、発生メカニズム(generation mechanism)、検証メカニズム(test mechanism)、制御原理(control strategy)の三者が必要であり、設計における知性は発生メカニズムか検証メカニズムかのいずれかに見出されるべきであるとしている。

ここで建築設計を問題解決行為と捉え、設計過程を解答案との逐次的な対話というアナロジーにおいてモデル化することが可能である。すなわち解答案の制御を行う進化計算アルゴリズムを援用して解答案の生成を行う計算機と、生成された案の評価を行う人間との組み合わせにおいて、それらの相互作用による対話型進化計算を行うことにより、解答案の探索空間の絞り込みを行いつつ創発的に探索を進めるモデルを設定し、これを設計の形態創発のモデルとして捉えた試行を行うことができる。

^{*1} 京都大学大学院工学研究科 大学院生・工修

^{*2} 京都大学大学院工学研究科 教授・工博

Graduate Student, Graduate School of Engineering, Kyoto University, M. Eng.
Prof., Graduate School of Engineering, Kyoto University, Dr. Eng.

2. 研究の目的

図形の体系はプリミティブ (Primitive: 基本図形記号) と、その操作を表す演算子 (Design Operator) から構成される^{文献(3)}と設定する。設定された図形の体系に、対話型進化計算 (Interactive Evolutionary Computation: 以下 I E C) を用い、モデルによる形態創発の可能性を示すことが本論の目的である。I E C では、設計者の嗜好を反映した形態構成規則が明示的に記述され、進化する。本論では、人工知能指向言語を用いた一階述語論理の概念により、形態表現と木構造言語表現の翻訳可能な枠組みを定義した。

図1 設計者と計算機の対話による形態表現と言語表現の進化

設計の初期段階では、三次元の建築形態を直接考察し、操作する場合や、より単純で操作対象の俯瞰が容易な平面上の抽象的な図形の操作により、建築形態のスタディを行う場合がある。本論では後者の場合について、制約条件が比較的少なく、設計者の嗜好が大きく影響する建物立面の窓配置パターンの平面上のスタディを想定した試行を行う。

3. 研究の概要

本論では形態獲得過程の体系は「プリミティブ」と「演算子」から成り、それらは人工知能指向言語である L I S P^{文献(4)}の S 式表現^{注(1)}、すなわち木構造で表される。表裏一体の関係にある形態表現と S 式表現としてモデル化された図形の体系に、I E C を適用する。I E C では遺伝的プログラミング (Genetic Programming: 以下 G P)^{文献(5)}を用いる。設計者は計算機により生成された図形の評価を行い、計算機は交叉 (Crossover)、突然変異 (Mutation)、転置 (Inversion)、複製 (reproduction) の G P オペレータを適用して逐次世代を生成する。これにより設計者と計算機の間で双方向に図形を評価・生成し、設計者の意図を反映した形態表現と S 式表現の進化を行う。得られた図形と木構造の進化の過程の分析により、形態に用いられた規則の獲得を検証しモデルの可能性を示す。

4. 既往研究

建築設計における形態構成規則の知識を事例の分析、類型化により獲得しようとする研究は盛んになされてきた。Roger^{文献(6)}は建築形態の図化された特性に注目し、形態に翻訳された概念的な操作を分類、分析した。Koning^{文献(7)}はフランク・ロイド・ライトの草原住宅の構成原理を構成規則によって分析した。Stiny^{文献(8)}はヴィラ・マルコテンタの平面形を分析し、その構成原理をパラメーターを持つ形態文法で記述している。

建築形態の原型的な概念を精密な事例の分析によって解明しようとする従来の研究に対し、形態表現を計算機に入力可能な言語表現に翻訳し、計算機による記号処理を用いた知識の構築や設計支援を提案す

る試みがある。Akin^{文献(9)}は建物の一般的な特性の収集によって建物と同定する計算機を用いた推論システム (AIM) を提案した。青木^{文献(9)}は建築形態記述の基本方程式により形態記述が可能であることを示し、建築形態情報を言語により計算機に入力する基本システムを提案した。渡辺^{文献(10)}は建築設計を認知科学的視点から捉え、幾何的制約に関する決定において人間の思考に従う形式で表現しうるモデルを提案した。

