

学位申請論文

建築設計プロセスの進化論的枠組みにおける
アルゴリズムック・デザインのプロトタイプ構築に関する研究

平成 29 年 03 月

松川 昌平

要旨

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

本論文は、近年進展しつつある「アルゴリズムック・デザイン」という建築の設計プロセスに関する基礎的研究である。建築設計プロセスに関する研究は、建築の工業化を背景として60年代に盛んに行われたが、その後70年代には一旦下火となった。しかしながら、95年以降の建築の情報化を背景として再び盛んに研究がなされてきており、今一度各種設計プロセスを総合的に研究することは、70年代に指摘された設計プロセス論の限界を情報テクノロジーによって乗り越え得る可能性があるという意味で現代的な意義があると考えられる。

本論文の目的は2つある。1つ目の目的は、(1) アルゴリズムック・デザインとその他の各種設計プロセスを「進化論的枠組み」という同じ枠組みの中に位置づけることを通して、それらの共通点や相違点を明らかにすることである。設計プロセスの分類に関する既往研究としては、C. アレグザンダーの『形の合成に関するノート』やJ.C. ジョーンズの『デザインの手法』、そして日本建築学会の設計方法小委員会編の『設計方法』パンフレットシリーズなどにおいて様々な方法が提案されているが、本論文では、それらコンピュータ導入以前の設計プロセスと共に、コンピュータ導入以後の設計プロセス——アルゴリズムック・デザイン、パラメトリック・デザイン、バイオミメティック・デザインなど——も含めて、同じ枠組みの中で総合的に分類を行っている点に新規性がある。また、各種設計プロセスにおける情報の流れに着目し、グラフ理論を用いたフローグラフとして設計プロセスを図式化することで、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを出来る限り定量的に明らかにしようとしている点に大きな特徴がある。

2つ目の目的は、(2) 上記枠組みに基いてアルゴリズムック・デザインを定式化し、その基礎的なシステムモデルとして「アルゴリズムック・デザインのプロトタイプ」を構築し、椅子や住宅のデザインへの工学的な応用可能性を示すことである。住宅へ応用する際には、与条件を満たした諸室の隣接関係を自動的に求める室配置問題を解くことが不可欠かつ重要である。本論では、2次元長方形分割図の列挙アルゴリズムを3次元へと拡張し、逆探索法を用いることができる範囲において3次元直方体分割図を列挙するアルゴリズムを提案している。室配置の基礎的なモデルとなる3次元直方体分割図に関しては、未だ漏れも重複もない列挙アルゴリズムは知られておらず、アルゴリズムック・デザインの住宅への応用可能性を示す上で要となる3次元直方体分割図を列挙するアルゴリズムの妥当性を詳細に検討した点が本論の主な成果である。

以上を踏まえ本論は大きく2部構成からなる。第1部（第2章～第5章）では目的(1)を、第2部（第6章～第11章）では目的(2)を扱う。すなわち本論において、第1部はコンピュータ導入前後の設計プロセスを総合的に俯瞰するための理論的な枠組みを提示した設計プロセス論の概論である。第2部はその枠組みに基づく幾つかの設計実験を試みるとともに、アルゴリズムック・デザインの住宅への応用可能性を示す

上で要となる「逆探索法を用いた直方体分割図の列挙アルゴリズム」を適用した部分を核にケーススタディを詳細に展開する。

以下各章の概略を述べる。

第1章では、本論におけるアルゴリズムック・デザインを提示し、本論全体の背景や目的について論じる。第1部、第2章では、最も単純化された設計プロセスのダイアグラムからスタートし、そのダイアグラムを、様々な理論を考察しながら書き換えていくことで、各種建築設計プロセスを位置づけるための「進化論的枠組み」を提案する。第3章では、第2章で提案した「進化論的枠組み」から演繹的に予想される帰結としてグラフ理論を用いた建築設計プロセスのフローグラフを導出する。第4章では、第一世代から第九世代までの各種設計プロセスを「進化論的枠組み」の中に帰納的に位置づける。また各種設計プロセスのフローグラフを作成することにより、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを明らかにすることを試みる。第5章では、各種設計プロセスのフローグラフをもとに分類を行い、それらの共通点や相違点を考察し、第1部の小結を述べる。第2部、第6章では、アルゴリズムック・デザインのフローグラフをもとに、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインの定式化を試みる。第7章では、第6章の定式化をもとに、アルゴリズムック・デザインのプロトタイプを構築する。第8章では、椅子のデザインへの工学的な応用可能性をケーススタディによって示す。第9章では、住宅のデザインへの工学的な応用可能性をケーススタディによって示す。第10章では、アルゴリズムック・デザインの住宅への応用可能性を示す上で要となる「逆探索法を用いた直方体分割図の列挙アルゴリズム」についての開発と検証とを行い、その妥当性を詳細に論じる。第11章では各種ケーススタディについて考察し、第2部の小結を述べる。最後に第12章では本論全体の結論を記し、今後の課題と展開について述べる。

以上から、アルゴリズムック・デザインとその他の各種設計プロセスを「進化論的枠組み」の中に位置づけることを通して、生成のフェイズ、評価のフェイズ、高適化のフェイズのいずれのフェイズにもアルゴリズムを用い、それらのフェイズを繰り返すことによって情報環境内で建築の試行錯誤を行っているという点が、他の設計プロセスにはないアルゴリズムック・デザインの特異性であることが明らかとなった。また、アルゴリズムック・デザインのプロトタイプを構築し、いくつかの限定的な条件の範囲ではあるが椅子や住宅への工学的な応用可能性を示すことができたことから、本論の目的は達成されたと結論付けている。

要旨

目次

序章

第1部

第2部

結章

後付

目次

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

1	序章	001
1.1	アルゴリズムックデザインの定義	002
1.2	設計プロセス論を研究テーマとする背景と意義	002
1.3	問題意識	004
1.4	目的	005
1.5	方法と章構成	005
第1部	建築設計プロセスの進化論適的枠組み	009
2	進化論的枠組みの提案	011
2.1	設計プロセスの分類に関する既往研究	012
2.2	実環境と設計者の環境	016
2.3	〈か〉〈かた〉〈かたち〉	018
2.4	〈かち〉	022
2.5	〈かち〉と建築の四層構造	025
2.6	利用者の環境	028
2.7	社会環境と情報環境	030
2.8	5つの環境の関係性	032
2.9	建築設計プロセスの進化論的枠組み	033
3	フローグラフの導出	035

3.1	デザインの定義	036	
3.2	フローグラフのフォーマット	036	要旨
3.3	設計プロセス	036	目次
3.4	フローグラフ	038	
3.5	環境	039	第1章
3.6	か・かた・かたち・かち	039	
3.7	ノード記号	040	第2章
3.8	ノード	040	
3.9	動力	041	第3章
3.10	動力の勢力	041	
3.11	かたちとかちの適合	042	第4章
3.12	かたちの出力	042	
3.13	設計プロセスの遺伝子	042	第5章
3.14	十進数	043	
3.15	設計プロセスの世代	043	第6章
			第7章
4	各種設計プロセスの位置づけ	045	第8章
			第9章
4.1	第一世代の設計プロセス：無自覚な状況、手工業的進化	046	
4.2	第二世代の設計プロセス：図面によるデザイン、自覚的な状況、モダニ ズムの設計プロセス	049	第10章
4.3	第三世代の設計プロセス：ポスト・モダニズムの設計プロセス	053	第11章
4.4	第四世代の設計プロセス：大量生産型の設計プロセス	057	
4.5	第五世代の設計プロセス：パターン・ランゲージ、超線型設計プロセス	060	第12章
4.6	第六世代の設計プロセス：BIM	063	
4.7	第七世代の設計プロセス：パラメトリック・デザイン、バイオ・ミメティッ	065	後付

ク・デザイン	
4.8 第八世代の設計プロセス：記号化の状況	069
4.9 第九世代の設計プロセス：アルゴリズムック・デザイン	074
5 第1部の小結	077
5.1 世代による分類	078
5.2 かたちの出力による分類	078
5.3 コンピュータライゼーションとコンピューテーションによる分類	079
5.4 コンピューテーショナルデザインの中での分類	080
5.5 第1部の小結	081
第2部 アルゴリズムック・デザインのプロトタイプ構築	083
6 アルゴリズムック・デザインの定式化	085
6.1 生成のフェイズ・評価のフェイズ・高適化のフェイズの定義	086
6.2 生成のフェイズ＝〈かたち〉の枠組み	088
6.3 評価のフェイズ＝〈かたち〉の適応度地形	090
6.4 高適化のフェイズ＝〈かたち〉の適応度地形を登る	093
7 アルゴリズムック・デザインのプロトタイプの構築	095

7.1 概要	096
7.2 生成のフェイズ	096
7.3 評価のフェイズ	098
7.4 高適化のフェイズ	104
7.5 アルゴリズムック・デザインのプロトタイプの実装	106
8 椅子への応用	109
8.1 概要	110
8.2 生成のフェイズ	110
8.3 評価のフェイズ	114
8.4 高適化のフェイズ	116
8.5 実験	118
9 住宅への応用	125
9.1 概要	126
9.2 生成のフェイズ	126
9.3 評価のフェイズ	138
9.4 高適化のフェイズ	156
10 逆探索法を用いた直方体分割図の列挙アルゴリズム	157

10.1 序	158
10.2 全域根付き木	162
10.3 列挙アルゴリズム	170
10.4 検証	186
10.5 結論と今後の課題	191

11 第2部の小結

193

11.1 生成のフェイズに関する考察	194
11.2 評価のフェイズに関する考察	194
11.3 高適化のフェイズに関する考察	195
11.4 第2部の小結	195

12 結章

197

12.1 結論	198
12.2 今後の課題と展望	200

参考文献

201

本研究に関連する論文、書籍、展覧会

211

謝辞

213

第1章

序章

本章では、研究の背景、問題意識、目的、方法などを述べる。

1.1 アルゴリズムックデザインの定義

本論文は、近年しつつある、「アルゴリズムック・デザイン」というデザインの新しい設計プロセス論に関する基礎的な研究である。まず本節では、アルゴリズムック・デザインの定義について述べる。日本建築学会編の『アルゴリズムック。デザイン—建築・都市の新しい設計手法』において、アルゴリズムック・デザインは、次のように定義される [1-1]。

本書では「アルゴリズムック・デザイン」を、「要求される課題を解くためのアルゴリズムを用い、解答としての形態や構成を生成する、設計方法」と定義する。

この定義のうち、「課題」は、いわゆる設計条件であり、「要求される」には、「設計者が自分で設計した」コンセプトなども含まれる。「アルゴリズムを用い」るのは手作業でもかまわない。しかし手作業はすぐ限界にあたる。コンピュータプログラムは不可欠だ。

「形態」は、狭い意味では文字どおり「かたち」であり、広い意味ではプランや配置、構造も含まれる。「生成」という言葉には、直接に成果品＝設計案を描くのではなく「アルゴリズム→プログラム」を通じて「間接的に」成果品を「生み出す」という意味を込めている。

「デザイン」としたのは、この方法は建築にかぎらず「設計」一般に共通するからである。

この定義をもう少し長く記すと、「設計に課せられた要求条件や設計者の意図を、解決（あるいは実現）するためのアルゴリズムを組み立て、そのアルゴリズムを実行するコンピュータプログラムを（一部あるいは全部に）用い、その課題を解決して目的をかなえた形態や構成（形態、配置、計画、構造、等）を成果品として生成する設計方法」のこととなる。

日本建築学会：アルゴリズムック・デザイン—建築・都市の新しい設計手法、鹿島出版会、2009、pp8（アンダーラインは筆者による）

1.2 設計プロセス論を研究テーマとする背景と意義

前節で述べたように、本研究のテーマは設計プロセス論である。しかしなぜ今また設計プロセス論をテーマにするのか？「また」と書いたのは、60年代に設計プロセス論が盛んに議論されたからである。本節では、その背景と意義を以下に述べる。

1960年に開催された「世界デザイン会議」においてメタボリズム・グループが結成された。そのメタボリズムのメンバーである菊竹清訓が、後に『代謝建築論——か・かた・かたち』[1-2]として結実する設計仮説をはじめて提出したのが1961年である。続く1962年には、世界初の設計方法に関する国際会議がロンドンで開催された[1-3]。この国際会議において先導的な役割を果たしたのが、J.C ジョーンズであった。設計方法に関する研究の世界的な潮流を受けて、1963年には日本建築学会の建築計画委員会に設計方法小委員会が設置された[1-4]。初代主査は池辺陽であった。そして同年には磯崎新による「プロセス・プランニング論」[1-5]が、さらに1964年にはクリストファー・アレグザンダーによる『形の合成に関するノート』(以下『ノート』)[1-6]が発表される。まさに設計プロセス論百花繚乱の時代である。

60年代に設計プロセス論が盛んに議論された背景には、戦後の高度経済成長による建築の工業化が挙げられる。それまでの手工業的な設計プロセスでは対処できない新たな環境としての「社会環境」が顕在化してきたときに、「設計過程における科学的方法の活用」と「工業的生産システムに対応する設計方法の確立」[1-7]をめざして設計プロセス論が要請されたのである。

しかし70年代に入ると、あれだけ華やかだった設計プロセス論が徐々に下火となる。池辺陽を恩師に持つ難波和彦は、『戦後モダニズムの極北——池辺陽試論』[1-8]のなかで、アレグザンダーが、『ノート』で示したような「客観的でシステマティックな方法」から「人間のあいまいな認知能力や感情にもとづくパターンランゲージを用いた方法」へと方向転換したことを例に挙げ、設計プロセス論の限界を指摘した。

そして時代は移って90年代。1990年にティム・バーナーズ＝リーによって初めてのWorld Wide Webが構築され、1995年に発売されたWindows 95が爆発的に普及すると、コンピュータ上はもちろんインターネット上にも新たな「情報環境」が顕在化してくる。

するとすぐに、実環境の物理的な制約から解き放たれた情報環境ならではの建築の可能性を模索する「サイバー・アーキテクチャー」が勃興し、日本でも雑誌等で特集が組まれた[1-9]。そして2002年、後にアルゴリズムック・デザインと呼ばれるようになるINDUCTION DESIGNを用いた先駆的なプロジェクト「誘導都市プロジェクト」が掲載された『建築は、柔らかな科学に近づく——INDUCTION DESIGN / 進化設計論』が渡辺誠によって上梓される[1-10]。その後アルゴリズムック・デザインは、主に構造分野で花開き、三井和男らによる『発見的最適化手法による構造のフォルムとシステム』[1-11]が2004年に、佐々木睦朗による『フラックス・ストラクチャー』[1-12]が2005年に、セシル・バルモンドによる『インフォーマル』[1-13]が2005年に相次いで発表された。そして2009年、先に紹介した『アルゴリズムック・デザイン——建築・都市の新しい設計手法』[1-1]において、アルゴリズムック・デザインという用語が初めて定義されると、2010年には、ハーバードGSDの准教授であるコスタス・テルジディ

スによる『アルゴリズムック・アーキテクチャ』[1-14]が翻訳され、2012年には渡辺誠による『アルゴリズムック・デザイン実行系 建築・都市設計の方法と理論』[1-15]が上梓された。こうしてアルゴリズムック・デザインは新しい設計プロセス論として注目を集めるようになる。

また同時期には、Achim Mengesらによる『Computational Design Thinking: Computation Design Thinking』(2011)[1-16]や、Wassim Jabiらによる『Parametric Design for Architecture』(2013)[1-17]、そしてIlaria Mazzoleniによる『Architecture Follows Nature-Biomimetic Principles for Innovative Design (Biomimetics)』(2013)[1-18]など、アルゴリズムック・デザインと類似の設計プロセス論である、コンピューテーショナル・デザイン、パラメトリック・デザイン、バイオミメティック・デザインなど、設計プロセスにコンピュータを介在させた様々な設計プロセス論が盛んに議論されるようになってきた。

このように、60年代の「社会環境」に加えて、90年代に「情報環境」という、ひとりの人間では情報処理しきれないような新たな環境が顕在化してきたときに、パラメトリック・デザイン、アルゴリズムック・デザインなどの設計プロセス論が再度議論されるようになってきたのである。

したがって今一度各種設計プロセスを総合的に研究することは、70年代に指摘された設計プロセス論の限界を情報テクノロジーによって乗り越え得る可能性があるという意味で現代的な意義があると考えられる。

1.3 問題意識

本研究の問題意識を以下にまとめる。

- (a) 日本建築学会編の『アルゴリズムック・デザイン—建築・都市の新しい設計手法』[1-1]においてアルゴリズムック・デザインの定義はされているものの、パラメトリック・デザイン、バイオミメティック・デザイン、コンピューテーショナル・デザインなどと呼ばれるコンピュータを用いた類似の設計プロセス論との共通点や相違点が明らかではない。
- (b) 60年代に盛んに研究されたコンピュータ導入以前の設計プロセス論との関係性も不明であるため、設計プロセスにコンピュータを介在させるのが、良いか/悪いという二項対立的な議論に陥りやすい。
- (c) アルゴリズムック・デザインの定式化されたシステムモデルはいまだ存在しないので、工学的に応用し難い。

1.4 目的

第1.3節で述べたような問題意識を解消するために、本研究では下記の2つの目的を設定する。

- (1) アルゴリズムック・デザインとその他の各種設計プロセスを同じ枠組みの中に位置づけることを通して、それらの共通点や相違点を議論するための理論的な枠組みを構築すること。
- (2) アルゴリズムック・デザインを定式化し、その基礎的なシステムモデルであるプロトタイプを構築することで、椅子や住宅のデザインへの工学的な応用可能性を示すこと。

目的(1)によって問題意識(a)(b)を、目的(2)によって問題意識(c)を解消することを目指す。

1.5 方法と章構成

本研究では、アメリカの論理科学者・科学哲学者 C. S. パース (C. S. Peirce) が示した科学的探求の方法を導入する。パースは科学的探究を三種類の推論——アブダクション (abduction)、演繹的推論 (deduction)、帰納的推論 (induction) ——からなるプロセスとしてモデル化している [1-19]。アブダクションは探求の第一段階であり、仮説を生成する。演繹は探求の第二段階であり、仮説からその帰結として予想を導く。帰納は探求の第三段階であり、予想と事実を照合して仮説を検証する [1-20]。

Fig. 1-1 に本論文の章構成を示す。本論では全体を大きく2部構成に分ける。第1部の「建築設計プロセスの進化論的枠組み」では、目的(1)を扱い、第2部の「アルゴリズムック・デザインのプロトタイプ構築」では、目的(2)を扱う。

第1部では、目的(1)を達成するために、パースの科学的探求の過程の三段階を踏襲する。まず第2章では、各種建築設計プロセスを位置づけるための「進化論的枠組み」を仮説的に提案する。第3章では、その「進化論的枠組み」から演繹的に予想される帰結としてグラフ理論を用いた建築設計プロセスのフローグラフを導出する。第4章では、各種設計プロセスを「進化論的枠組み」の中に帰納的に位置づける。また各種設計プロセスのフローグラフを作成することにより、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを明らかにする。第5章では、各種設計プロセスのフローグラフをもとに分類を行い、それらの共通点や相違点を考察し、第1部の小結を述べる。

第2部では、目的(2)を達成するために、同じくパースの科学的探求の過程の三段階を踏襲する。第6章では、アルゴリズムック・デザインのフローグラフをもとに、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインの定式化を試みる。第7章では、その定式をもとにアルゴリズムック・デザインのプロトタイプを構築する。そして第8章では椅子への、第9章では住宅への工学的な応用可能性を示す。第10章では、逆探索法を用いた直方体分割図の列挙アルゴリズムについてその詳細を述べる。第11章では各種ケーススタディについて考察し、第2部の小結を述べる。

そして最後に第12章では本論全体の結論と今後の展開について述べる。

パースの科学的探求の三段階に対応させていえば、第2章は目的(1)を探求するための第一段階＝アブダクションであり、第3章は目的(1)を探求するための第二段階＝演繹であり、第4章は目的(1)を探求するための第三段階＝帰納である。また同様に、第6章は目的(2)を探求するための第一段階＝アブダクションであり、第7章は目的(2)を探求するための第二段階＝演繹であり、第8,9章は目的(2)を探求するための第三段階＝帰納である。

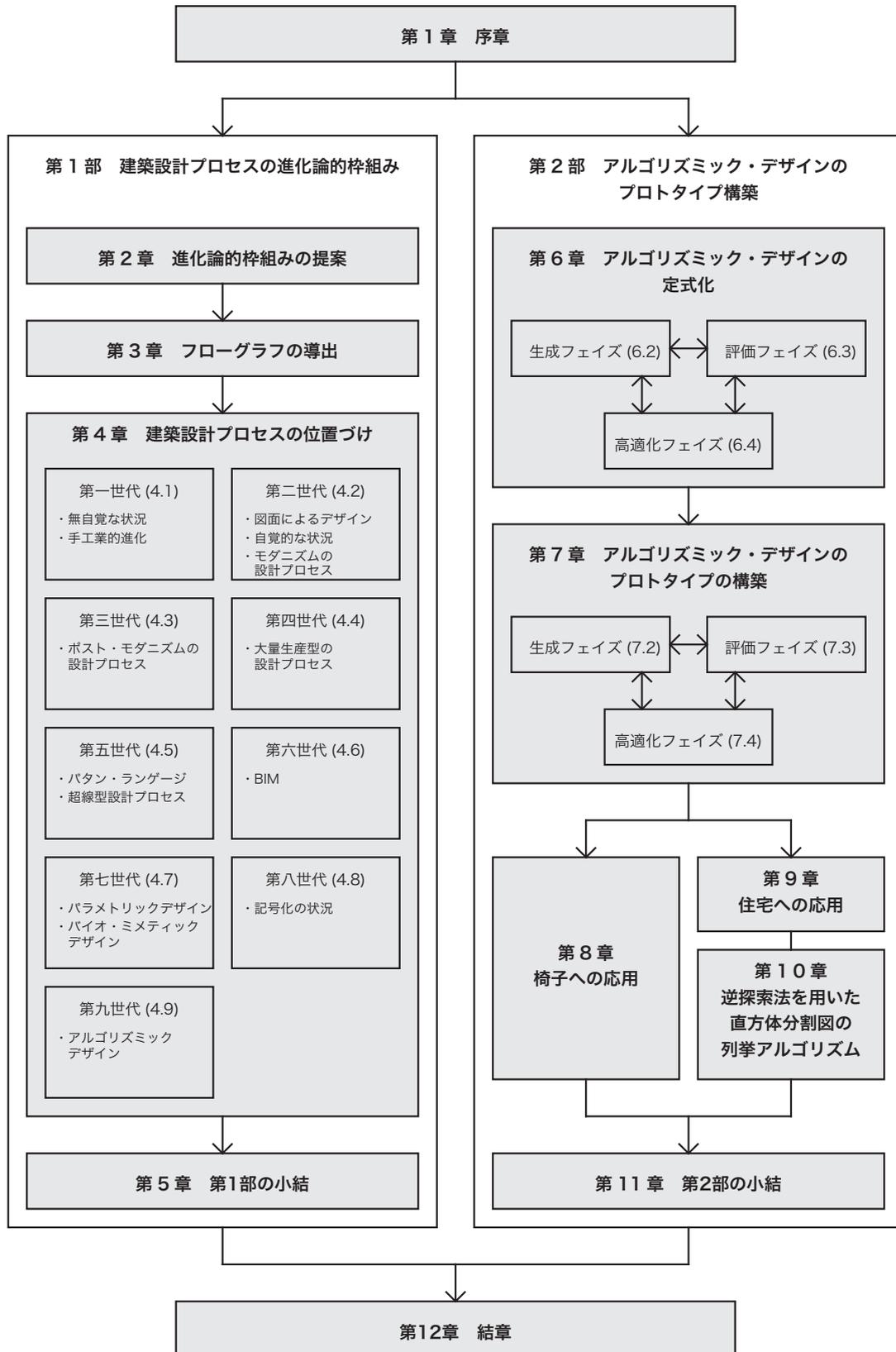


Fig. 1-1 本論文の章構成

要旨

目次

序章

第1部

第2部

結章

後付

第1部

建築設計プロセスの進化論的枠組み

要旨

目次

序章

第1部

第2部

結章

後付

第2章

進化論的枠組みの提案

本章では、最も単純化された設計プロセスのダイアグラムからスタートし、そのダイアグラムを、様々な理論を考察しながら書き換えていくことで、各種建築設計プロセスを位置づけるための「建築設計プロセスの進化論的枠組み」を提案する。

2.1 設計プロセスの分類に関する既往研究

設計プロセスの分類に関する既往研究としては、C. アレグザンダーの『形の合成に関するノート』(Fig. 2-1) [2-1] や J.C. ジョーンズの『デザインの手法』(Fig. 2-2) [2-2]、そして日本建築学会の設計方法小委員会編の『設計方法』パンフレットシリーズ (Fig. 2-3) [2-3] などにおいて様々な方法が提案されている。

本論文では、それらコンピュータ導入以前の設計プロセスと共に、コンピュータ導入以後の設計プロセス——アルゴリズムック・デザイン、パラメトリック・デザイン、バイオミメティック・デザインなど——も含めて、同じ枠組みの中で総合的に分類を行っている点に新規性がある。また、各種設計プロセスにおける情報の流れに着目し、グラフ理論を用いたフローグラフとして設計プロセスを図式化することで、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを出来る限り定量的に明らかにしようとしている点に大きな特徴がある。

本章では、最も単純化された設計プロセスのダイアグラムからスタートし、そのダイアグラムを、様々な理論を参照しながら随時書き換えていくことで、各種建築設計プロセスを位置づけるための「建築設計プロセスの進化論的枠組み」を提案する。

様々な理論を参照するのではなく限定された理論に依拠して論じるべきであるという意見もあるだろうが、第4.3節で述べるようなポスト・モダンな現代においては、様々な理論を参照しつつも、それらを紡ぎ合わせるようなひとつの仮説を組み立てることに意義があると考えられる。

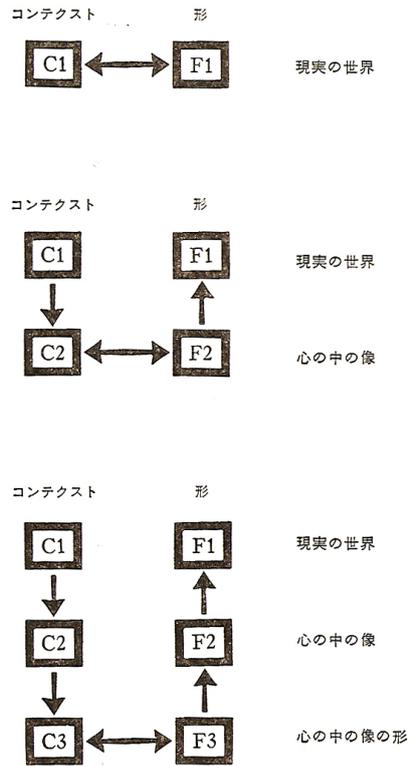


Fig. 2-1 「無自覚な状況」(上)、「自覚的な状況」(中)、
「記号化の状況」(下)

引用出典=クリストファー・アレグザンダー『形の合成に関するノート』(稲葉武司訳、
鹿島出版会、1978) p65

アウトプット → インプット ↓	2 調査されたデザイン状況	3 把握された、または変換された問題の構造	4 つきつめられた境界、記述された二次的な解と見きわめられた矛盾	5 選択性のあるデザイン案に組合わされた二次的な解	6 評価された選択性のあるデザイン案および選択された最終的デザイン
1 与条件	3-1 目標設定 3-2 文献調査 3-3 目に見える矛盾 3-4 ユーザーとの重接 4-1 プレレンストレーミング	3-2 文献調査 3-3 目に見える矛盾の検討 3-4 ユーザーとの重接 4-1 プレレンストレーミング 4-2 シネクティクス	3-3 目に見える矛盾の検討 4-1 プレレンストレーミング 4-4 形態学的図表	3-3 目に見える矛盾の検討 4-1 プレレンストレーミング 4-2 シネクティクス	2-1 軌跡転換 2-2 MatchettのFDM
2 調査されたデザイン状況		3-1 目標設定 3-9 データ整理 5-1 相関マトリックス 5-2 相互作用ネットワーク 5-8 分類 6-4 仕様の作成		5-4 システム変更 5-6 機能の革新 5-7 Alexanderの方法	
3 把握された、または変換された問題の構造	3-2 文献調査 3-5 質問形式 3-6 ユーザーの動作研究 3-7 システム的な試験 3-8 測定尺度の選択 3-9 データ収集		1-5 限界探索 3-7 システム的な試験 4-1 プレレンストレーミング 4-4 形態学的図表 6-2 評価基準の選択 6-3 ランキングと重みづけ 6-4 仕様の作成	4-1 プレレンストレーミング 4-2 シネクティクス 5-4 システム変更 5-5 境界移動	1-1 システムティックな改良 1-2 価値分析 1-3 システム工学 1-4 人間・機械系のデザイン 1-5 限界探索 1-6 Pageの戦略 1-7 CASA
4 つきつめられた境界、記述された二次的な解と見きわめられた矛盾		4-2 シネクティクス 4-3 思考障害の除去 5-3 AIDA 5-4 システム変更 5-5 境界移動 5-6 機能の革新 5-7 Alexanderの方法		4-1 プレレンストレーミング 4-2 シネクティクス 4-3 思考障害の除去 5-3 AIDA	5-3 AIDA
5 選択性のあるデザイン案に組合わされた二次的な解					1-2 価値分析 3-5 質問形式 3-6 ユーザーの動作研究 3-7 システム的な試験 3-8 測定尺度の選択 3-9 データ収集と整理 6-1 チェックリスト 6-2 評価基準の選択 6-3 ランキングと重みづけ 6-4 仕様の作成 6-5 Quirkの信頼性指標
6 評価された選択性のあるデザイン案、および選択された最終的デザイン					

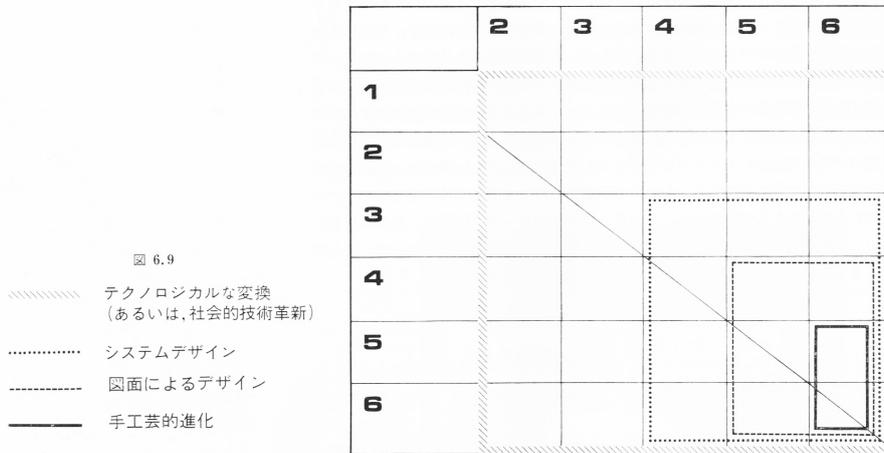
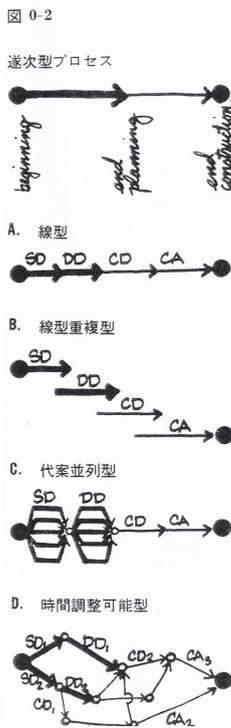
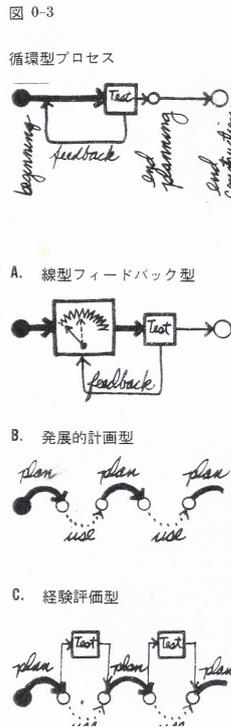


Fig. 2-2 「手工業的進化」、「図面によるデザイン」、
「システムデザイン」、「テクノロジカルな変換」
引用出典 = J. C. ジョーンズ『デザインの手法』
(池辺陽訳、丸善株式会社、1973) p82



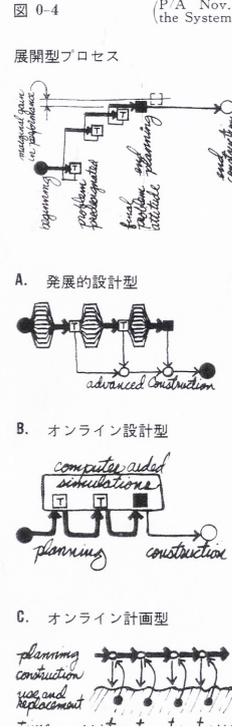
このプロセスは計画と実施に分離している。設計上の諸決定は、最初の決定を絶対的なものとして逐次進められる。その理由は、それらの諸決定が変更できないためである。設計の評価に対処する機構は備えられていない。

A. 建築家実務ハンドブック (ワシントン州アメリカ建築家協会 1963) B. SOM シカゴ事務所 製作マネジメントのシステム統合 マーチンゲネスター (ニュージャージー州イングルウッドクリフパブリシティスホール社 1964) 「製作マネジメントモデル」P. 112, 139 C. ジョンカール ワーネック サンフランシスコ設計事務所 D. 宇宙空間産業の計画コントロール (ロッキードミサイルおよび宇宙部門 カリフォルニア州 サニ



一連の設計案が、あらかじめ設定された性能基準に対して、そのうちの一つの解決案が基準を満たすまで、それぞれ、フィードバック・テストが行なわれる。そうすることによってプロセスが完了する。計画プロセスとしては設計と実施の評価が時間との関係で強調される。

A. サイバネティクス入門 ロス W. アシュビー (ニューヨーク:ジョン Wiley 社 1967) B. 開発計画 リチャードメイヤー (ニューヨーク:マクローヒル社 1965) C. パークレイ寮の環境分析 シムパジデルリン, マルセルバステイン (ニューヨーク, 教育施設研究所 1967) 性能測定技術 ホルストリッテル, ロバートフェッセンドン, ヘンリーサノフ (バークレイ カリフォルニア大学



設計は、環境して明らかに人間のさした問題の解試みるより式プロセスで具される。たぶ題の定式化のと、問題解決果の両者をシレートする能増大するであ

A. 未来計画 ニューヨーク事務カウドソット スコット事務所 B. EPT 設計シム ポストン所によって屏計されたハーエアとソフトアを含む広いビュターシム (グラフィディスプレイ) C. 「トポテントロジ来選択のため画理論」著者れを1969年3ニューヨークソンスデサスクールで初発表した。再一変化に基て 著 ロバートアスト (イリ州エルハスト

Fig. 2-3 「逐次型プロセス」「循環型プロセス」「展開型プロセス」

引用出典 = P/A Nov. '69 "What is the Systems Approach?"

日本建築学会建築計画委員会『設計方法Ⅱ 設計プロセス/ケーススタディ』

(彰国社、1973) p4

2.2 実環境と設計者の環境

建築設計プロセスの進化論的枠組みを提案するにあたって、最も単純化された設計プロセスのダイアグラムから考える (Fig. 2-4) [2-4]。本論では自然環境と人工環境をあわせた環境を**実環境 (Real Environment)**と呼ぶこととする。実環境から設計要求が情報として入力され、ブラックボックス化された設計者の脳内のなかで何らかの情報処理を経て、実環境に設計解としての物理的な建築物が出力される。本論では実環境に出力される**建築物**を *RI* と表す。

ここで情報処理を行なうのは設計者である。しかし Fig. 2-4 のように実環境から設計者へと情報が入力されているように描くと、あたかも実環境にすでに意味のある情報が存在し、それを入力すれば我々人間も機械に置き換えることができるという素朴な人間機械論を想起させる。このことに警笛を鳴らしたのはドイツの生物学者・哲学者ヤーコブ・フォン・ユクスキュルである。ユクスキュルは『生物から見た世界』 [2-5] において、生物それぞれの知覚器官に応じた主観的な環境のことを環世界 (Umwelt) と呼び、ありのままの環境 (Umgebung) と区別した。ここでの文脈で言い換えると、実環境 (Umgebung) から意味のある情報が天下りに設計者に入力されるのではなく、設計者が実環境へ主体的に働きかけることによって意味ある情報へと変換され、設計者の内に環世界 (Umwelt) が構築されるのである。本論では、このように設計者の脳内に構築された主観的な環世界のことを**設計者の環境 (Architect's Environment)**と呼ぶこととする。

以上を表現するために、Fig. 2-4 の入力の矢印の向きを反転させ、Fig. 2-5 のように表す。このとき情報が実環境から天下りに**入力**されるのではなく、設計者が実環境から情報を**取得**することを意味する。本論では、Fig. 2-4 のような矢印を**入力の矢印**と呼び、Fig. 2-5 のような矢印を**取得の矢印**と呼ぶ

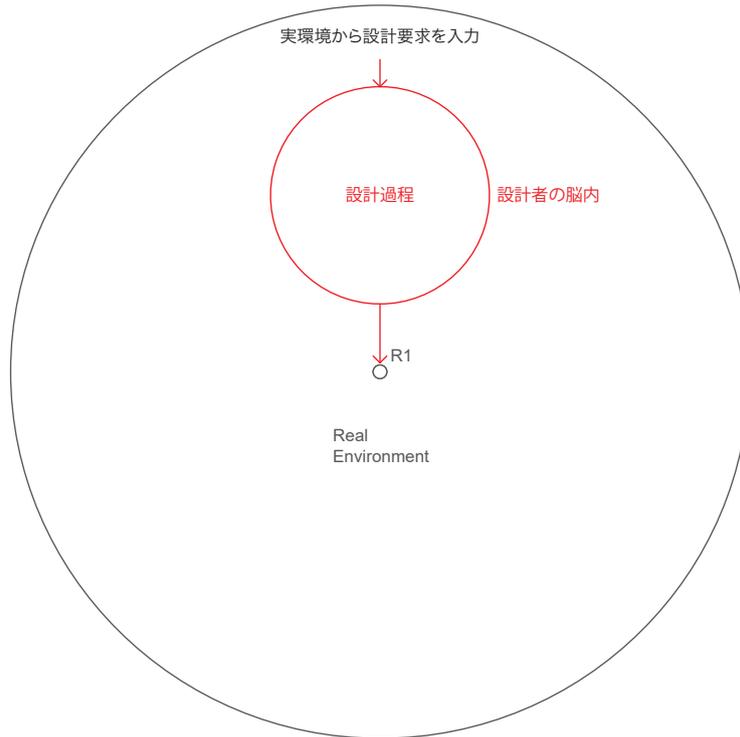


Fig2-4 基本的な設計プロセス 1

引用出典=吉川弘之「設計とはなにか——一般設計学の試み」(日本機械学会誌、第48巻、第749号、1981)をもとに筆者作成

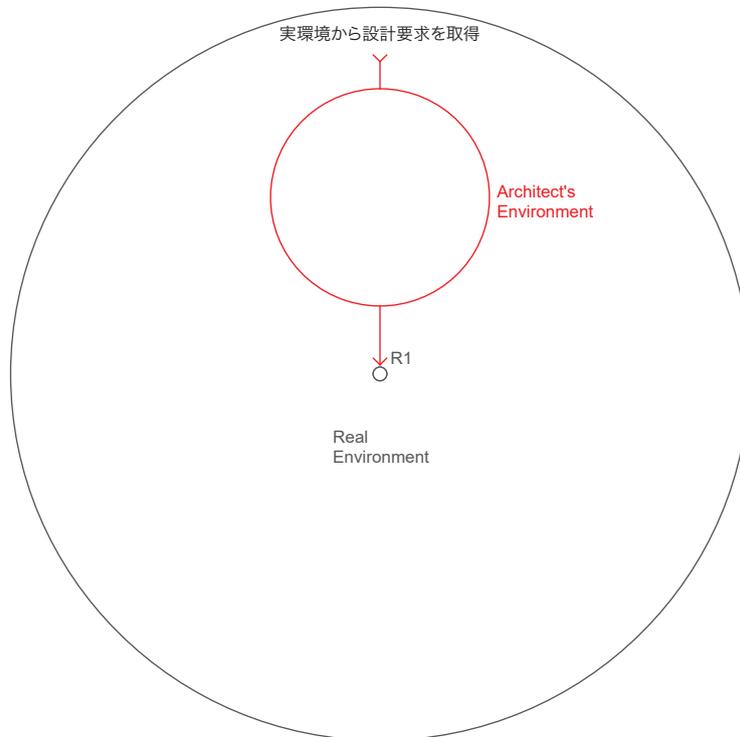


Fig2-5 基本的な設計プロセス 2

2.3 〈か〉〈かた〉〈かたち〉

本節では、Fig. 2-5 の設計者の環境内でブラックボックス化された設計プロセスとはどのようなものかを考察する。

先述した 60 年代の設計方法論のなかで、世界に先駆けて設計者の設計プロセスを鮮やかに提示したのが菊竹清訓の「か・かた・かたちの三段階論」である。菊竹は『代謝建築論』の序文 [2-6] のなかで、「か・かた・かたちの三段階論」は、武谷三男の『弁証法の諸問題』[2-7] に影響を受けたと言及している。また川添登の「国民文化の形成」[2-8] やルイス・カーンの「Order・Form・Shape」[2-9] との関連性も指摘されている。ここでは、暗黙的な設計プロセスを科学的に検証していくにあたって、「武谷三段階論」に着目する。

武谷は『弁証法の諸問題』に収められている「ニュートン力学の形成について」の中で、人間の自然認識は「現象論的段階」「実体論的段階」「本質論的段階」という三段階を経て発展すると論じている [2-10]。第一段階の「現象論的段階」は、ティコ・ブラーエが膨大な天体観測を記録したように、現象の知識を集める段階。第二段階の「実体論的段階」は、ヨハネス・ケプラーがブラーエの観測データをもとにケプラーの三法則を得たように、現象が起こるべき実体的な構造を知り、この構造の知識によって現象の記述が整理されて法則性を得る段階。そして第三段階の「本質論的段階」は、アイザック・ニュートンが天体だけでなく質量を持つ任意の物体がニュートンの三法則に従って運動することを示したように、任意の構造の実体は任意の条件の下にいかなる現象を起こすかということを明らかにする段階である。このように三段階を説明した後で、武谷は「物理学的認識は（中略）一律に進むのではなく、この三つの段階の環をくりかえして進むのである」と述べている。ここで三段階を、フィードバック・ループを備えた環構造として提示している点に留意したい。

菊竹の「か・かた・かたちの三段階論」の鮮やかな点は、日本語の〈かたち〉という言葉の成り立ちを丹念に調べ上げ、「現象論的段階」「実体論的段階」「本質論的段階」の各段階に対応して、〈かたち〉〈かた〉〈か〉という日本語を当てはめたことである (Fig2-6) [2-11]。

〈かたち〉を現象として感覚する段階から、〈かたち〉のなかにある普遍的技術あるいは法則性を理解する〈かた〉の段階を経て、〈かたち〉の原理ともいえるべき本質的問題をあつかう〈か〉の段階へ至るといような、デザインの「認識のプロセス」を提示する [2-12]。つまり人間の自然認識における三段階が、建物の認識における〈かたち〉に対しても同様に成り立つと考えたのである。この「認識のプロセス」は、武谷と同様に、三角形の環構造として図式化されている。さらに巧妙なのは、〈かたち〉→〈かた〉→〈か〉と進む「認識のプロセス」の流れを逆転させて、〈か（構想的段階）〉→〈かた（技術的段階）〉→〈かたち（形態的段階）〉という「実践のプロセス」

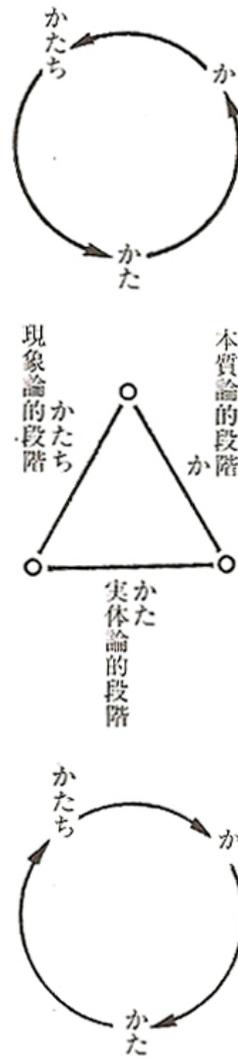


Fig. 2-6 「認識のプロセス」(上)と「実践のプロセス」(下)

引用出典＝菊竹清訓『代謝建築論』p14

を提示したことである [2-13]。つまり、〈か〉〈かた〉〈かたち〉という三角形の環構造はそのまま、「認識のプロセス」と「実践のプロセス」の違いを、時計回りと反時計回りという流れの違いだけで示したのである。

以上を踏まえて、Fig. 2-5 においてはブラックボックス化されていた設計者の環境の中に〈か〉〈かた〉〈かたち〉を明記し、設計者の環境を Fig. 2-7 のようにガラスボックス化する。

本論では設計プロセスの流れをグラフ理論を用いて図式化する。すなわち〈か〉〈かた〉〈かたち〉をノードとし、〈かたち〉は $A1$ 、〈かた〉は $A2$ 、〈か〉は $A3$ と表す。本論において、〈かたち〉は**形態**であり、〈かた〉は**型、法則、手順**であり、〈か〉は**構想、主義**である。〈かたち〉→〈かた〉→〈か〉という**認識のプロセス**は、ノードとノードを方向をもったエッジで結ぶことで、時計回りに表現する。また同様に、〈か〉→〈かた〉→〈かたち〉という**実践のプロセス**は、反時計回りの有向グラフで表現する。ただし、図が煩雑になることを避けるために、本論の図中では実践のプロセスのみ表現するが、認識のプロセスも含意されていることに留意したい。

設計者は実環境から設計要求を取得し、設計者の環境の中で、〈か〉 $A3$ → 〈かた〉 $A2$ → 〈かたち〉 $A1$ という実践のプロセスを何度も繰り返し試行錯誤することで、最もよいと思われる〈かたち〉 $A1$ を選び、実環境に建築物 $R1$ を出力する。

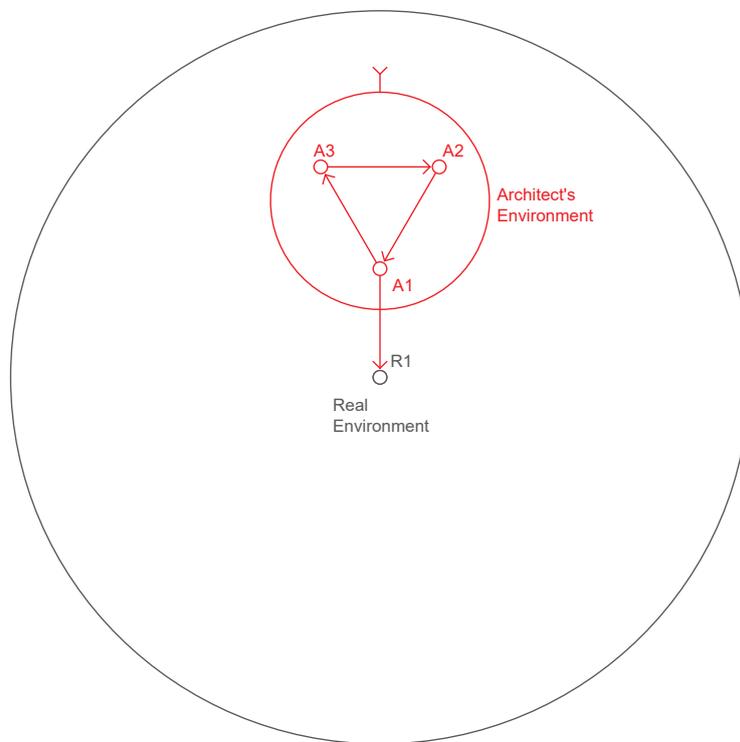


Fig. 2-7 基本的な設計プロセス 3

2.4 〈かち〉

本節では、菊竹の「か・かた・かたちの三段階論」を敷衍し、さらに、〈かち〉の段階を明示する。以下で〈かち〉の段階とはどのような段階なのかを論じる。

パースは、一般的に科学的論理的思考の方法として知られる「演繹 (deduction)」と「帰納 (induction)」に加えて、科学的発見や創造的思考において重要な役割を果たす「アブダクション (abduction)」という仮説的推論の方法を提唱した。アブダクションは次のように定式化される [2-14]。

驚くべき事実 B が観察される (B)、
 しかしもし A が真であれば、 B は当然の事柄であろう ($A \supset B$)、
 よって、 A が真であると考えべき理由がある ($\therefore A$)

この形式から明らかなように、アブダクションは後件 B を肯定することによって先件 A を肯定しているので、論理学でいう「後件肯定の誤謬」を冒している。したがって大いに間違える可能性のある論証力の弱いタイプの推論である。しかしパースは、未知について推測するという目的にとって、アブダクションは最も重要な働きをすると論じたのだ。

『アブダクション——仮説と発見の論理』の著者である米盛裕二は、「アブダクションは事実の発見、法則の発見、理論の発見のあらゆるレベルにおける発見にかかわる」と述べている [2-15]。この3つのレベルは武谷のいう「現象論的段階」「実体論的段階」「本質論的段階」に他ならない。しかし武谷は、それぞれの段階がどのように発見されるかについては触れていない。それは菊竹の〈か〉〈かた〉〈かたち〉においても同様である。対して米盛は同書のなかで、武谷と同じようにブラーエ→ケプラー→ニュートンの科学的発見のなかでいかにアブダクションが機能したかを検証していった。

パースはアブダクションによる仮説の形成は二段階あると考えた [2-16]。第一段階は、最初にいくつかの仮説を思いつくままに提起する示唆的な段階。第二段階は、それらの仮説の中からもっとも正しいと思われる仮説を選ぶ熟慮的な推論の段階である。例えばケプラーは、ブラーエの観察データをもとにしてその観察データを説明するような仮説を思いつくままに列挙するために、人間の本能的な能力であるアブダクティブな示唆を用い、その仮説の中から最も正しいと思う仮説を選択するという段階を、何度も繰り返すことでケプラーの三法則という仮説発見に至ったのである。

「か・かた・かたちの三段階論」に即して言い換えると、「認識のプロセス」においては、驚くべき〈かたち〉が観測されたときに、その〈かたち〉が当然であると説明できるような〈かた〉を思いつくままに列挙し、その〈かた〉の中から最も正しいと思われる〈かた〉を選択する。この〈かたち〉から〈かた〉へと至る中間段階、つま

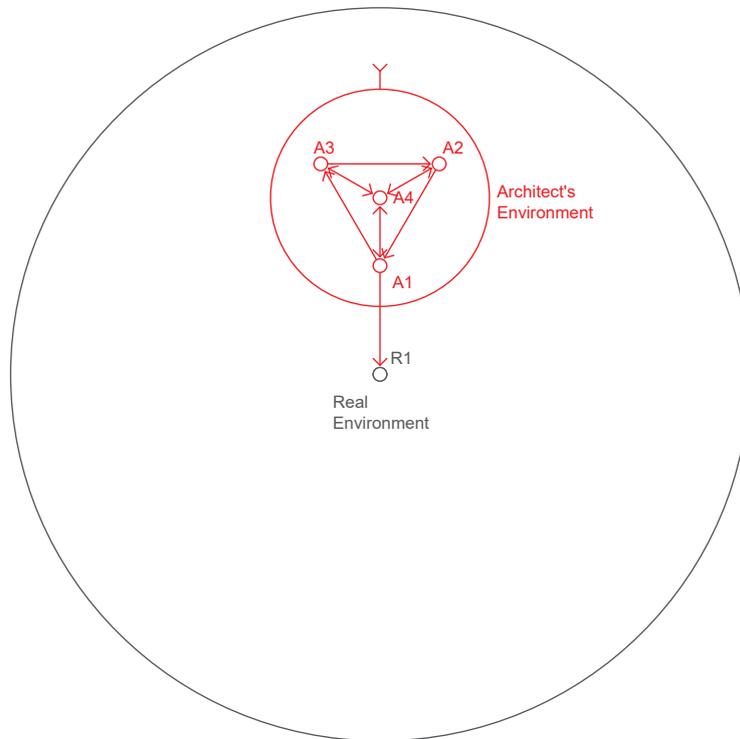


Fig. 2-8 基本的な設計プロセス 4

り**アブダクティブな推論の段階**が、先に提案した〈かち〉の段階である。同様に、驚くべき〈かた〉が観測されたときに、その〈かた〉が当然であると説明できるような〈か〉を思いつくままに列挙し、その〈か〉の中から最も正しいと思われる〈か〉を選択する。この〈かた〉から〈か〉へと至る中間段階、つまりアブダクティブな推論の段階も、〈かち〉の段階である。また「実践のプロセス」においては、複数の〈か〉〈かた〉〈かたち〉の中からより良い〈か〉〈かた〉〈かたち〉をそれぞれ**評価する段階**が、〈かち〉の段階である。アブダクションが事実の発見、法則の発見、理論の発見のあらゆるレベルにおける発見にかかわるように、〈かち〉の段階は、〈か〉〈かた〉〈かたち〉というそれぞれの段階を繋ぐための中間段階である。

以上を踏まえて、Fig. 2-7においては自明すぎて省略されていた**〈かち〉**を設計者の環境の中に明示し、Fig. 2-8のように書き換えることとする。追加された〈かち〉はA4と表現し、〈か〉〈かた〉〈かたち〉の三角形だった構造の真ん中に〈かち〉を加え、三角錐の構造で表現する。

2.5 〈かち〉と建築の四層構造

本節では、難波和彦の提唱する「建築の四層構造」[2-17]を参照し、〈かち〉を4つの層に分類する。

難波和彦は、建築を総合的に捉える視点を4つに分類し「建築の四層構造」を提唱している (Fig. 2-9)。第1層は建築を物理的なモノとして見る物理性の層である。第2層は建築をエネルギーの制御装置として見るエネルギー性の層である。第3層は建築を社会的な機能として見る機能性の層である。第4層は建築を意味を持った記号として見る記号性の層である。

難波自身「四層構造は建築それ自体に内在する属性ではない」[2-18]と注意を喚起している点からみても、「四層構造」は、設計者の環境の中に構築される〈かち〉の分類であるといえる。したがって本論では「建築の四層構造」に基いて、〈かち〉を4つの層に分類する。第1層の〈かち〉は〈かたち〉を物理的なモノとして評価する**物理的評価**の層とする。第2層の〈かち〉は〈かたち〉をエネルギーの制御装置として評価する**エネルギー的評価**の層とする。第3層の〈かち〉は〈かたち〉を社会的な機能として評価する**機能的評価**の層とする。第4層の〈かち〉は〈かたち〉を意味を持った記号として評価する**記号的評価**の層とする。

以上を踏まえて、Fig. 2-8において追加された〈かち〉を4つの層に分類し、Fig. 2-10のように書き換える。第1層の物理的評価である〈かち〉をA4、第2層のエネルギー的評価である〈かち〉をA5、第3層の機能的評価である〈かち〉をA6、第4層の記号的評価である〈かち〉をA7というノードで表現する。

四つの層 (建築学の領域)	視点 (建築の様相)	プログラム (デザインの条件)	技術 (問題解決の手段)	時間 (歴史)	サステイナブルデザインの プログラム
第1層・物理性 (材料・構法・構造学)	物理的な モノとして見る	材料部品 構造構法	生産運搬 組立施行	メインテナンス 耐久性風化	再利用リサイクル 長寿命化
第2層・エネルギー性 (環境工学)	エネルギーの 制御装置として見る	環境気候 エネルギー	気候制御装置 機械電気設備	設備更新 エントロピー	省エネルギー LCA 高性能化
第3層・機能性 (計画学)	社会的な 機能として見る	用途目的 ビルディングタイプ	平面計画 断面計画組織化	機能変化 ライフサイクル	コンバージョン 生活様式の変化
第4層・記号性 (歴史・意匠学)	意味を持った 記号として見る	形態空間 表象記号	様式幾何学 コード操作	街並記憶 ゲニウスロキ	リノベーション 保存と再生

Fig. 2-9 建築の四層構造

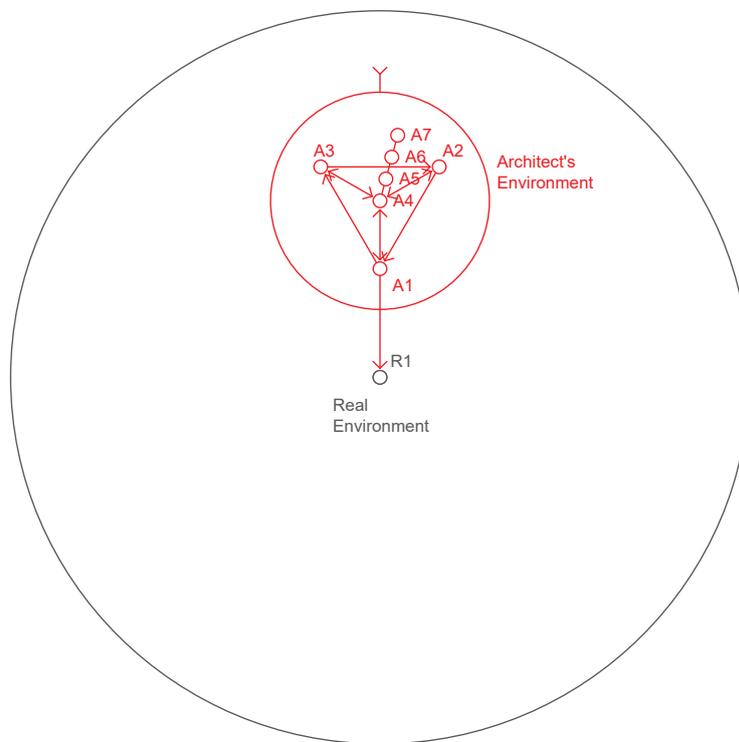


Fig. 2-10 基本的な設計プロセス 5

2.6 利用者の環境

日本建築学会に設計方法小委員会が設けられたのは、建築計画委員会の発足と同時の1963年であった。その研究成果は設計方法パンフレットシリーズにまとめられている。第I集『設計方法』[2-19]の発刊は1968年であった。その後、第II集の『設計プロセス／ケーススタディ』[2-20]を1971年に、第III集の『設計プロセス／道具の提案』[2-21]を1974年に、第IV集の『設計方法論』[2-22]を1981年に、そして第V集の『設計方法と設計主体』[2-23]は1989年に刊行された。

第4集までの設計プロセスモデルにおいては、建築物を設計する主体として設計者が意識されていたとしても、その設計の属性までを問題とすることはなかった。しかし第V集では、設計者の属性とともに、建築物の利用主体までも研究の対象となっている。

第1部の目的は、アルゴリズムック・デザインとその他の各種設計プロセスを同じ枠組みの中に位置づけることを通して、それらの共通点や相違点を議論するための理論的な枠組みを構築することである。したがって各種設計プロセスの共通点や相違点が議論できれば、それほど細かな設計主体の分類を必要としない。

そこで本論では、設計主体を個人や組織といった設計者の数による設計プロセスの差までは考慮しない。また意匠設計者、構造設計者、設備設計者など、設計の分野による設計プロセスの差も考慮しないものとする。したがって本論で設計者といったとき、個人や組織、そして意匠や構造や設備を含めた建築物を設計する専門的な技能を有する設計者一般を想定するものとする。

ただし、各種設計プロセスの共通点や相違点が議論する上で、建築物の利用主体を考慮することは有効であると考えられる。したがって本論では、建築物を設計する専門的な技能を有しない利用者は考慮する。その利用者にも設計者と同じように、利用者の脳内に構築された主観的な環世界が存在すると考えられる。したがってその環境のことを**利用者の環境 (User's Environment)**と呼ぶこととする。

設計者も最初は利用者だったわけだから、利用者の環境の中にも、潜在的には設計者と同じように、〈か〉〈かた〉〈かたち〉〈かち〉が存在すると考えられる。また〈かち〉も4層に分類できると考えられる。したがって、Fig. 2-11のように、利用者の環境も明示することとする。その際利用者の環境内の〈かたち〉をU1、〈かた〉をU2、〈か〉をU3、物理的評価の〈かち〉をU4、エネルギー的評価の〈かち〉をU5、機能的評価の〈かち〉をU6、記号的評価の〈かち〉をU7と表す。

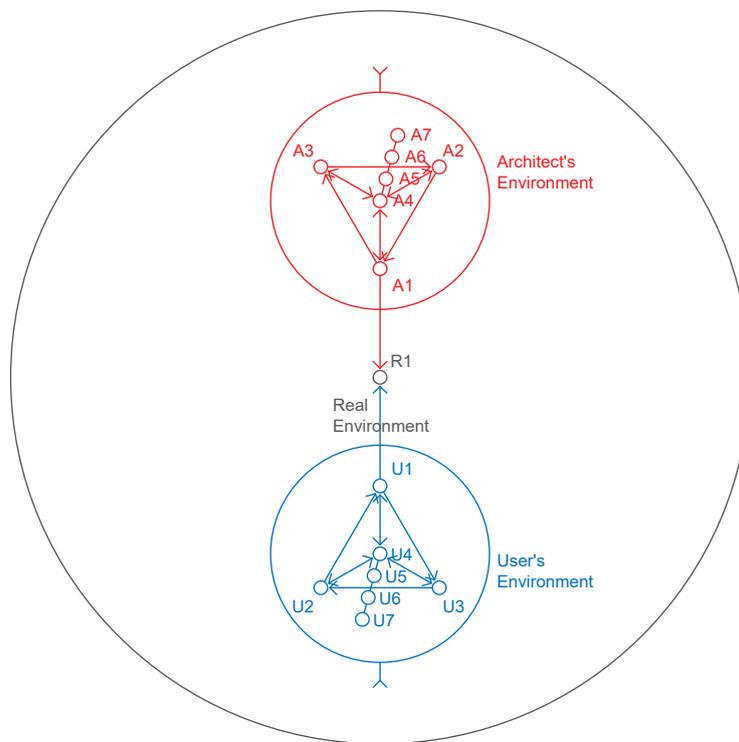


Fig. 2-11 基本的な設計プロセス 6

2.7 社会環境と情報環境

先述した日本建築学会の設計方法小委員会では、70年代以降も設計プロセスの探求が続けられ、90年代以前をシステムティックな設計方法を探求する第一シリーズとして総括した。そして90年代以降、設計方法を環境形成活動というより広範な文脈のもとに位置づけ、門内輝行を中心として「人間—環境系のデザイン」という第二シリーズが展開された [2-24]。門内は『人間—環境系のデザインの展望』という論考の中で、人間と自然環境だけの関係から、人工環境と社会—文化環境、さらに情報環境へと、段階的に環境が重層化されていく図を参照している [2-25] (Fig. 2-12)。したがって、自然環境と人工環境を実環境とすれば、本論においても、社会—文化環境と情報環境を考慮する必要がある。

本論では、社会—文化環境を「社会環境 (Social Environment)」、情報環境を「情報環境 (Information Environment)」とする。設計者や利用者の総体が社会環境や情報環境を構築しているわけだから、社会環境や情報環境の中へも、設計者や利用者の〈か〉〈かた〉〈かたち〉や4層の〈かち〉が客観的に外部化される可能性がある。したがって、Fig. 2-13のように、社会環境や情報環境も明示することとする。その際社会環境内の〈かたち〉をS1、〈かた〉をS2、〈か〉をS3、物理的評価の〈かち〉をS4、エネルギー的評価の〈かち〉をS5、機能的評価の〈かち〉をS6、記号的評価の〈かち〉をS7と表す。また同様に、情報環境内の〈かたち〉をI1、〈かた〉をI2、〈か〉をI3、物理的評価の〈かち〉をI4、エネルギー的評価の〈かち〉をI5、機能的評価の〈かち〉をI6、記号的評価の〈かち〉をI7と表す。

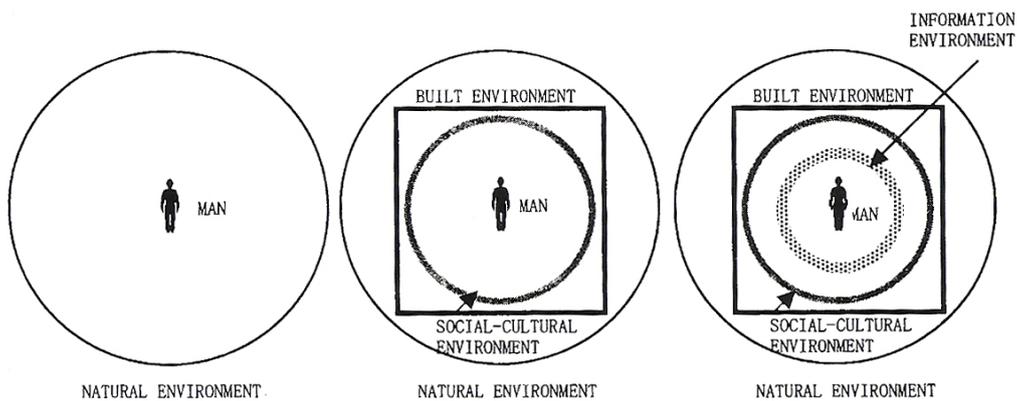


Fig. 2-12 人間—環境系の多層性

引用出典＝門内輝行『人間—環境系のデザインの展望——21世紀のデザインビジョン』
(日本機械学会、No. 06-5、Design シンポジウム 2006 講演論文集)

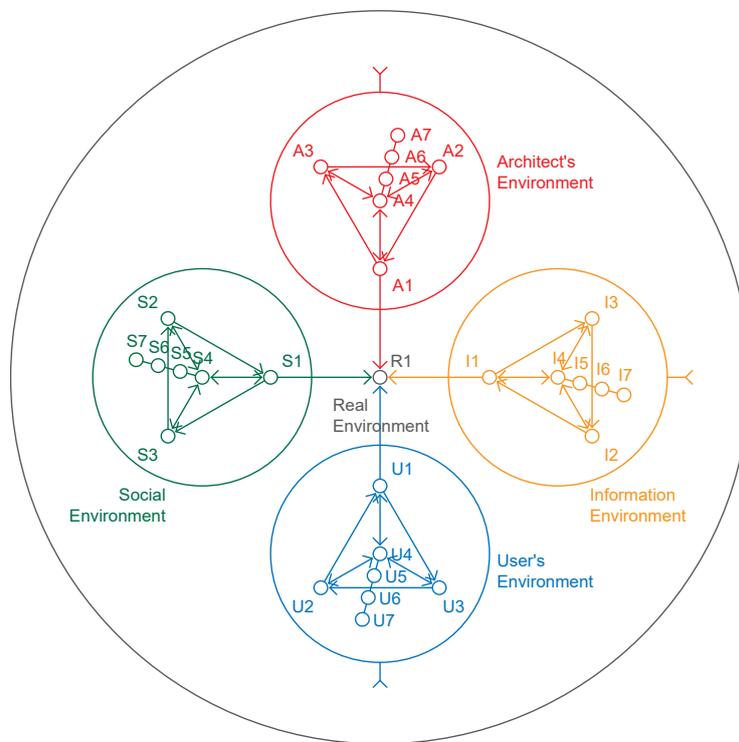


Fig. 2-13 基本的な設計プロセス 7

2.8 5つの環境の関係性

本節では、これまでに登場した5つの環境—すなわち「実環境」「設計者の環境」「利用者の環境」「社会環境」「情報環境」の関係性について、さらに考察する。

先述したように、アレグザンダーは『ノート』のなかで、設計プロセスを「無自覚な状況」「自覚的な状況」「記号化の状況」の三段階に分けて図式化している (Fig. 2-1)。形とコンテキストの関係については後述するが、ここでは暗黙的な設計プロセスが段階的に外部化されていく様子に注目する。5つの環境との関係性という文脈で捉え直せば、「無自覚な状況」では「実環境」と「設計者の環境」や「利用者の環境」の関係性のみで設計プロセスは依然としてブラックボックスのままなのに対して、「自覚的な状況」では「設計者の環境」や「利用者の環境」から「社会環境」へとプロセスが外部化され、さらに「記号化の状況」では「情報環境」にまで外部化されていると見ることができる。

さらに、前述した世界初の「設計方法に関する国際会議」において中心的な役割を果たした J.C. ジョーンズは『デザインの手法』のなかで、設計プロセスの進化を「手工業的進化」「図面によるデザイン」「システムデザイン」「テクノロジカルな変換」という4段階に分けている (Fig. 2-2)。「システムデザイン」と「テクノロジカルな変換」を共に「記号化されたプロセス」とみれば、先のアレグザンダーの図式に対応づけることができる。

このように見てくると、まず「実環境」があって、建築物を設計する専門的な技能を有しない「利用者の環境」の中で、〈か〉〈かた〉〈かたち〉〈かち〉が「自覚」されると建築物を設計する専門的な技能を有した「設計者の環境」が構築され、さらにその自覚された〈か〉〈かた〉〈かたち〉〈かち〉が、段階的に「社会環境」や「情報環境」へと外部化されていくことによって、設計プロセスが進化してきた状況が浮かび上がってくる。このことは、実環境の設計者としてのアーキテクトから、社会環境の設計者としてのアーキテクト、そして情報環境の設計者としてのアーキテクトというように、現在アーキテクトという職能が拡張してきていることから理解できるだろう。

2.9 建築設計プロセスの進化論的枠組み

以上の考察を踏まえて、本節ではまず「建築設計プロセスの進化論的枠組み」を提案し、さらに、「各種設計プロセスは建築設計プロセスの進化論的枠組みの中で進化してきたのである」というアブダクティブな仮説を生成する。

Fig. 2-13の基本的な設計プロセス7で示したように、設計プロセスでは実環境に建築物が出力される。建築物が出力されれば値1を、出力されなければ値0を割り当てる。また「利用者の環境」と「設計者の環境」には〈か〉〈かた〉〈かたち〉、そして4層の〈かち〉という7つの段階があった。しかしこの7段階は、利用者や設計者のなかで必ずしも自覚されているとは限らない。そこで、自覚されていれば値1、自覚されていなければ値0というように自覚の状態をビット化する。さらに、「社会環境」と「情報環境」にもそれぞれ7つの段階が外部化される可能性がある。ここでも同様に外部化の状態をビット化する。

そうすると、実環境で1ビット、4つの環境のそれぞれに7ビットあるので、設計プロセスの状態空間は、0000000 0000000 0000000 0000000 0から1111111 1111111 1111111 1111111 1までの29ビットで表現できることがわかる。本研究では、このビット列を「設計プロセスの遺伝子 (DP-GENE)」と呼ぶ。「設計プロセスの遺伝子」のありべきパターン数は、全部で2の29乗すなわち536,870,912パターンとなる。この536,870,912パターンの設計プロセスが「建築設計プロセスの進化論的枠組み」である (Fig. 2-15)。すなわち、無限にあるであろう設計プロセスを、有限の設計プロセスにフレーミング (枠組み) したのである。

先述したアブダクションの定式化にしたがって、本節の仮説を定式化すると次のようになる。

驚くべきことに、アルゴリズムック・デザインは現在最新の設計プロセスとして注目されている (B)。

しかしもし、各種設計プロセスが「建築設計プロセスの進化論的枠組み」のなかで進化してきた (A) ならば、アルゴリズムック・デザインが注目される (B) のは当然の事柄であろう (A ⊃ B)。

よって、各種設計プロセスが建築設計プロセスの進化論的枠組みのなかで進化してきたのであると考えるべき理由がある (∴ A)

第3章

フローグラフの導出

本章では、第2章で提案した「建築設計プロセスの進化論的枠組み」から演繹的に予想される帰結として、グラフ理論を用いた建築設計プロセスのフローグラフを導出する。

3.1 デザインの定義

本節ではまず議論の前提として、デザインの定義をしておく。アレグザンダーは『ノート』のなかで、建物がもつ調和のとれた全体性を、「形」と「コンテキスト」に区分したうえで、デザインの定義をこう述べる。「デザインの最終の目的は形である。(中略) どのデザインの問題も、求められている形と、その形全体との脈絡、すなわちコンテキストという二つの存在を適合 (fit) させようとする努力で始まるという考え方に基づいている」[3-1]。

本論の文脈に沿って言い換えると、「デザインの最終の目的は実環境に建築物としての〈かたち〉を出力することである。どの建築設計においても、求められる〈かたち〉と、その〈かたち〉全体との脈絡、すなわちコンテキストという二つの存在を適合させようとする努力で始まるという考え方に基づいている」となる。ここで〈かたち〉と環境のコンテキストがよりよく適合しているかどうかを評価する主体は設計者か利用者である。すなわち設計者あるいは利用者が〈かたち〉の〈かち〉を評価するのである。

したがって本論におけるデザインとは、「**実環境、環世界、社会環境、情報環境**という複合的な〈かち〉に適合するような〈かたち〉を出力すること」と定義する。

3.2 フローグラフのフォーマット

本章では、第2章で提案した建築設計プロセスの進化論的枠組み (Fig. 2-13) を元に、対象とする設計プロセスの流れをグラフ理論を用いて図式化することを試みる。本論では、そのグラフを**フローグラフ**と呼ぶ。

次章の第4章からは、対象とする設計プロセスを建築設計プロセスの進化論的枠組みの中に位置づけながらフローグラフを用いて表現することで、対象とする設計プロセスが、実環境、環世界、社会環境、情報環境という複合的な〈かち〉に適合するような〈かたち〉をどのように出力しているのか、そのメカニズムを明らかにしていく。

Fig. 3-1 に対象とする設計プロセスのフローグラフを記述するためのフォーマットを示す。以降、このフォーマットのそれぞれの項目を定義していく。

3.3 設計プロセス

設計プロセスの項目には、次章の第4章において、進化論的枠組みの中に位置づけていく設計プロセスを記載する。

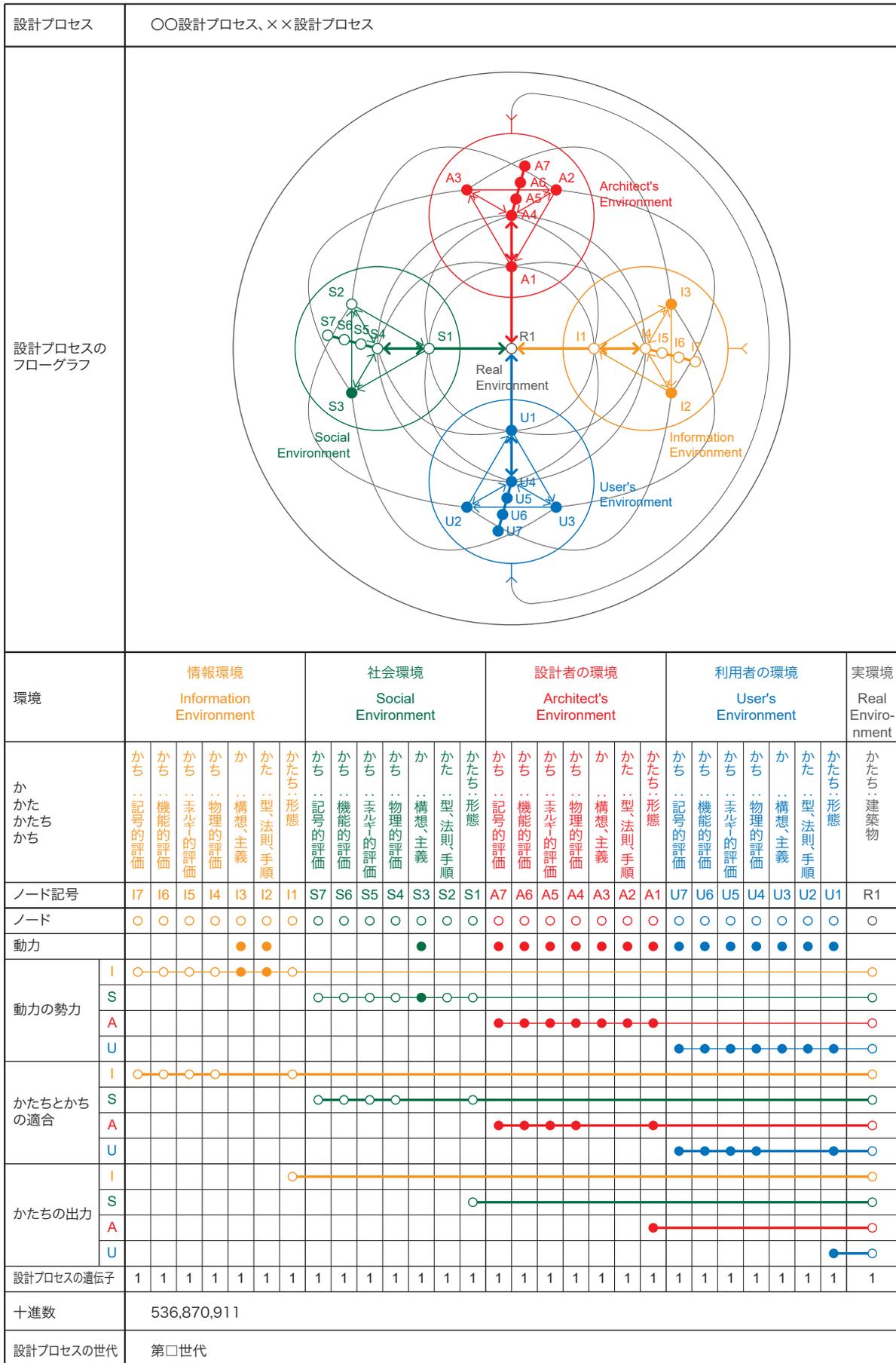


Fig. 3-1 フローグラフのフォーマット

本論では、その位置づける対象となる設計プロセスは以下とおりとする。

設計プロセス論を牽引したクリストファー・アレグザンダーの設計プロセスから、

- ・無自覚な状況 [3-1]
- ・自覚的な状況 [3-1]
- ・記号化の状況 [3-1]
- ・パターン・ランゲージ [3-2]

同じく J.C. ジョーンズの『デザインの手法』から、

- ・手工業的進化 [3-3]
- ・図面によるデザイン [3-3]

また、大きな時代区分の代表として、

- ・モダニズムの設計プロセス
- ・ポスト・モダニズムの設計プロセス

アノニマスな設計プロセスの代表として、

- ・大量生産型の設計プロセス

1995 以後の設計プロセスの代表として、

- ・超線型設計プロセス [3-4]

コンピュータ導入以後の設計プロセスの代表として、

- ・BIM[3-5]
- ・パラメトリック・デザイン [3-6]
- ・バイオ・ミメティック・デザイン [3-7]
- ・アルゴリズムック・デザイン [3-8]
- ・コンピューテーショナル・デザイン [3-9]

を対象とする。

3.4 フローグラフ

フローグラフの項目には、対象とする設計プロセスのフローグラフを描く。任意のノード X から任意のノード Y へ情報が入力される時、ノード X からノード Y へと入

力の矢印を描く。一方ノード X からノード Y の情報を取得するとき、ノード X からノード Y へと取得のの矢印を描く。ノード X とノード Y の間に情報の流れが発生しないが結びついているとき、ノード X とノード Y を単にエッジで結ぶ。本論ではそのときのエッジを結合のエッジと呼ぶ。ノード X とノード Y の間に情報の流れも結びつきもないときは、ノード X とノード Y をエッジで結ばない。

〈かたち〉 → 〈かた〉 → 〈か〉 という認識のプロセスは、ノードとノードを方向をもったエッジで結ぶことで、時計回りに表現する。また同様に、〈か〉 → 〈かた〉 → 〈かたち〉 という実践のプロセスは、反時計回りの有向グラフで表現する。ただし、図が煩雑になることを避けるために、本論の図中では実践のプロセスのみ表現するが、認識のプロセスも含意されていることに留意したい。

3.5 環境

第2.7節ですでに述べたが、本論では5つの環境—すなわち「実環境 (Real Environment)」「設計者の環境 (Architect's Environment)」「利用者の環境 (User's Environment)」「社会環境 (Social Environment)」「情報環境 (Information Environment)」を対象とする。

この5つの環境をどの順番で記載するかによって、設計プロセスの遺伝子の桁が変わる。第2.7節ですでにみたように、「実環境 (Real Environment)」 → 「設計者の環境 (Architect's Environment)」 → 「利用者の環境 (User's Environment)」 → 「社会環境 (Social Environment)」 → 「情報環境 (Information Environment)」の順に〈か〉〈かた〉〈かたち〉〈かち〉が「自覚」あるいは「外部化」されると考えられるので、この順に桁が上がっていくものとする。

3.6 か・かた・かたち・かち

第2.2節ですでに述べたが、本論において、〈かたち〉は形態であり、〈かた〉は型、法則、手順であり、〈か〉は構想、主義である。

また第2.4節ですでに述べたように、第1層の〈かち〉は物理的評価の層であり、第2層の〈かち〉はエネルギー的評価の層であり、第3層の〈かち〉は機能的評価の層であり、第4層の〈かち〉は記号的評価の層である。

環境と同じように、この〈か〉〈かた〉〈かたち〉と4層の〈かち〉をどの順番で記載するかによって、設計プロセスの遺伝子の桁が変わる。〈か〉〈かた〉〈かたち〉に関しては、〈かたち〉 → 〈かた〉 → 〈か〉 という認識のプロセスによって「自覚」あ

るいは「外部化」されると考えられるので、この順に桁が上がっていくものとする。

〈かち〉の4層に関しては、第4層の記号性が歴史的に最も遅くに明らかにされたので、1層から4層の順に桁が上がっていくものとする。

最後に4層の〈かち〉が〈かたち〉よりも桁が下か、〈か〉よりも桁が上か議論の余地はあるが、これも第4層の記号性の〈かち〉が歴史的に最も遅くに明らかにされたので、4層の〈かち〉は〈か〉よりも桁が上であるものとする。

3.7 ノード記号

実環境に建築物のノード記号は *RI* と表す。

利用者の環境の〈かたち〉を *U1*、〈かた〉は *U2*、〈か〉は *U3*、第1層の物理的評価である〈かち〉を *U4*、第2層のエネルギー的評価である〈かち〉を *U5*、第3層の機能的評価である〈かち〉を *U6*、第4層の記号的評価である〈かち〉を *U7* というノード記号で表す。

設計者の環境の〈かたち〉を *A1*、〈かた〉は *A2*、〈か〉は *A3*、第1層の物理的評価である〈かち〉を *A4*、第2層のエネルギー的評価である〈かち〉を *A5*、第3層の機能的評価である〈かち〉を *A6*、第4層の記号的評価である〈かち〉を *A7* というノード記号で表す。

社会環境の〈かたち〉を *S1*、〈かた〉は *S2*、〈か〉は *S3*、第1層の物理的評価である〈かち〉を *S4*、第2層のエネルギー的評価である〈かち〉を *S5*、第3層の機能的評価である〈かち〉を *S6*、第4層の記号的評価である〈かち〉を *S7* というノード記号で表す。

情報環境の〈かたち〉を *I1*、〈かた〉は *I2*、〈か〉は *I3*、第1層の物理的評価である〈かち〉を *I4*、第2層のエネルギー的評価である〈かち〉を *I5*、第3層の機能的評価である〈かち〉を *I6*、第4層の記号的評価である〈かち〉を *I7* というノード記号で表す。

3.8 ノード

設計プロセスでは実環境に建築物が出力される。建築物が出力されれば *RI* のノードを表示する。また「利用者の環境」と「設計者の環境」には〈か〉〈かた〉〈かたち〉、そして4層の〈かち〉という7つの段階があった。しかしこの7段階は、利用者や設計者のなかで必ずしも自覚されているとは限らない。そこで、自覚されていればそれぞれのノードを表示する。さらに、「社会環境」と「情報環境」にもそれぞれ7つの段階が外部化される可能性がある。ここでも同様に外部化されているノードを表示する。

3.9 動力

円環状に繋がった高低差のないプールに水をはっただけでは流れは生まれない。動力があって始めてプールに水の流れが生まれ、流れるプールとなる。同様に、フローグラフに情報の流れが生まれるためには、**動力**が必要である。

設計者あるいは利用者は人間であるから、〈か〉〈かた〉〈かたち〉そして4層の〈かち〉のいずれのノードも動力になり得る。本論では動力となるノードは、ノードの中を塗りつぶして●のように表現する。動力ではないノードはノードの中を塗りつぶさないで○のように表現する。

問題は、社会環境と情報環境の〈か〉〈かた〉〈かたち〉そして4層の〈かち〉のノードが動力になり得るかということである。

ローレンス・レッシングは『CODE』[3-10]の中で、他者を規制（コントロール）する方法として「規範」「法」「市場」「アーキテクチャ」があると述べている。したがって、社会環境においては、「規範」や「法」や「市場」としての〈か〉が外部化されていれば、それを動力として、社会環境から設計者や利用者へ情報が流れることがあり得ると考えられる。

情報環境では、電源が供給されていさえすれば、コンピュータアルゴリズムとしての〈かた〉がプログラミング言語で実装され情報の流れが常に生み出される。したがって、情報環境の〈かた〉は動力になり得ると考えられる。また昨今人工知能が囲碁棋士のトッププロを破るなど、人工知能の技術的発達が目覚ましい。現在のところ人工知能が設計者や利用者のような〈か=構想、主義〉を持ち得ていないが、将来的には持ちうる可能性は否定できない。したがって、情報環境の〈か〉も動力になり得ると考えられる。

3.10 動力の勢力

設計プロセスのフローグラフは、常に動力のノードを起点として情報の流れが生まれる。動力のノードと他のノードとの間が入力の矢印あるいは単にエッジで連結されていた場合、情報の流れはそれらのノード間を辿って流れ続ける。しかしもしノードXからノードYが取得の矢印で連結されていた場合、ノードYからノードXへ情報は自動的に流れず、ノードX側の動力を使ってノードYから情報を取得しなければならない。

本論では、動力のノードを起点として情報の流れが生まれる範囲のノードを**動力の**

勢力とする。動力の勢力は、動力のノードが属する環境の行に記載するものとする。

3.11 かたちとかちの適合

本論でフローグラフを用いるのは、デザインの定義である〈かたち〉と〈かち〉がどのように適合されているのか、そのメカニズムを明らかにするためである。

フローグラフの中で、〈かたち〉と〈かたち〉、〈かち〉と〈かち〉、そして〈かたち〉と〈かち〉のノードが連結されていれば、そのエッジを太線で表す。このことによって、〈かたち〉と〈かち〉がどのように適合されているのかが可視化されることとなる。

3.12 かたちの出力

デザインの最終の目的は実環境に建築物としての〈かたち〉*RI*を出力することである。*RI*に連結されているノードにおいて、情報の流れを逆順に辿れば動力のノードに行き着くはずである。その動力のノードが属する環境の行に、〈かたち〉のノードの連結だけを取り出して記載する。