建築形態の解の探索を創発という点から進化計算により獲得するものとして 3 D ソリッドオブジェクトに G A (遺伝的アルゴリズム) を適用し、進化計算が最適化のみでなく創発にも有効であることを示した Bentley^{文献(11)}の研究や G P (遺伝的プログラミング) を用いて 3 次元形態の進化計算を行っている Una-may^{文献(12)}の研究がある。

建築設計の概念や設計者の嗜好、認知プロセスの生成等を進化計算を用いて生成したものとして、環境との関係の下に行動、進化するエージェントが獲得するふるまいを分析し、意図を伴う認知プロセスが生成されることを示した藤井^{文献(13)}の研究や、数種類の規則の組み合わせから G A を用いて発生的に設計解を生成し、そこに規則性が生じることをもって設計の概念が構成されうることを示した Gero^{文献(14)}の研究や、家具デザインにおけるユーザーの嗜好を定量化した瀧澤^{文献(15)}の研究がある。

設計支援の目的に対して計算機と設計者の対話関係の必要性から対話型最適化を行ったものとして宗本^{文献(16)}の研究がある。

以上のように、形態表現の記述が可能であることを示したり、予め高度に設定された形態構成の枠組みの中で計算機を用いた設計支援を行ったり、認知プロセスの生成を示したりする研究はなされているが、形態構成規則の知識を予め恣意的に設定することなく、形態表現とその言語表現による記述を設計者と計算機の双方向の対話を通じて進化させることによって、形態構成の知識を創発的に獲得するモデルを提示した研究はない。

5. デザイン体系の設定

デザイン体系 (design world) の基本的な構造は、最初に用意された形のヴォキャブラリーと、それらに適用される演算子、およびその結果生じる代数の系によって与えられる^{文献(3)}とされ、汎用 C A D の形態記述・処理システムにおいても同様の概念に基づいているものがある。例えばある 3 次元 C A D のモデリングでは、基本図形記号である球、立方体等のプリミティブを選択し、次に個々のプリミティブに移動、拡大縮小、回転等の操作を加える手法がとられている^{注(1)}。

幾何学的変形は図2のように階層的に分類され、この中の各々の種類の変形は自身より下位にある変形を包含する。

図2 デザイン演算 ウィリアム・ミッチェル著「建築の形態言語」より

本論では2次元座標系で表される図形を扱い、プリミティブとして、「線分(line)」と「矩形(rectangle)」を、演算子として、アフィン変形より下位の「移動(translate)」「回転(rotate)」「拡大縮小(scale)―鏡像変形含む」を設定する(図3)。ここで設定された図形の体系は、プリミティブと演算子、すなわち図形記号とそれらの関係の組み合わせにより多様な2次元の図形表現が可能であり、モデルの十分な一般性を見込んだ^{注2参照}。

6. 形態表現とS式表現の対応関係

(1) S式表現の概要

S式表現はグラフ表現の木構造をLISPの記法で表したもので、本論ではこの言語表現を形態表現の翻訳として進化させる。木構造とS式と通常の式との関係は図のようになる^{文献5)}。ここでA、B、Cを「終端記号」、-、*を「関数記号」という。最も上位のノードである-は根という。Aの親は*で、*の子(引数)はA、B。*、A、Bで構成される部分を部分木、*はその部分木の根という。

(2) 形態表現とS式表現の対応関係

形態表現を言語表現で表すために、デザイン体系は基本図形記号であるプリミティブと演算子から成るという假定から、これらを一階述語論理の概念^{文献17)}を用い、人工知能指向言語であるS式表現で表す枠組みを定義する。