3.13 設計プロセスの遺伝子

設計プロセスでは実環境に建築物が出力される。建築物が出力されれば値1を、出力されなければ値0を割り当てる。また「利用者の環境」と「設計者の環境」には〈か〉〈かた〉〈かたち〉、そして4層の〈かち〉という7つの段階があった。しかしこの7段階は、利用者や設計者のなかで必ずしも自覚されているとは限らない。そこで、自覚されていれば値1、自覚されていなければ値0というように自覚の状態をビット化する。さらに、「社会環境」と「情報環境」にもそれぞれ7つの段階が外部化される可能性がある。ここでも同様に外部化の状態をビット化する。

そうすると、実環境で1ビット、4つの環境のそれぞれに7ビットあるので、設計プロセスの状態空間は、0000000 0000000 0000000 0000000 0から1111111 1111111 1111111 1111111 1までの29ビットで表現できることがわかる。本研究では、このビット列を「設計プロセスの遺伝子 (DP-GENE)」と呼ぶ。「設計プロセスの遺伝子」のありうべきパターン数は、全部で2の29乗すなわち536,870,912パターンとなる。この536,870,912パターンの設計プロセスが「設計プロセスの進化論的枠組み」である

3.14 十進数

設計プロセスの遺伝子は 29 ビットすなわち 2 進数で現れる数字だが、これを十進数で表した数字を記載する。

3.15 設計プロセスの世代

対象とする設計プロセスの十進数を比較して、数字の小さい順に設計プロセスの世代を割り当てる。

要旨

目次

序章

第1部

第2部

結章

後付

第4章

各種設計プロセスの位置づけ

本章では、第一世代から第九世代までの各種設計プロセスを「建築設計プロセスの進化論的枠組み」の中に帰納的に位置づける。また各種設計プロセスのフローグラフを作成することにより、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを明らかにすることを試みる。

4.1 第一世代の設計プロセス：無自覚な状況、手工業的進化

本節では、アレグザンダーのいう「無自覚な状況」、J・C・ジョーンズのいう「手工業的なプロセス」を位置づける。

無自覚な状況というくらいなので、設計者の環境の7つのノードの値はすべて0。もちろん社会環境や情報環境にもプロセスは外部化されていないので値はすべて0となる。

しかしながら、設計するという自覚なしに建物を作ることができるのであろうか。アレグザンダーは『ノート』のなかで「現存する形の批判なら、我々のすべてにできる」と力強く言い切る [4-1]。確かに我々は、実環境の建物を模倣することで実際に建物を作ることができる。しかし設計の経験がなければそこで問題が起こるのは当然である。だから現在設計を自覚している設計者は、実環境のコンテキストに適合している建物を「事前的に」予測し設計する。しかし「事後的に」ならば、我々のすべては現存する形を体験することで、形とコンテキストの不適合を認知し、それを漸進的に修正していくことができる。

したがってフローグラフで表現すると、実空間に出力された建築物としての〈かたち〉 RI の情報を、利用者が事後的に体験することで、物理的評価である〈かち〉 $U4$ や、エネルギー的評価である〈かち〉 $U5$ や、機能的評価である〈かち〉 $U6$ と〈かたち〉との不適合を認知し、その不適合を修正したいというインセンティブが動力となり、 RI を漸進的に修正していくこととなる。

ここに $RI \rightarrow U4, U5, U6 \rightarrow RI$ というフィードバックを見出すことができる。これはすなわち実環境における試行錯誤 [4-2] である。

フィードバックには正と負の2種類があるが、ここではどちらのフィードバックが働いているのだろうか。アレグザンダーは「フィードバックはコントロールされ、また幾分鈍らせねばならない」と言う。無自覚な状況では、形の不適合を修正したいというインセンティブがシステムを回す動力となり、試行錯誤というフィードバックが形の変化を促し、そして実環境からの制約という「システムに粘りを与えるもの」が形の他の側面に変化が波及するのを防ぎ、実環境のコンテキストに適応した形が、ゆっくりと時間をかけて出力される。つまりここでは、生命が持つ重要な性質のひとつである「ホメオスタシス (homeostasis)」のような「負のフィードバック」が働いているのである。アレグザンダーは、このような変化と抵抗との相互作用を「動的平衡 (homeostatic)」と呼んだ [4-3]。

こう見ると、あたかも実環境から環世界へとコンテキストがトップダウン的に入力されているように見える。設計に無自覚な利用者が実環境のコンテキストを読み取る

ことができるのは、「システムに粘りを与えるもの」によって、有限なコンテキストへと制限されているからである。中谷礼仁は、この形に対する意味解釈の有限性を「セヴェラルネス (severalness)」と表現し、前近代的な社会だけでなく現代社会における事物の転用においても、その背景には「セヴェラルネス」という法則性が潜んでいることを明らかにした [4-4]。形に対する意味解釈がひとつしかないならば、修正する余地がない。逆に、無限に解釈可能ならば、とりつく島がない。それらの間、つまり「セヴェラル (いくつか)」の意味解釈に制限されていることによって、設計するという自覚なしに個人が設計プロセスに参画できるのである。

このように無自覚な状況においては、本来自律的な心的システムである環世界（個人）が、「セヴェラルネス」という法則性によって、あたかも都市システムに従属する下層システムのひとつとなる [4-5]。バーナード・ルドフスキーは、このような特定の主体の設計によらない、自然発生的な建築群を「建築家なしの建築 (Architecture without Architects)」と名づけたのであった [4-6]。

したがって、本来ならば $U4$ から RI へと取得の矢印が描かれるはずであるが、ここでは、 RI から $U4$ へと入力矢印を描くこととする。

以上を踏まえ、無自覚な状況、手工業的進化のフローグラフを Fig. 4-1 に示す。設計プロセスの遺伝子は 0000000 0000000 0000000 0111000 1 となるので、これを十進数で表すと 113 となる。後述する他の設計プロセスの十進数を比較すると、もっとも数字が小さいので、これらの設計プロセスは、第一世代の設計プロセスと呼ぶこととする。

4.2 第二世代の設計プロセス：図面によるデザイン、自覚的な状況、モダニズムの設計プロセス

本節では、個別の設計プロセスを「進化論的枠組み」に一对一で位置づけるというよりは、〈か〉〈かた〉〈かたち〉の各段階を社会環境に外部化する設計プロセスとして一括りに論じる。したがって〈か〉〈かた〉〈かたち〉のいずれかの段階を設計者の環境のなかで自覚しているという意味では、アレグザンダーのいう「自覚的な状況」を位置づけることとなる。もちろん各段階をそれぞれ組み合わせる異なる設計プロセスとして位置づけることはできる。しかしそうはしないのは、設計者の環境において〈かち〉の段階を自覚していないという点で、共通の設計プロセスであると考えからである。

第一世代で見たように、実際に建物を作って「事後的に」修正するような試行錯誤はたいへんに有効な方法である。しかし現実にはそのような試行錯誤は、「あまりにも高価でありあまりにも遅い」[4-6]。そこで図面やスケッチや模型などの〈かたち〉を社会環境に外部化することとなる ($A1 \rightarrow S1$)。J・C・ジョーンズのいう「図面によるデザイン」である。ここで重要なのは、「生産行為から試行錯誤を分離させた」[4-7] ことである。つまり、現実態としての建物が実際に建てられる前に、可能態としての〈かたち〉 $S1$ が社会環境へと外部化されることによって、「事前的に」試行錯誤できるようになるわけである。生産行為と試行錯誤が分離すると、スケールを自由に行き来することで個人で試行錯誤できる空間的な範囲が増大し、そしてまた時間的にも試行錯誤のスピードは増大する。その結果〈かたち〉の可能性も増大していく。しかしそれは同時に、無自覚な状況において、システムに粘りを与えていた実環境の物理的な制約が、溶けてなくなっていくということである [4-8]。コンテキストに適応した形を出力する構造であった「動的平衡」が崩れてのである。しかし〈かたち〉が社会環境に外部化されるということは、利用者の〈かち〉 $U4, U5, U6$ から〈かたち〉 $S1$ の情報を取得することで、試行錯誤ができるということである。ここでは、この「他者性」が、システムに粘りを与えるものとして「動的平衡」を担うこととなる。

〈かたち〉が社会環境にある程度ストックされてくると、「認識のプロセス」によって「〈かたち〉のなかにある普遍的技術あるいは法則性を理解する」〈かた〉の段階が設計者の環境の中で自覚される ($A2$)。そこで次は、〈かた〉を社会環境に外部化する ($S2$)。それらの〈かた〉 $S2$ を取得し組み合わせることによって、より多様な〈かたち〉 $A1$ の可能性を試行錯誤することができる。そしてさらに、他の設計者とのコミュニケーションを通して、〈かた〉 $S2$ をより洗練させることができる。〈かた〉 $S2$ が洗練されれば、〈かたち〉 $A1, S1$ のありうべき可能性全体も洗練されていく。

このような $S2 \rightarrow A2 \rightarrow A1 \rightarrow S1$ の機構は、正/負どちらのフィードバックとしても

用いることができる。例えば、難波和彦は「箱の家シリーズ」という〈かた〉S2を用いて「箱の家」という〈かたち〉SIを展開している。敷地やクライアントが変わっても、同じ〈かた〉S2から質の高い〈かたち〉SIを安定して提供しているという意味で、S2→A2→AI→SIの機構は負のフィードバックといえる。またSANAAに代表されるように、夥しい数の模型を人海戦術的に作るという〈かた〉S2は、人工的に淘汰と突然変異を作り出すことによって多様な〈かたち〉SIを創発させるという意味で、正のフィードバックといえる。このように、環境は多様なことから、〈かたち〉はどんな環境でも安定しているべきと見るか、環境が多様だからこそ、その反映である〈かたち〉も多様であるべきと見るか、という設計者の〈か〉A3の違いによって、フィードバックの正／負が使い分けられているのである。

設計者の環世界で自覚されたこのような〈か〉A3が、同時代的にある傾向を帯びてくると、キャッチーなタームやアフォーリズムが割り当てられ、主義（イズム）に転化する。〈か〉の社会環境への外部化である（S3）。なかでも、最も有名で、現在でも根強く影響を及ぼしている〈か＝イズム〉は、ルイス・サリバンによる「形態は機能にしたがう／Form Follows Function(FFF)」によって端的に示された機能主義であろう。

ここまでは、菊竹の「認識のプロセス」のように、設計者の環境だけでなく社会環境においても、〈かたち〉SI→〈かた〉S2→〈か〉S3が段階的に外部化され、主義（イズム）が形成されていく過程を見てきた。では今度は、「実践のプロセス」のように、社会環境に外部化された主義（イズム）を規範として、〈か〉S3→〈かた〉S2→〈かたち〉SIという逆の流れをたどって設計するべきなのだろうか。少なくとも、菊竹の「か・かた・かたちの三段階論」は、そのような示唆を与えてくれる。実際例えば、ミース・ファン・デル・ローエの「ユニヴァーサル・スペース」にしても、ル・コルビュジェの「住宅は住むための機械である」にしても、「形態は機能にしたがわねばならない」という規範に基づいている点で、両者は一致している [4-9]。

機能主義のような規範は、なにもモダニズム特有のものではない。日笠直彦は「ポストモダニズムと建築」という論考のなかで、一般にモダニズムからポストモダニズムへ移行したとされる1968年を特異点として、「『68年』に崩壊したもとのものは、最も広義に言えばこの古典性ではないだろうか」という興味深い指摘をしている [4-10]。日笠は、「ローマ的伝統の崩壊において建築を開始するロマネスクと、近代主義の崩壊において建築を開始するわれわれ」を類比させ、モダニズムにおける機能主義や抽象性・幾何学性という規範が、ギリシア・ローマの古典主義にまで遡ることを明らかにしていく。そしてモダニズム以前を古典主義と捉え直し、ポストモダニズムから現在をポスト古典主義と位置づけたのである。なるほどそのような視点に立つてみると、古典主義を復興させたルネサンスにおいて頂点を極めたマニエラ（手法）がマニエリスムを産み出し、それがマンネリズム化して前時代の主義（イズム）を転倒させていくように、あるいはまた、ギリシア・ローマ→ルネサンス→新古典主義→モダニズム

と、古典主義という規範が時代を超えて螺旋状に繰り返していくように、主義（イズム）の栄枯盛衰の歴史が見えてくる。「建築設計プロセスの進化論的枠組み」に沿ってもう少し形式的に言い換えると、〈かたち〉S1 → 〈かた〉S2 → 〈か=イズム〉S3 → 〈かた〉S2 → 〈かたち〉S1 → 飽きる → 〈かたち〉S1 → 〈かた〉S2 → 〈か=イズム〉S3……というように、〈か=イズム〉S3 が次々と社会環境に産み出されるプロセスと見ることができる。

このように、本節で対象とする設計プロセスを個別の設計プロセスではなく一括りに論じてきたのは、〈か=イズム〉S3 を産み出し、その〈か=イズム〉S3 が何であれ、特定の〈か=イズム〉S3 に基づいて設計しなければならないという規範がある、という意味で、共通の設計プロセスだと考えるからである。

したがって、本来ならばA3 からS3 へと取得の矢印が描かれるべきであるが、S3 を動力として、S3 からA3 への入力矢印で連結することとする。

以上を踏まえ、図面によるデザイン、自覚的な状況、モダニズムの設計プロセスのフローグラフをFig. 4-2 に示す。設計プロセスの遺伝子は0000000 0000111 0111111 0111000 1となるので、これを十進数で表すと245,617となる。他の設計プロセスの十進数と比較すると、2番目に数字が大きいので、これらの設計プロセスは、第二世代の設計プロセスと呼ぶこととする。

メタボリズムという〈か=イズム〉S3 は、〈かたち〉S1 を生物の新陳代謝というアナロジーで捉え、半ば永久的に建物が存続する〈かた〉S2 を提示した。しかし最近メタボリズム建築が、相次いで取り壊しの危機に瀕している。一般に、実際に建っている建物に転用を施すのは最初の設計者ではなく他の設計者である。ポスト・モダン以降の現在からみれば、メタボリズムという〈か=イズム〉S3 に基づいて設計しなければならないという規範はもはや存在しない。このように、建築家の社会環境における啓蒙的な役割が低下した現在においては、〈か=イズム〉S3 に基づいて設計することの限界が明らかになってきたと考えられる [4-11]。

磯崎新の「プロセス・プランニング論」[4-12] は、このような限界を先取りした建築の終末論であった。すなわち本質的に設計（コントロール）できない他者性を「事前的に」織り込んで設計しなければいけないという矛盾を指摘したのだと考えられる。動的に回り続けるプロセスを「切断」ということは、建物の未来を試行錯誤すると同時に、その終末をも内包せざるをえないということなのである。

4.3 第三世代の設計プロセス：ポスト・モダニズムの設計プロセス

本節では、設計者が第4層の記号的評価である〈かち〉A7を自覚することによって、社会環境に外部化された〈か=イズム〉S3が相対化していくようなポスト・モダニズムの設計プロセスを位置づける。

先述したように、設計者の環境における第4層の〈かち〉A7は〈かたち〉A1を意味を持った記号として評価する**記号的評価**の自覚である。アブダクションを提唱したチャールズ・パースは、記号論 (semiotics) の創始者として知られている。パースは記号の意味解釈の過程 (semiosis) を、「記号表現 (representamen)」「対象 (object)」「解釈内容 (interpretant)」という三項関係で表わす [4-13]。たとえば、「通行人がある家の窓から煙が上がっているのを見て火事だと思った」という状況は、煙という記号表現によって、通行人の環世界のなかに火事のイメージという意味 (解釈内容) が立ち上がっていると考えられる。実際に家が燃えている (対象) ならば、煙 (記号表現) と火事のイメージ (解釈内容) は一致する。しかし、もしかすると、キッチンで秋刀魚を焼いた煙が窓から出ているだけかもしれない。つまり、記号表現とその解釈内容は必ずしも一致するとは限らないのである。この過程は先に見たアブダクションに他ならない。このようにパースは、人間のすべての認識と思考を記号の意味解釈の過程と捉え、その本質は推論にあると論じたのであった。

ここでの枠組みに即して言い換えると、これから設計する建物 *RI* が対象、そのモデルである〈かたち〉A1が記号表現、そして〈かち〉A7が解釈内容と考えられる。ここで重要なのは、〈かたち=記号表現=形〉A1と〈かち=解釈内容=コンテキスト〉A7との間には必ずズレがあるということを知覚するということなのである。

記号学 (semiology) の創始者であるフェルディナン・ド・ソシュールは、言語という記号が「シニフィアン (記号表現)」と「シニフィエ (記号内容)」とのセットから成り、両者を結びつける対応規則 (コード) が恣意的であることを明らかにした。近代建築を規定していた機能主義では、機能から形態が一意に導かれたが、ポストモダン建築においては、記号表現は〈かたち〉A1に、記号内容は〈かち〉A7へと読み替えられ、両者を結びつける普遍的・絶対的なコードがないことが自覚されたのである [4-14]。〈かたち〉と〈かち〉のズレを知覚するということは、両者を結びつけるコードを設計者が意図的・恣意的に操作できるということである。こうして1970年代以降、一方では、言語の統辞論のように、さまざまな〈かた〉S2を駆使して過剰な装飾や奇抜な形態の〈かたち〉S1を生成するような形式主義・形態主義といった〈か=イズム〉S3が表われ、また他方では、言語の意味論のように、さまざまな歴史的引用を〈かたち〉S1に織り込む歴史主義・折衷主義などの〈か=イズム〉S3が見直された。いず

れも近代建築において抑圧された〈か=イズム〉 $S3$ が堰を切ったように溢れ出てきたのである [4-15]。

このように第三世代の設計プロセスは、建物を記号としてとらえ、設計者がそのコードを意図的・恣意的に操作することで、建物全体の意味を創発あるいは異化させようとする。「建築は意味を生産する機械である」(磯崎新、1976) [4-16] というわけである。つまり、「〈かち〉に適合する〈かたち〉を出力すること」というデザインの定義が、「〈かたち=記号表現〉と〈かち=記号内容〉を結びつけるコードを操作すること」と再定義されたのである。この定義に沿って先の世代を振り返ってみると、第一世代は、実環境の物理的な制約によってコードが「セヴェラル (いくつか)」に規定されていた世代であり、第二世代は、社会環境に外部化された〈か=イズム〉 $S1$ によって、コードが単一に規定できると信じられていた世代であると、逆に定義できる。

難波は「機能主義・再考」という論考のなかで、機能 (function) を関数 (function) と読み替えて、社会環境に外部化された〈か=イズム〉 $S3$ を「関係主義」として数学的に定式化しようと試みる [4-17]。ここで、第1層の物理的評価である〈かち〉 $A4$ 、第2層のエネルギー的評価である〈かち〉 $A5$ 、第3層の機能的評価である〈かち〉 $A6$ 、第4層の記号的評価である〈かち〉 $A7$ を変数として、社会環境に外部化された〈か=イズム〉 $S3$ との関係性を $f_1()$ で表わすと、関係主義は下記のように定式化される。

$$S3 = f_1(A4, A5, A6, A7) = 0 \quad \text{—— (4-3-1 式) 関係主義}$$

難波は、(4-3-1 式)において並列的に扱われている〈かち〉の変数のなかから、ある特定の〈かち〉を独立変数として優先的に取り出し、残りを従属変数とすることによって、設計者の〈か=イズム〉 $A3$ が相対化していく様子を明瞭に形式化した [4-18]。

$$A3 \Leftrightarrow A6 = f_2(A4, A5, A7) \quad \text{—— (4-3-2 式) 機能主義・プログラム主義}$$

$$A3 \Leftrightarrow A4 = f_3(A5, A6, A7) \quad \text{—— (4-3-3 式) 技術主義・構造主義}$$

$$A3 \Leftrightarrow A4+A6 = f_2(A5, A7) \quad \text{—— (4-3-4 式) モダニズム}$$

$$A3 \Leftrightarrow A7 = f_3(A4, A5, A6) \quad \text{—— (4-3-5 式) ポストモダニズム}$$

$$A3 \Leftrightarrow A5 = f_2(A4, A6, A7) \quad \text{—— (4-3-6 式) サステイナブル・デザイン}$$

〈かたち〉 $S1$ も〈かち〉 $S2$ も機械的にコピーされる「複製技術時代」(ヴァルター・ベンヤミン)の近代においては、かろうじて〈か=イズム〉 $S3$ を社会に向けて外部化することだけが建築家のオリジナルな芸術として機能していた。しかし第三世代に至っては、その〈か=イズム〉 $S3$ までもが社会環境にデータベース化され、一握りのスター建築家以外の設計者でもそれらを組み合わせることによって、シミュラクルとしての〈か=イズム〉 $S3$ を量産できるという極めてポストモダン的な状況となった

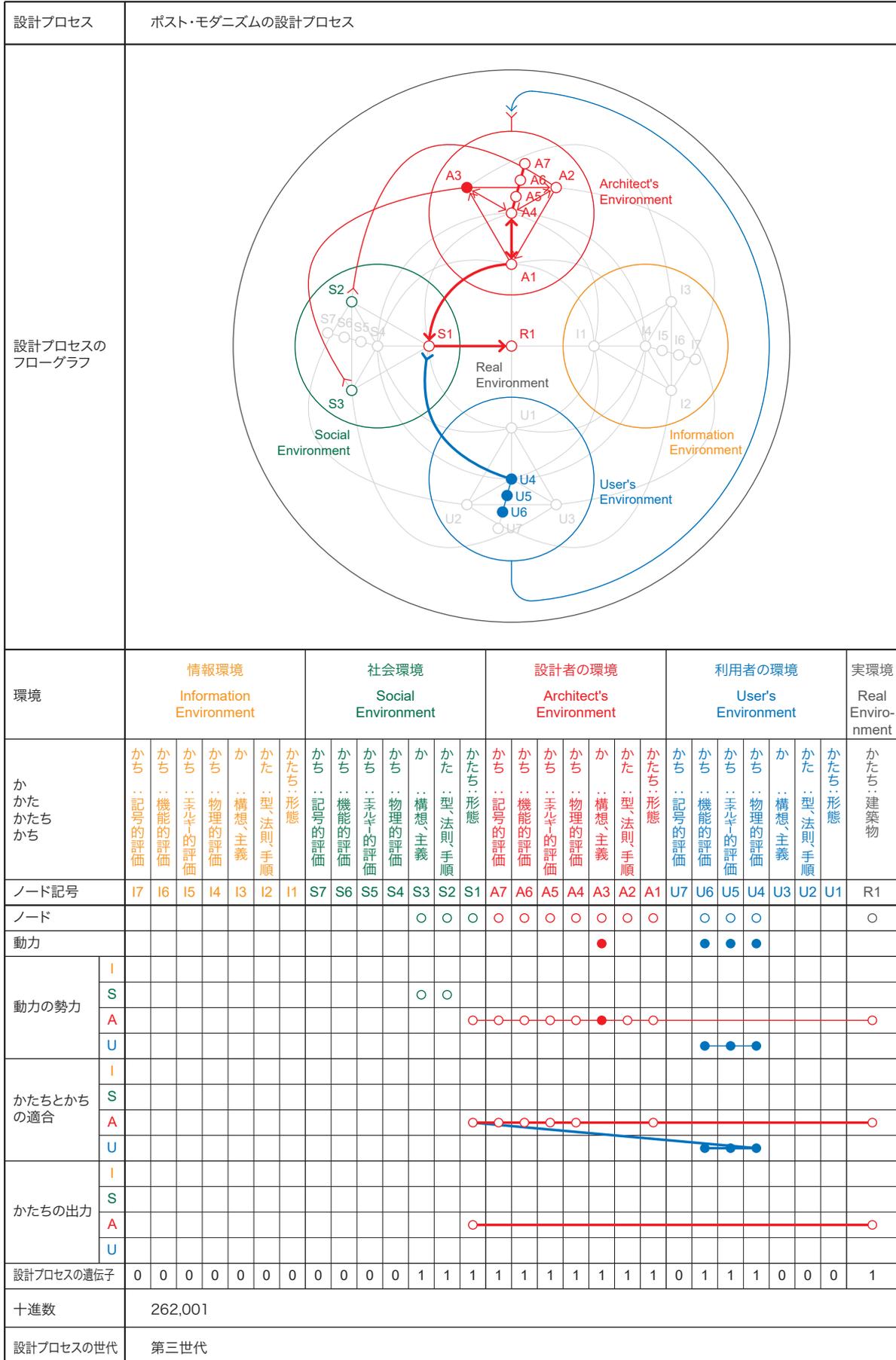


Fig. 4-3 第三世代のフローグラフ

のである。

現代建築はもはや表現すべき主題をもっていない。(…中略…)誰もが主題を、みずから主張できるようになったのだ。ということは、現代において、決定的な主題が失われてしまったことを意味しよう。

磯崎新「なぜ手法なのか」[4-19]

以上を踏まえ、ポスト・モダニズムの設計プロセスのフローグラフをFig. 4-3に示す。設計プロセスの遺伝子は0000000 0000111 1111111 0111000 1となるので、これを十進数で表すと262,001となる。他の設計プロセスの十進数と比較すると、3番目に数字が大きいため、これらの設計プロセスは、第三世代の設計プロセスと呼ぶこととする。

4.4 第四世代の設計プロセス：大量生産型の設計プロセス

本節では、社会環境へ〈かち〉を外部化する設計プロセスとして、大量生産型の設計プロセスを位置づける。

より多くの人に建物を供給しなければならないと考えたときに、少品種になるのはある意味当然である。第二・三世代のように設計者が個別のユーザーに対し毎回試行錯誤していたのでは、コストがかかりすぎるからである。そこで、いくつかの〈かた〉を規格化・標準化し、それらを組み合わせた〈かたち〉をカタログ化し、そして利用者が設計プロセスに関わる前に社会環境へと外部化することとなる (*S1, S2*)。

ここで問題なのは、第二世代でシステムの動的平衡を担う重要な役割を果たしていた利用者からのフィードバックがなくなってしまうことと、設計者の「より多くの人に建物を供給したい」という〈か〉 *A3* によって、その他の機構にも負のフィードバックが働くことである。こうして環境から新たなコンテキストを読み込む機構がなくなり、設計者の環境と社会環境における〈か〉〈かた〉〈かたち〉の可能性が固定化していく。

こうみると全く良いところがないように思えるが、実際には、都市を形成している建物の大部分が少品種大量生産される工業化建築で占められている。その理由のひとつは、〈かち〉をすべてお金に換算し、社会環境に外部化しているからだと考えられる。設計を自覚していない利用者は社会環境に外部化された〈かたち〉と〈かち〉を正しく適合させることはできない。しかし、お金という単一の基準で〈かち〉が社会環境へと外部化されることによって、容易に〈かたち〉を選択できるようになるのである。

この社会環境に外部化された〈かち〉を、第三世代の社会環境の〈か〉のように定式化することを試みる。社会環境に外部化された〈かち〉 *S4, S5, S6, S7* を、お金に換算してすべて足しあわせる関数を *SUM()* と表現すれば、〈かち〉は下記のように表すことができる。

$$\langle \text{かち} = \text{お金} \rangle = \text{SUM}(S4, S5, S6, S7) \quad \text{—— (4-4-1 式)}$$

しかし一般的に、第4層の記号的評価である〈かち〉 *S7* は定量化するのが難しい。したがって、記号的評価である〈かち〉 *S7* は抑圧され、実際に社会環境に外部化される〈かち〉は下記のようなになる。

$$\langle \text{かち} = \text{お金} \rangle = \text{SUM}(S4, S5, S6) \quad \text{—— (4-4-2 式)}$$

それはつまり、高さが高い、面積が大きい、部屋数が多い、素材がよい、工期が早いなど、お金の換算できるような定量的な〈かち〉だけが即物的に〈かたち〉へと変

換されるということである。パースは記号の種類を、「類似記号／アイコン (icon)」「指標記号／インデックス (index)」「象徴記号／シンボル (symbol)」という3つに分類したが、少品種大量生産される建物は「インデックス」としての記号を備えている [4-20]。お金という〈かち〉と〈かたち〉のあいだには直接的な類似性はない。しかし、温度計が気温のインデックスとなるように、高さが高いという〈かたち〉*SI*がお金という〈かち〉*SUM(S4, S5, S6)*と即物的な対応関係を結ぶのである。だとすると、ジャン・ボードリヤールが指摘したように、利用者自らが〈かたち〉*SI*を自由に選択しているように見えて、実は、都市システムがユーザーの差異化の欲望を喚起し、お金という記号を纏った〈かたち〉の消費を促しているとみるべきである [4-21]。つまり第一世代でも見たように、本来自律的な個人の環世界が、あたかも都市システムに従属する下層システムとなっているのである。しかしここには動的平衡を担っていた「フィードバックによる変化」も「システムの粘り気」もない。あるのは、R. レッシングのいう市場 [4-22] としての動力〈か〉*S3*と、無意識のうちにシステムにコントロールされている利用者の〈かたち〉*UI*の消費だけである。こうして、半自動的にシステムは回り続け、少品種大量生産された建物が大量消費され続けていくこととなる。

したがって、本来ならば *UI* から *SI* へ、そして *U4* から *S4* へと取得の矢印が描かれるはずであるが、ここでは、*SI* から *UI* へ、そして *S4* から *U4* へと入力矢印を描くこととする。

以上を踏まえ、大量生産型の設計プロセスのフローグラフを Fig. 4-4 に示す。設計プロセスの遺伝子は 0000000 0111111 0111111 0111101 1 となるので、これを十進数で表すと 2,080,635 となる。他の設計プロセスの十進数と比較すると、4番目に数字が大きいのので、これらの設計プロセスは、第四世代の設計プロセスと呼ぶこととする。

4.5 第五世代の設計プロセス：パターン・ランゲージ、超線型設計プロセス

本節では、第四世代とは異なる方法で社会環境へ〈かち〉を外部化する設計プロセスとして、クリストファー・アレグザンダーのパターン・ランゲージ [4-23] と、藤村龍至の超線型設計プロセス [4-24] を位置づける。

アレグザンダーは、後述する「記号化の状況」において、頻繁に現われる〈かち〉と〈かたち〉のセットを「パターン」としてまとめ、それらの関係性を自然言語でデータベース化した。すなわち *S1* と *S4, S5, S6, S7* を部分的に連結した上で社会環境に外部化したのである。

この「パターン・ランゲージ」による設計プロセスは、自然言語が持つコードの恣意性を利用して、後述する「記号化の状況」においてツリー構造だった〈かたち〉を、再度セミラティス構造の〈かたち〉へと合成することを狙ったものである [4-25]。しかし結局は、部分的な〈かち〉 *S4, S5, S6, S7* に適合した部分的な〈かたち〉 *S1* を組み合わせ、利用者によって全体の〈かたち〉 *UI* を合成しようとすることに変わらない。それは実際に「パターン・ランゲージ」を適用してつくられた《盈進学園東野高等学校》(1984) の加算的なデザインを見れば明らかである。「部分を総和しても全体にならない」という要素還元的なデザイン手法の限界は、「形態は機能に従う」という近代建築のテーゼの限界でもある。

藤村龍至は、第四世代のような半ば自動的に大量生産される工業化建築を「形骸化した工学主義」として、それを批判的に乗り越えるために「批判的工学主義」という〈か=イズム〉 *S3* を提唱している。その具体的な〈かた〉 *S2* が「超線形設計プロセス」である。藤村によれば「超線形設計プロセスとは、『ジャンプしない』『枝分かたれしない』『後戻りしない』という3つの大きな原則に基づき、最初にゴールイメージを抱かずに、漸進的な改良を重ねながら徹底して線形的に設計を進めるもの」と定義される [4-26]。

この定義を聞いて、コンピュータ・プログラミングに詳しい人ならば、すぐさまケント・ベックが提唱したソフトウェアの開発手法である「エクストリーム・プログラミング (XP)」を思い浮かべるに違いない [4-27]。従来のソフトウェア開発は、最初にしかりとした仕様を固め、その仕様に基づいて開発を進める「ウォーターフォール」と呼ばれる開発手法が主流だった。対して、頻繁に変更される仕様にも柔軟に対応するために、従うべき規範を「アジャイルマニフェスト」としてまとめ、漸進的に、そしてアジャイル (俊敏) に開発を進める XP という手法が提唱された。

これは建築分野が抱える問題とまったく同型である。藤村は、第四世代の設計プロセスのもつスピード感を維持すると同時に、捨象されていた利用者とのフィードバック

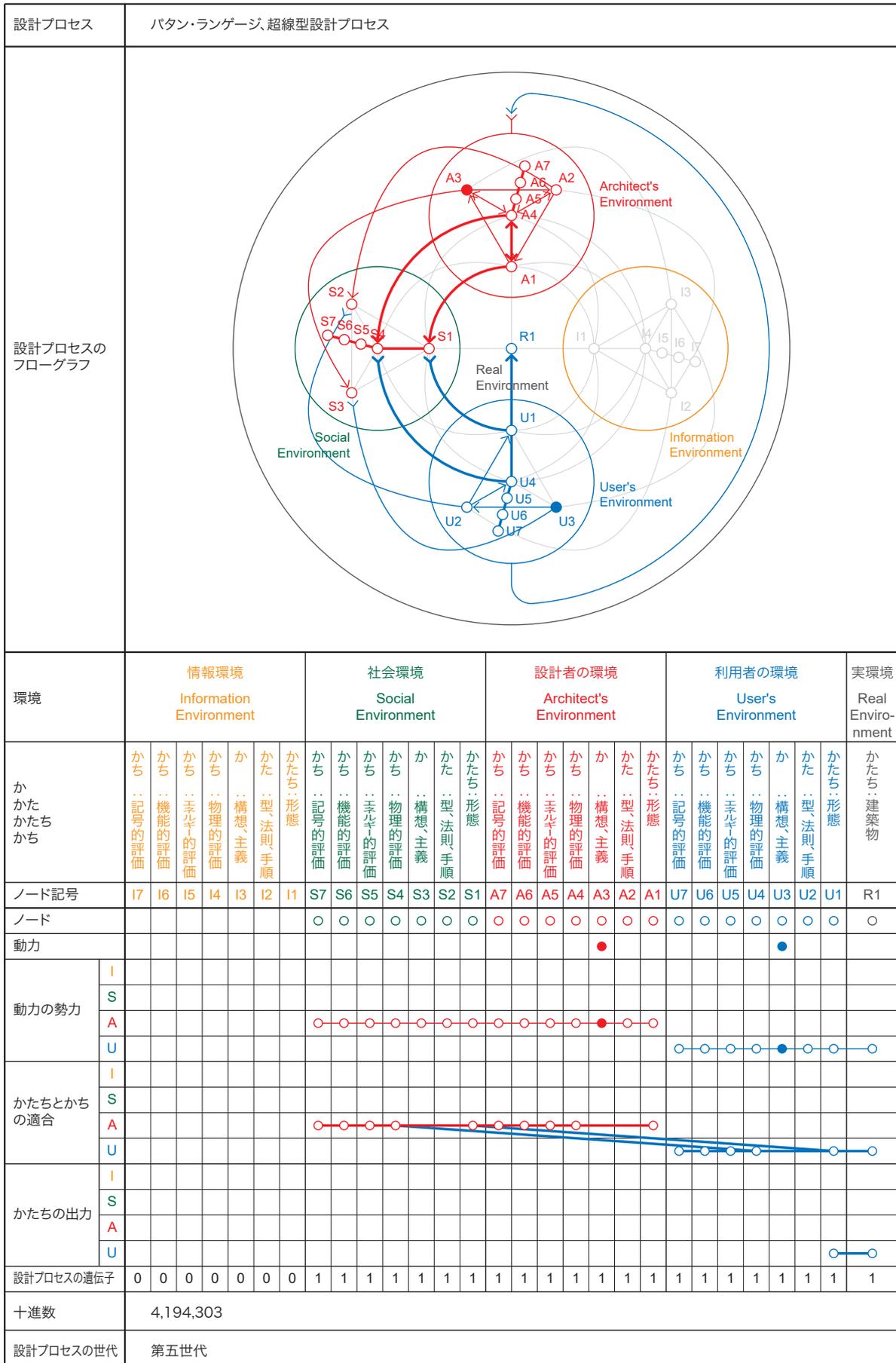


Fig. 4-5 第五世代のフローグラフ

グループを回復し、環境のコンテキストをより濃密に読み込んだ形をつくりだそうと試みているのである。

「超線形設計プロセス」では、環境から取得した〈かち〉*A4, A5, A6, A7*をひとつずつ〈かたち〉*A1*へと合成し、模型として社会環境に〈かたち〉*S1*の履歴を残していく。これは、部分的な〈かち〉*S4, S5, S6, S7*に適合した部分的な〈かたち〉*S1*がセットになったアレグザンダーの「パタン」を踏襲するものである [4-28]。

「超線形設計プロセス」「パタン・ランゲージ」など〈かた〉は、社会環境に外部化されている (*S2*)。その〈かた〉*S2*を利用者が扱うことを考えた場合、それを利用する前に、社会環境の規範としての〈か〉*S3*を、利用者が共有し内面化するプロセスが必要となる [4-29]。すなわち *U3*を動力として *S3*から取得の矢印を連結しなければならない。したがって第二世代のように、どうしても設計者の〈か=イズム〉*A3*を社会環境へとマニフェストするアヴァンギャルド的傾向を帯びてきてしまうのである。もちろん、一旦その規範を共有し内面化してしまえば、非常に有効な設計プロセスとなる。しかし第三世代で見たように、社会環境に外部化された〈か〉や〈かた〉は、相対的に選択できるオプションのひとつにすぎないのである [4-30]。

以上を踏まえ、パタン・ランゲージと超線型設計プロセスのフローグラフを Fig. 4-5 に示す。設計プロセスの遺伝子は 0000000 1111111 1111111 1111111 1 となるので、これを十進数で表すと 4,194,303 となる。他の設計プロセスの十進数と比較すると、5番目に数字が大きいため、これらの設計プロセスは、第五世代の設計プロセスと呼ぶこととする。

4.6 第六世代の設計プロセス：BIM

本節では、〈かたち〉を情報環境へと外部化する設計プロセスの代表として「BIM」を位置づける。

90年代以降に情報環境が顕在化してくると、手書きの図面がCAD化され、スケッチがCGになり、模型も3Dプリンターを用いてデジタルデータから直接つくれるようになった(AI→II)。その結果〈かたち〉IIの可能性は増える。そのようなソフトウェア群は、現在さらに高機能化・効率化・統合化され、建物のライフサイクル全体をデジタルに管理できるようになった。BIMの登場である(AI→II) [4-31]。

BIMの特性のひとつは、〈かたち〉の様々なメディアをシームレスに繋げることにあり、その結果さらに〈かたち〉IIの可能性は増えることとなる。BIMのもうひとつの重要な特性は、インターネットを介した他の設計者とのコラボレーション機能にあるといえる。3DCGで表現された〈かたち〉に加えて、今後、強化現実(Augmented Reality)の技術もBIMに統合されることが期待できる。そうなれば、設計者だけでなく、第一世代でみたような設計を自覚していない利用者でも、あたかも「事後的に」試行錯誤することができるようになるだろう。またBIMの本質は、ひとつの建物の設計における効率性というよりは、多様な〈かたち〉IIのデータベース構築にある。BIMがより普及し、〈かたち〉のデータベースが充実・洗練してくると、萌え要素をサンプリングすることでキャラ化した建物(シミュラクル)がより多くあられるだろう [4-32]。その是非はともかく、第四世代で抑圧されていた第4層の記号的評価である〈かち〉A7が再評価され、画一化した建物が多様化していく可能性はありそうである。

以上を踏まえ、BIMのフローグラフをFig. 4-6に示す。設計プロセスの遺伝子は0000001 0000111 1111111 0111000 1となるので、これを十進数で表すと4,456,305となる。他の設計プロセスの十進数と比較すると、6番目に数字が大きいので、これらの設計プロセスは、第六世代の設計プロセスと呼ぶこととする。

4.7 第七世代の設計プロセス：パラメトリック・デザイン、 バイオ・ミメティック・デザイン

本節では、〈かた〉と〈かたち〉を情報環境へ外部化している設計プロセスとして、「パラメトリック・デザイン」と「バイオミメティック・デザイン」を位置づける。

第五世代の〈かたち〉IIと同様に、〈かた=アルゴリズム〉I2も社会環境へと外部化すると、ここで初めてIIとI2によるデジタルメディア同士の情報の流れが起こる。この流れはこれまでのような自然言語ではなく、機械言語を介して行なわれる。機械言語による情報の流れは、自然言語のそれに比べて圧倒的に早いのが特徴のひとつである。さらにここでは、設計者の〈か=イズム〉AIのように気まぐれな動力は必要ない。電力が供給される限り〈かた=アルゴリズム〉I2が動力となり、飽きることなく、そして正確に、システムは回り続ける。こうして、〈かた=アルゴリズム〉I2から膨大な数の〈かたち〉IIが情報環境に生成されることとなる。

個別の〈かたち〉を規定していた定数を「変数（パラメーター）」のまま保持することで、多様な〈かたち〉を生成するという設計手法は「パラメトリック・デザイン」と呼ばれている。たとえば、従来はモデリングするのが比較的難しかった複雑な曲面でも、〈かたち〉を制御する可視化された変数を直感的にスライドさせるだけで、平滑な平面から複雑な曲面まで多くのヴァリエーションを生成することができる。しかし、これまでできなかったことが「できるから」という理由だけで複雑な〈かたち〉をつくるのは、パラメトリックなツールの制約にコントロールされている「ツール・ユーザー」である[4-33]。ローレンス・レッシングが『CODE』のなかで指摘したように、我々はつねに、情報環境の「アーキテクチャ」によって無意識のうちにコントロールされるということを、自覚しなくてはならない[4-34]。一方、第三世代でも見たように、ある限界や制約を自覚するということは、設計者がそれを逆手にとることができるということである（＝「ツール・メーカー」）。濱野智史は『アーキテクチャの生態系』のなかで、通常否定的に捉えられがちな「アーキテクチャ＝環境権利型権力」による制約を、情報環境に社会秩序を生み出す手法のひとつとして肯定的に捉え直すことを提案した。同じようにここでも、「アーキテクチャ」が持つ「いちいち価値観やルールを内面化することなしに人を無意識のうちに操作できる」という特徴を用いて、設計に無自覚な他者でも容易に設計プロセスに参画できるようなシステムを築くことができるかもしれない[4-35]。それはつまり、第一・四世代でもみたように、本来自律的な個人の環世界が、あたかも都市システムに従属する下層システムとなるということである。またそれは、第五世代の「パターン・ランゲージ」や「超線形設計プロセス」のような「規範としての〈かた=手法〉」の限界を、「アーキテクチャとしての〈かた=アルゴリズム〉」によって乗り越える可能性を示唆するものといえる。

このように、設計者の〈か=イズム〉という動力なしにシステムが回り続け、多様な〈かたち〉が生成され続けるところまではよい。しかしここで問題なのは、機械言語による $I1$ と $I2$ の機構は、〈かたち〉 $I1$ が環境の〈かち〉に適合しているかどうかを何も保証してくれないということである。コンピュータは〈かち〉を（まだ）理解できない。「パラメトリック・デザイン」によって、いくら膨大な〈かたち〉の可能性を生成したとしても、玉石混淆の〈かたち〉の中から、その〈かち〉を評価するのは、やはり設計者なのである。したがって設計者による〈かち〉 $A4, A5, A6, A7$ から情報環境の〈かたち〉 $I1$ へと入力の手印が連結され、 $I1$ の動力の勢力は設計者の環境の属性となり、結果的に、設計者の〈かち〉に適合した〈かたち〉しか出力されなくなる。

これは「バイオミメティック（生体模倣）・デザイン」と呼ばれる設計手法にも当てはまる。自然や生命の〈かたち〉の背後に隠された「黄金比」「フィボナッチ数」「フラクタル」「 $1/f$ のゆらぎ」といった〈かた=法則性〉をアルゴリズム化し、建物の〈かた=生成原理〉として適用する際にも注意が必要である。自然の形の法則性は実環境のコンテキストに適応した結果なのであって、その因果を逆にして、無批判にその法則性を模倣しても、建物の〈かち〉を考慮しない限りメタファーにとどまる。逆にいえば、設計者の正しい〈かち=アブダクション〉に基づいて〈かた=生成原理〉が選ばれたときにのみ、「自」ずから「然」るべき〈かたち〉が創発するという意味で、設計主体の恣意性を超えた「自然」な〈かたち〉が生成される可能性がある。

今後、「デジタル・ファブリケーション」による生産技術がより普及するのは間違いと思われる。そうすると、従来は実現が困難だった「〈かち〉なき〈かたち〉」でも容易に出力されてしまう。その是非は現時点ではわからないが、「デジタル・ファブリケーション」のより重要な可能性は、第一世代のような、設計に無自覚な他者による「事後的な」試行錯誤の復活にある。「デジタル・ファブリケーション」が「パーソナル・ファブリケーション」となったとき、ニール・ガーシェンフェルドのいう「ものづくり革命」が起こるかもしれない [4-36]。

そのような生産技術に着目すれば、第一世代は、手工業的な設計プロセスだった。第二・三世代は、設計者の複雑な〈かた〉から複雑な〈かたち〉をつくる一品生産型の設計プロセスだった。第四世代は、単純な〈かた〉から単純な〈かたち〉をつくる少品種大量生産型の設計プロセスだった。第五世代は、基本的には一品生産型の設計プロセスを踏襲しながらも、アジャイルで漸進的な設計プロセスを導入した中品種少量生産型の設計プロセスだった。だとすれば、第六世代は、自然の法則性とその形の多様性からは想像できないくらい単純なように、単純な〈かた=アルゴリズム〉から複雑で多様な〈かたち〉を生成するような多品種少量生産型の設計プロセスであるといえる。しかし同時にここで問われるべきなのは、多品種の〈かち〉である。

以上を踏まえ、「パラメトリック・デザイン」と「バイオミメティック・デザイン」のフローグラフを Fig. 4-7 に示す。設計プロセスの遺伝子は 0000011 0000111

1111111 0111010 1 となるので、これを十進数で表すと 12,844,917 となる。他の設計プロセスの十進数と比較すると、7番目に数字が大きいため、これらの設計プロセスは、第七世代の設計プロセスと呼ぶこととする。

要
旨

目
次

序
章

第
1
部

第
2
部

結
章

後
付

4.8 第八世代の設計プロセス：記号化の状況

第六世代の設計プロセスにおいて、情報環境に膨大な玉石混淆の〈かたち〉 II が生成されてしまうのならば、〈かたち〉 II を生成する前に情報環境に〈かち〉を外部化し、その〈かち〉に適合した〈かたち〉だけを生成すればいいように思える。つまり、〈かち〉を分析する〈かた＝アルゴリズム〉に着目し、〈かた〉→〈かち〉というプロセスを情報環境へ外部化するのである。その嚆矢が、『ノート』においてアレグザンダーが提示した「記号化の状況」である (Fig. 2-7 参照)。

「記号化の状況」においても、実際に環境の〈かち〉を取得するのは設計者だが、意匠性や歴史性といった第四層の記号性に関する〈かち〉は、定量化し外部化するのが難しいことは、第四世代ですすでに見てきた。しかしここでアレグザンダーが秀逸なのは、良く適合する形とは何かを直接問わずに、不適合を除いた形のすべてが良い適合である [4-37] と、デザインの定義を逆説的に読み替えることによって、定量化しにくい第四層の記号性に関する〈かち〉でさえも、他の〈かち〉と同様に不適合変数として外部化したことである。

ここで i 個の不適合変数の集合を $M = \{ m_1, m_2, \dots, m_i \}$ とする。不適合変数 M は、不適合があれば値 1、不適合が取り除かれていれば値 0 というように、形とコンテキストの関係をビットで表現できる。さらに、不適合変数 M をノード、それらの相互関係 L をエッジとすると、グラフ $G(M, L)$ で表現できる。このグラフ G は、設計者の環世界にモデル化された〈かち＝コンテキスト〉に他ならない。アレグザンダーによれば、良いデザインとは、この不適合変数 M をすべて 0 にするような形を与えることである。しかし通常、不適合変数は複雑に絡み合っているので、グラフ G の不適合変数をすべて 0 にするような全体の〈かたち〉を一気に創造するのは困難である。そこで、全体の〈かたち〉を作る前に不適合変数 M を分解しようと試みる。アレグザンダーは、このセミラティス状に絡みあったグラフ G を解きほぐすために、HIDECS と呼ばれる相関分析の〈かた＝アルゴリズム〉を用いた。つまり、相関が強い不適合変数から順にサブ・セットへと分解し、セミラティス状のグラフ G を、ツリー状のグラフ $G'(M, L)$ へと再構築した上で情報環境に外部化したのである。この際、相関が少ないエッジ L のいくつかは、削除されてしまうことに留意したい。

ツリー・グラフ $G'(M, L)$ は上から下へとサブ・セットに分解されるので、もはや他のサブ・セットの不適合変数とは絡み合っていない。したがって、最下層のサブ・セット内の不適合変数をすべて 0 にするような部分的な〈かたち〉を、設計に無自覚な他者でも容易に創造することができるのである。そして今度は分解と逆の流れをたどって、その部分的な〈かたち〉を、下から上へと順に合成することによって、不適合変数 M がすべて 0 となるような全体の〈かたち〉が出力される。

1964年、『ノート』においてこのような革新的な設計プロセスが発表されると、日本でも翻訳が出る前から海賊版が出まわるほど大きな話題となった [4-38]。しかし翌1965年、同じくアレグザンダーによって「都市はツリーではない」という論考が発表される。その内容は、自然発生的に形成された都市はセミラティス状の構造を持つのに対し、ひとりの設計主体によって人工的に構想された都市は必ずツリー構造を持つてしまうことを論証したものだだったが、それは同時に、『ノート』で示した「記号化の状況」の限界を、自ら論証してしまうものだったといえる。

現在から見れば、「記号化の状況」の限界は明らかである。「全体とは、部分の総和以上のなにかである」という複雑系科学の知見は、部分的な〈かたち〉が、たとえ部分的な〈かち〉と適合したとしても、その部分的な〈かたち〉をもう一度全体へと統合したときに、まったく異なる〈かち〉へと創発する可能性があることを示している。

アレグザンダーはその後、ツリーの最下層において、頻繁に現われる〈かた=コンテキスト〉と〈かたち〉のセットを「パタン」としてまとめ、それらの関係性を自然言語でデータベース化した [4-39]。この「パタン・ランゲージ」による設計プロセスは、自然言語が持つコードの恣意性を利用して、「記号化の状況」においてツリー構造だった形を、再度セミラティス構造の形へと合成することを狙ったものである [4-40]。しかし結局は、部分的な〈かち〉に適合した部分的な〈かたち〉を組み合わせ、全体の〈かたち〉を合成しようとすることに変わらない。それは実際に「パタン・ランゲージ」を適用してつくられた《盈進学園東野高等学校》(1984)の加算的なデザインを見れば明らかである。「部分を総和しても全体にならない」という要素還元的なデザイン手法の限界は、「形態は機能に従う」という近代建築のテーゼの限界でもある。『ノート』において、設計プロセスを徹底的に形式化することによって初めて明らかになったこのような限界を、「アルゴリズムック・デザイン」は、「パタン・ランゲージ」とは異なる方法で乗り越えようとするのである。

以上を踏まえ、「記号化の状況」のフローグラフを Fig. 4-8 に示す。設計プロセスの遺伝子は 1111010 0000111 1111111 0000111 1 となるので、これを十進数で表すと 511,966,991 となる。他の設計プロセスの十進数と比較すると、8番目に数字が大きいため、これらの設計プロセスは、第八世代の設計プロセスと呼ぶこととする。

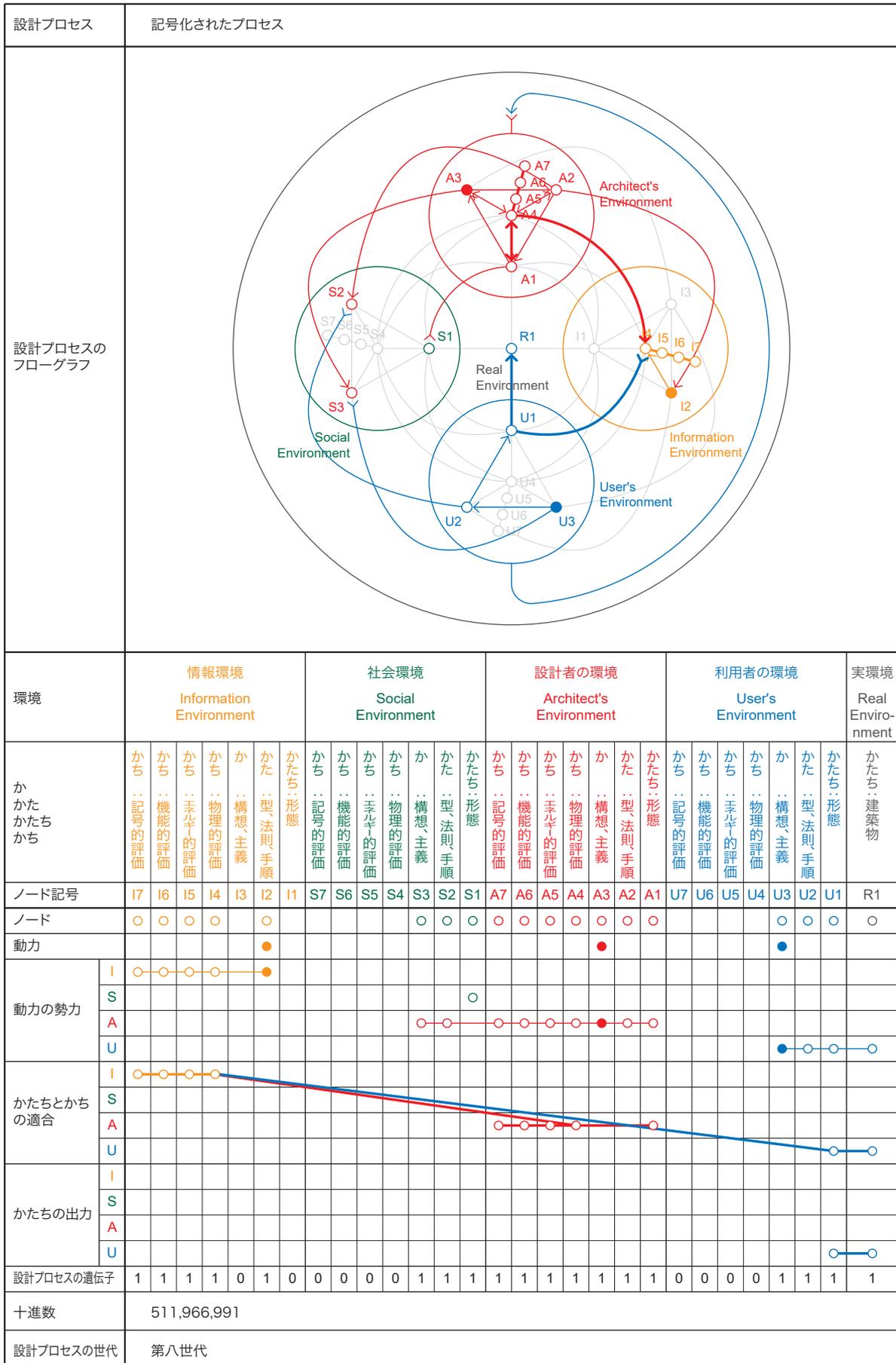


Fig. 4-8 第八世代のフローグラフ

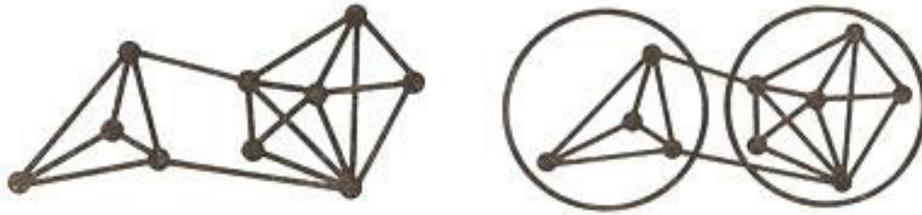


Fig. 4-8-1 セミ・ラティス状に絡み合った〈かち〉(左)と
相関する〈かち〉のサブセット(右)

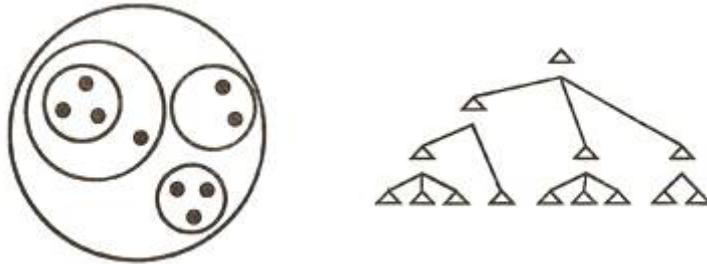


Fig. 4-8-2 ツリー状に分解された〈かち〉

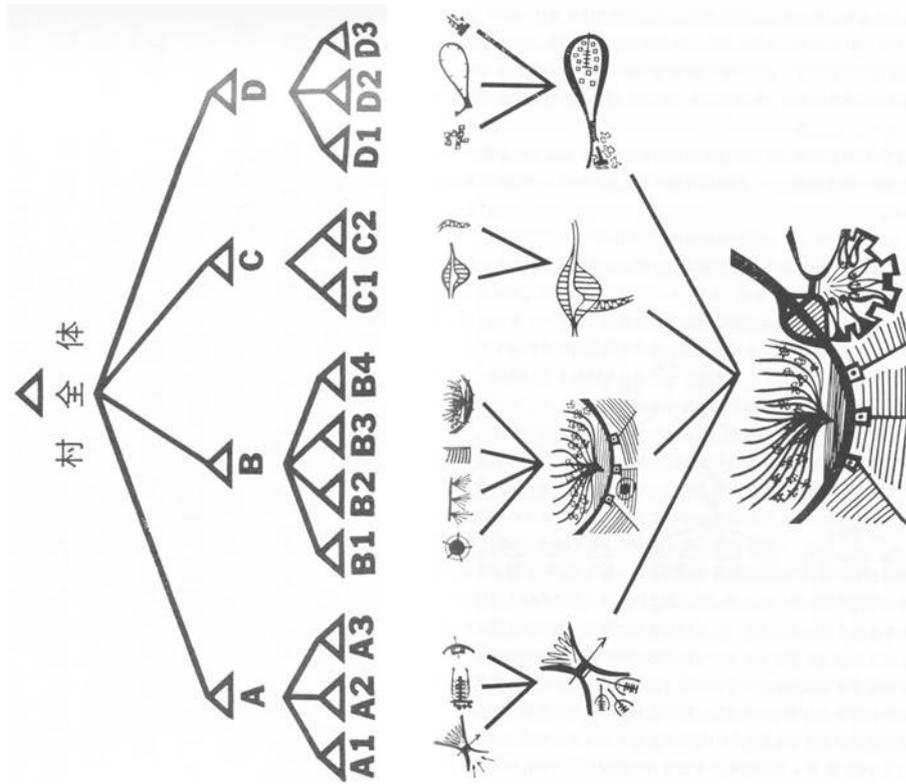


Fig.4-8-3 ツリー状に分解した〈かち〉(左)に対応する部分的な〈かたち〉を作成し全体の〈かたち〉を合成する(右)

4.9 第九世代の設計プロセス：アルゴリズムック・デザイン

第七世代では、〈かち〉が情報環境へ外部化されていないので、〈かた＝アルゴリズム〉から生成された〈かたち〉は玉石混淆となった。かといって第八世代でみたように、〈かたち〉を生成する前に〈かち〉を分析することにも限界があった。ではどうすればいいのか。「たくさん作って選ぶ」のである。〈かち〉に適合する〈かたち〉を事前的に「創造」あるいは「計画」するのではなく、ありうべき全体の〈かたち〉を「生成」してから、〈かち〉に適合する〈かたち〉を事後的に「選択」するのである。情報環境に外部化されたこのようなプロセスを俯瞰してみれば、〈かたち〉の可能性が進化論的に「淘汰」されているように見えるだろう。「形態は機能に従う」のではなく「生き残った形態が機能的」なのである。「創造」から「生成→淘汰」へ。この一見単純な設計概念の転回こそが、「アルゴリズムック・デザイン」の本質なのである [4-41]。

本節では、「アルゴリズムック・デザイン」を位置づける。

たとえば、前述した「敷地Aに建設できる最大ヴォリュームを求めよ」という問題には、「ヴォリュームは大きいほど良い」という設計者が考える〈かち〉が事前的に含まれている。したがってこの問題を解く〈かた＝アルゴリズム〉から生成される〈かたち〉は、その設計者が考える〈かち〉にしか答えることができない。しかし当然、ヴォリュームが小さいほど〈かち〉があると思う人もいるはずである。第三世代において、〈かたち〉は〈かち〉と切り離された自立した存在であることをすでに見てきた。そこでまずは、問題の中から〈かち〉を取り除き、良いも悪いも問わないで、ありうべき〈かたち〉の可能性をすべて生成できるような〈かた＝アルゴリズム〉を情報環境へ外部化する (I2) [4-42]。

次に、第八世代の「記号化の状況」と同じように、考慮すべき〈かち〉を情報環境へ外部化する。しかしここでは、第八世代のようにお互いに絡みあった〈かち〉をツリー構造に分解する必要はない。ここでの〈かち〉は二値数ではなく、環境にどの程度適応しているかによって0から1の間の実数値を取り得るからである。この値を「適応度」と呼ぶ。すなわちここでは適応度を計算するための〈かた＝アルゴリズム〉I2を情報環境に外部化するのである。

したがって、定量的に評価することができる〈かち〉I4, I5, I6は評価関数という〈かた＝アルゴリズム〉I2によって適応度が計算され、情報環境にされることとなる。また第四世代でみたお金に関する〈かち〉も、ここでは適応度のひとつになり得る。さらに、意匠性や歴史性といった第四層の記号性に関する〈かち〉も、事前的に評価関数を定義するのは難しいとしても、〈かたち〉はすでに情報環境へ外部化されているので、設計に無自覚な他者でもそれを仮想的に体験し、事後的に適応度を定めること

ができる。

このように〈かち=適応度〉と〈かたち〉がどちらも情報環境に外部化されているので、第四世代と同じく、設計者以外の他者でも〈かたち〉を選択することができる。

しかし、このままではまだ、第四世代で見たような資本主義的な市場原理によって、結局は画一的な建物が量産される可能性が残る。そこで、設計者がある程度〈かち〉をコントロールできるように、それぞれの適応度に重みづけを施し、その値を補正する。この補正は、第三世代で定式化した関係主義のような、設計者の〈か=イズム〉に他ならない。したがって、情報環境に補正係数として〈か〉I3が外部化されることとなる。このように、設計者の〈か=イズム〉によって、〈かたち〉の〈かち〉をある程度コントロールすることができるのである。ただしここでの〈か=イズム〉は、第二世代のように社会環境へとマニフェストする規範としての〈か=イズム〉ではなく、情報環境に外部化されたアーキテクチャとしての〈か=イズム〉であり、システム内部に隠蔽されてしまうことに留意したい。

このように〈かた〉I2から〈かたち〉I1を生成し、その〈かたち〉I1を評価関数〈かた〉I2によって、環境にどのくらい適合しているかを〈かち〉である適応度I4, I5, I6, I7を算出することで評価を行うのである。この評価の際に設計者の〈か〉によって〈かち〉の値が補正されている。

ここでもし〈かたち〉の〈かち=適応度〉が高ければその〈かたち〉をR1へと出力する。しかし〈かたち〉の〈かち=適応度〉が低ければ、もう一度〈かたち〉を生成して〈かち=適応度〉を算出しなければならない。このとき、なんのあてもなくランダムに〈かたち〉を生成しては非常に効率が悪い。したがってより〈かち=適応度〉が高くような〈かたち〉に改善する必要がある。そのような〈かた=アルゴリズム〉はヒューリスティクスと呼ばれる[4-43]。詳しくは第6章で触れる。このヒューリスティクスという〈かた=アルゴリズム〉を情報環境へ外部化することで、アルゴリズムミック・デザインでは〈かち〉に適合した〈かたち〉が出力されるのである。

このようなアルゴリズムは、これまで各世代で見てきた「試行錯誤」と本質的には何も変わらない。「アルゴリズムミック・デザイン」では、この単純で素朴な試行錯誤を膨大な回数繰り返すことによって、〈かち〉に適合した〈かたち〉を出力することを可能にするのである。

以上を踏まえ、「アルゴリズムミック・デザイン」のフローグラフをFig. 4-9に示す。設計プロセスの遺伝子は1111111 0000111 1111111 0111001 1となるので、これを十進数で表すと532,938,611となる。他の設計プロセスの十進数と比較すると、9番目に数字が大きいため、これらの設計プロセスは、第九世代の設計プロセスと呼ぶこととする。

第5章

第1部の小結

本章では、各種設計プロセスのフローグラフをもとに分類を行い、それらの共通点や相違点を考察し、第1部の小結を述べる。

5.1 世代による分類

第4章で作成した各種設計プロセスのフローグラフの中で、「設計プロセスの世代」の項目に着目すると下記のような分類ができる。

- (a) 第一世代：「無自覚なプロセス」、「手工業的なプロセス」
- (b) 第二世代：「図面によるプロセス」、「自覚的なプロセス」、モダニズムの設計プロセス
- (c) 第三世代：「ポストモダンの設計プロセス」
- (d) 第四世代：「大量生産型の設計プロセス」
- (e) 第五世代：「パタン・ランゲージ」、「超線型設計プロセス」
- (f) 第六世代：「BIM」
- (g) 第七世代：「パラメトリック・デザイン」、「バイオ・ミメティック・デザイン」
- (h) 第八世代の「記号化されたプロセス」
- (i) 第九世代の「アルゴリズムック・デザイン」

この中で、世代と設計プロセスの年代にギャップがあるのは、第八世代の「記号化されたプロセス」のみであり、アレグザンダーの先見性が際立つ結果が見て取れる。

5.2 かたちの出力による分類

第4章で作成した各種設計プロセスのフローグラフの中で、「かたちの出力」の項目に着目すると下記のような分類ができる。

- (a) かたちの出力の属性が「利用者の環境」である設計プロセス
 - ・第一世代：「無自覚なプロセス」、「手工業的なプロセス」
 - ・第五世代：「パタン・ランゲージ」、「超線型設計プロセス」
 - ・第八世代：「記号化されたプロセス」
 - ・第九世代：「アルゴリズムック・デザイン」
- (b) かたちの出力の属性が「設計者の環境」である設計プロセス
 - ・第三世代：「ポストモダンの設計プロセス」
 - ・第六世代：「BIM」
 - ・第七世代：「パラメトリック・デザイン」、「バイオ・ミメティック・デザイン」
- (c) かたちの出力の属性が「社会環境」である設計プロセス

- ・第二世代の「図面によるプロセス」、「自覚的なプロセス」、モダニズムの設計プロセス

- ・第四世代の「大量生産型の設計プロセス」

(d) かたちの出力の属性が「情報環境」である設計プロセス

- ・第二世代の「図面によるプロセス」、「自覚的なプロセス」、モダニズムの設計プロセス

- ・第四世代の「大量生産型の設計プロセス」

(a) の中に、アレグザンダーの設計プロセスがすべて入っていることは、本論で提案した進化論的枠組みが、分類方法として妥当であることを示していると考ええる。

またコンピュータ以前と以後の設計プロセスが混在している分類ができていることは、本論で提案した進化論的枠組みが、分類方法として妥当であることを示していると考ええる。

5.3 コンピュータライゼーションとコンピューテーションによる分類

コスタス・テルジディスは、コンピュータライゼーションとコンピューテーションは異なると述べているが [5-1]、両者をコンピュータを用いた設計プロセスの中で分類すると下記のようなになる。

(a) コンピュータライゼーション

- ・第六世代：「BIM」

(b) コンピューテーション

- ・第七世代：「パラメトリック・デザイン」、「バイオ・ミメティック・デザイン」

- ・第八世代：「記号化されたプロセス」

- ・第九世代：「アルゴリズムック・デザイン」

第七世代では、情報環境に〈かたち〉I1 を生成するための〈かた＝アルゴリズム〉I2 が外部化された。また第八世代では、情報環境に〈かち〉I4, I5, I6, I7 をツリー構造へ分解する〈かた＝アルゴリズム〉が外部化された。そして第九世代では、〈かたち〉I1 を生成するための〈かた＝アルゴリズム〉I2、〈かたち〉I1 を評価する〈かた＝アルゴリズム〉I2、〈かち〉I4, I5, I6, I7 によりよく適合するような〈かたち〉I1 を探索する〈かた＝アルゴリズム〉I2 が外部化された。

このように情報環境に〈かた＝アルゴリズム〉が外部化されている設計プロセスを総称しているのがコンピューショナル・デザインであると考えられる。

5.4 コンピューショナルデザインの中での分類

- (a) 〈かたち〉の生成に〈かた＝アルゴリズム〉を用いている設計プロセス
 - ・第七世代：「パラメトリック・デザイン」、「バイオ・ミメティック・デザイン」
 - ・第九世代：「アルゴリズムック・デザイン」
- (b) 〈かち〉の分析や評価に〈かた＝アルゴリズム〉を用いている設計プロセス
 - ・第八世代：「記号化されたプロセス」
 - ・第九世代：「アルゴリズムック・デザイン」
- (c) 〈かち〉に適合するような〈かたち〉の探索に〈かた＝アルゴリズム〉を用いている設計プロセス
 - ・第九世代：「アルゴリズムック・デザイン」

(a) (b) (c) のいずれにも属しているのがアルゴリズムック・デザインの特異性であるということがわかる。

5.5 第1部の小結

第1部の小結を下記にまとめる。

- (1) 第2章において、各種建築設計プロセスを位置づけるための「進化論的枠組み」を仮説的に提案した。
- (2) 第3章では、その「進化論的枠組み」から演繹的に予想される帰結としてグラフ理論を用いた建築設計プロセスのフローグラフを導出した。
- (3) 第4章では、各種設計プロセスを「進化論的枠組み」の中に帰納的に位置づけた。また各種設計プロセスのフローグラフを作成することにより、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを明らかにした。
- (4) 第5章では、各種設計プロセスのフローグラフをもとに分類を行い、それらの共通点や相違点を考察した。

以上から、アルゴリズムック・デザインとその他の各種設計プロセスを「進化論的枠組み」の中に位置づけることを通して、生成のフェイズ、評価のフェイズ、高適化のフェイズのいずれのフェイズにもアルゴリズムを用い、それらのフェイズを繰り返すことによって情報環境内で建築の試行錯誤を行っているという点が、他の設計プロセスにはないアルゴリズムック・デザインの特異性であることが明らかとなった。

要旨

目次

序章

第1部

第2部

結章

後付

要
旨

目
次

第
1
章

第
2
章

第
3
章

第
4
章

第
5
章

第 2 部

第
6
章

アルゴリズムック・デザインのプロトタイプ構築

第
7
章

第
8
章

第
9
章

第
1
0
章

第
1
1
章

第
1
2
章

後
付

要旨

目次

序章

第1部

第2部

結章

後付

第6章

アルゴリズムック・デザインの定式化

本章では、アルゴリズムック・デザインのフローグラフをもとに、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインの定式化を試みる。

6.1 生成のフェイズ・評価のフェイズ・高適化のフェイズの定義

第4.9節において、アルゴリズムック・デザインを第九世代に位置づけ、Fig. 4-9に示したようなフローグラフを得た。そのアルゴリズムック・デザインのフローグラフの中で、情報環境を拡大したものをFig. 6-1に示す。情報環境内に着目すると、アルゴリズムック・デザインでは下記の3つのアルゴリズムが情報環境に外部化されていた。

- (1) 〈かたち〉 I1 を生成するための 〈かた=アルゴリズム〉 I2
- (2) 〈かたち〉 I1 を評価する 〈かた=アルゴリズム〉 I2
- (3) 〈かち〉 I4, I5, I6, I7 によりよく適合するような 〈かたち〉 I1 を探索する 〈かた=アルゴリズム〉 I2

本論では、上記のそれぞれに対応したフェイズを、下記のように定義する。

- (1) 生成のフェイズ
- (2) 評価のフェイズ
- (3) 高適化のフェイズ

以下それぞれのフェイズに分けて、アルゴリズムック・デザインの定式化を試みる。

6.2 生成のフェイズ＝〈かたち〉の枠組み

ここで、第九世代のフローグラフ Fig. 4-9 に記された情報環境の〈かたち〉の集合 I1 に注目する。この I1 をもっと拡大したものが Fig. 6-2 である。無数にある白丸は情報環境に外部化された〈かたち〉の可能態である。〈かたち〉の可能態は無限にあると考えて良いだろう。本節ではこの無限にある〈かたち〉の可能態の中から有限の〈かたち〉をフレーミングしていく。

たとえば、「敷地 A に建設できる最大ヴォリュームを求めよ」という問題には、「ヴォリュームは大きいほど良い」という設計者が考える〈かち〉が事前的に含まれている。したがってこの問題を解く〈かた＝アルゴリズム〉から生成される〈かたち〉は、その設計者が考える〈かち〉にしか答えることができない。しかし当然、ヴォリュームが小さいほど〈かち〉があると思う人もいるはずである。第三世代において、〈かたち〉は〈かち〉と切り離された自立した存在であることをすでに見てきた。そこでまずは、問題の中から〈かち〉を取り除き、良いも悪いも問わないで、ありうべき〈かたち〉の可能性をすべて生成できるような〈かた＝アルゴリズム〉を情報環境へ外部化する。

ここで、全体の〈かたち〉を制御する j 個の**設計変数**を $N = \{n_1, n_2, \dots, n_j\}$ で表わし、〈かたち〉を生成する関数を KATACHI() とすると、〈かたち〉は次のように定式化することができる。

$$\langle \text{かたち} \rangle = \text{KATACHI}(N) \quad \text{—— (6-1 式)}$$

ただし、たとえば「敷地 A からはみ出さない」というような、必ず守らなければいけない「**制約条件**」があるので、それぞれの設計変数は無限の値を取るわけではなく、ある範囲に制限される。設計変数は連続変数の場合もあるが、ここでは簡易のために変数を離散化して、それぞれの設計変数が 10 通りの値を取りうると仮定する。そうすると、(6-1 式) によって生成される〈かたち〉は全部で 10 の j 乗個となる。

ここで Fig. 6-2 にもう一度着目する。無数にある白丸は情報環境に外部化された〈かたち〉の可能態であった。オレンジの枠で囲まれた領域が、(6-1 式) で定式化された〈かたち〉の可能態である。すなわちこの集合のなかには現在、10 の j 乗個の〈かたち〉が点で表現されている。しかしこの集合の外側には、平面上に無限に広がっているであろう〈かたち〉の可能性はあるはずである。つまりここでは、設計変数と制約条件によって、無限の〈かたち〉のなかから有限の〈かたち〉を切り取り、いわば「〈かたち〉の枠組み」を設計しているのである。いかに少ない設計変数で多様な〈かたち〉を生成できるか、それが問われているのである。

以上のようなフェイズが「生成のフェイズ」である。

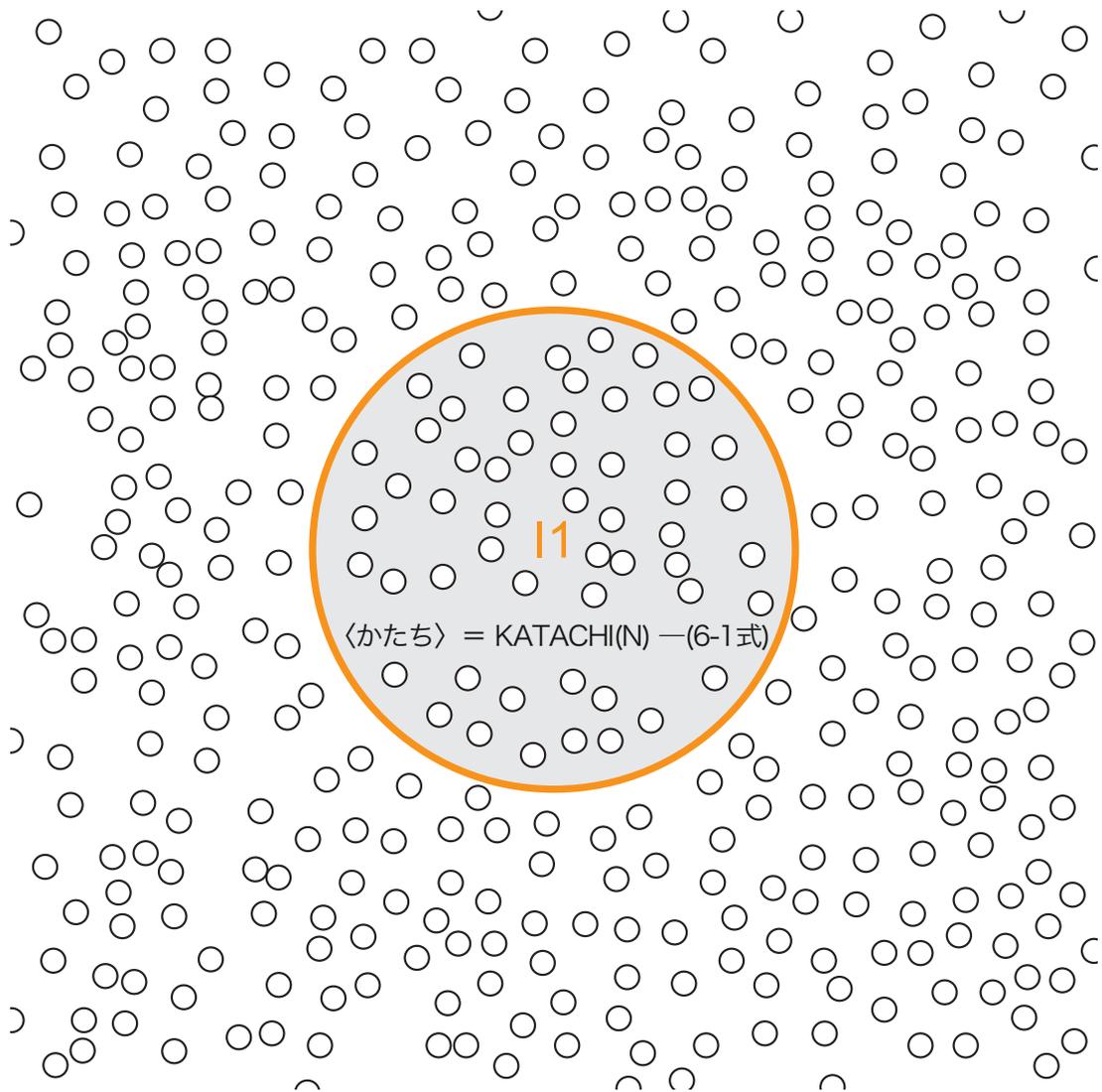


Fig. 6-2 生成のフェイズ
第九世代のフローグラフの情報環境の I1 を拡大

6.3 評価のフェイズ＝〈かたち〉の適応度地形

次に、第八世代の「記号化されたプロセス」と同じように、考慮すべき i 個の〈かち〉の変数を $M = \{m_1, m_2, \dots, m_i\}$ で表わす。しかしここでは、第八世代のようにお互いに絡みあった変数 M をツリー構造に分解する必要はない。ここでの〈かち〉は二値数ではなく、環境にどの程度適応しているかによって 0 から 1 の間の実数値を取り得るからである。変数 M を「適応度」と呼ぶ。さらに、設計変数 N と適応度 M を結びつける「評価関数」を $KACHI()$ で表わせれば、適応度 M は次のように定式化できる。

$$\text{適応度 } m_i = KACHI_i(N) \quad \text{—— (6-2 式)}$$

たとえば、 m_i が変位やひずみに関する構造力学的な適応度だとすれば、10 の j 乗個の〈かたち〉それぞれに対して、0 から 1 までの点数を付けることができるような評価関数 $KACHI_i$ を定義し、情報環境に外部化することができる。また第四世代でみたお金に関する〈かち〉も、ここでは適応度のひとつになり得る。さらに、意匠性や歴史性といった第四層の記号性に関する〈かち〉も、事前的に評価関数を定義するのは難しいとしても、〈かたち〉はすでに情報環境へ外部化されているので、設計に無自覚な他者でもそれを仮想的に体験し、事後的に適応度を定めることができる。

ここでもう一度、フローグラフ Fig. 6-2 の〈かたち〉II の集合に注目する。〈かたち〉を表わす点それぞれが、適応度に応じて高さをもっていると仮定する。平面的にはなく立面的に視るのである。このような視点でみた図を Fig. 6-3 に示す。そうすると、〈かたち〉の集合全体は、0 から 1 までの標高をもった「適応度地形」として捉えることができる。同じ〈かたち〉でも、検討する〈かち〉によって適応度が高いときもあれば低いときもあるので、 i 個の適応度地形の起伏はすべて異なる。

次に、波を重ね合わせるように、 i 個の適応度地形を重ね合わせて全体の適応度地形をつくる。このとき、ひとつの〈かたち〉における全体の〈かち＝適応度〉は、(6-2 式)の適応度をすべて足しあわせ、その個数 i で割った平均値として定式化できる。

$$\langle \text{かち} = \text{適応度} \rangle = \text{SUM}(M)/i \quad \text{—— (6-3 式)}$$

このように〈かち＝適応度〉と〈かたち〉がどちらも情報環境に外部化されているので、第四世代と同じく、設計者以外の他者でも〈かたち〉を選択することができる。そのとき、適応度地形のより高い場所にある〈かたち〉は建物として出力される確率が高くなり、その逆に、〈かち〉なき〈かたち〉は適応度地形の谷に落ち込んで出力され難くなる。可能態としての〈かたち〉が現実態としての建物になる際の確率分布

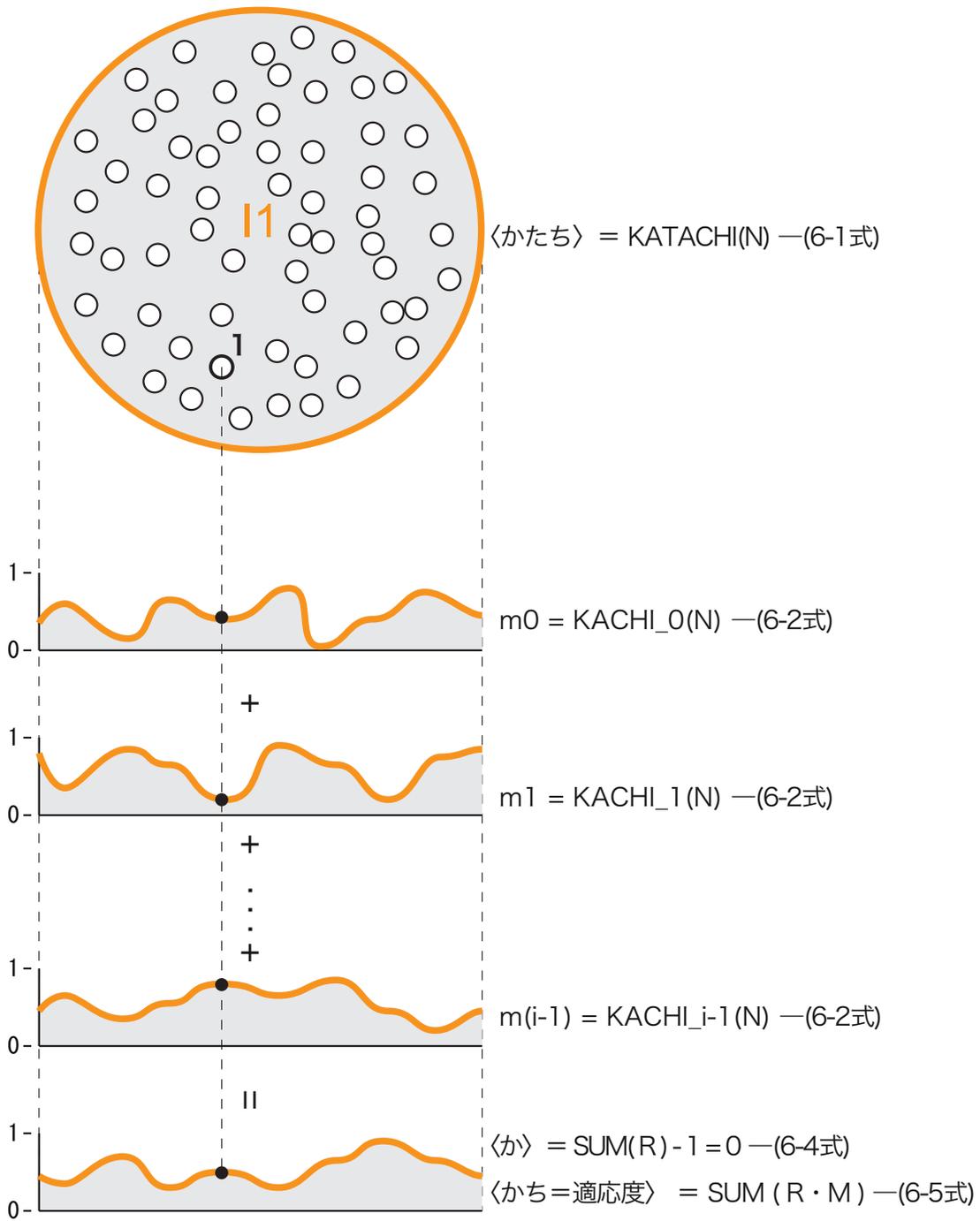


Fig. 6-3 評価のフェイズ
第九世代のフローグラフの情報環境の I1 を拡大

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

が適応度地形なのである。ただ、ある人の適応度地形において、ある〈かたち〉の適応度が低いからといって、その〈かたち〉が完全に淘汰されるわけではない。また逆に、適応度が高いからといって、誰にでも選択されるとは限らない。現実の地形は誰から見ても同じだが、適応度地形は選択する人によって、動的に変化するからである。つまりここでは、適応度地形がシステムの動的平衡を担うのである。

しかし、このままではまだ、第四世代で見たような資本主義的な市場原理によって、結局は画一的な建物が量産される可能性が残る。そこで、設計者がある程度〈かち〉をコントロールできるように、それぞれの適応度に重みづけを施し、その値を補正する。ここではその補正係数を $R=\{r_1, r_2, \dots, r_i\}$ と表す。適応度の数だけ補正係数があるのでその個数は i 個である。補正後も適応度の値は0から1の間の値をとるとすると、補正係数をすべて足し合わせると1となる。さらに言えば、この補正係数 R は、第三世代で定式化した関係主義のような、設計者の〈か=イズム〉に他ならない。したがって補正係数は下記のように定式化できる。

$$\langle \text{か=イズム} \rangle = \text{SUM}(R) - 1 = 0 \quad \text{--- (6-4 式)}$$

そして (6-3 式) は、 $\text{SUM}(1/i \cdot M)$ と変形できるので、補正後の適応度は、下記のように表わすことができる。

$$\langle \text{かち=適応度} \rangle = \text{SUM}(R \cdot M) \quad \text{--- (6-5 式)}$$

たとえば、設計者が意匠性に関する適応度を m_2 とし、その適応度を重視するのであれば、補正係数 r_2 を $1/i$ よりも高く設定することで、他の適応度を相対的に抑制することができる。このように、設計者の〈か=イズム〉によって、適応度地形をある程度コントロールすることができるのである。ただしここでの〈か=イズム〉は、第二世代のように社会環境へとマニフェストする規範としての〈か=イズム〉ではなく、情報環境に外部化されたアーキテクチャとしての〈か=イズム〉であり、システム内部に隠蔽されてしまうことに留意したい。