1) 演算子によるデザイン体系の状態の変化

デザイン体系において、ある演算子により、ある状態が次の状態に移行する場合、それらは下図のような関係にある^{文献3)}。

設定されたデザイン体系において、その初期状態であるプリミティブと演算子の組が定義されれば、初期状態に個々の演算子を繰り返し適用することで生成される図形の全ての選択肢が表現可能となる^{文献3)}。図6の樹状図の丸印は各段階での図形の状態、矢印は演算子による操作を表す。樹状図の全体は解空間全体に相当する。

ウィリアム・ミッチェル著「建築の形態言語」より

2) プリミティブと演算子によるS式表現

6. で設定された「移動」「回転」「拡大縮小(鏡像変形含む)」という幾何学変形を表す演算子は、それらにより状態(x, y)が次の状態(X, Y)に移動するとき、a～eのパラメータを用いて3行3列の行列で表され、さらにそれらは図7のようにまとめられる。

移動	回転	拡大縮小
$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$ <p>(x方向にa y方向にb移動)</p>	$\begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix}$ <p>(c/2 πラジアン回転)</p>	$\begin{bmatrix} d & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & 1 \end{bmatrix}$ <p>(x方向にd y方向にe倍)</p>

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} d \cdot \cos c & -\sin c & a \\ \sin c & e \cdot \cos c & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

移動	Translate X : a	Translate Y : b
回転	Rotate θ : c	
拡大縮小	Scale X : d	Scale Y : e

図7 設定された演算子の行列での表記

S式表現では、演算子、プリミティブという順序による前置記法で、式1のように定める。

$$(\text{演算子}[\text{パラメータ}] \text{プリミティブ}) \dots (\text{式1})$$

例えば辺の長さ各1の矩形プリミティブが、①y方向に2倍、②x方向に3移動、③y方向に-2移動するとき(図8)式2のようになる。

$$(b[-2] (a[3] (e[2] \text{rectangle}))) \dots (\text{式2})$$

7. GPへのコーディング

(1) GPの概要

遺伝的アルゴリズムを構造的表現に拡張し、機械学習によりプログラムが進化するようにしたものを遺伝的プログラミング(GP)といい、「探索のための的確な部分構造の把握」、「問題の表現形式に基づいた効果的な探索の実現」、「より高次の知識の適応的な学習システムの構築」といった長所^{文献5)}が知られている。

GPでは関数記号と終端記号から成る木構造を個体として扱い、「交配」、「突然変異」、「転置」、「複製」という遺伝的オペレータが用いられる(図9)。なお複製は親と同一の子を作成する。

図9 遺伝的オペレータ

(2) 関数記号、終端記号の設定

既に定義されたS式表現において、プリミティブは終端記号に、演算子は関数記号に対応させる。演算子は図7における行列の要素を構成する変数(a~e)に加えてプリミティブの表示・非表示を切り替えるスイッチのPRINT関数を関数記号に設定する。さらに終端記号として定数乱数を扱い、関数記号として四則演算、IFLTE関数を扱う(表1, 2)。これにより演算子を構成する変数を根とする部分木の値がパラメータとして用いられる。

表1 関数記号の設定

記号	引数の数	戻り値(引数 ①、②、③、④)
+	2	① + ②。
-	2	① - ②。
*	2	① * ②。
%	2	②が0でない場合: ① / ②。0の場合: 1。
A	1	①。
B	1	①。
C	1	①。
D	1	①。
E	1	①。
PRINT	1	①。
IFLTE	4	① < ② の場合: ③。そうでない場合: ④。

表2 終端記号の設定

記号	意味
line	単位長さの線分。0を返す。
rectangle	単位長さの辺を持つ矩形。0を返す。
1, -2, 8等	-5以上5以下の浮動小数点型定数乱数。

(3) 木構造の形態表現への翻訳

木構造では、終端記号から根に向かって数値が渡されながら計算され、a~eの関数記号のパラメータとなり、それらを根とする部分木が図形での1つの操作を表す(表3)。各プリミティブから部分木の根の間のPRINT関数は表示・非表示の切り替えを行う。すなわち各プリミティブから部分木の根の間のPRINT関数が奇数個の場合にそのプリミティブは表示され、偶数個の場合は表示されない。