このように第八世代においては、「コンテキストによりよく適合する形を出力すること」というデザインの定義は、「適応度地形をより高く登ること」と同義であり、スチュアート・カウフマンの言葉を借りて言い換えれば、デザインとは「高地への冒険」だといえる [6-1]。

以上のようなフェイズが「評価のフェイズ」である。

6.4 高適化のフェイズ＝〈かたち〉の適応度地形を登る

設計変数 N や適応度 M の数が少ない場合は、すべての〈かたち〉に対して〈かち＝適応度〉を求めることができるので、考慮した〈かち＝コンテキスト〉に対して、必ず最適な〈かたち〉を選択することができる。しかし実際には、その変数の組み合わせは指数関数的に爆発するので、たとえ (6-1 式) や (6-5 式) を定義できたとしても、すべての〈かたち〉の可能性を一望することは不可能である。それはたとえば、真夜中のアルプス山脈を、サーチライト片手に登山しているような状況である。見える範囲 (近傍) で最も高い地点 (局所最適解) に登ったとしても、全体が見渡せないのだから、すべての適応度地形のなかでその地点が最も高い (大域最適解) という保証はどこにもない。しかし逆にいえば、適応度地形の全貌を見渡すことができないからこそデザインは面白いのだといえる。

そもそもデザインに正解などないと達観して、設計者の経験と感性と運を頼りに適応度地形を登るのもいいが、全貌を見渡せなくてもより高地へと冒険できるような経験的手法を、〈かち＝アルゴリズム〉として情報環境へ外部化できれば、コンピュータでも適応度地形を登ることができる。そのような経験的手法のことを「ヒューリスティクス (heuristics)」という [6-2]。たとえば、局所探索法 (local search) [6-3]、擬似焼きなまし法 (Simulated Annealing) [6-4]、遺伝的アルゴリズム (Genetic Algorithm) [6-5] など、さまざまな探索アルゴリズムが試みられている。ここではその詳細に立ち入ることはしないが、たとえば局所探索法は、下記のようなアルゴリズム (手順) となる。

- STEP1 (6-1 式) によって〈かたち〉を生成する。
- STEP2 (6-5 式) によって、その〈かたち〉の適応度を求める。
- STEP3 以前の暫定解と比較して、より高い適応度ならば、その〈かたち〉を暫定解とする。
- STEP4 終了条件を満たしていれば暫定解の〈かたち〉を出力する。満たしていなければ STEP1 へ戻る。

このようなアルゴリズムは、これまで各世代で見てきた「試行錯誤」と本質的には何も変わらない。たくさん作って選ぶ。「生成のフェイズ」と「評価のフェイズ」を繰り返す。「アルゴリズムック・デザイン」では、この単純で素朴な試行錯誤を膨大な回数繰り返すことによって、高地への冒険を可能にするのである。

以上のようなフェイズが「高適化のフェイズ」である。

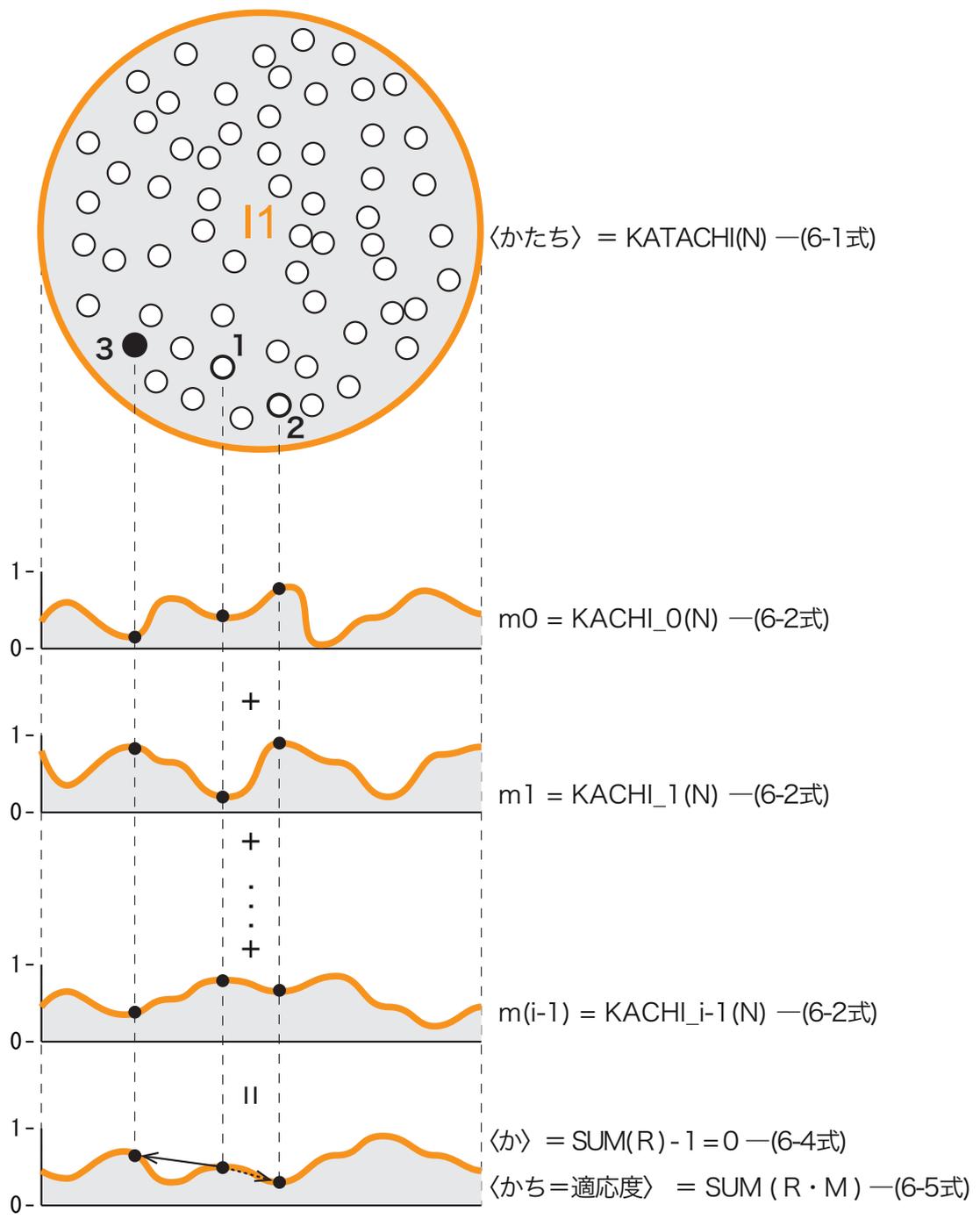


Fig. 6-4 高適化のフェイズ
第九世代のフローグラフの情報環境の I1 を拡大

第7章

アルゴリズムック・デザインのプロトタイプの構築

本章では、第6章の定式化をもとに、アルゴリズムック・デザインのプロトタイプを構築する。

7.1 概要

アルゴリズムック・デザインのプロトタイプを構築するにあたって、我々はデザインの対象を Fig. 7-1 のような 2 次元構築物とした [7-1][7-2]。後にこのプロトタイプを機能的なものへと応用することを考慮すると、プロトタイプでは機能性を省いたほうが汎用性が高いこと、構造計算によって物理的な評価を定量的に行うことができることが主な理由である。また次元を 3 次元から 2 次元へと落とすことについては、計算を簡易化することと、ユーザーによって「〇〇っぽい」というような定性的な評価を事後的に施しやすいようにしたいという狙いがある。

以下、第 6 章と同様に、「生成のフェイズ」「評価のフェイズ」「高適化のフェイズ」に分けて、アルゴリズムック・デザインのプロトタイプの詳細を述べる。

7.2 生成のフェイズ

Fig. 7-1 のような 2 次元構造体を〈かたち〉として生成する。この 2 次元構築物は、接合部としてのノード (Node)、部材としてのパス (Path) から構成される。

ノードの生成規則と制約条件について下記にまとめる。

- ・ ノードの数は 50 個とする。
- ・ かたちの多様性を表現するために、50 個すべてが表示されるわけではなく、ノードの表示／非表示は、初期設定ではランダムに割り振られる。
- ・ ノードが表示の場合は 1 を、非表示の場合は 0 を記す。ノードの遺伝子は Fig. 7-2 の *Node_GENE* のように 50 ビットでコーディングする。
- ・ すべてのノードの XY 座標は、 $0 < x < 1000$ 、 $0 < y < 1000$ に制約されるものとする。初期設定では制約の範囲内でランダムに座標が割り振られる。座標の集合を *Node_Coordinate* とする。
- ・ 少なくとも 1 つのノードが地面に接していなければならない。
- ・ 接合部はすべて剛接合とする。

パスの生成規則と制約条件について下記にまとめる。

- ・ 表示されているノード同士を、初期設定ではランダムにパスで結ぶ。
- ・ その際にパス同士は交差しないように制約する。
- ・ ノードの次数は 4 以下に制約する。

- ・ Fig. 7-2 の *Path_Matrix* で示すように、Node 同士が Path で結ばれていれば 1 を結ばれていなければ 0 を記す。
- ・ パスの断面は円形とし、半径は 5mm とする。
- ・ パスの素材は木材（ヒノキ）を想定し、ヤング係数は 9000N/mm²、比重は 0.4g/cm³ とする。
- ・ パスの自重を両端の接点に均等に振り分けて外力とする。積載荷重と短期荷重は考慮しない。
- ・ 少なくとも 1 つのバスが地面に接地していなければならない。

以上を、第 4.9.1 項で示した (4-9-1 式) に当てはめると以下のようになる。

〈かたち〉 = KATACHI(Node_GENE, Node_Coordinate, Path_Matrix) —— (7-1 式)

(7-1 式) の関数を実際にコンピュータプログラムで実装し、Node_GENE, Node_Coordinate, Path_Matrix を引数として渡しながらこの関数を呼び出すことで 2 次元構築物としての 〈かたち〉 を生成することができる。

7.3 評価のフェイズ

第 2.3 節で述べたように、〈かち〉は、難波の四層構造を参照することで、第 1 層の物理的評価の 〈かち〉、第 2 層のエネルギー的評価 〈かち〉、第 3 層の機能的評価の 〈かち〉、第 4 層の記号的評価の 〈かち〉 の 4 層で構成される。ここでは、定量的な評価基準として第 1 層の物理性に関する 〈かち〉 と、定量的な評価基準として第 4 層の記号性に関する 〈かち〉 を取り扱う。取り扱う 〈かち〉 の例を Fig. 7-3 に示す。

第 1 層の物理性に関する 〈かち〉 に関しては下記を取り扱う。

- ・ 適応度 $m_1 = \text{number_of_node}$: ノードの数
- ・ 適応度 $m_2 = \text{number_of_path}$: パスの数
- ・ 適応度 $m_3 = \text{total_path_length}$: パスの総長さ
- ・ 適応度 $m_4 = \text{max_displacement}$: ノードの最大変位
- ・ 適応度 $m_5 = \text{max_axial_force}$: 最大軸方向力
- ・ 適応度 $m_6 = \text{max_shearing_force}$: 最大せん断力
- ・ 適応度 $m_7 = \text{max_bending_moment}$: 最大曲げモーメント

LAYER 1 PHYSICS			0
number_of_node	19 node (s)	0.62 pts	
number_of_path	28 path (s)	0.70 pts	
total_path_length	5784.46 mm	0.71 pts	
max_displacement	32.02 mm	0.77 pts	
max_axial_force	9.72 N	0.88 pts	
max_shearing_force	4.79 N	0.89 pts	
max_bending_moment	0.51 kNm	0.68 pts	
LAYER 2 ENERGY			0
None	-	- pts	
LAYER 3 FUNCTION			0
None	-	- pts	
LAYER 4 SIGN			0
croissant	9.77 %	0.10 pts	
desk light	9.70 %	0.10 pts	
desk	4.32 %	0.04 pts	
kettle	1.77 %	0.02 pts	
plant	1.04 %	0.01 pts	
bicycle	0.80 %	0.01 pts	
cave	0.59 %	0.01 pts	
scissors	0.29 %	0.00 pts	
crane	0.13 %	0.00 pts	
egg	0.12 %	0.00 pts	
ostrich	0.06 %	0.00 pts	
turkey	0.05 %	0.00 pts	
2F	0.00 %	0.00 pts	
3F	0.00 %	0.00 pts	
alien	0.00 %	0.00 pts	
others	0.00 %	0.00 pts	
TOTAL FITNESS			0.11 pts

Fig. 7-3 〈かち〉の例

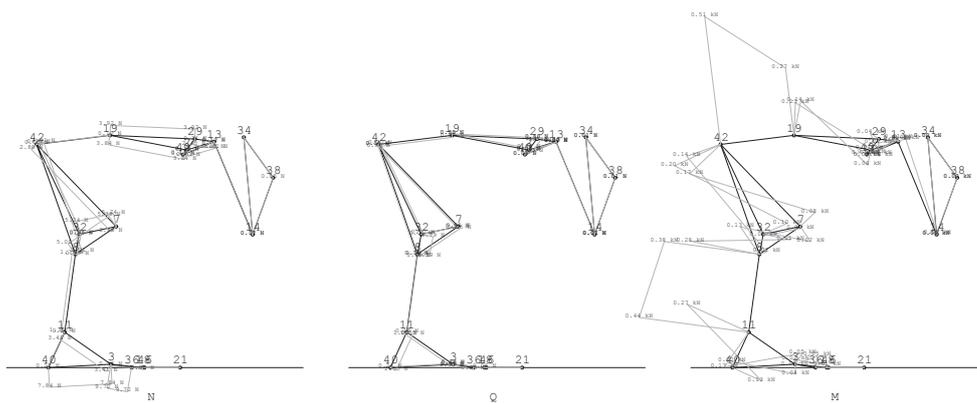


Fig. 7-4 軸力図 (左)、せん断力図 (中)、曲げモーメント図 (右)

上記すべての評価において、すべての〈かたち〉の中での最大値と最小値を保持しておくことで、各評価を0から1の間の値へと正規化する。なお、構造計算に関しては、マトリクス変位法による応力解析プログラムを自作し構造計算を行う。Fig7-4に構造計算によって求めた応力図の例を示す。

第4層の記号性に関する〈かち〉に関しては、あらかじめ評価指標を決めておくのではなく、ユーザーがタグを随時追加し、その〈かたち〉の評価を0から1の間で採点する。例えば、2次元構築物の〈かたち〉が「デスクライトっぽい」と思えば、*desk_light* というタグを追加し、その〈かたち〉がどの程度「デスクライトっぽい」かを0から1の間で入力する。

以上を、第4.9.2項で示した(4-9-2式)に当てはめると以下のようなになる。なお取り扱う*i*個の〈かち〉の適応度を $M=\{m_1, m_2, \dots, m_i\}$ で表わす。

適応度 $m_i = \text{KACHII}(\text{Node_GENE}, \text{Node_Coordinate}, \text{Path_Matrix})$ —— (7-2式)

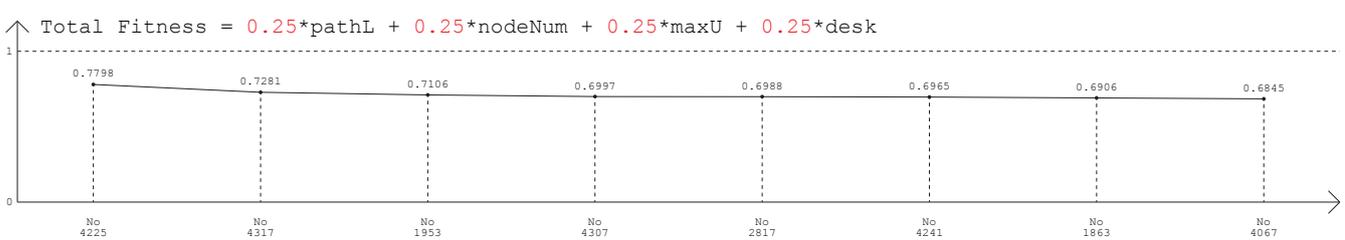
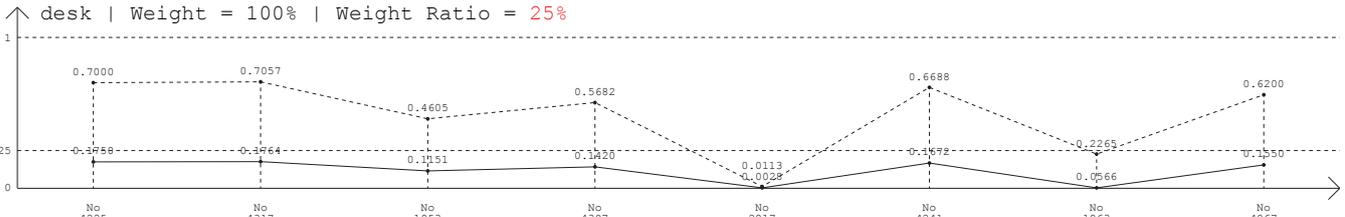
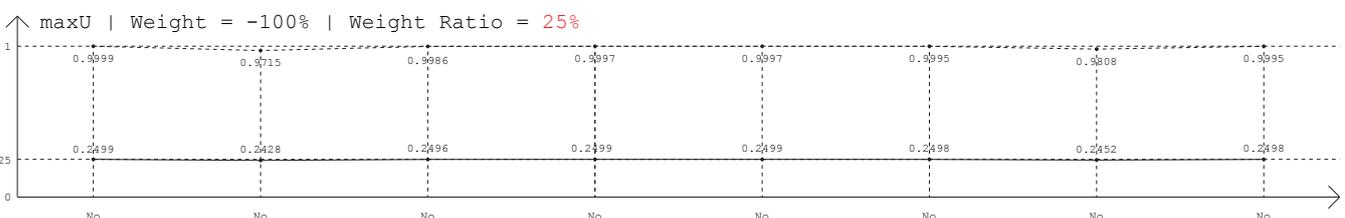
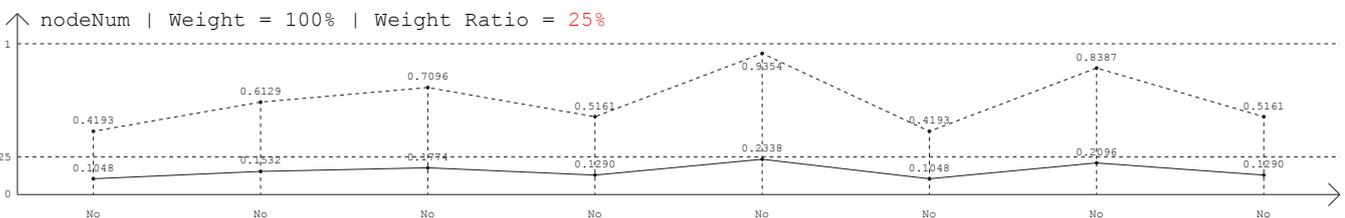
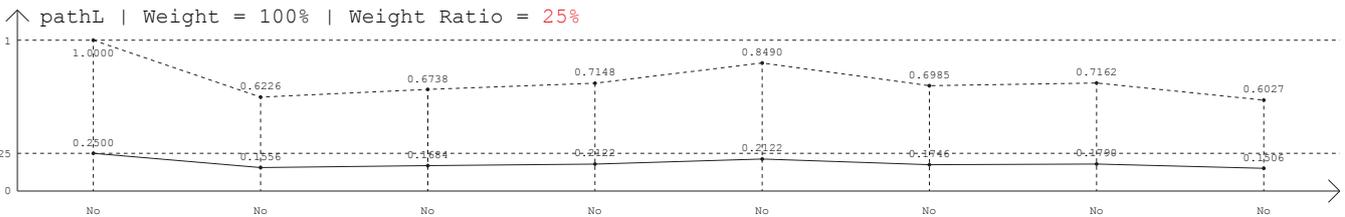
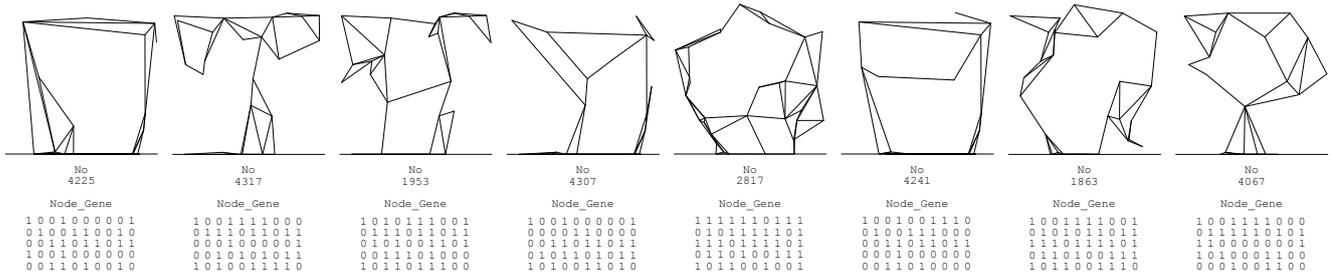
(7-2式)の関数を実際にコンピュータプログラムで実装し、*Node_GENE*, *Node_Coordinate*, *Path_Matrix* を引数として渡しながらかこの関数を呼び出すことで、適応度 m_i を計算することができる。

また、第4.9.2項の(4-9-5式)で見たように、それぞれのユーザーの〈か〉に応じた補正係数 R を考慮することで、すべての適応度を重ね合わせることができる。

〈かち = *Total_Fitness*〉 = $\text{SUM}(R \cdot M)$ —— (7-3式)

例えば、下記のような補正係数とした場合の *Total_Fitness* とその順位を Fig. 7-5 に示す。

- ・ 補正係数 $r_1 = 0$: *number_of_node* : ノードの数
- ・ 補正係数 $r_2 = 0$: *number_of_path* : パスの数
- ・ 補正係数 $r_3 = 0.25$: *total_path_length* : パスの総長さ
- ・ 補正係数 $r_4 = 0.25$: *max_displacement* : ノードの最大変位
- ・ 補正係数 $r_5 = 0$: *max_axial_force* : 最大軸方向力
- ・ 補正係数 $r_6 = 0$: *max_shearing_force* : 最大せん断力
- ・ 補正係数 $r_7 = 0$: *max_bending_moment* : 最大曲げモーメント
- ・ 補正係数 $r_8 = 0.25$: *desk* : 机っぽい



#1 #2 #3 #4 #5 #6 #7 #8

Fig. 7-5 補正係数を含めた適応度の例 1

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

要旨

目次

序章

第1部

第2部

結章

後付

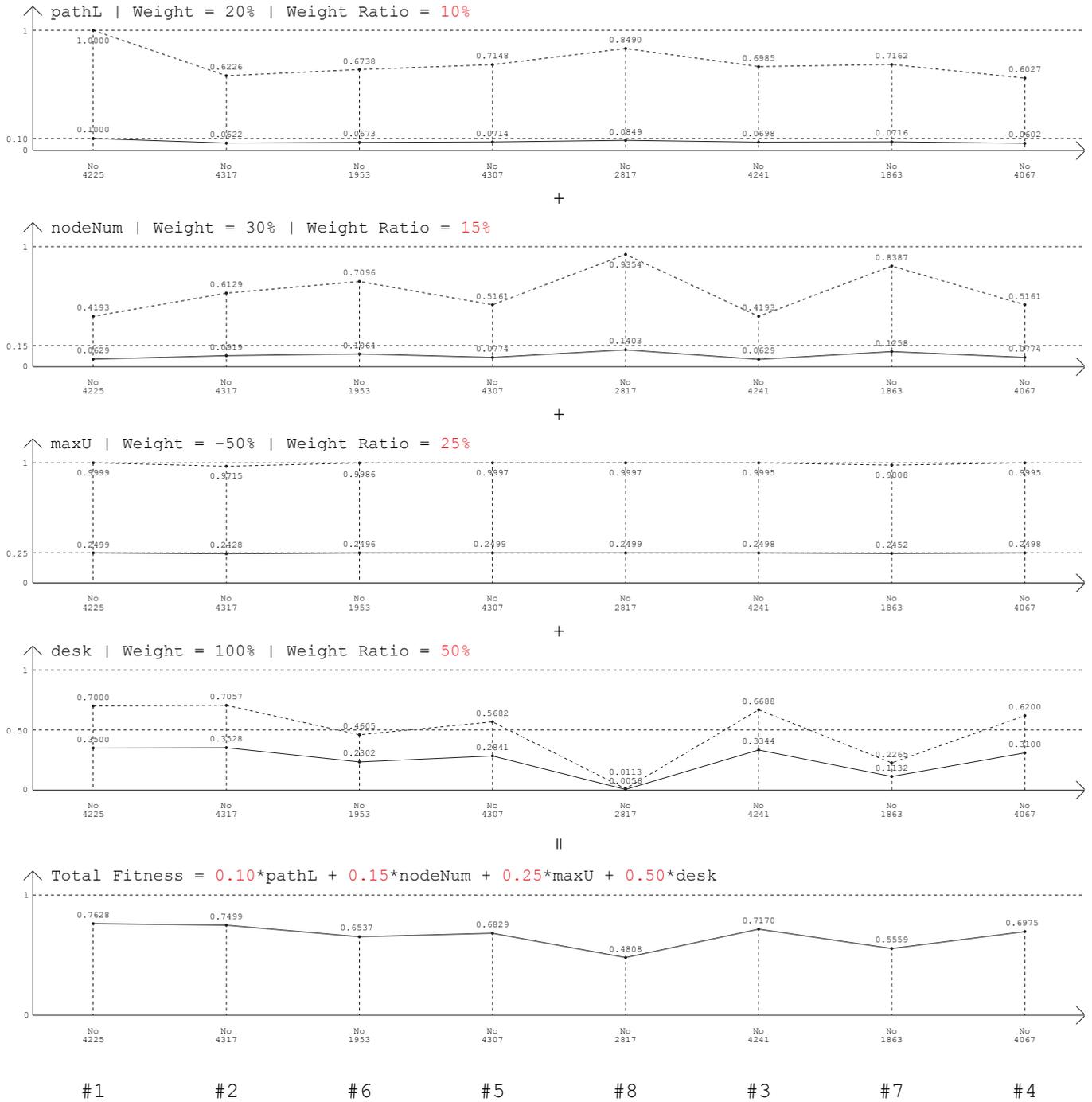
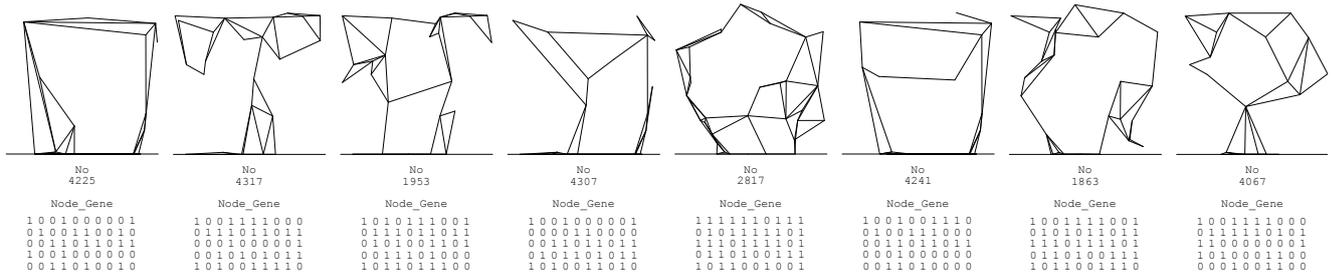


Fig. 7-6 補正係数を含めた適応度の例2

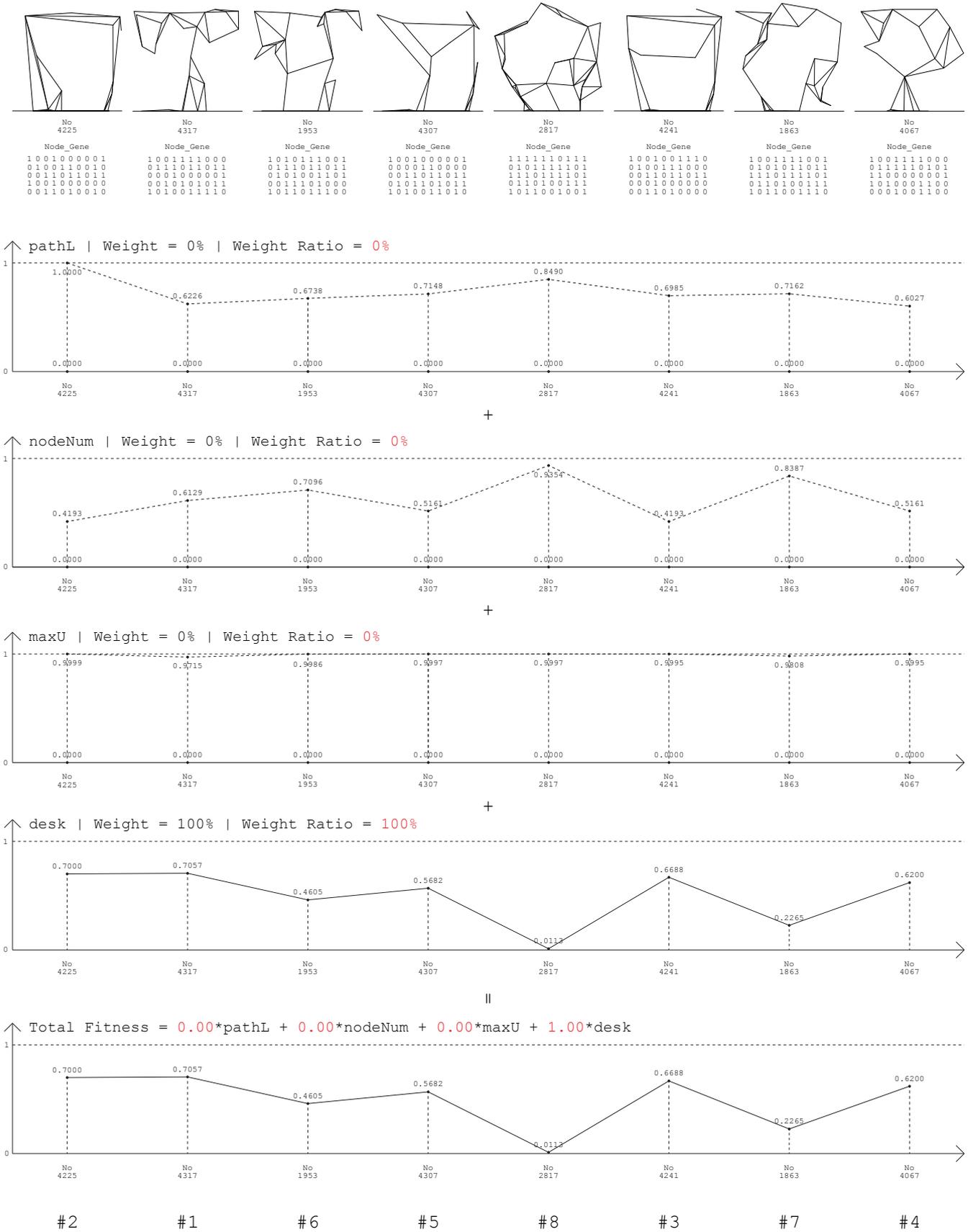


Fig. 7-7 補正係数を含めた適応度の例 3

同様に、補正係数を変更した例を Fig. 7-6 と Fig. 7-7 に示す。これらを比較すると、適応地形はユーザーの〈か=補正係数〉によって動的に変化し、それによって、〈かたち〉の順位も変化することがわかる。

(7-3 式) の関数を実際にコンピュータプログラムで実装し、この関数を呼び出すことで、補正係数 R を考慮した適応度 $Total_Fitness$ を計算することができる。

7.4 高適化のフェイズ

第 4.9.3 節で見たように、アルゴリズム化したヒューリスティクスがうまく適応地形を登ることができるならば、採用する手法はどれも構わない。ここでは、ノードやパスをコード化することが容易なことと、実装が比較的容易なこと、汎用性があることなどを理由に、対話型遺伝的アルゴリズム (Genetic Algorithm、以下 GA) [7-3] [7-4] [7-5] [7-6] を採用する。

高適化のフェイズの手順を以下に示す。

- STEP1 (7-1 式) を呼び出し、 N 個の〈かたち〉を生成する。
- STEP2 (7-3 式) により、各〈かたち〉の適応度 $Total_Fitness$ を計算する。
- STEP3 適応度が高い順に〈かたち〉をソートして表示する。
- STEP4 ユーザーが好きな〈かたち〉を 2 つ選択し親とする。
- STEP5 選択した 2 つの〈かたち〉を交叉 (後述) させ、新たな遺伝子を得る。
- STEP6 新たな遺伝子に対して、ある確率で突然変異 (後述) を行う。
- STEP7 こうして得た新たな設計変数 $Node_GENE$, $Node_Coordinate$, $Path_Matrix$ を引数として (7-1 式) から〈かたち〉を生成し、子とする。
- STEP8 子の数が指定した K 個未満ならば STEP4 に戻る。
- STEP9 子を出力
- STEP10 ユーザーが好きな回数だけ STEP2 からを繰り返す。

STEP5 の交叉の例を Fig. 7-8 に示す。親の〈かたち〉のノードは、 $Node_GENE$ として遺伝子化されている。 $Node_GENE$ のある 1 点で遺伝子列を切断し (切断する点はランダム)、片親の前半の遺伝子列と、もう片親の後半の遺伝子列を繋げることで、子の $Node_GENE$ を新たに生成する。この交叉は一点交叉と呼ばれる。

STEP6 では、全体の遺伝子 50 ビットのうち、5% 以下の確率でビットを反転し、突然変異させる。

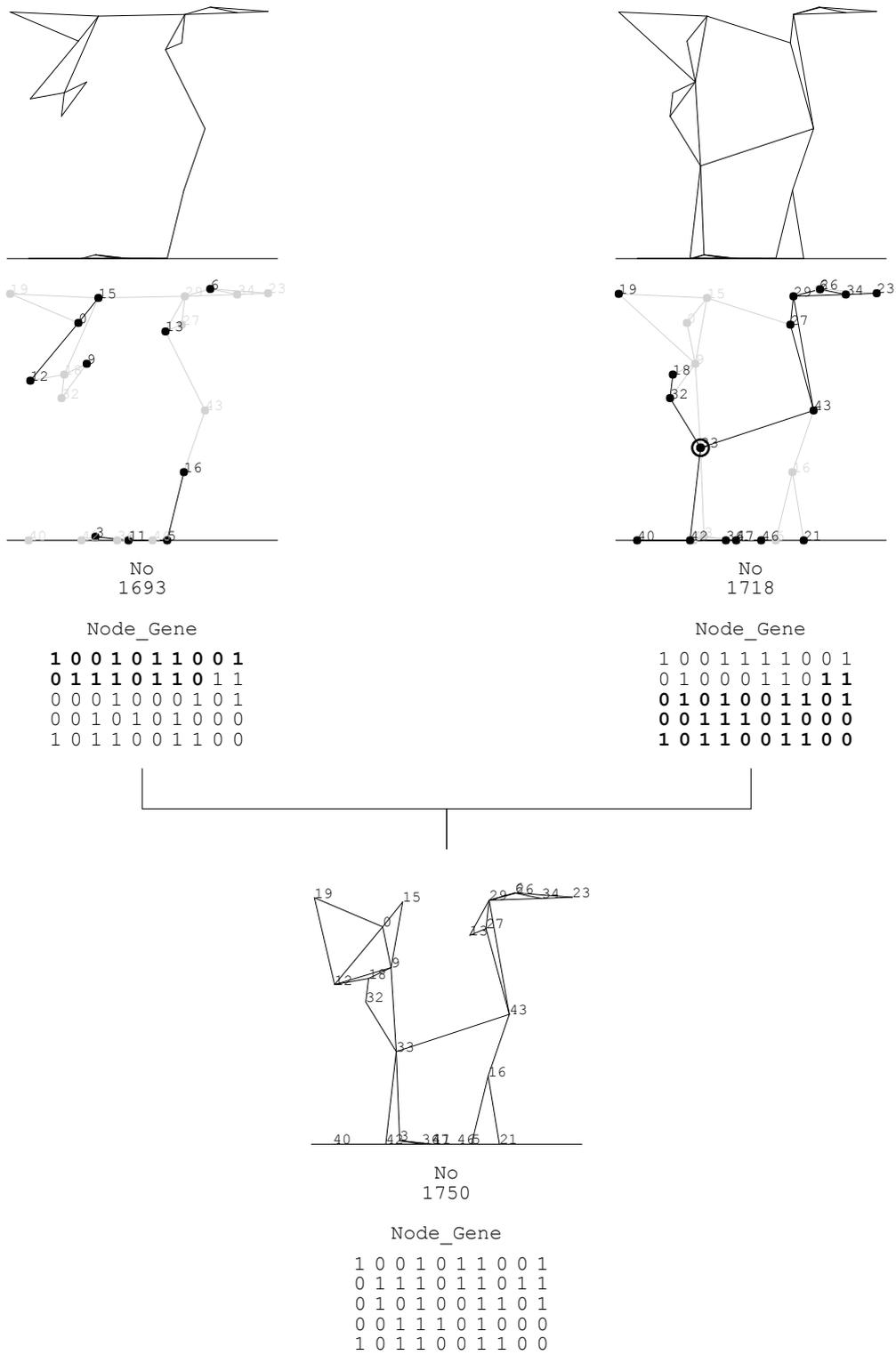


Fig. 7-8 〈かたち〉の交叉の例

継承したノードが両端にあるパスはそのまま継承する。継承したノードがもともと持っていた次数になるまで、新たにパスを追加する。その際、パス同士が交差しないように制約する。こうして新たな設計変数である *Node_GENE*, *Node_Coordinate*, *Path_Matrix* を得る。

7.5 アルゴリズムック・デザインのプロトタイプの実装

以上の「生成のフェイズ (7.2)」「評価のフェイズ (7.3)」「高適化のフェイズ (7.4)」を元に、アルゴリズムック・デザインのプロトタイプの実装を行った。Fig. 7-9 を参照しながら、下記にその概要を述べる。

- (1) ブラウザ (現状は Google Chrome のみ対応) で、<http://arkhitektome.com> [7-7] にアクセスする。
- (2) データベースに保存されている〈かたち〉の可能態。
- (3) ページを送っていけば、全ての〈かたち〉を見ることができる。2017年02月10日の時点で、10155個の〈かたち〉がデータベース化されている。
- (4) 〈かたち〉の〈かち〉— すなわち〈かたち〉に対する評価基準 — の重み付けをパラメトリックに操作できる。ノードの総数や、構造計算によって求められた変位などの「定量的な評価基準」だけでなく、机っぽい、椅子っぽいといった「定性的な評価基準」によって、〈かたち〉を検索することができる。
- (5) 例えば、犬っぽく、構造的に強く、ノードの数が小さい〈かたち〉を検索したければ、定量的な評価基準のなかの「*nodeNum*」と「*MaxU*」と「*MaxM*」を「-100」に設定し、定性的な評価基準のなかの「dog」を選択した上でその重み付けの値を「100」に設定した上で「Set」をクリック。机っぽく、構造的に強く、ノードの数が小さいという〈かち〉をよりよく満たす〈かたち〉を検索することができる。図は2014年12月11日時点での検索結果。
- (6) 父親と母親を選択し、交配させることで、両親の遺伝子を継承した子供を入力した数だけ生成することができる。ある確率で、突然変異したノードが生成されることもある。親を選択しなかった場合は、ランダムな子供が生成される。
- (7) 選択した〈かたち〉に定性的な〈かち〉を付与することができる。例えば選択した〈かたち〉が犬っぽいと思えば、「dog」を入力し、どのくらい犬っぽいかを0-1の間で入力する。
- (8) 一覧表示の〈かたち〉をクリックすれば、それぞれの個別ページを閲覧できる。
- (9) 〈かたち〉のデータ。

parents : 遺伝子を継承した父親と母親

tag : 定性的な〈かち〉 — 評価基準一覧

tag value : 定性的な〈かち〉に対する適応度

gene : ノードの遺伝子

nodesid : ノードの ID。1 はノードあり。0 はノードなし

nodePt : ノードの座標

paths : パス。始点のノード ID と終点のノード ID

drawPathCnt : ノードの次数

- (10) 〈かたち〉の〈かち〉 — すなわち〈かたち〉に対する評価基準 とその適応度。
適応度は、0 ~ 1 の実数値をとる。

pathNum : パスの総数

pathL : パスの総長さ

nodeNum : ノードの数

maxU : 最大変位

maxM : 最大モーメント

- (11) すでに入力した定性的な〈かち〉の適応度を更新したり、新たな定性的な〈かち〉を付与することができる。例えば、この〈かたち〉が犬っぽいと思えば、「dog」を入力し、どのくらい犬っぽいかを 0-100% の間で入力する。

1 <http://arkhitektome.com>

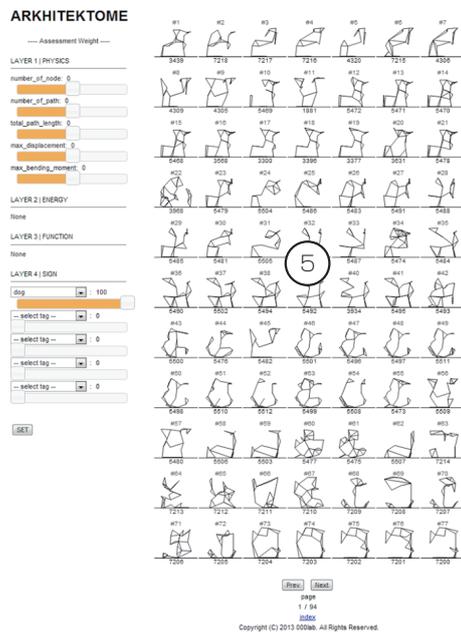
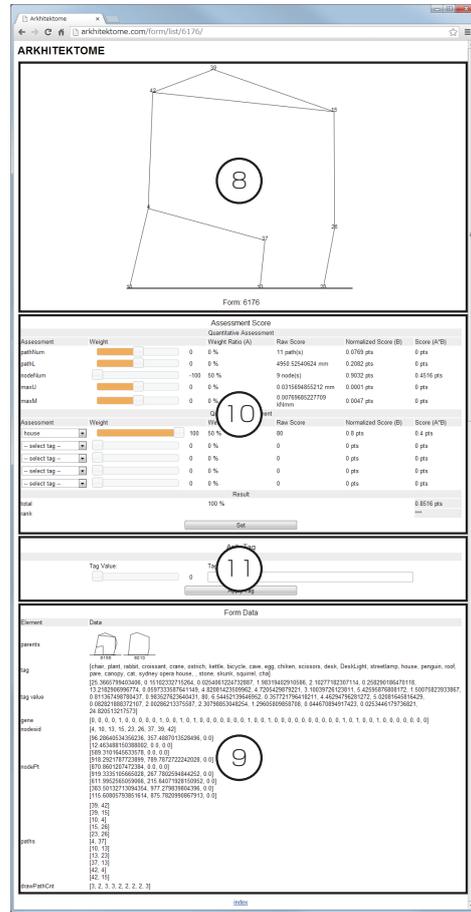
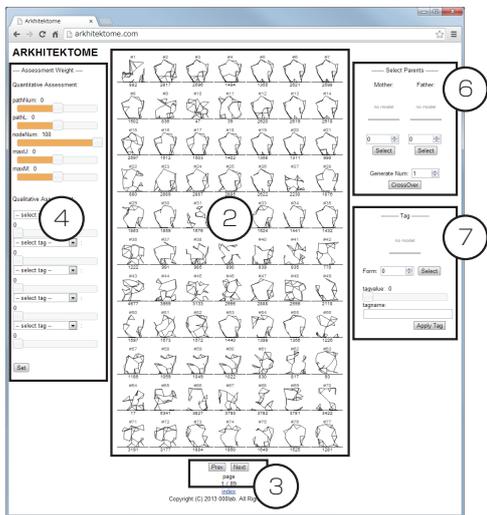


Fig. 7-9 アルゴリズムック・デザインのプロトタイプの実装

第8章

椅子への応用

本章では、椅子のデザインへの工学的な応用可能性をケーススタディによって示す。

8.1 概要

第5章で構築したアルゴリズムック・デザインのプロトタイプを、椅子のデザインへと適用する。ここでは、環境に適応した椅子を作成するシステムを **STOOLOME** と呼ぶ [8-1]。Fig. 8-1 に示すように、「生成のフェイズ」「評価のフェイズ」「高適化のフェイズ」に分けて、**STOOLOME** の詳細を述べる。

8.2 生成のフェイズ

8.2.1. 仕様

基本的な仕様について以下にまとめる。

- ・ 用途は椅子とする。
- ・ 部材はスチールの丸棒で、断面は 3 φ とする。全ての接合部は溶接による剛接合とする。
- ・ 想定使用者は 20 歳の男子とする
- ・ 身長は 20 歳の男子の平均身長である 171.65cm とする。
- ・ 体重は 20 歳の男子の平均身長である 65.63kg とする。

8.2.2. 制約条件

制約条件について以下にまとめる。

- ・ 水色：すべてのノードはこの範囲内に生成される。
- ・ 緑色：座面を構成するノードをあらかじめ入力しておく。
- ・ オレンジ色：背面を構成するノードをあらかじめ入力しておく
- ・ マゼンダ色：柱脚を構成するノードはこの範囲内に少なくともひとつ生成される。

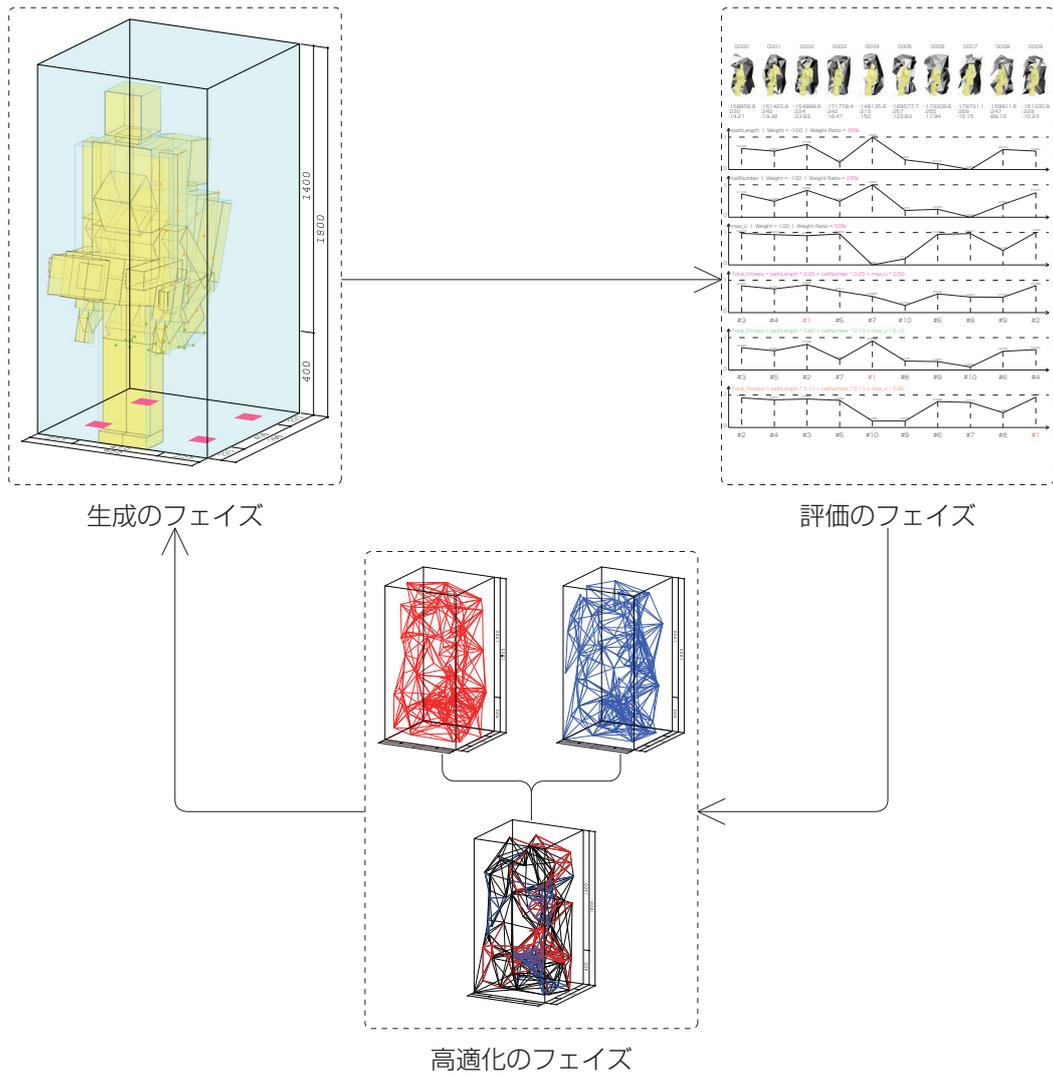


Fig. 8-1 環境に適応するための椅子を育てるための3つのフェイズ

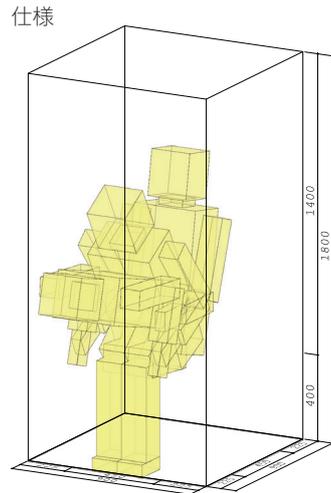
8.2.3. ノードの生成

あらかじめ入力された座面、背面のノードの他に、100個のノードをランダムに生成する。その際、マゼンダ色で示された範囲内に少なくともひとつのノードを生成する。

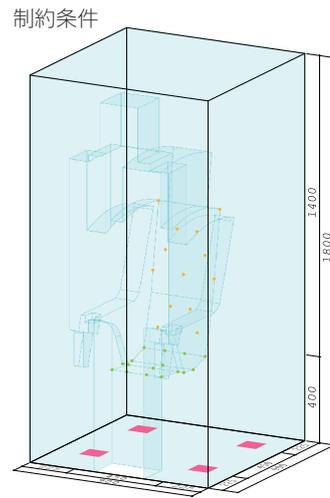
8.2.4. エッジの生成

すべてのノードを母点として、3次元ドロネー四面体を求める。すべてのドロネー四面体の内、水色の範囲から一部でもはみ出ている四面体は非表示にする。また、四面体を構成するエッジの中で、200mmを超えるエッジを持つ四面体も非表示にする。

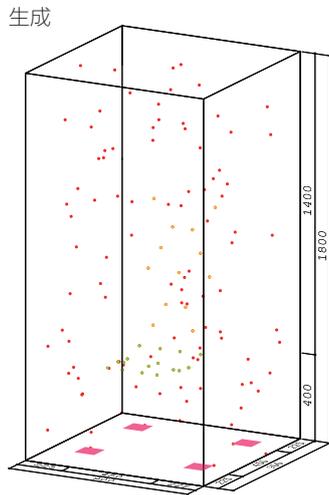
以上のフェイズを、Rhinoceros5.0という3Dモデリングソフト上において、Python言語で実装を行った。



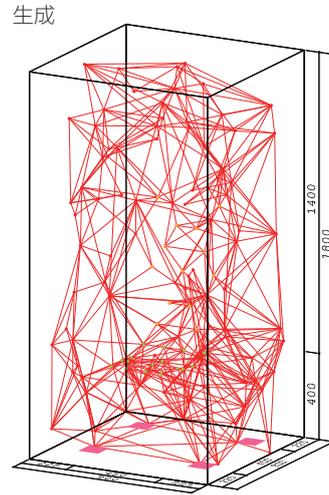
用途仕様 : 椅子
 部材仕様 : スチール 丸棒 3φ
 使用者設定 : 20歳男子平均
 身長 : 171.65cm
 体重 : 65.63kg



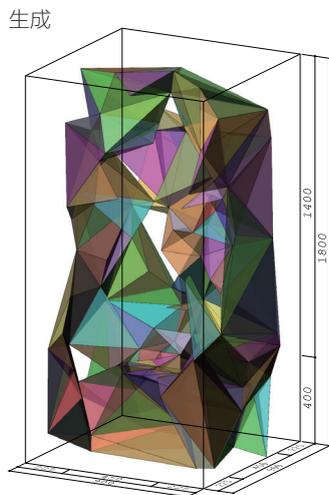
クリアランス : 人間行動範囲
 座面支持位置 : 体重負荷
 背面支持位置 : 衝撃負荷
 椅子支持位置 : 全体支持



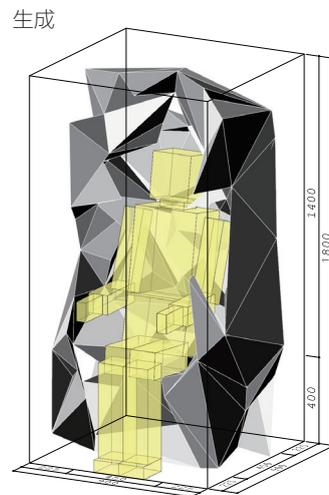
form_id : 0001
 parents : None, None
 generation_Number : 0001
 gene : 100



form_id : 0001
 parents : None, None
 generation_Number : 0001
 gene : 100



form_id : 0001
 parents : None, None
 generation_Number : 0001
 gene : 100



form_id : 0001
 parents : None, None
 generation_Number : 0001
 gene : 100

Fig. 8-2 STOOLOME の生成のフェイズ

8.3 評価のフェイズ

STOOLOME で取り扱う〈かち=評価〉を以下にまとめる。

- ・ 適応度 $m_1 = pathLength$: パスの総長さ
- ・ 適応度 $m_2 = cellNumber$: セルの総数
- ・ 適応度 $m_3 = max_U$: ノードの最大変位

上記すべての評価において、すべての〈かたち〉の中での最大値と最小値を保持しておくことで、各評価を0から1の間の値へと正規化する。なお、構造計算に関しては、マトリクス変位法による応力解析プログラムを自作し構造計算を行う。

上記のそれぞれの適応度に応じた補正係数を下記にまとめる。

- ・ 補正係数 $r_1 : pathLength$: パスの総長さ
- ・ 補正係数 $r_2 : cellNumber$: セルの総数
- ・ 補正係数 $r_3 : max_U$: ノードの最大変位

例えば、 $r_1=0.25$, $r_2=0.25$, $r_3=0.50$ としたときの適応度を Fig. 8-3-1 に示す。また同じ〈かたち〉で、 $r_1=0.10$, $r_2=0.10$, $r_3=0.80$ としたときの適応度を Fig. 8-3-2 に示す。これらを比較すると、適応地形はユーザーの〈か=補正係数〉によって動的に変化し、それによって、〈かたち〉の順位も変化することがわかる。

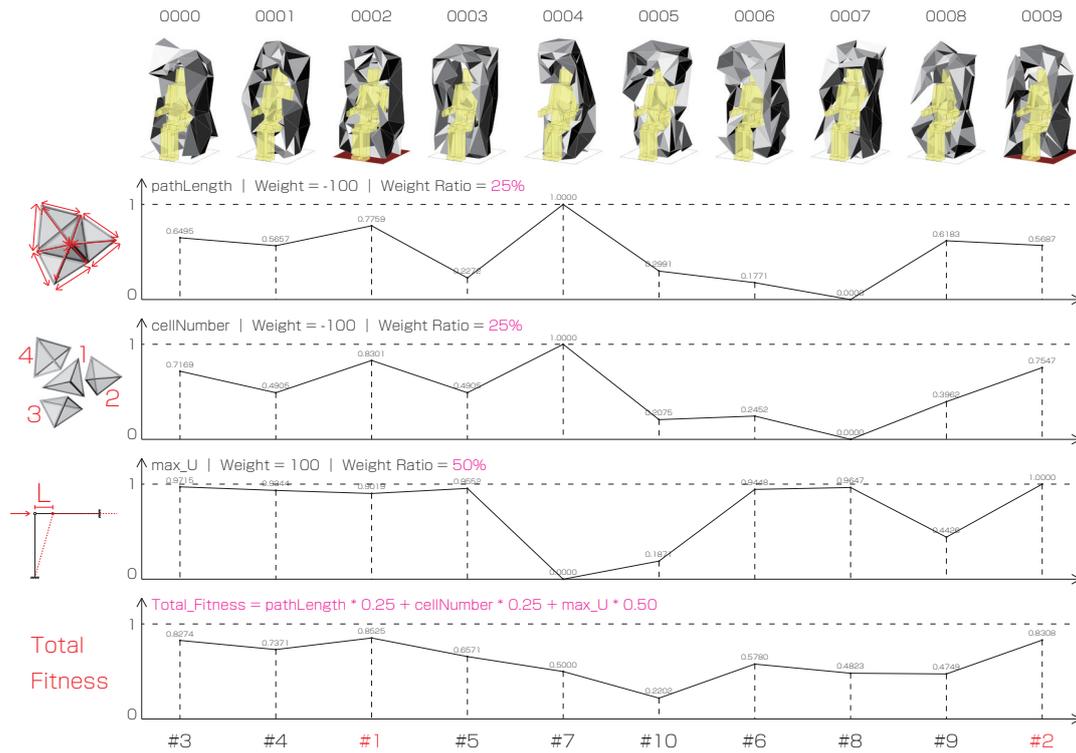


Fig. 8-3-1 補正係数を含めた適応度の例1

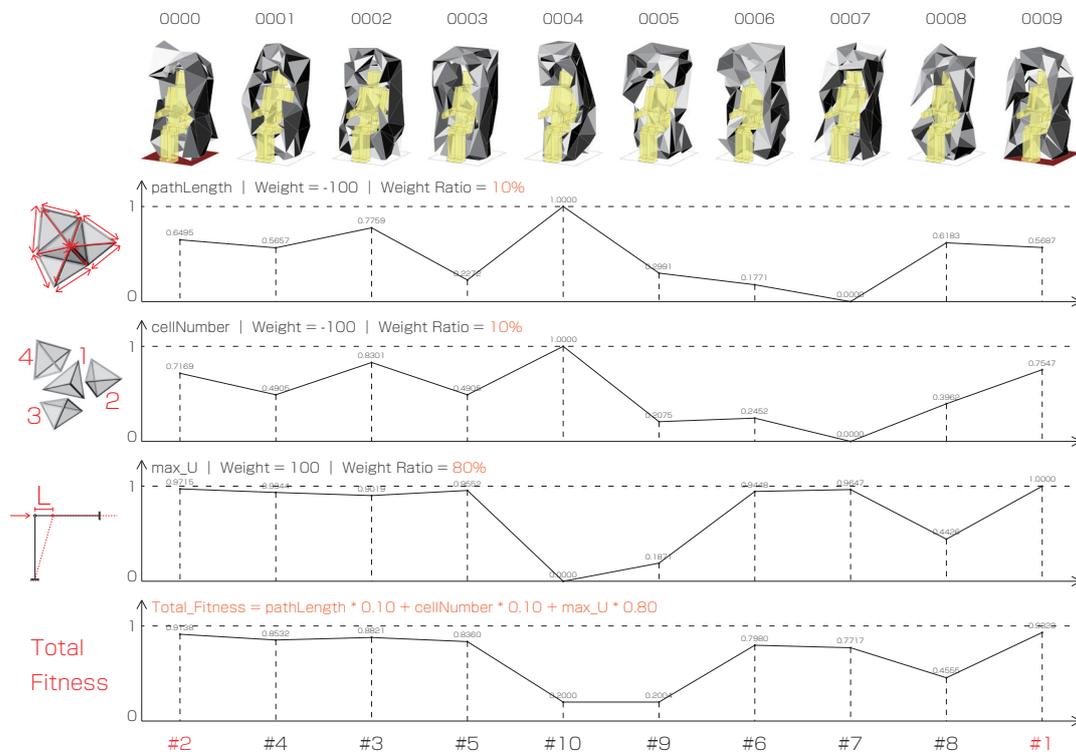


Fig. 8-3-2 補正係数を含めた適応度の例2

8.4 高適化のフェイズ

STOOLOME においても、高適化を行うためのヒュリスティクスとして、遺伝的アルゴリズム (Genetic Algorithm、以下 GA) を採用する。

高適化のフェイズの手順を以下に示す。(Fig. 8-4-1)

- STEP1 生成のフェイズにより、10 脚の椅子を生成する。
- STEP2 評価のフェイズにより、各〈かたち〉の適応度 *Total_Fitness* を計算する。
- STEP3 適応度が高い順に〈かたち〉をソートして表示する。
- STEP4 適応度が高い椅子を2つ選択し親とする。
- STEP5 選択した2つの椅子を交叉(後述)させ、新たな遺伝子を得る。
- STEP6 新たな遺伝子に対して、ある確率で突然変異(後述)を行う。
- STEP7 こうして得た新たな設計変数を引数として椅子を生成し、子とする。
- STEP8 子の数が指定した10個未満ならばSTEP4に戻る。
- STEP9 子を出力
- STEP10 30世代未満ならば、STEP2からを繰り返す。以上ならば終了する。

STEP5の交叉の例をFig. 8-4-2に示す。親の椅子のノードは、プロトタイプと同様に *Node_GENE* として遺伝子化されている。*Node_GENE* のある1点で遺伝子列を切断し(切断する点はランダム)、片親の前半の遺伝子列と、もう片親の後半の遺伝子列を繋げることで、子の *Node_GENE* を新たに生成する。この交叉は一点交叉と呼ばれる。

STEP6では、全体の遺伝子50ビットのうち、5%以下の確率でビットを反転し、突然変異させる。こうして新たな設計変数である *Node_GENE*, *Node_Coordinate*, *Path_Matrix* を得る。

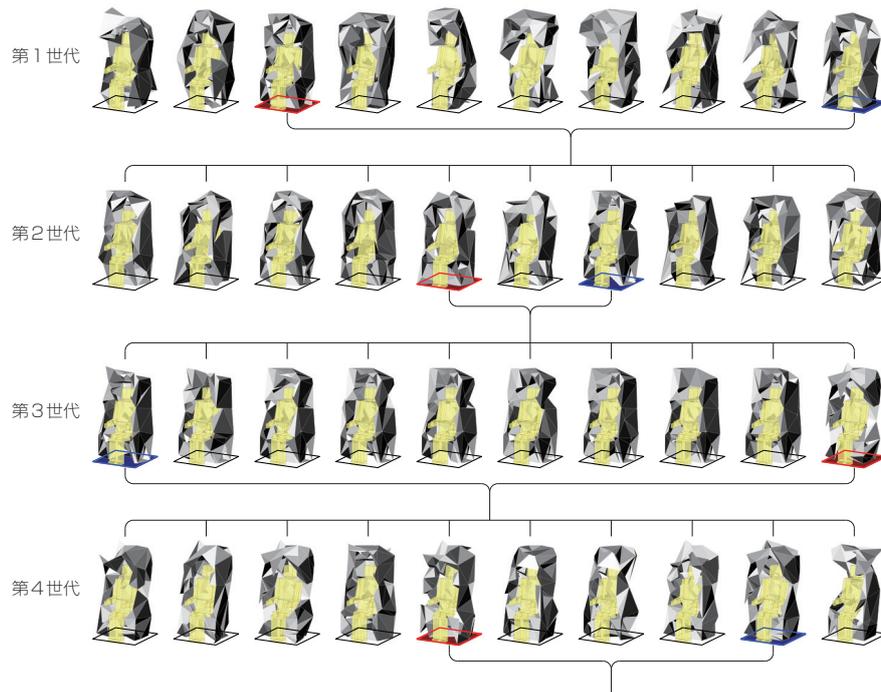


Fig. 8-4-1 椅子の高適化のフェイズ

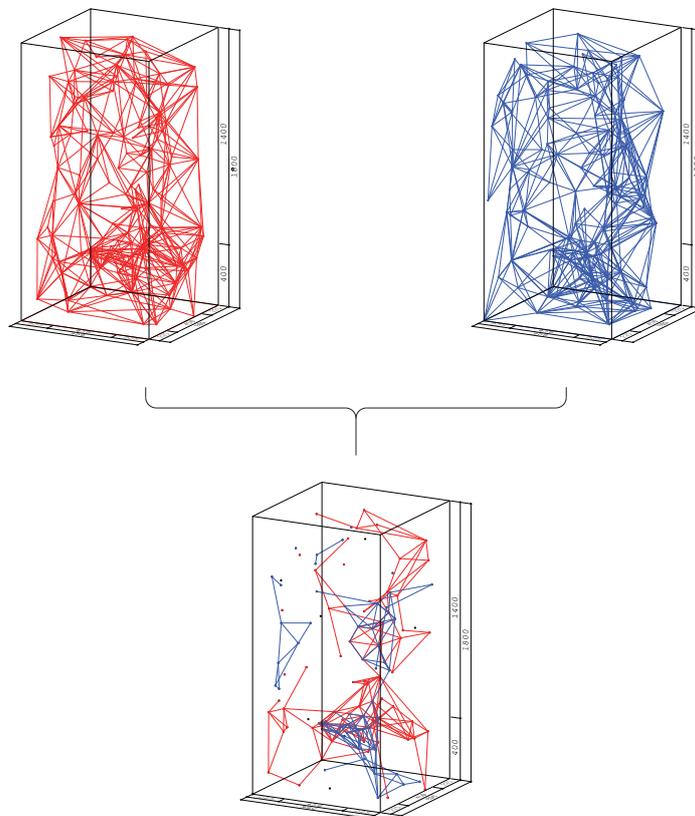


Fig. 8-4-2 椅子の交叉の例

8.5 実験

以上の「生成のフェイズ (8.2)」「評価のフェイズ (8.3)」「高適化のフェイズ (8.4)」を元に、STOOLOME の実装を行い、生成した椅子を3つの環境に適応させた。

8.5.1. 環境1で椅子を育てる

1つ目の環境を Fig. 8-5-1 に示す。生成のフェイズで述べた仕様や制約条件と同じであるが、背面を構成する各ノードに対して、後ろ向きに 10N の外力を与えた。また、補正係数は $r_1=0.20$, $r_2=0.00$, $r_3=0.80$ と設定した。

Fig. 8-5-2 に、各世代で最も適応度が高かった椅子の適応度の推移を示す。部材長さが順調に減っているのに対して、構造が上手く高適化されていないために、*Total_Fitness* もあまり上手く高適化されていない。これは背面のノードの外力が小さすぎたためと考えられる。

30 世代目で最も環境1に適応した椅子を Fig. 8-5-3 に示す。

環境設定_1

物理的環境設定

用途仕様 : 椅子
 部材仕様 : スティール丸棒 3φ
 使用者設定 : 20 歳男子平均
 身長 : 171.65cm
 体重 : 65.63kg
 負荷荷重 : y 方向 10kg

クリアランス : 人間行動範囲
 座面支持位置 : 体重負荷
 背面支持位置 : 衝撃負荷
 椅子支持位置 : 全体支持

定性的環境設定

number_of_Generation : 30
 gene : 100
 children_Number : 100
 path_Length : 20%
 cell_Number : 0%
 max_U : 80%

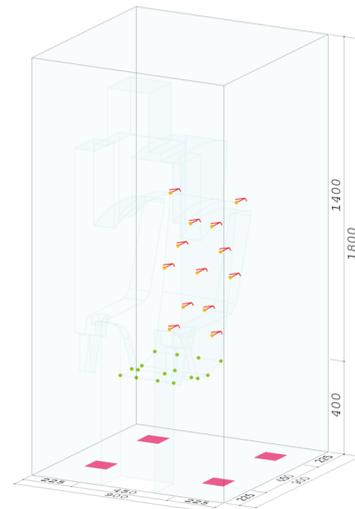


Fig. 8-5-1 環境1

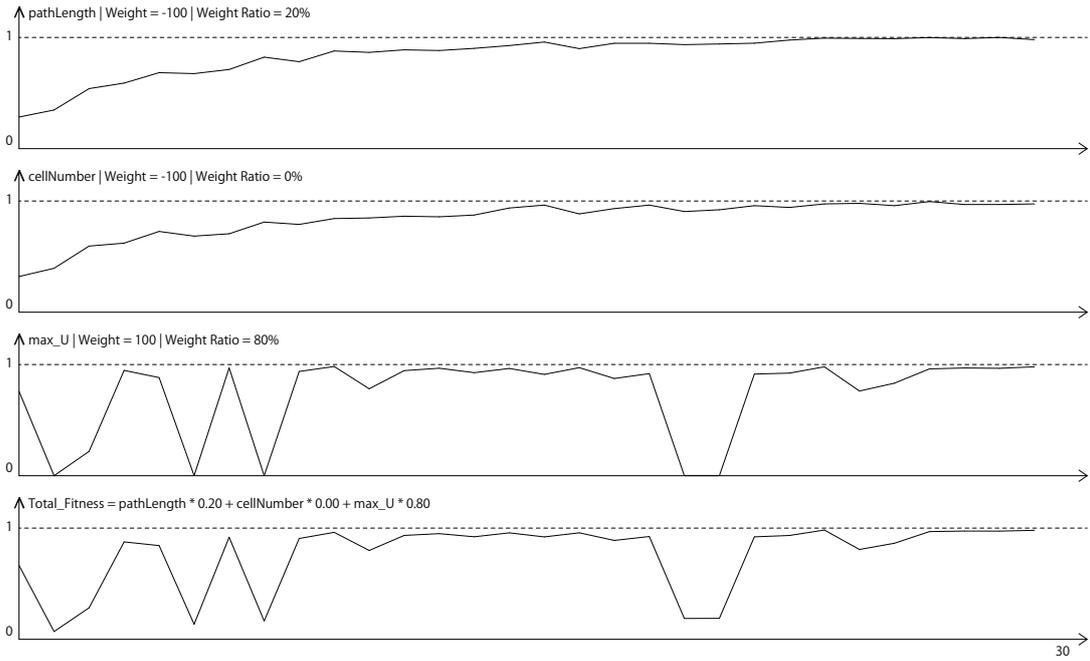


Fig. 8-5-2 適応度の推移

環境設定_1

form_id :2916
 generation :29
 parents :2811,2892
 path_Length :67539.117
 cell_Number :102
 max_U :3.0050918

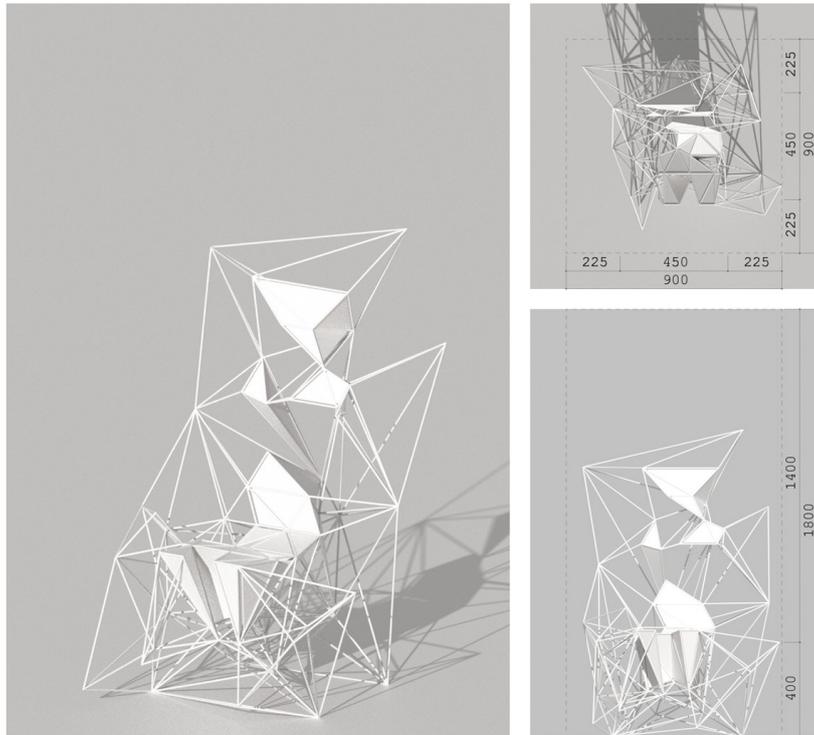


Fig. 8-5-3 30 世代目で最も環境 1 に適応した椅子

8.5.2. 環境2で椅子を育てる

2つ目の環境を Fig. 8-5-4 に示す。生成のフェイズで述べた仕様や制約条件と同じであるが、背面を構成する各ノードに対して、後ろ向きに 300N の外力を与えた。また、補正係数は $r_1=0.20$, $r_2=0$, $r_3=0.8$ と設定した。

Fig. 8-5-5 に、各世代で最も適応度が高かった椅子の適応度の推移を示す。変位が最後に少し適応度が乱高下していることを考慮すると、一世代の生成個体数と世代数ともに、まだファインチューニングされてはいえない。しかしながら、全体的には順調に適応度が上がっていることがわかる。

30 世代目で最も環境2に適応した椅子を Fig. 8-5-6 に示す。Fig. 8-5-3 の椅子と比較して、背面の部材が多く、背後に膨らんでいることがわかる。また全体の重心が低く安定したかたちに適応していることが見て取れる。

環境設定_2

物理的環境設定

用途仕様 : 椅子
 部材仕様 : スティール丸棒 3φ
 使用者設定 : 20歳男子平均
 身長 : 171.65cm
 体重 : 65.63kg
 負荷荷重 : y方向 300kg

クリアランス : 人間行動範囲
 座面支持位置 : 体重負荷
 背面支持位置 : 衝撃負荷
 椅子支持位置 : 全体支持

定性的環境設定

number_of_Generation : 30
 gene : 100
 children_Number : 100
 path_Length : 20%
 cell_Number : 0%
 max_U : 80%

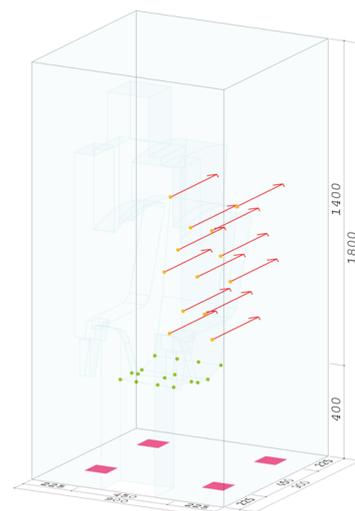


Fig. 8-5-4 環境2

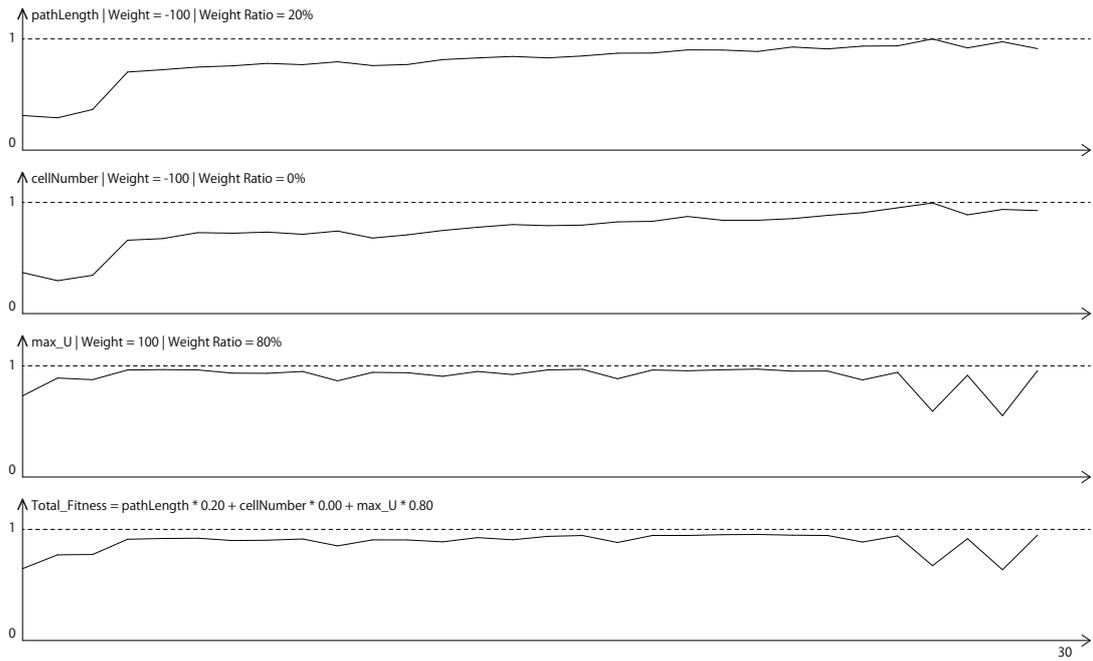


Fig. 8-5-5 適応度の推移

環境設定_2

form_id :2909
 generation :29
 parents :2871,2816
 path_Length :64625.305
 cell_Number :92
 max_U :6.3834375

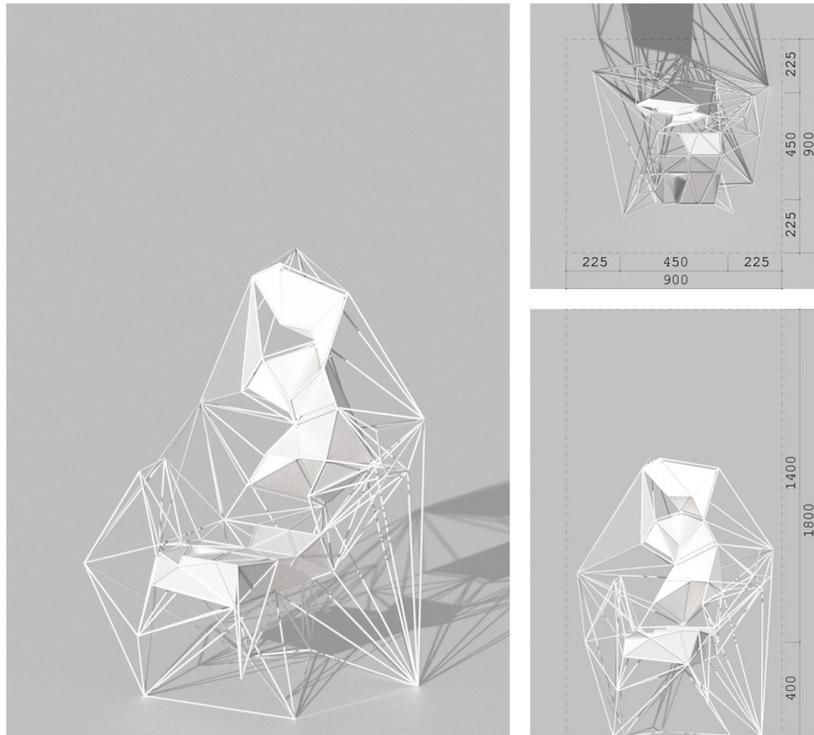


Fig. 8-5-6 30 世代目で最も環境 2 に適応した椅子

8.5.3. 環境3で椅子を育てる

3つ目の環境を Fig. 8-5-7 に示す。生成のフェイズで述べた仕様や制約条件と同じであるが、背面を構成する各ノードに対して、右から左に 300N の外力を与えた。また、補正係数は $r_1=0.20$, $r_2=0.00$, $r_3=0.80$ と設定した。

Fig. 8-5-8 に、各世代で最も適応度が高かった椅子の適応度の推移を示す。順調に適応度が上がり、環境3に適応しているのがわかる。

30 世代目で最も環境3に適応した椅子を Fig. 8-5-9 に示す。Fig. 8-5-6 の椅子と比較して、座って左側の部材が多く、また柱脚部も左側に大きく張り出していることから、外力に抵抗しようとしていることがわかる。

環境設定_3

物理的環境設定

用途仕様 : 椅子
 部材仕様 : スティール丸棒 3φ
 使用者設定 : 20歳男子平均
 身長 : 171.65cm
 体重 : 65.63kg
 負荷荷重 : -x方向 300kg

クリアランス : 人間行動範囲
 座面支持位置 : 体重負荷
 背面支持位置 : 衝撃負荷
 椅子支持位置 : 全体支持

定性的環境設定

number_of_Generation : 30
 gene : 100
 children_Number : 100
 path_Length : 20%
 cell_Number : 0%
 max_U : 80%

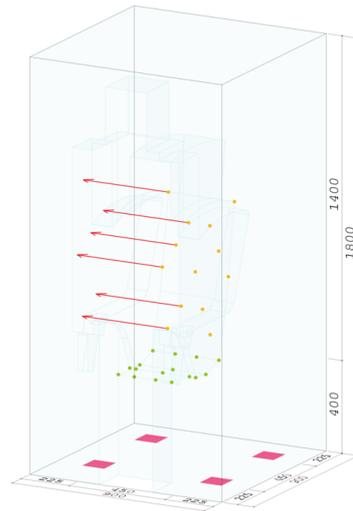


Fig. 8-5-7 環境3

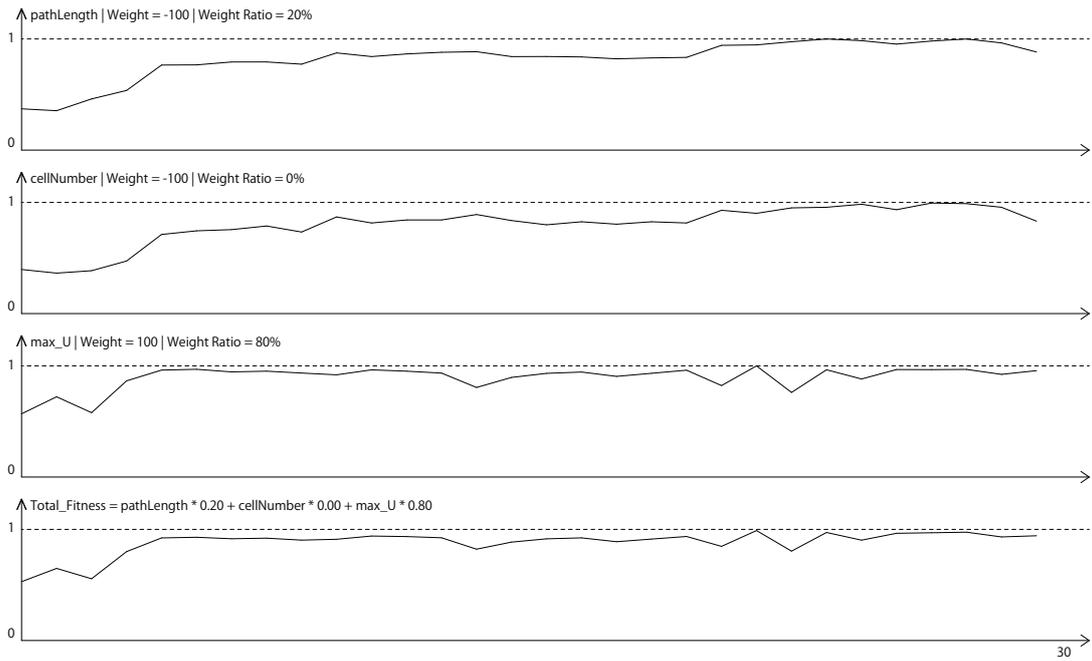


Fig. 8-5-8 適応度の推移

環境設定_3

form_id :2978
 generation :29
 parents :2869,2884
 path_Length :91721.706
 cell_Number :145
 max_U :6.3863430

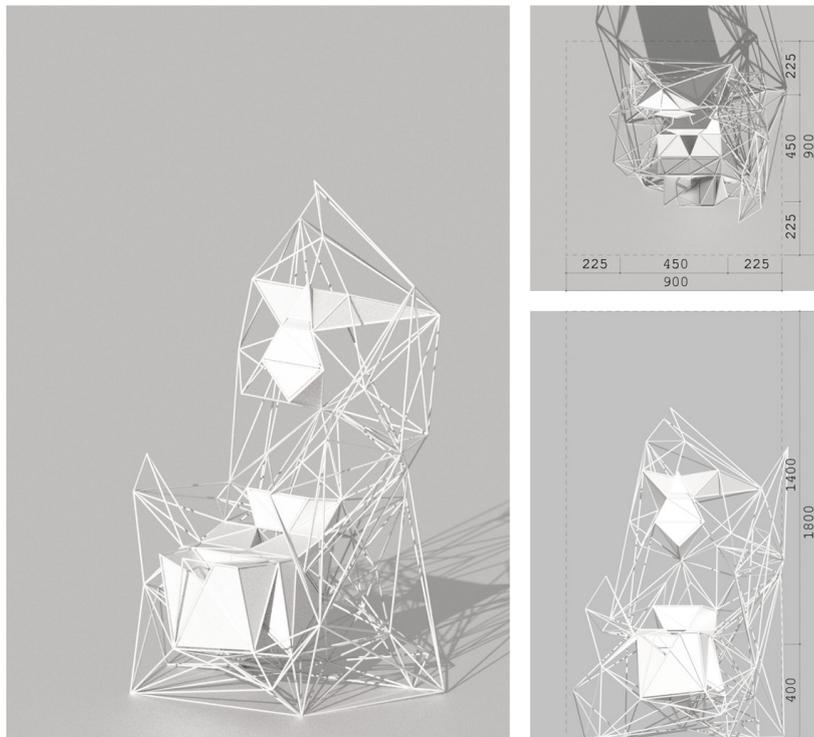


Fig. 8-5-9 30 世代目で最も環境 3 に適応した椅子

要旨

目次

序章

第1部

第2部

結章

後付

第9章

住宅への応用

本章では、住宅のデザインへの工学的な応用可能性をケーススタディによって示す。

9.1 概要

第7章で構築したアルゴリズムック・デザインのプロトタイプを、住宅のデザインへと適用する。ここでは、環境に適応した住宅を作成するシステムを **ARKHITEKTOME** と呼ぶ [9-1][9-2]。Fig. 9-1 に示すように、「生成のフェイズ」「評価のフェイズ」「最適化のフェイズ」に分けて、**ARKHITEKTOME** の詳細を述べる。

9.2 生成のフェイズ

本節では、**ARKHITEKTOME** の生成のフェイズについて述べる。以下に生成のフェイズの手順をまとめる。

- STEP01 与えられた隣接グラフを入力する
- STEP02 敷地／最大ボリューム／モジュールを入力する
- STEP03 入力された隣接グラフを満たすトポロジカルグリッドを計算する
- STEP04 トポロジカルグリッドに割当可能な寸法を計算する
- STEP05 入力された隣接グラフを満たす開口部を計算する
- STEP06 境界条件を計算する
- STEP07 積載荷重を計算する
- STEP08 応力と変位を計算する
- STEP09 屋根の境界線を計算する
- STEP10 真隅寄棟屋根を計算する
- STEP11 ワイヤフレームモデルを生成する
- STEP12 ビルディングエレメントを生成する
- STEP13 図面を出力する

9.2.1 STEP01：与えられた隣接グラフを入力する (Fig. 9-2-1)

クライアントあるいは設計者が満たして欲しい空間相互の隣接関係を入力する。入力には Rhinoceros5.0 という 3D モデリングソフトを用いた。必要な機能単位空間を閉じたサーフェイスで描く。描くべき機能単位空間は全部で三種類ある。ひとつ目は必要な室内の諸室である。図では濃いグレーの背景で描かれている。このときサーフェイスの面積が、必要な機能単位空間の面積の目安となる。2つ目は必要な外部空間である。図では、点線の外形線で描かれている。外部空間の面積は面積割には反映され