表3 木構造の形態表現への翻訳におけるa~e、print関数の意味

関数	意味
A	x方向にA移動。(Translate X)
B	y方向にB移動。(Translate Y)
C	cラジアン回転。(Rotate θ)
D	x方向に比率Aで拡大縮小。(Scale X)
E	y方向に比率Bで拡大縮小。(Scale Y)
PRINT	プリミティブの表示・非表示のスイッチ。

図10に木構造の形態表現への翻訳の一例を示す。ここでは一つのrectangleプリミティブに3つの演算子が適用されている。このように木構造の上位の演算子は下位の演算子を包含して適用され、プリミティブに連続的に操作がなされる。なおlineプリミティブと根の間にはPRINT関数がないため、この場合非表示となる。

図10 木構造の形態表現への翻訳の一例

8. IECの試行

(1) 試行の概要

はじめに初期個体がそれぞれ画面に表示される。設計者は自らの嗜好に応じた得点付けを行い、計算機に入力する。その得点に基づいて次世代が生成され、画面に表示される(図11)。望ましい個体が得られたところで試行を終了する。GPのパラメータは次のように設定した注3)。

集団数=12

初期に生成される木の最大深さ=6

交叉で生成される木の最大深さ=17

突然変異で生成される木の最大深さ=4

木の生成規則=ランダム、最大深度まで生成

選択方式=ルーレット方式

関数ノードでの交叉の確率=5%

関数および終端ノードでの交叉の確率=75%

複製の確率=5%

突然変異の確率=15%

小さい木を優先するための適合度の変換係数=5%

図11 GPのパラメータ設定と計算機の画面表示

(2) 試行

1) 初期個体をランダムに発生させる場合

IECの試行の際、初期個体として予め設計者により作成された図形を用いることも可能であるが、ここでは形態構成規則を与えない状態、すなわち低次元の知識から何らかの設計行為が関与したと考えられるより高次元の知識を含んだ図形を獲得することを目標とするため、初期個体として事前に設定したものを用いず、GPオペレータによるランダムな生成によって得られた初期個体を用いる。

ここでは個体の評価において、表示図形のない空白の個体は50、意味のない図形の個体は30、体を成しているが好ましくない個体は20、好ましい個体は10、特別好ましい個体は5という原則に基づいて、世代内の相対評価で得点を与えた。なお評価が高いほど得点の数値は小さくなる。

得点 個体1 = 20, 個体2 = 5, 個体3 = 20, 個体4 = 30,
個体5 = 50, 個体6 = 30, 個体7 = 30, 個体8 = 30,
個体9 = 50, 個体10 = 50, 個体11 = 50, 個体12 = 50
世代 8 生の適合度の平均: 34.954167 最良個体の適合度: 5.900000

図12 初期個体 (第0世代)

得点 個体1 = 10, 個体2 = 50, 個体3 = 50, 個体4 = 20,
個体5 = 20, 個体6 = 50, 個体7 = 50, 個体8 = 30,
個体9 = 20, 個体10 = 50, 個体11 = 30, 個体12 = 30
世代 14 生の適合度の平均: 35.612500 最良個体の適合度: 11.800000

図15 第14世代

得点 個体1 = 20, 個体2 = 50, 個体3 = 30, 個体4 = 10,
個体5 = 50, 個体6 = 10, 個体7 = 50, 個体8 = 50,
個体9 = 20, 個体10 = 10, 個体11 = 50, 個体12 = 20
世代 8 生の適合度の平均: 31.683334 最良個体の適合度: 11.000000

図13 第4世代

得点 個体1 = 50, 個体2 = 50, 個体3 = 50, 個体4 = 20,
個体5 = 10, 個体6 = 10, 個体7 = 50, 個体8 = 10,
個体9 = 50, 個体10 = 10, 個体11 = 10, 個体12 = 50
世代 60 生の適合度の平均: 33.575000 最良個体の適合度: 13.350000