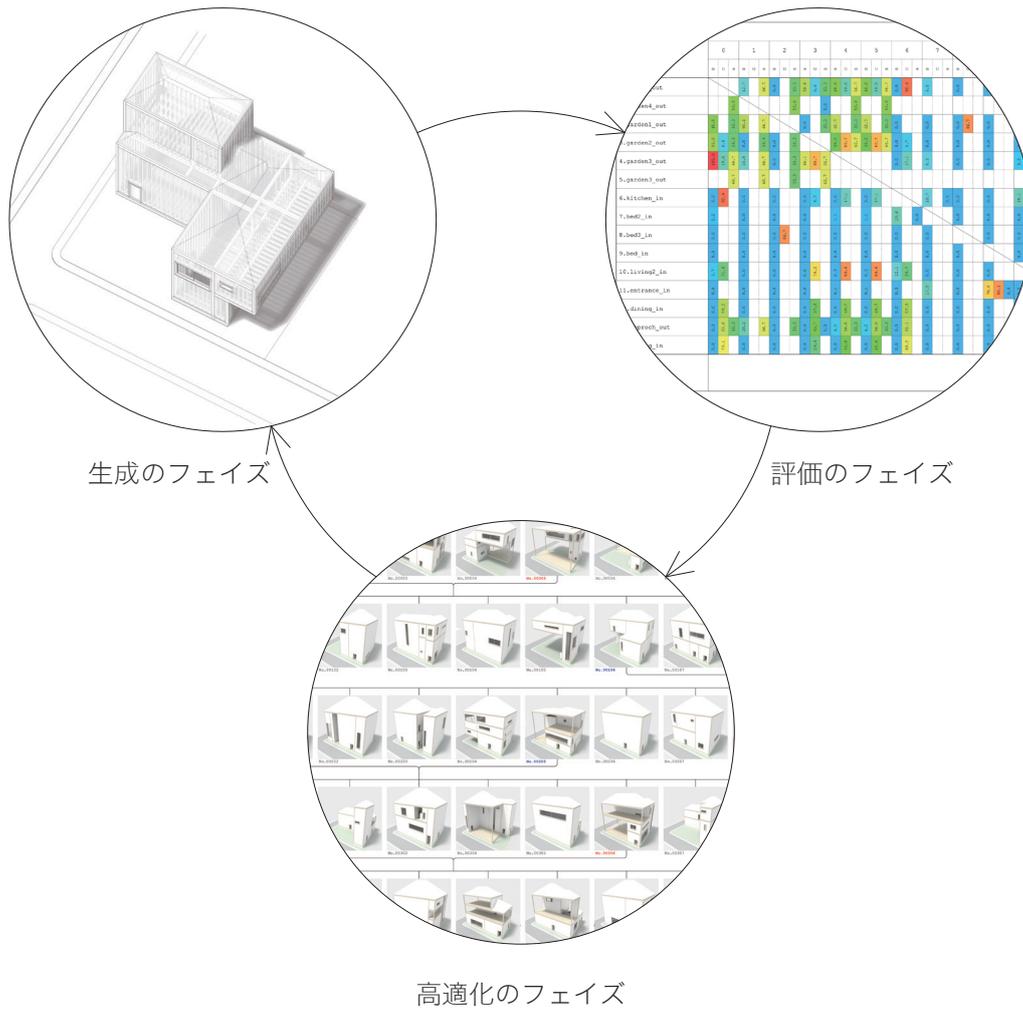


Fig. 9-1 環境に適応したを育てるための3つのフェイズ

ない。3つ目は方角である。考慮したい方角（東西南北上下）を同じく閉じたサーフェースで描く。図では薄いグレーで描かれている。サーフェースを描いた後、隣接して欲しい単位空間同士を交差するように移動させる。これらのサーフェースを選択し、プログラムの中に読み込むことで、隣接グラフを作成する。

9.2.2 STEP02：敷地／最大ボリューム／モジュールを入力する (Fig. 9-2-2)

敷地の形状を同じく Rhinoceros を用いて描く。図の例では、2つの道路に接道した角地の敷地とした。敷地内における住宅の最大ボリュームは、幅 12m 奥行き 9m 高さ 9m とした。使用する最小モジュールは 300mm とした。最大ボリュームの直方体を選択することで、プログラムの中に読み込むことができる。

9.2.3 STEP03：入力された隣接グラフを満たす直方体分割図を計算する (Fig. 9-2-3)

外形が直方体で、機能的な単位空間もそれぞれ直方体である建築物を想定して、壁や床、天井などのビルディングエレメントを抽象化すると、直方体がいかに隣接しているかを示すモデルが得られる。このモデルは、外形の直方体をさらにいくつかの小直方体に分割する形をしているので、直方体分割図と呼ばれる [9-3]。また、このような建築物の空間構成を、与えられた様々な要求に基づいて、自動的に決定する問題を室配置問題と呼ぶ [9-4]。

筆者は、中野の提案した長方形分割図の列挙アルゴリズム [9-5] を拡張し、逆探索法 [9-6] を用いた直方体分割図を列挙するアルゴリズムを提案した [9-7]。逆探索法を用いた列挙アルゴリズムに関しては第 10 章で詳しく説明する。

このアルゴリズムを拡張することで、入力された隣接グラフを満たす直方体分割図を得ることができる。STEP01 において外部空間として入力した直方体は点線で表す。

9.2.4 STEP04：トポロジカルグリッドに割当可能な寸法を計算する (Fig. 9-2-4)

STEP03 で得た直方体分割図に寸法を割り付ける。その際、STEP01 で入力した室内書室の床面積に近づくように高適化を行う。ここでは焼きなまし法 (Simulated

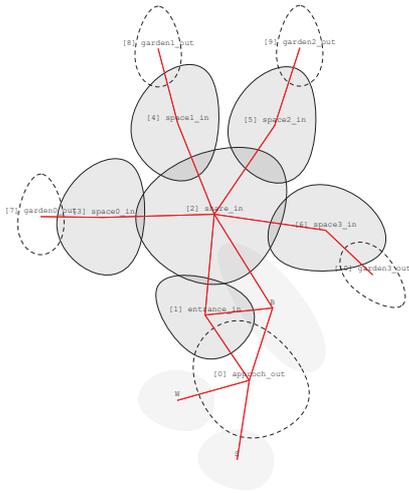


Fig. 9-2-1 STEP01 : 与えられた隣接グラフを入力する

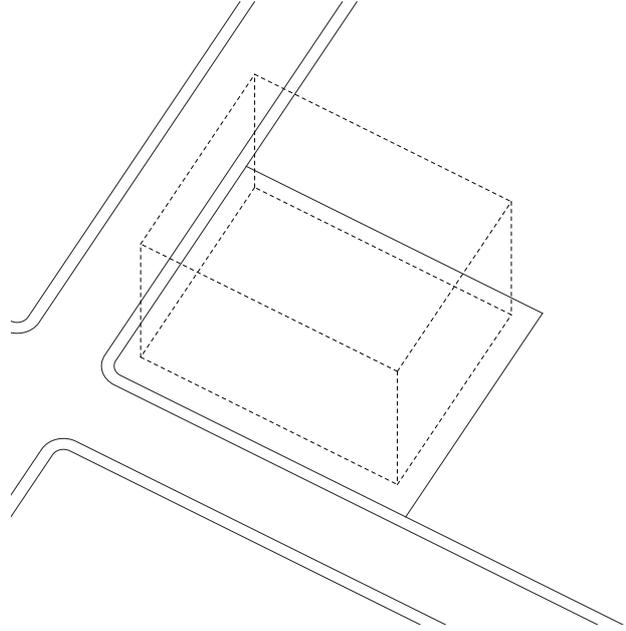
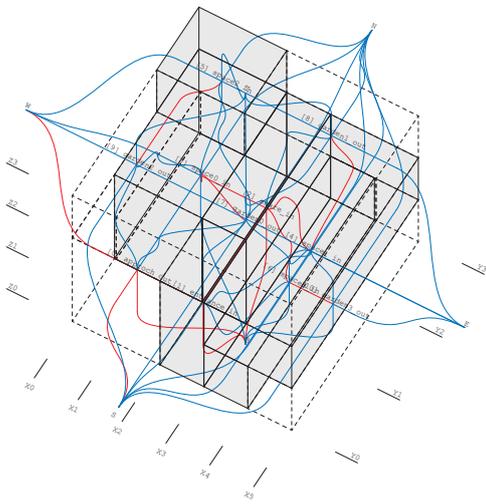


Fig. 9-2-1 STEP02 : 敷地/最大ボリューム/モジュールを入力する



建築の遺伝子 = EX0T0GY1T0DX0+OPX1T0HZ4T0B20T0FY0T1GX0+1EX0T0BX0T0

Fig. 9-2-3 STEP03 : 入力された隣接グラフを満たす直方体分割図を計算する

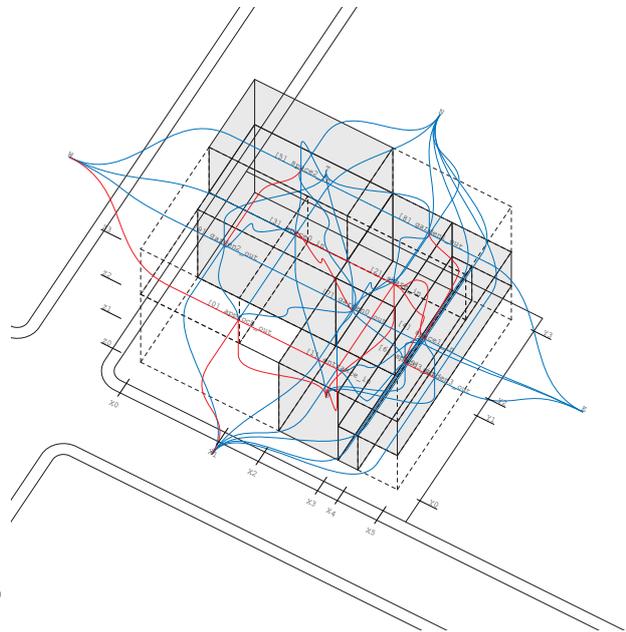


Fig. 9-2-4 STEP04 : トポロジカルグリッドに割当可能な寸法を計算する

Annealing/SA) を用いて最適化を行った。その際、STEP01 で入力した隣接グラフに基づき、隣接している諸室間は少なくとも幅 900mm 以上、高さ 2100 以上で接していなければならないという制約条件を設ける。

9.2.5 STEP05 : 入力された隣接グラフを満たす開口部を計算する (Fig. 9-2-5)

STEP01 で入力した隣接グラフに基づき、隣接している諸室間のインターフェイスに導線を確保するための開口部を生成する。水平方向での隣接の場合は、開口幅の最小寸法は 900mm とし、開口高さは 2100mm とする。鉛直方向の場合は、開口幅の最小寸法は 900mm とし、開口高さは 900mm とする。室内諸室は、直方体の 6 面の内、少なくとも 2 面は外部に面していなければならないという制約を設け、開口をランダムに空ける。その際、開口幅の寸法は 600mm 以上とし、開口高さは 600mm 以上とする。

9.2.6 STEP06 : 境界条件を計算する (Fig. 9-2-6)

構造計算が行えるように、通し柱、通し梁を自動的に定める。接合部はすべてピン接合とする。

9.2.7 STEP07 : 積載荷重を計算する (Fig. 9-2-7)

長期の積載荷重を自動的に設定する。小梁は長方形の短辺方向に架け、ピッチを 300mm とする。ここでは、短期荷重は考慮しない。

9.2.8 応力と変位を計算する (Fig. 9-2-8)

マトリクス変位法を実装し、応力と変位を求める。柱の断面寸法は 105 角、梁は 105x300 とする。柱梁材は、米松集成材 E105-F300 を想定する。

9.2.9 屋根の境界線を計算する (Fig. 9-2-9)

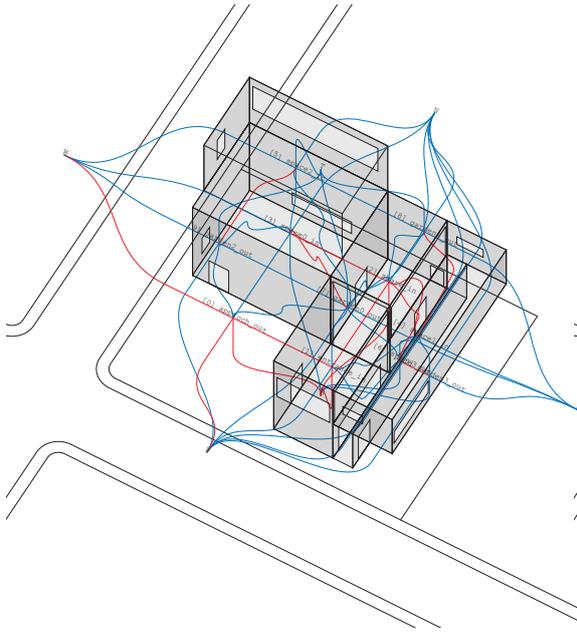


Fig. 9-2-5 STEP05 : 入力された隣接グラフを満たす開口部を計算する

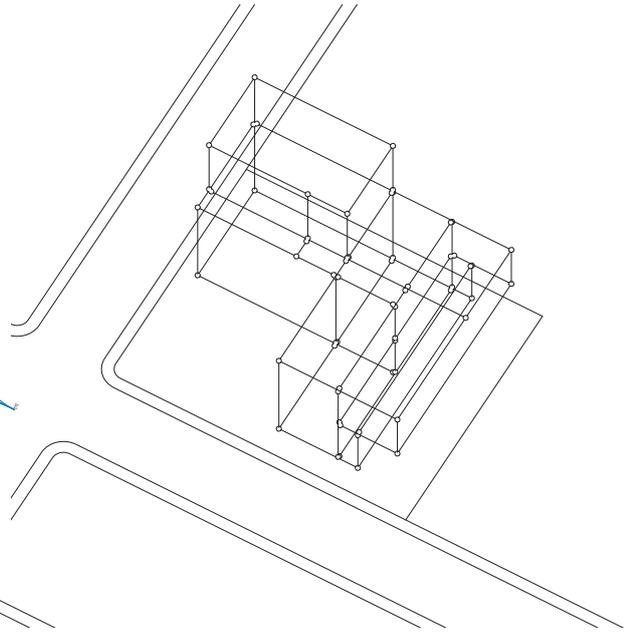


Fig. 9-2-6 STEP06 : 境界条件を計算する

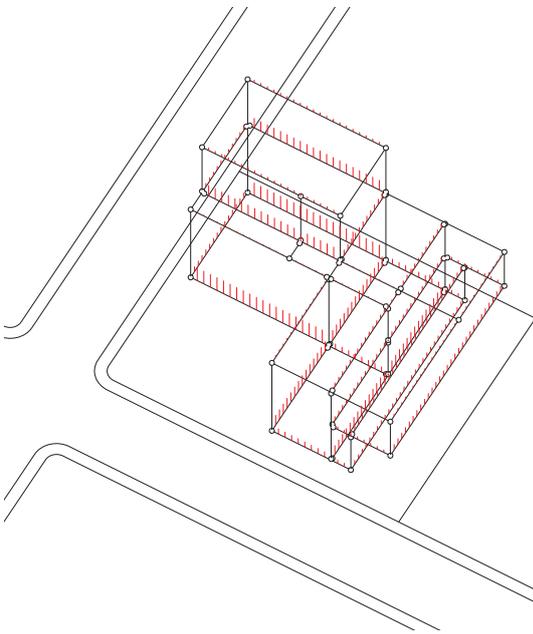


Fig. 9-2-7 STEP07 : 積載荷重を計算する

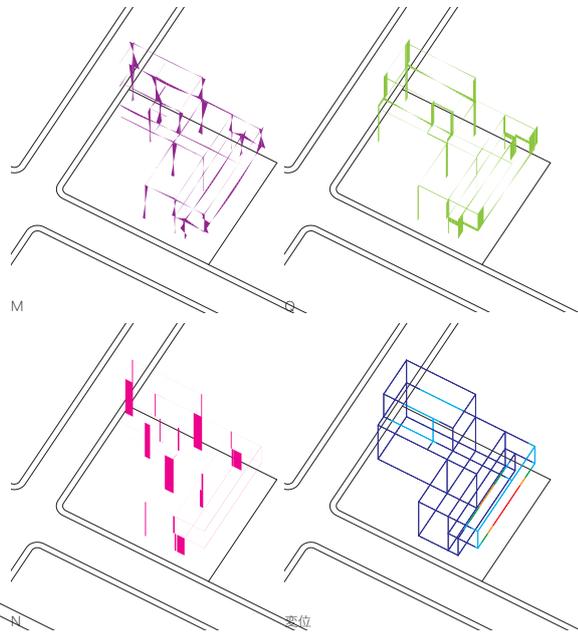


Fig. 9-2-8 STEP08 : 応力と変位を計算する

各室内諸室の上方向隣接関係を調べ、隣接する単位空間が外部だった場合、そのインターフェイスを屋根面とする。ここでは、多角形の屋根の外形を 300mm のグリッドに区切る。さらにそのグリッドのひとつの正方形セルを十字と対角によって 8 つに区切る。そのようにして得られた頂点をノード、線分をエッジとしてグラフを作成する。すべてのノードに 0 のラベルを振っておく。

9.2.10 真隅寄棟屋根を計算する (Fig. 9-2-10)

屋根は真隅寄棟屋根とする。真隅寄棟屋根は外形が矩形の多角形ならば、矛盾なく屋根を架けられることが知られている。

本研究では、従来提案されていた真隅寄棟屋根の生成アルゴリズムとは異なる、新たなアルゴリズムを用いる。

外周部のノードからスタートし、エッジで接続されたノードを随時辿り、辿ったステップ数の中で一番小さいステップ数をノードのラベルとする。すべてのノードを辿り終わったら、150mm x ステップ数だけノードを鉛直方向に移動する。そうすると、真澄寄棟屋根の形状を得ることができる。

9.2.11 ワイヤーフレームモデルを生成する (Fig. 9-2-11)

以上により、厚みを持たないワイヤーフレーム状のモデルを生成する。

9.2.12 ビルディングエレメントを生成する (Fig. 9-2-12)

基礎、土台、柱、梁、開口、間柱、根太、床板、垂木、屋根野地板などのビルディングエレメントを生成する。

9.2.13 図面を出力する

生成された 3D モデルを用いて、外観パース、内観パースを自動レンダリングし、さらに平面図、断面図、立面図、構造図などを自動生成・自動レイアウトした上で出力する。

Fig. 9-2-13 から Fig. 9-2-20 まで出力される図面の例を示す。

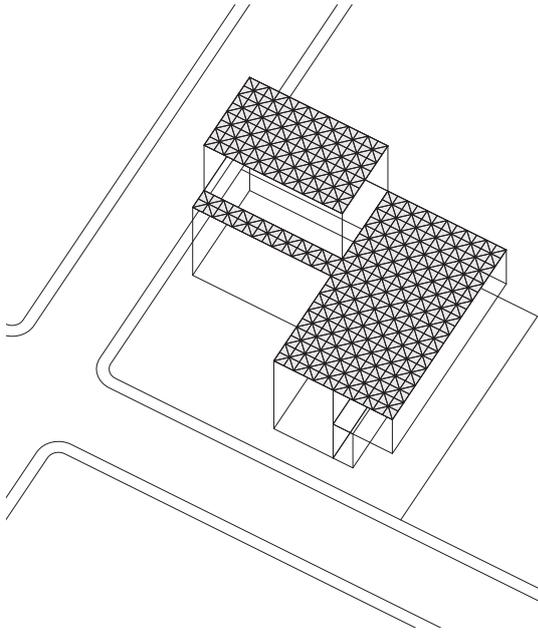


Fig. 9-2-9 STEP09 : 屋根の境界線を計算する

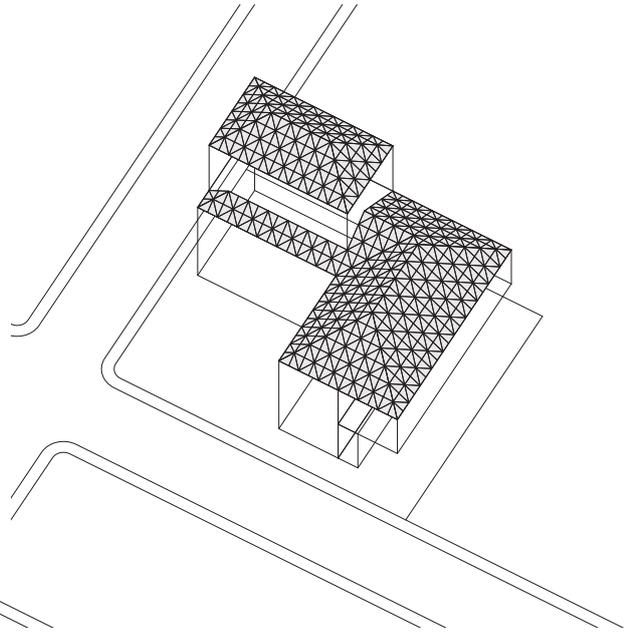


Fig. 9-2-10 STEP10 : 真隅寄棟屋根を計算する

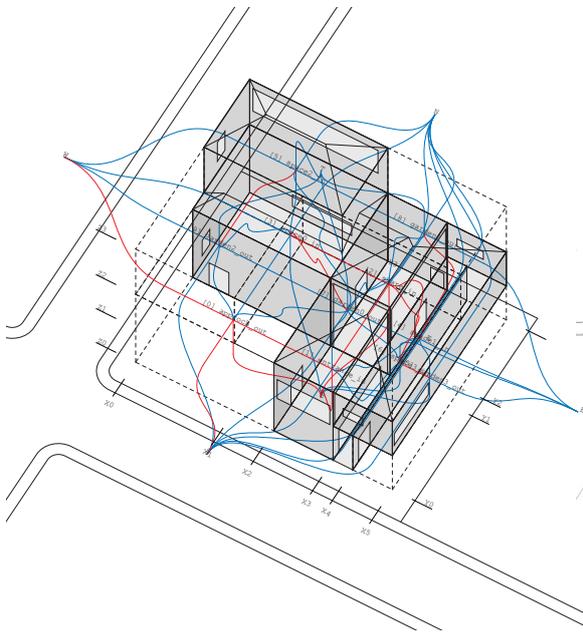


Fig. 9-2-11 STEP11 : ワイヤーフレームモデルを生成する

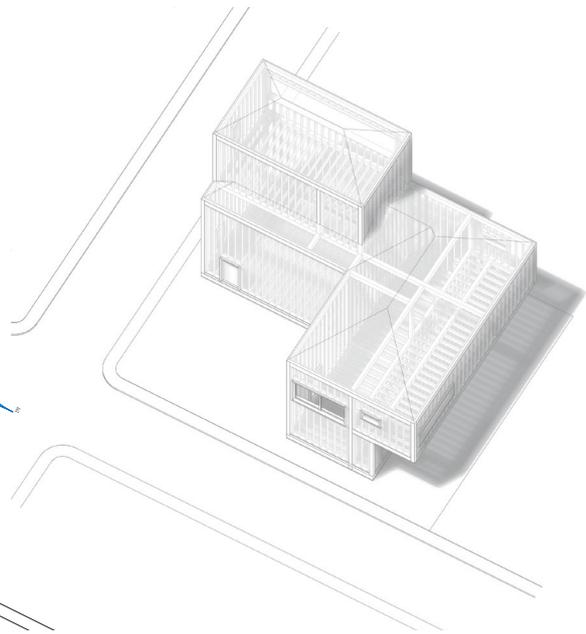


Fig. 9-2-12 STEP12 : ビルディングエレメントを生成する

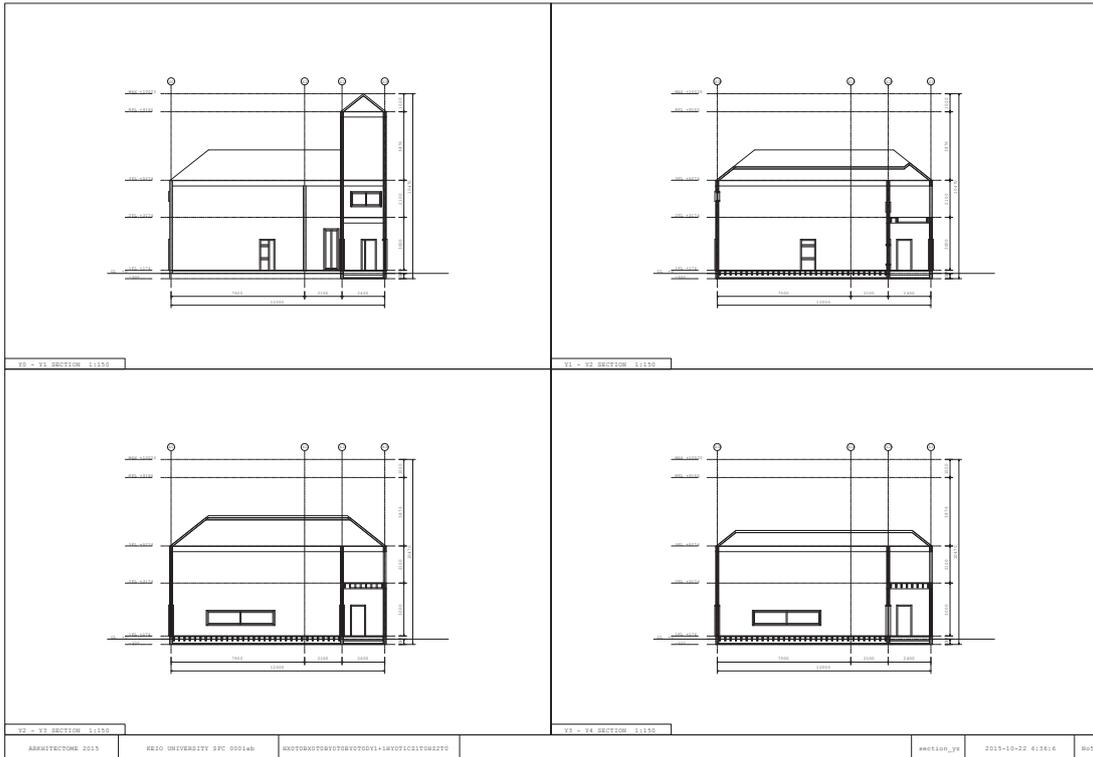


Fig. 9-2-17 断面図1

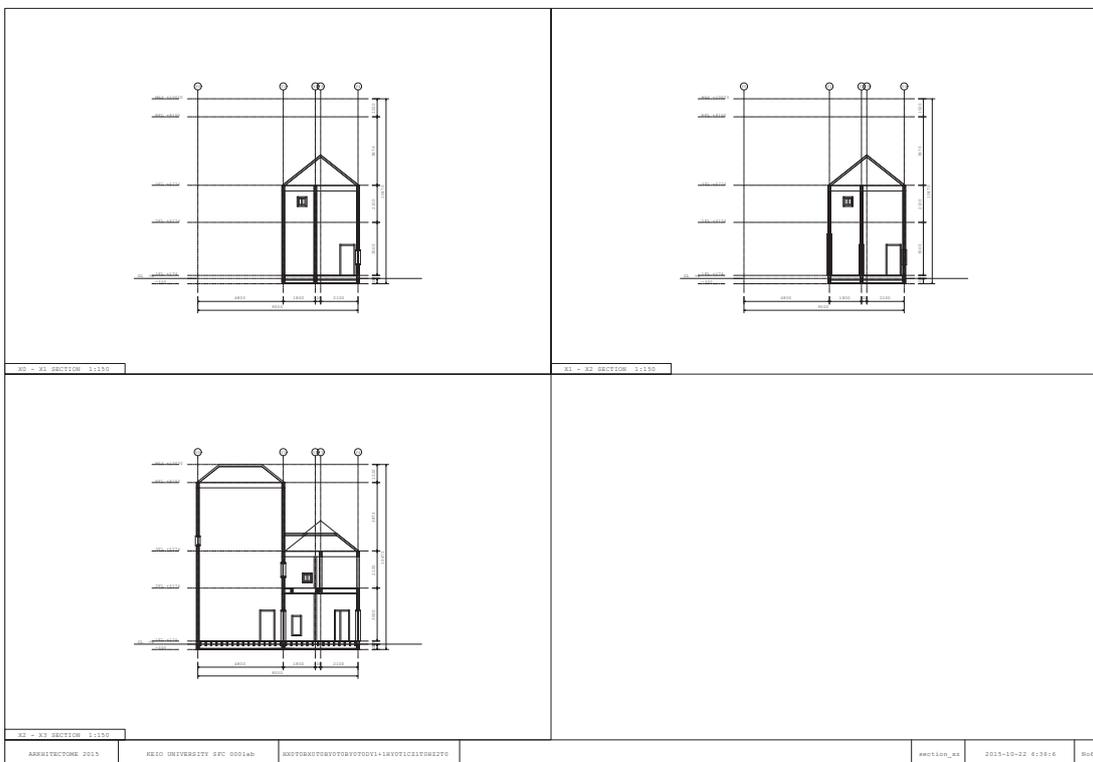


Fig. 9-2-18 断面図2

9.3 評価のフェイズ

本節では、ARKHITEKTOME の評価のフェイズについて述べる。既往研究としては、内田らによるビルディング・エレメント論や [9-8] ~ [9-16] や Bill Hillier らによる「Space Syntax 理論」がある [9-17]。本節は、「建築の空間構成における、視線、動線、空気の関係性を定量的に評価する手法の提案」 [9-18] で発表した内容に基いている。

9.3.1 評価に関する3つの指標

不動産サイトで Fig. 9-3-1 のようなふたつの住宅を検索したとする。左右どちらも、3LDK で 110 平米程度の同規模な面積である。通常の検索では、通常の検索パラメータでは、間取や面積といった情報しかないので、このようなふたつの空間の質を区別できない。しかし、このような空間性の違いは、隣接している空間との関係性によって質が異なると考えられる。そこで、本研究では、隣接する空間間において、

1. 視線がどれくらい通じているか
2. 動線がどれくらい通じているか
3. 空気がどれくらい通じているか

という3つの指標によって空間構成の評価を行う。

9.3.2 評価に関する3つの段階

下記の三段階で評価していく (Fig. 9-3-2)。

1. 住宅全体に電磁場のような「観測場」を設定し、各観測点ごとに評価を行う。
2. 各観測点の評価値を、外部空間も含めた「単位空間」へと統合する。
3. 各単位空間の評価値を、「建築全体」の評価へと統合する。

3TARGETS OF EVALUATION/1-1. 評価の3対象

立地：東京都目黒区緑が丘一丁目22-19
 間取：3LDK
 面積/バルコニー：125.32㎡/0㎡
 印象：大きな一室空間 パブリック 中心性



視線：5 動線：7 空気：4

立地：東京都目黒区碑文谷四丁目22-3
 間取：3LDK
 面積/バルコニー：108.63㎡/31.09㎡
 印象：個室 プライベート 内庭



視線：1 動線：1 空気：0

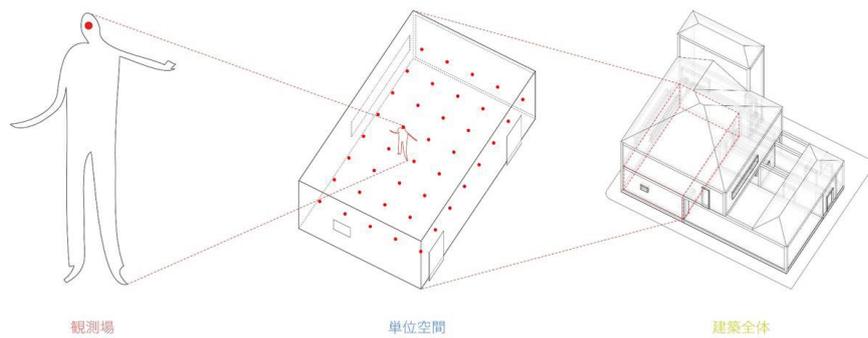
参照：R-STOREホームページ

© ARCHITECTURE 2013

© 2015- Shohhei MATSUMURA Lab., (0001Lab.) in RIZO RICE

Fig. 9-3-1 視線、導線、空気による空間構成の評価

3PHASE OF EVALUATION/1-2. 評価の3段階



© ARCHITECTURE 2013

© 2015- Shohhei MATSUMURA Lab., (0001Lab.) in RIZO RICE

Fig. 9-3-2 評価の三段階

9.3.3 評価に関する5つの関係性

各単位空間の種類によって下記の5種類の関係性がある。

1. 内部／内部
2. 内部／外部
3. 外部／外部
4. 内部／敷地外
5. 外部／敷地外

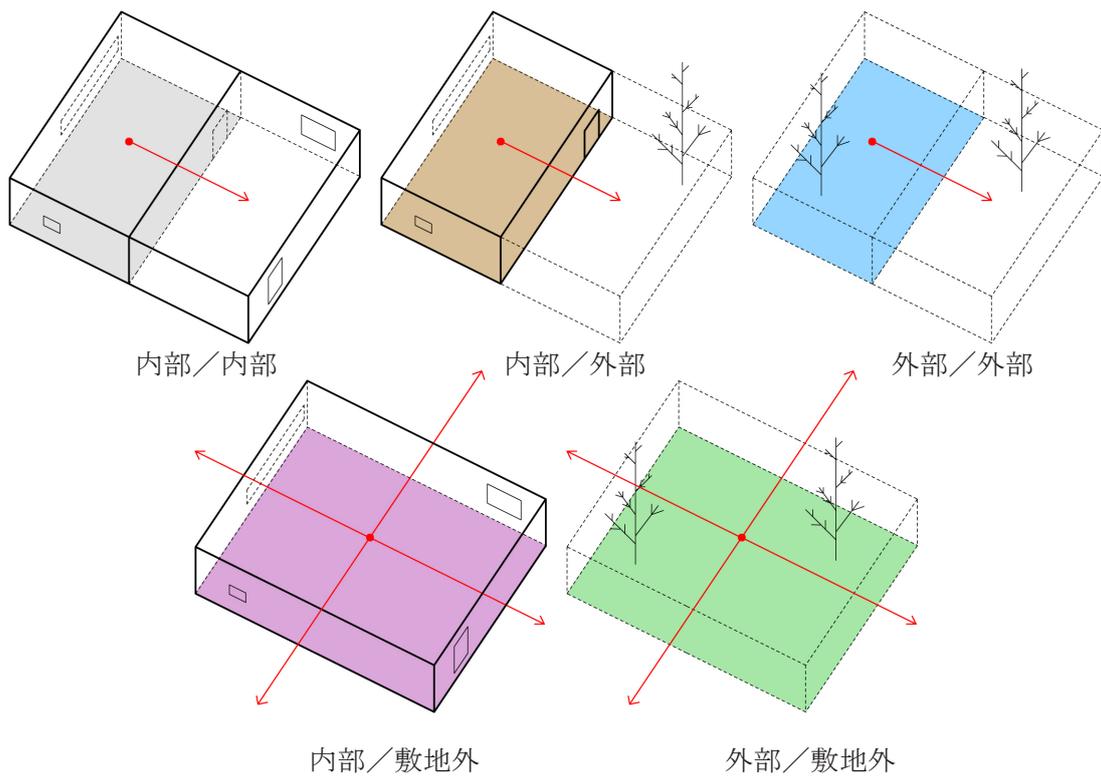


Fig. 9-3-3 5種類の空間相互の関係性

9.3.4 結果の表現

計算によって得られた評価値は、0 から 100 の点数で表される。点数に応じて結果がわかりやすいよう、ヒートマップのグラデーションによって表現する。Fig. 9-3-4 に例を示す。マトリクスは下記の3つの観点に従って構成されている。

- (1) 第9.3.1項で述べた評価の指標は、Fig. 9-3-5のようにマトリクス中に表される。
 1. 視線
 2. 動線
 3. 空気

- (2) 第9.3.2項で述べた評価の3段階は、Fig. 9-3-6のようにマトリクス中に表される。
 1. 単位空間相互の関係性の評価値
 2. 単位空間単体の評価値
 3. 建築全体の評価値

- (3) 第9.3.3項で述べた5種類の関係性は、Fig. 9-3-7のようにマトリクス中に表される。
 1. 内部／内部
 2. 内部／外部
 3. 外部／外部
 4. 内部／敷地外
 5. 外部／敷地外

9.3.5 視線による評価

視線による評価は、下記の手順により、任意の単位空間からどのくらい他の単位空間が見える化を0から100の間で点数化し評価する。

1. 人間の目の高さ(1600mm)の位置に均等(300mmピッチ)に観測場を配置する。個室1に配置したそれぞれの観測場から、個室2内に配置した観測場が何個見えるかを算出する。(Fig. 9-3-8 左上)
2. 各観測場の点数をすべての合計する。(Fig. 9-3-8 中上)
3. 合計値を全パターン数で割る。例えば、個室1の観測場が12個、個室2の観測場が20個の場合、全パターン数は $12 \times 20 = 240$ となる。個室1のすべての観測場から個室2のすべての観測場が視えた場合は100、全く見えなかった場合は0となる。(Fig. 9-3-8 右上)

9.3.6 動線による評価

動線による評価は、下記の手順により、任意の単位空間から他の単位空間へ、どれくらい人間が移動しやすいかを0から100の間で点数化し評価する。

1. 個室1の床から100mmの位置に900ピッチで均等に観測場を配置し、隣接する観測場や上下動線が繋がる観測場を結んで経路グラフを作成する。ダイクストラ法を用いて、それぞれの観測場から他の観測場までの最短経路を算出する。(Fig. 9-3-9 左上)
2. 個室1に配置したそれぞれの観測場から、相手側の個室2内に配置した観測場までの最短距離の合計を計算していく。次に、その観測場ごとに保持している最短経路の合計値を、更に合計する。その合計値を、個室1内の観測場から個室2内の観測場へのマッチングの全パターン数で割る。すると、個室1の観測場から、個室2への観測場への平均距離が計算できる。(Fig. 9-3-9 中上)
3. 建築内の「動線」の観測場の関係の中で、最も遠い最短経路の距離を割り出す。その建築内の最長な最短経路で、動線の平均距離を割る。すると、建築内の最長な最短経路を100とした時に、距離がどれくらい近いかを計算することができる。(Fig. 9-3-9 右上)

9.3.7 空気による評価

空気による評価は、下記の手順により、任意の単位空間から他の単位空間へ、どれくらい空気が移動しやすいかという空間の距離を0から100の間で点数化し評価する。

1. 個室1、2に三次元的に900ピッチで観測場を配置する。隣接する観測場や上下動線が繋がる観測場を結んで経路グラフを生成する。ダイクストラ法を用いて、それぞれの観測場から他の観測場までの最短経路を算出する。(Fig. 9-3-10 左上)
2. 個室1に配置したそれぞれの観測場から、相手側の個室2内に配置した観測場までの最短距離の合計を計算していく。次に、その観測場ごとに保持している最短経路の合計値を、更に合計する。その合計値を、個室1内の観測場から個室2内の観測場へのマッチングの全パターン数で割る。すると、個室1の観測場から、個室2への観測場への平均距離が計算できる。(Fig. 9-3-10 中上)
3. 建築内の「空気」の観測場の関係の中で、最長な最短経路の距離を割り出す。その建築内の最長な最短経路で、空気の経路の平均距離を割る。すると、建築内の最長な最短経路を100とした時に、距離がどれくらい近いかを計算することができる。(Fig. 9-3-10 右上)

9.3.8 評価実験1：難波和彦「箱の家」

上記で示した評価のフェイズが有効に働くかどうかを、難波和彦の箱の家を対象として検証する。

難波和彦の箱の家は一室空間として計画されている。したがって、内部-内部の視線の評価、内部-内部の空気の評価の点数が高いと想定される。また前面道路に向かって大きな開口が設えてあるので、内部-外部の視線の点数も比較的高いと想定される。

実際に評価を行い、その結果を Fig. 9-3-11 に示す。

評価結果から、内部-内部の視線の評価、内部-内部の空気の評価の点数が高く、内部-外部の視線の点数も比較的高いことが確認された。また視線と空気の評価を見ると、箱の家001から箱の家017まで、竣工順に視線の点数が上がっていることが伺える。これは一室空間性を高めるために、設計者の試行錯誤が結果として現れたと考えられる。このような目には見えない微細な差異も分析できることは、本論文で提案する評価のフェイズが有効に働いている証左といえる。

9.3.9 評価実験2：安藤忠雄「住吉の長屋」

上記で示した評価のフェイズが有効に働くかどうかを、安藤忠雄の住吉の長屋を対象として検証する。

住吉の長屋は外部の周辺環境に対して閉鎖的であるため、内部－外部の視線の評価は非常に低いと想定される。また各内部空間が中庭を介して繋がっているため、内部－内部の動線の評価は低く、内部－外部の動線の評価は高いと想定される。

実際に評価を行い、その結果を Fig. 9-3-12 に示す。

評価結果から、内部－外部の視線の評価が低いこと、内部－内部の動線の評価が低いこと、内部－外部の動線の評価が高いことが確認された。

9.3.10 評価実験3：西沢立衛「森山邸」

上記で示した評価のフェイズが有効に働くかどうかを、西沢立衛の森山邸を対象として検証する。

森山邸は複数の建物が分散配置され、開口部も大きいので、一見すると内部－外部の視線や外部－外部の視線の評価が高いと想定される。

実際に評価を行い、その結果を Fig. 9-3-13 に示す。

評価結果から、想定とは逆に、内部－外部の視線や外部－外部の視線の評価が低いことが確認された。これは、一見オープンに視界が開かれているように見えながらも、実は建物の配置や開口部の位置や大きさを注意深く配置することによって、視線の抜けを制御し、プライバシーを保っていることを意味する。

9.3.11 評価実験の考察

第9.3.8-10項の3つの実験により、本論で提案した評価のフェイズは、評価する対象となる建築物の視線・動線・空気に関する空間構成を定量的に評価できることが確認できた。