図16 第60世代

得点 個体1 = 30, 個体2 = 20, 個体3 = 10, 個体4 = 30,
個体5 = 20, 個体6 = 50, 個体7 = 20, 個体8 = 20,
個体9 = 30, 個体10 = 20, 個体11 = 20, 個体12 = 50
世代 8 生の適合度の平均: 26.104167 最良個体の適合度: 11.000000

図14 第8世代

得点 個体1 = 20, 個体2 = 50, 個体3 = 30, 個体4 = 10,
個体5 = 50, 個体6 = 30, 個体7 = 20, 個体8 = 30,
個体9 = 20, 個体10 = 20, 個体11 = 20, 個体12 = 20
世代 73 生の適合度の平均: 29.162500 最良個体の適合度: 16.650000

図17 第73世代

得点 個体 1 = 20, 個体 2 = 20, 個体 3 = 30, 個体 4 = 50,
個体 5 = 20, 個体 6 = 20, 個体 7 = 30, 個体 8 = 10,
個体 9 = 30, 個体 10 = 20, 個体 11 = 20, 個体 12 = 20
世代 75 生の適合度の平均: 27.150000 最良個体の適合度: 14.200000

図 18 第 75 世代

図 20 91 世代の個体 1 の形態表現

図 21 個体 1 の S 式表現

rectangle1
(b[-2.95] (a[-2.95] (d[-2.28] (b[-2] (a[-2] (print (a[-0.67] rectangle1)))))))
rectangle2
(b[-2.95] (a[-2.95] (d[-2.28] (print (a[0.0] rectangle2))))))

図 22 rectangle1 と rectangle2 に関わる操作を纏めた S 式表現 (下)

図 19 第 91 世代

standardized = 23.299999, adjusted = 0.041152, norm = 0.096973
standardized = 24.750000, adjusted = 0.038835, norm = 0.091513

図 23 個体 4 の形態表現と S 式表現

個体 4
rectangle1
(b[-3.02] (a[-3.02] (d[-3.02] (b[-2.0] (a[-2.0] (print (a[-0.67] rectangle1)))))))
rectangle2
(b[-3.02] (a[-3.02] (d[-3.02] (print (a[0.0] rectangle2))))))
個体 8
rectangle1
(b[-2.67] (a[-2.67] (d[-2.0] (b[-2.0] (a[-2.0] (print (a[-0.67] rectangle1)))))))
rectangle2
(b[-2.67] (a[-2.67] (d[-2.0] (print (a[-0.67] rectangle2))))))

図 25 操作を纏めた S 式表現

図 24 個体 8 の形態表現と S 式表現

第91世代まで試行を行い、その間に得られた個体群を各特徴的な世代ごとに示した(図12-19)。初期個体では、各個体は単純なプリミティブのみか、表示される形態がない状態であったが、試行を進めるにつれ次第に複雑さを増し、あるパターンとそれから派生する類似の形態構成が見られるようになった。世代が小さい初期の間は、全く無秩序な形態構成を呈する個体や、形態が表示されない個体が多く現れたが、世代を経るにしたがい一定の規則性を示すものが増え、設計者の嗜好に全く合致しない無秩序な個体や空白を示す個体は減少した。同時に、前世代で評価の高かった優秀な個体が、次世代において前世代の優秀な個体の形態構成に類似しつつも部分の形態構成の異なる個体へと世代交代し、評価がより高くなる過程が認められた。同様に設計者の嗜好に合う個体が複数共存する状態になり形態構成のパターンが分岐したり、それらがある規則で組み合わせられたりする過程があった(図17, 18参照)。

第91世代のうち評価の高かった3個体についてそれぞれ形態表現とS式表現(図20, 21, 23, 24)、さらに図面に表示されているプリミティブ、rectangle1とrectangle2の操作に直接関わる部分をまとめてS式で表現したものを示した(図22, 25)。

3個体の間には類似点と相違点の両者が認められる。

a) 類似点

・3つの扁平な矩形で構成されるパターンが左上(以下パターン1)と右下(以下パターン2)に存在し、両者を連携するように同形の小さい矩形が上下(以下パターン3)に存在している。左上と右下のパターンを連携するように、同形の2つの矩形が上下に存在している(図26)。

図26 部分のパターン

b) 相違点

・パターン1の上のy座標の等しい2つの矩形は、個体1、個体8では一定の間隔を開けているが、個体4では接している。
・矩形の高さは、個体8では0.67と一定であるが、個体1、個体4ではパターン1は高さ1、パターン2とパターン3は高さ0.67となっており、異なる。
・パターン3の上の矩形は、個体1、個体4ではパターン1の下に接しているが、個体8では、パターン1の矩形に内接している。

一方、S式表現について3個体いずれにも共通する部分として(b(a(d(b(a(print))))))があり、さらにこれら3個体以外にもこの部分を有するものが多く、各世代において支配的であった。前述のように、試行が進み世代を経るに従い図形に見られた一定の規則性はこの共通部分によって形成されており、図形の変換のパラメータの値や細部の部分木が交叉、突然変異等のG Pオペレータによって変化しつつも、設計者の評価が常に高い部分が逐次的に継承され、形成されて

いったと考えられる。この規則性はリズム、比例、対称性といった形態的特性によって説明されると言える^{文献3)}。これらの基本的な形態構成規則は、設計者の嗜好を反映しつつ、I E Cの試行を通して明示的に獲得された。

なお、3個体のS式に同様に共通するIFLTE以下の部分木では、IFLTE関数の、第1引数と第2引数を比較して第1引数が小さければ第3引数を、そうでなければ第4引数を親に返すという特性から、木の末端付近の数値情報が変化することによって、木の上部に渡される数値が変化し、それが形態表現の部分的な相違点をもたらししていることが分かる。前世代において評価の高かった優秀な個体が、次世代において前世代の評価の高い構造を維持しつつ、部分の異なる代替案を生じさせ、それが設計者の試行により近い個体となる過程は、このように木構造で表された知識としての形態構成規則が適当に進化していったためと考えられる。

2) 設計者により初期個体を与える場合

予め設計者により作成された図形をS式表現に翻訳し、これを初期個体として設定することにより試行を開始する。

ここでは設計者は図左上に示される、同じ大きさの正方形が直行グリッド上に等間隔で配置される図形を描き、試行においては左右対称の図形を創発的に獲得して行くことを目的とした。特に定まった目標図形はなく、対称という概念に沿った図形を設計者の嗜好に基づいて評価した。

第40世代まで試行を進め、その間に得られた左右対称かつ特徴的な個体を以下に示す(図26)。

<p>(第0世代) 設計者が与えた初期個体</p>	<p>(第5世代)</p>	<p>(第13世代)</p>	<p>(第23世代)</p>
<p>(第24世代)</p>	<p>(第26世代)</p>	<p>(第28世代)</p>	<p>(第34世代)</p>
<p>(第36世代)</p>	<p>(第38世代)</p>	<p>(第40世代)</p>	<p>(第42世代)</p>

図27 設定された初期個体(左上)と試行により得られた個体(左上以外)

図のように左右対称という設計者の想定した図形の概念に則った図形が、初期個体との類似点と相違点を様々に変化させながら得られていった。ここでも、世代を経るに従い設計者の嗜好、ここでは左右対称性を満たす図形が頻出するようになり、左右対称でない図形の生成される割合は減少した。木構造の分析により、 $[d[-1.0]]$ というX軸に関する鏡像変形を表す部分を有する個体が増加し、世代交代によっても失われずに次世代に継承されていた。その他の部分では交叉、突然変異を経て多様な木構造を形成していた。

9. まとめ

建築設計を問題解決行為と捉え、平面上の形態獲得過程を解答案との対話を通して進めるモデル化を行った。人工知能指向言語を用いた I E C の試行を通して図形の設計者の嗜好を反映した形態構成規則を獲得した。I E C では初期個体を恣意的に与えない場合と設計者により設定した場合を試行した。前者では初めは無秩序を呈していた個体群も対話による世代交代を経るに従いある一定の規則性を示すようになり、それらの木構造には共通した部分構造が形成されていた。後者では左右対称等の設計者の概念が保たれつつ、設計者の嗜好に基づき多様な解答案が生成されて行く過程が認められた。

進化計算による発生的な試行を行う場合は一般に、前提的な一定の制約条件、評価関数等が用いられる。本論では評価する者の嗜好に基づいて、対称、リズム、比例等の一定の規則が木構造における支配的な部分構造として制約的に形成され、その枠組みの中で多様な解答案の探索がなされる過程が示された。提示した対話的なモデルを用いることにより、非明示的な設計者の嗜好を明示的な形態構成規則として記述し、平面上の形態獲得過程における形態創発が可能となることが示された。

注

注1) Symbolic Expression (L I S P の記法、本論 7.-(1)参照)。

注2) Alias Wavefront MAYA(R) プリミティブ ポリゴンキューブのプロパティー

pCube1		
TranslateX	0	TranslateY 0
TranslateZ	0	TranslateX 0
RotateX	0	RotateY 0
RotateZ	0	RotateX 0
ScaleX	1	ScaleY 1
ScaleZ	1	ScaleX 1
Visibility	on	

注3) Walter 他 の sgpc(Simple Genetic Programming in C) を元に実装した。

参考文献

- 1) Akin O.: Psychology of Architectural Design, Pion, 1986.
- 2) Schon D. A.: The Reflective Practitioner, How Professionals Think In Action, Basic Books, 1982.
- 3) ウィリアム・ミッチェル著、長倉威彦訳:「建築の形態言語」、鹿島出版会、1991.
- 4) Gerald Jay Sussman 他著、和田英一訳、「計算機プログラムの構造と解釈」、ピアソン、2000.
- 5) Wolfgang Banzhaf, Peter Nordi, Robert E. Keller, Frank D. Francone 著、新田徹、伊庭斉志訳、「遺伝的プログラミング」、科学技術出版、2000.
- 6) Roger H. Clark and Michael Pause: Precedents in Architecture, Van Nostrand Reinhold, 1985.
- 7) Koning H, Eizenberg J: The language of the prairie " Frank Lloyd Wright's prairie houses" Planning and Design, 8 295-323, 1981
- 8) Stiny, George, and James Gips: Shape Grammars and the Generative Specification of Painting and Sculpture, Information Processing 71, C. V. Freeman, 1972
- 9) 青木義次:「建築形態言語の基本システムー建築CADのための基礎的研究その4」、日本建築学会計画系論文集、第418号、pp41-50、1990.12
- 10) 渡辺俊:「建築設計プロセスにおける幾何的知識の表現に関する研究」、日本建築学会計画系論文集、第496号、pp263-268、1997.6
- 11) Peter J. Bentley: The Evolution of Solid Object Designs using Genetic Algorithms, Modern Heuristic Search Methods, chap12, 1996
- 12) Una-May O' Reilly, Girich Ramachandran: A Preliminary Investigation of Evolution as a Form Design Strategy, Artificial Life IV, 1998
- 13) 藤井晴行、谷本潤:「意図的となせざる行動の発生に関する思考実験」、日本建築学会計画系論文集、第547号、pp149-155、2001.9
- 14) John S. Gero, Concept Formation in Design, Knowledge-Based Systems, 10(7-8), 429-435, 1998

- 15) 瀧澤重志、河村廣、谷明敷:「対話型進化計算法による家具デザインー遺伝的プログラミングによるユーザーの嗜好の定量化と予測」、日本建築学会情報システム技術委員会 第23回情報システム利用シンポジウム、pp175-180、2000.12
- 16) 宗本晋作、加藤直樹、今村元一:「直行グラフ描画法を用いた室配置手法ータブー探索法を用いた対話型多目的最適化」、日本建築学会計画系論文集、第529号、pp279-286、2000.3
- 17) 有川節夫他、「述語論理と論理プログラミング」、オーム社、1988

(2002年2月10日原稿受理、2002年7月5日採用決定)