EVALUATION MATRIX/2-3. 予想される結果

住吉の長屋_安藤忠雄

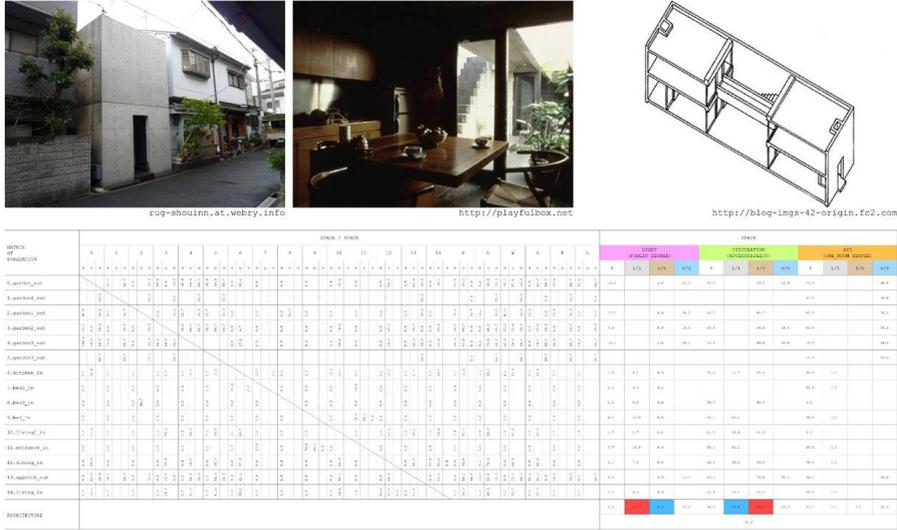


Fig. 9-3-12 安藤忠雄「住吉の長屋」の評価

EVALUATION MATRIX/2-3. 予想される結果

森山邸_西沢立衛

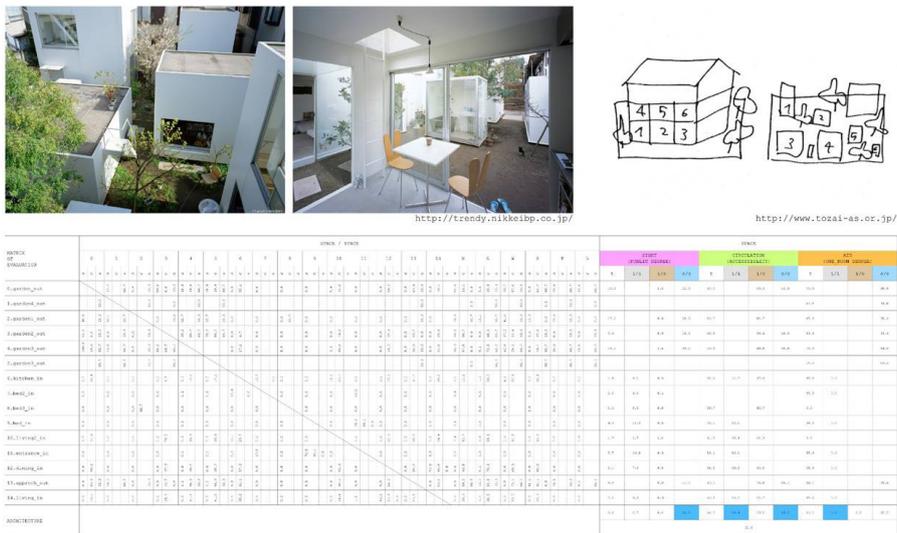


Fig. 9-3-13 西沢立衛「森山邸」の評価

9.3.12 評価のデータシート出力

生成された住宅を評価した結果をまとめたデータシートの例を Fig. 9-3-14 から Fig. 9-3-17 までに示す。

要旨

目次

序章

第1部

第2部

結章

後付

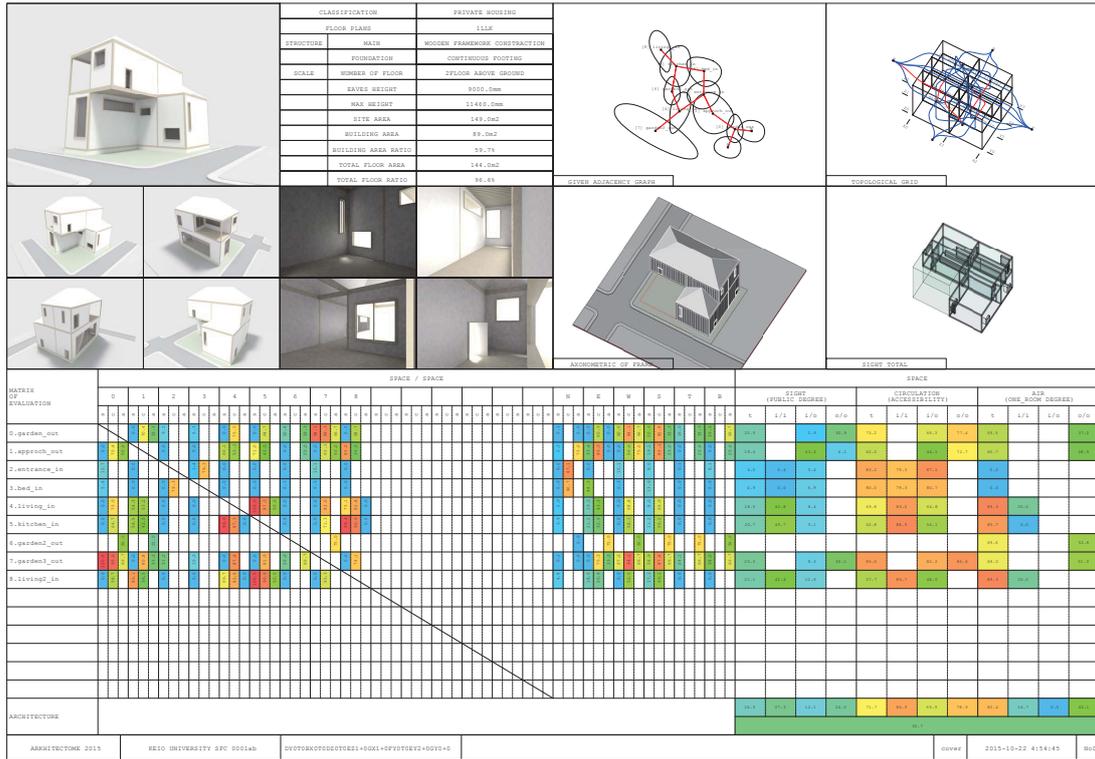


Fig. 9-3-14 評価データシート 1

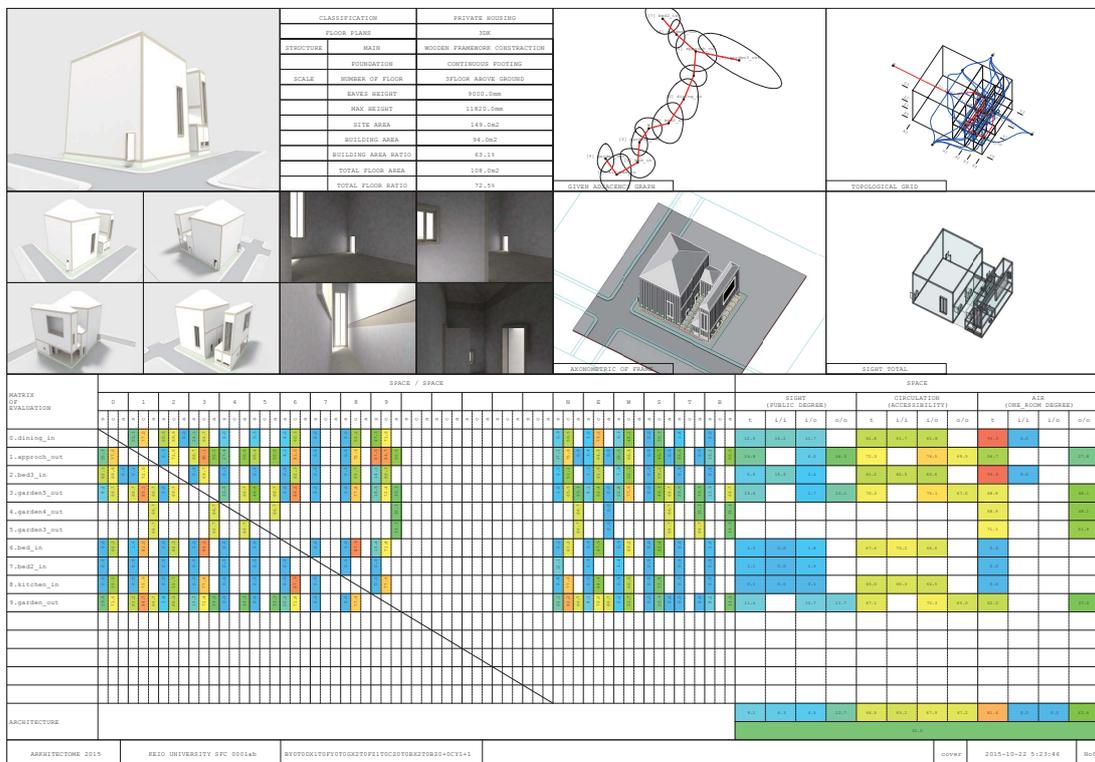


Fig. 9-3-15 評価データシート 2

9.3.13 Total_Fitness の算出

補正係数 R を考慮した *Total_Fitness* を算出する。(Fig. 9-3-18)

Fig. 9-2-18 の例では、視線を例にとると、より視線が抜けている空間同士が高い適応度を示すように計算している。しかしながら、実際にはより視線が抜けている空間同士が高い適応度であるとは限らない。

したがって、実際に住宅の評価に用いるときには、例えば、住吉の長屋の空間構成が好きなユーザーがいたと仮定すると、住吉の長屋の空間構成に近い点数ほど良い評価となるような計算方法が考えられる。今後の課題である。

EVALUATION.6 WEIGHTING AND RANKING

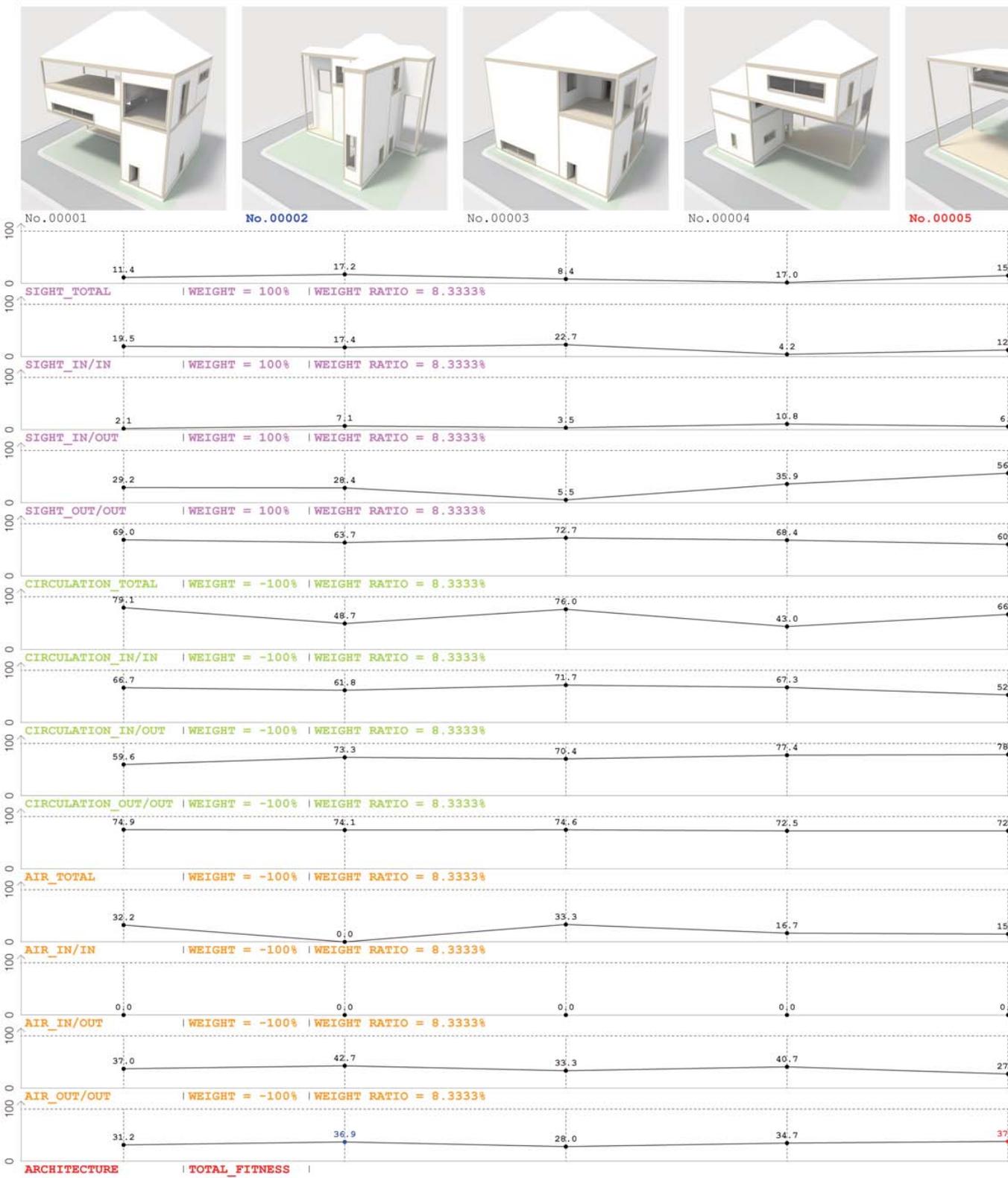
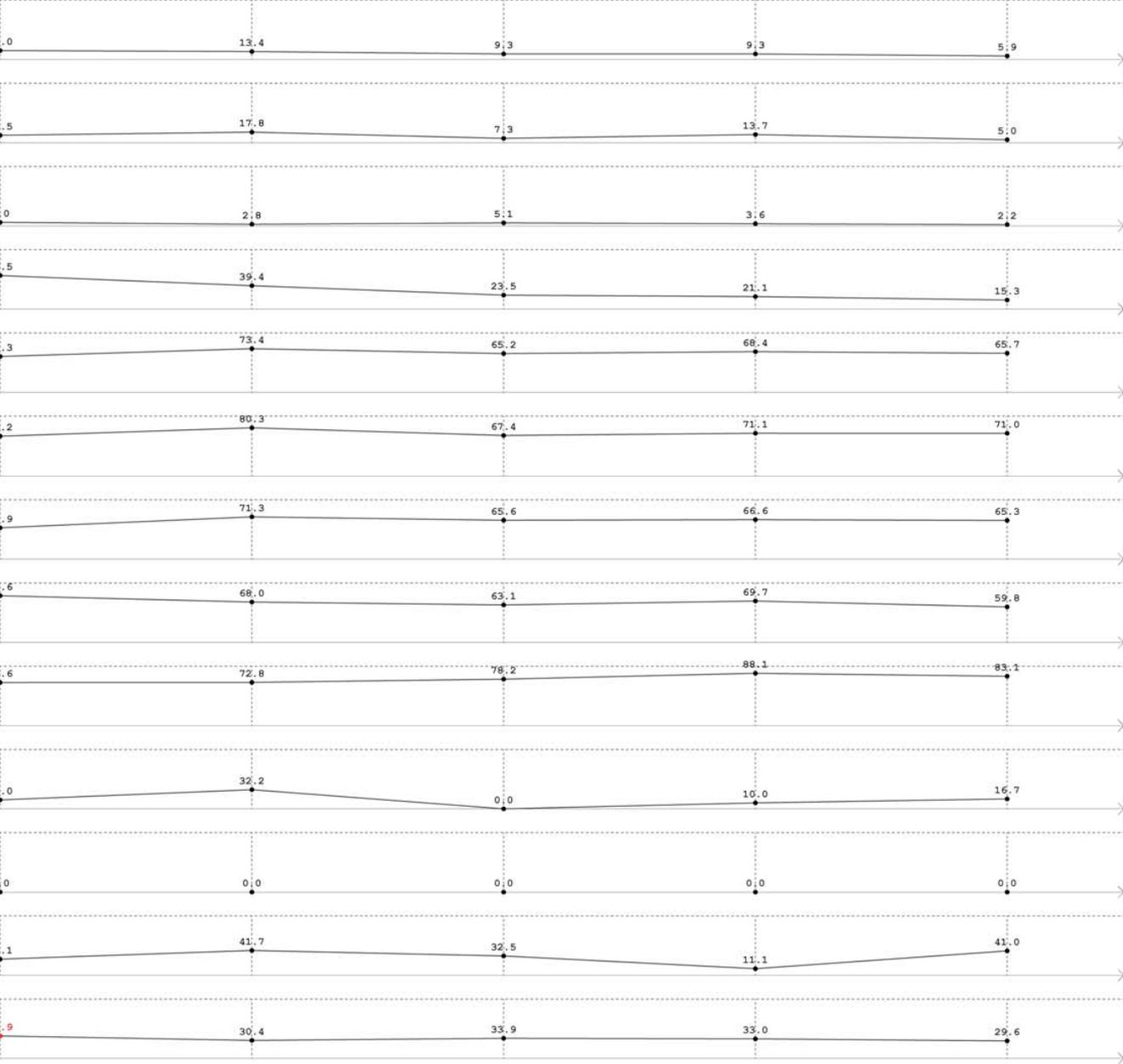
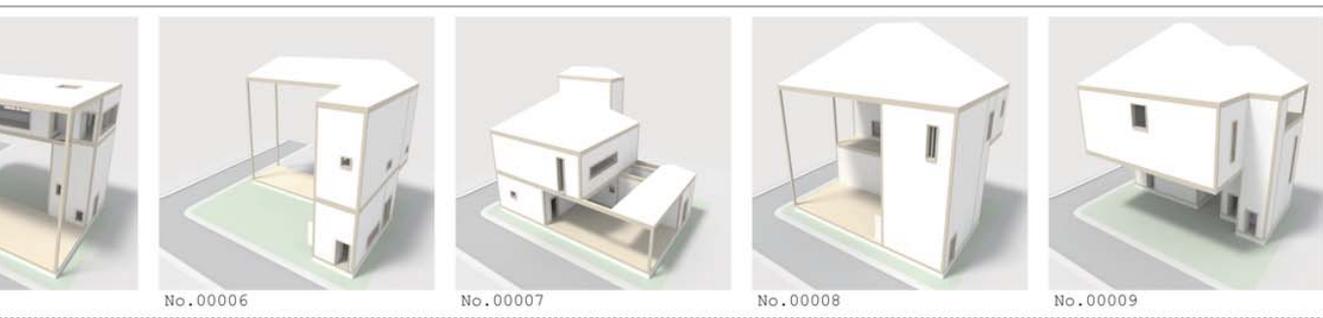


Fig.9-3-18 Total_Fitness の算出



9.4 高適化のフェイズ

ARKHITEKTOME では、Fig. 9-4 のような高適化のフェイズは、まだ実装されていない。今後の課題である。

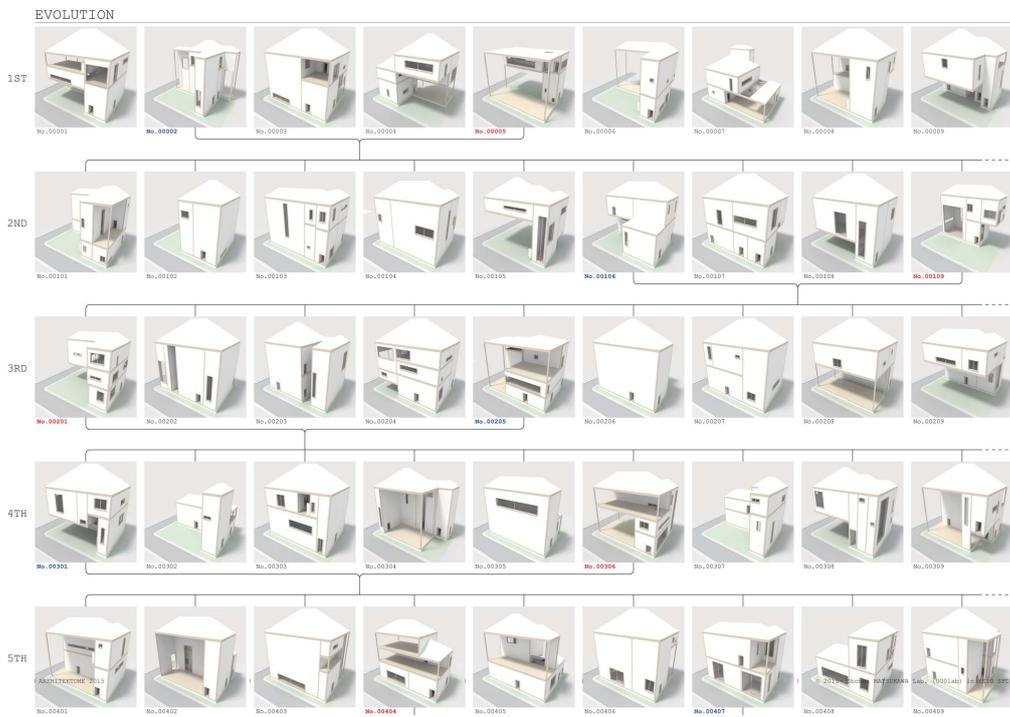


Fig. 9-4 高適化のフェイズ (未実装)

要旨

目次

序章

第1部

第2部

結章

後付

第 10 章

逆探索法を用いた直方体分割図の列挙アルゴリズム

本章では、逆探索法を用いた列挙アルゴリズムについて詳説する。アルゴリズムック・デザインのプロトタイプを住宅へ応用する際には、与条件を満たした諸室の隣接関係を自動的に求める室配置問題を解くことが不可欠かつ重要である。本章では、2次元長方形分割図の列挙アルゴリズムを3次元へと拡張し、逆探索法を用いることができる範囲において3次元直方体分割図を列挙するアルゴリズムを提案する。室配置の基礎的なモデルとなる3次元直方体分割図に関しては、未だ漏れも重複もない列挙アルゴリズムは知られておらず、アルゴリズムック・デザインの住宅への応用可能性を示す上で要となる3次元直方体分割図を列挙するアルゴリズムの妥当性を詳細に検討した点が本論の主な成果である。

10.1 序

10.1.1 背景

敷地の外形が直方体で、機能的な単位空間もそれぞれ直方体である建築物を想定して、壁や床、天井などのビルディングエレメントや寸法を抽象化すると、直方体がいかに隣接しているかを示すモデルが得られる。このモデルは、外形の直方体をさらにいくつかの小さな直方体に分割する形をしているので、**直方体分割図**と呼ばれる [10-1][10-2]。また、このような建築物の空間構成を、与えられた様々な要求に基づいて、自動的に決定する問題を**室配置問題**と呼ぶ。

直方体分割図は、外形や室の形状が直方体に限定されるものの、建築物として具体化する際に、室を外部と見做したり、室と室を組み合わせたりすることで、より複雑な形状の建築物を表現できる可能性がある。また直方体分割図は、その分割数に応じて定まる種類の数が有限であるので、あらかじめすべての直方体分割図を列挙することができれば、その中から設計条件に適合する直方体分割図を抜き出し、適正な面積や容積を割り付けることで設計初期段階のボリュームモデルを得ることができる。さらに、直方体分割図は新しい建築物を生成する際に用いられるだけでなく、既存の建築物を直方体分割図に応じて分類する際に用いることもできる。したがって直方体分割図の列挙は、建築計画上、基礎的で重要な問題であると考えられる。

2次元の分割図に関しては、後述する逆探索法 [10-3] を用いることで、漏れも重複もないすべての長方形分割図を列挙するアルゴリズムが中野 [10-4][10-5] らによって提案されている。しかしながら、3次元の分割図に関しては、漏れも重複もないすべての直方体分割図を列挙するアルゴリズムはいまだ知られていない。一般的な建築物は2次元の室配置を複層階に積層させることで表現できるので、神山ら [10-6] が提案したように、中野らの2次元長方形分割図の列挙アルゴリズムを用いて室配置を行うことができる。しかし近年、藤本のHouse H [10-7] やHouse NA [10-8] などで見られるように、各室は直方体であるものの、単純な階の積層では表現し難い建築物が多く実現されている。したがって、可能な限り多くの種類の3次元直方体分割図を列挙するアルゴリズムを提案することには現代的な意義があると考えられる。

10.1.2 既往研究

神山ら [10-6] は室配置問題に関する既往研究を、室の数学的表現によって、(1) 単位格子による表現 [10-9][10-10][10-11][10-12]、(2) 箱詰め問題としての表現 [10-13][10-14][10-15][10-16]、(3) グラフ理論による表現という3つの方法に分類し、さらに上記(3)を、探索型とカタログ型という2つのアプローチに分類している。探索

型は、与条件が先にあつて、その与条件を満たすような直方体分割図だけを生成するアプローチである [10-17][10-18][10-19]。カタログ型は、先にあり得べき直方体分割図を列挙しておいて、その中から条件に適合するものを探し出すアプローチである [10-20][10-21]。

本研究は、室配置をグラフ理論を用いて表現し、可能な限り多くの種類の 3 次元直方体分割図を列挙することを目指すもので、カタログ型に属する。以下、3 次元直方体分割図に関するカタログ型の既往研究を概観する。

Fig. 10-1 の (イ) のような直方体分割図は、外形の直方体を矩形分割面で 2 つの直方体に分割する操作を再帰的に行うことで得られるが、このようなスライス型 [10-22] だけでは表現できる直方体分割図の種類は少なく、例えば Fig. 10-1 の (ロ) に示すような卍字型は表現できない。

Ohta ら [10-23] は、小直方体の数 n に対して $O(n)$ 時間でエンコード・デコードが可能である 0-sequence という直方体分割図の表現方法を提案した。0-sequence は、スライス型と卍字型のどちらも包含した直方体分割図を表現できる。その一例を Fig. 10-1 の (ハ) に示す。このように直方体分割図を構成するすべての頂点の次数 (頂点から伸びている線分の数) が 4 以下である直方体分割図を本論では **T 字型** と呼ぶこととする。T 字型はスライス型と卍字型を包含している。対して、Fig. 10-1 の (ニ) に示すように、直方体分割図を構成するいずれかの頂点の次数が 4 より大きい直方体分割図を本論では **十字型** と呼ぶこととする。0-sequence は、T 字型を表現できるが、十字型を表現することはできない。

吉田 [10-1] は、立体格子から不必要な分割面を削除することで T 字型の直方体分割図を列挙するアルゴリズムを提案している。ただし、Fig. 10-1 の (ホ) と (へ) のように、回転および鏡面反転による置換によって同じとみなせるふたつの直方体分割図は、同じ種類の直方体分割図であるとして取り除かれている。このことは列挙する数を少なく抑えることができるなどの利点はあるが、建築計画では室の東西南北上下に意味があるため、設計者によっては、数は増えても異なる種類の直方体分割図として列挙したほうが望ましいと考えられる。また十字型については、Fig. 10-1 の (イ) のような T 字型の直方体分割図の各小直方体に適正な寸法を割付けることで、Fig. 10-1 の (ニ) のような十字型が表現可能であるため、十字型の直方体分割図は不必要であるとして列挙されていない。しかし、実際多くの建築物は十字型であり、直方体分割図を新しい建築物の生成に用いる際には、寸法を割り付けた後も室と室あるいは室と方角がどのように隣接しているかを表す隣接グラフ (定義は第 10.2.1 項参照) が変化しないほうが望ましいと考えられる。また、直方体分割図を既存の建築物の分類に用いるという観点からも、数は増えても十字型の直方体分割図を列挙したほうが望ましいと考えられる。さらに吉田のアルゴリズムでは、Fig. 10-1 の (ト) と (チ) のように、隣接グラフは同じであるが分割面の基準座標 (定義は第 10.2.1 項参照) が

異なるふたつの直方体分割図を、異なる種類の直方体分割図としている。しかし、隣接グラフに応じて直方体分割図を建築物の生成あるいは分類に用いるという観点からは、隣接グラフが同じであるふたつの直方体分割図を、同じ種類の直方体分割図としたほうが望ましいと考えられる。

10.1.3 目的

本研究では、中野らの提案した 2 次元長方形分割図の列挙アルゴリズムを 3 次元へ拡張し、逆探索法を用いることができる範囲において、T 字型だけではなく十字型も含めた 3 次元直方体分割図を列挙するアルゴリズムを提案することを目的とする。

逆探索法 [10-3] とは、探索したい対象をグラフのノードとし、探索経路をノードを結ぶエッジとしたときに、対象間の親子関係と根を上手く定義することで、対象全域を**全域根付き木 T [10-24]** として定義し、この T を根から**深さ優先探索 [10-25]** することで対象を列挙する手法である。探索したい対象の親子関係と根を上手く定義できれば、 T 自身を知ることなく T 上の深さ優先探索が行えるようになる。逆探索法が有効な手法とされるのは、対象の局所的な情報のみに基づいて探索が実行されるからである。

上記の逆探索法の定義より、本研究で探索の対象となる直方体分割図は、直方体分割図の間に親子関係が定義できたものに限られる（詳しくは第 10.2.6 項で述べる）。しかしながら、逆探索法を用いて 3 次元直方体分割図の列挙を試みた既往研究はこれまで存在しない点、また、本研究のアルゴリズムで列挙される直方体分割図は、これまで知られていない種類の直方体分割図を多く含む点に既往研究との違いがある。

以降において、まず第 10.2 節では対象となる直方体分割図の間に親子関係と根を定義し、 n 個の小直方体を持つ直方体分割図の全域根付き木 T_n を定義する。第 10.3 節では直方体分割図の全域根付き木 T_n を探索するためのアルゴリズム、すなわち直方体分割図の列挙アルゴリズムを提案する。そして第 10.4 節においてアルゴリズムの妥当性を検証し、第 10.5 節で結論と今後の課題について述べる。

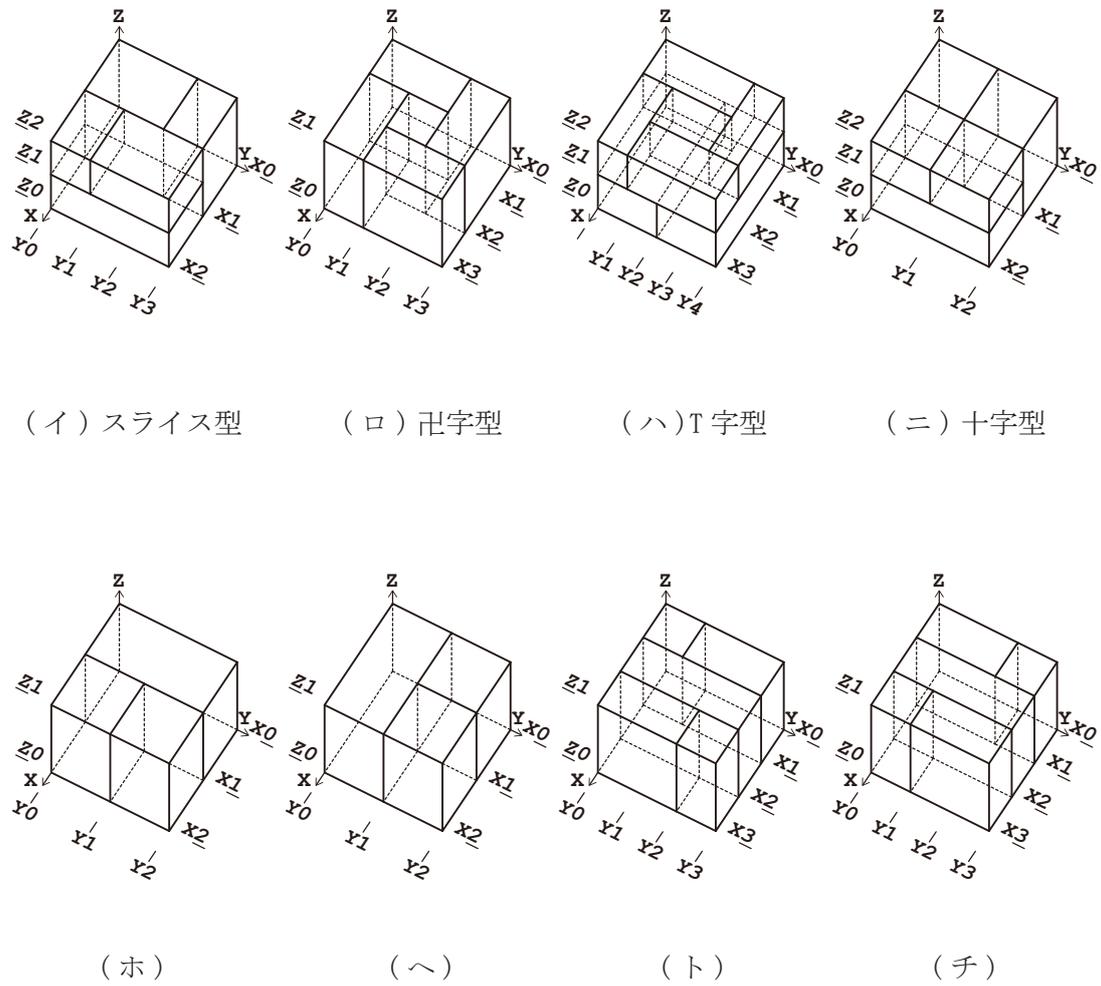


Fig. 10-1 直方体分割図の種類

10.2 全域根付き木

本節では、対象となる直方体分割図の間に親子関係と根を定義し、 n 個の小直方体を持つ直方体分割図の全域根付き木 T_n を定義する。

10.2.1 直方体分割図の定義

3次元空間の直交右手座標系（右手の親指が $X+$ 軸、人差し指が $Y+$ 軸、中指が $Z+$ 軸）において、 $X-$ 軸方向を西、 $X+$ 軸方向を東、 $Y-$ 軸方向を南、 $Y+$ 軸方向を北、 $Z-$ 軸方向を下、 $Z+$ 軸方向を上とする。

XYZ 軸に沿って置かれている直方体（**外形直方体**）を、 X 軸、 Y 軸、 Z 軸のいずれかに垂直な平面（**分割面**）によって複数の直方体領域（**小直方体**）に分割したものを**直方体分割図**と呼ぶ。小直方体をひとつだけ持つ直方体分割図を R_i と書く。 R は R_i を除く任意の直方体分割図とする。またそのすべての集合を R_n とする。 R は頂点、線分、面、そして小直方体で構成された立体的なグラフとして表現される。

ある頂点からみて、他の頂点と線分で結ばれているとき、線分の伸びている方向に**接続**していると呼ぶ。

各小直方体を構成する 6 面の内、各小直方体の内側から見て $X-$ 軸方向の面を **$X-$ 面** と呼ぶ。同様に、 $X+$ 軸方向の面を **$X+$ 面**、 $Y-$ 軸方向の面を **$Y-$ 面**、 $Y+$ 軸方向の面を **$Y+$ 面**、 $Z-$ 軸方向の面を **$Z-$ 面**、 $Z+$ 軸方向の面を **$Z+$ 面** と呼ぶ。

ある小直方体からみて、他の小直方体あるいは外形直方体と面を共有しているとき、その面の方向に**隣接**していると呼ぶ。小直方体をノードとし、隣接している小直方体同士あるいは隣接している小直方体と方角をエッジで結ぶと、小直方体と小直方体あるいは小直方体と方角がどのように隣接しているかをグラフで表現できる。このグラフを**隣接グラフ**と呼ぶ。直方体分割図と隣接グラフは双対である。

ふたつの R が与えられたとき、回転や鏡像反転を許さず、隣接グラフが同じであれば、 R の外形直方体や各小直方体の大きさが異なっても、ふたつの R は同一の直方体分割図とみなす。

X 軸に垂直な分割面の X 座標の集合を X 軸の**基準座標**と呼ぶ。同様に Y 軸に垂直な分割面の Y 座標の集合を Y 軸の**基準座標**、 Z 軸に垂直な分割面の Z 座標の集合を Z 軸の**基準座標**と呼ぶ。

10.2.2 直方体分割図の7つのTYPE

任意の直方体分割図 R を構成する小直方体のうち、XYZ 座標が最大である点 G (図中では白丸で表す) を頂点とする小直方体はひとつだけ存在する。その小直方体のことを第 n 小直方体と呼ぶ。

第 n 小直方体を構成する頂点の中でXYZ座標が最小の点を V (図中では黒丸) とする。 V を端点とする線分のうち第 n 小直方体を構成しない線分を E_v とすると、任意の直方体分割図 R は E_v の数と方向によって7つのTYPEに分類することができる (Fig. 10-2)。

TYPE1 は E_v の数が1本でその方向はX-軸方向である。TYPE2 は E_v の数が1本でその方向はY-軸方向である。TYPE3 は E_v の数は1本でその方向はZ-軸方向である。TYPE4 は E_v の数が2本でその方向はX-軸方向とY-軸方向である。TYPE5 は E_v の数が2本でその方向はY-軸方向とZ-軸方向である。TYPE6 は E_v の数が2本でその方向はX-軸方向とZ-軸方向である。TYPE7 は E_v の数が3本でその方向はX-軸方向とY-軸方向とZ-軸方向である。

次に、第 n 小直方体のX-面、Y-面、Z-面のそれぞれを E_v に沿ってスライドすることを考える。TYPE1 の V はX-軸方向に1本の E_v を持つので、その E_v に沿って第 n 小直方体のX-面をX軸方向にスライドさせることができる。同様にTYPE2 はY-面をY軸方向に、TYPE3 はZ-面をZ軸方向にスライドさせることができる。TYPE4, 5, 6, 7 はそれぞれ複数通りのスライド方向があるが、TYPE4 はX-面をX軸方向に、TYPE5 はY-面をY軸方向に、TYPE6 はZ-面をZ軸方向に、TYPE7 はX-面をX軸方向にスライドさせるものとする。いずれのTYPEにおいてもスライドする面を I とする (図中では濃いグレーで表す)。

また、 V の次数が4以下 (すなわちTYPE1, 2, 3) の場合、 V を **T型** と呼び、一方、 V の次数が4より大きい (すなわちTYPE4, 5, 6, 7) 場合、 V を **+型** と呼ぶ。

以上をまとめると、Fig. 10-2 で示した7つのTYPEは、それぞれ [E_v の数、 I のスライド方向、 V の型] のように表すことができる。

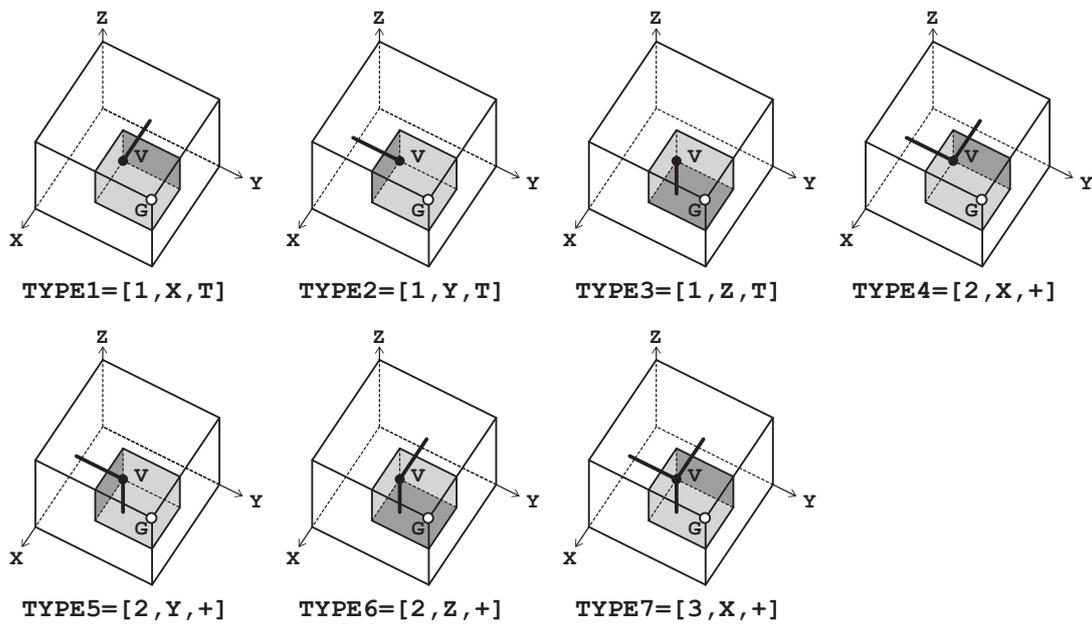


Fig. 10-2 直方体分割図の7つのTYPE

10.2.3 消去可能な直方体分割図

任意の直方体分割図 R が与えられたとき、TYPE1, 4, 7 は、 I 面を $X+$ 軸方向にスライドさせながら押しつぶしていくと同時に、 I 面を共有する他の小直方体を押し広げていくことにより、第 n 小直方体だけを消去できる場合がある。このとき R は X 軸に消去可能であるという。 X 軸に消去可能な直方体分割図 R の例を Fig. 10-3 に示す。同様に、TYPE2, 5 においては Y 軸に消去可能、TYPE3, 6 においては Z 軸に消去可能な R が存在する。

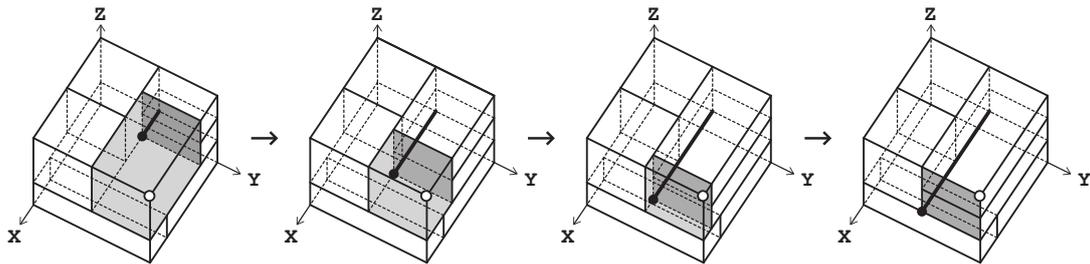


Fig. 10-3 X 軸に消去可能な直方体分割図の例

10.2.4 消去不可能な直方体分割図

前節で第 n 小直方体が消去可能な直方体分割図をみたが、消去不可能な直方体分割図も存在する。その一例を Fig. 10-4 に示す。この例は TYPE2 の直方体分割図だが、 I 面を $Y+$ 軸方向にスライドさせながら押しつぶしていくと同時に、 I 面を共有する他の小直方体を押し広げていくと、第 n 小直方体だけでなく、他の小直方体も消去されてしまう。このように第 n 小直方体を消去するとき、他の小直方体も消去されてしまう場合、 R は消去不可能であるという。

第 n 小直方体と I 面を共有する他の小直方体を C_i とする。 C_i を構成する面の中で、 I 面を含む分割面を F_{ci} とする。第 n 小直方体を消去するとき、他の小直方体も消去されてしまうのは、 I 面と分割面 F_{ci} が異なるからである。逆にいえば、 I 面と分割面 F_{ci} が一致する場合、 R は必ず消去可能である。

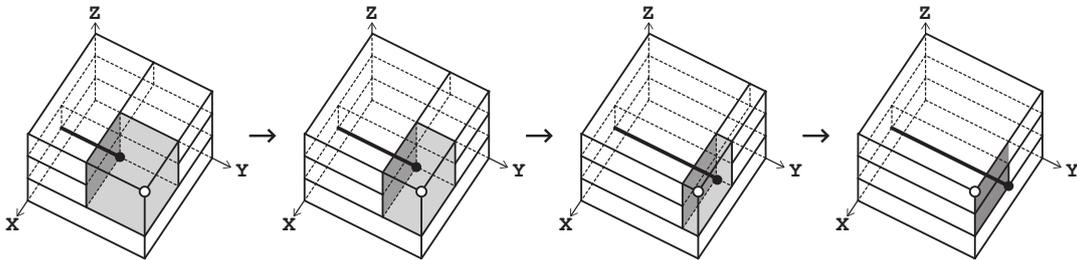


Fig. 10-4 第 n 小直方体が消去不可能な直方体分割図の例

10.2.5 対象とする直方体分割図

R の中で、第 n 小直方体を X, Y, Z 軸のいずれかで消去可能な任意の直方体分割図を S とする。またそのすべての集合を S_n とする。消去可能な直方体分割図の集合 S_n はすべての直方体分割図の集合 R_n の部分集合である。本研究で提案する直方体分割図の列挙アルゴリズムは、消去可能な直方体分割図の集合 S_n を対象とし、消去不可能な直方体分割図は対象外とする。

10.2.6 親子関係と根の定義

S の第 n 小直方体を消去すれば、小直方体の数が1つ少ない S_n 中の直方体分割図が得られる。そのような直方体分割図を $P(S)$ と書き、 S の親であると定める。また逆に、 S は $P(S)$ の子であると定める。 S_n 中の任意の直方体分割図 S が与えられたとき、繰り返し第 n 小直方体を消去することにより $S, P(S), P(P(S)), \dots$ なる直方体分割図の親子関係を表す列が得られる。この列は必ず R_1 で終了するので、 R_1 を根と定める。Fig. 10-5 に親子関係列の例を示す。

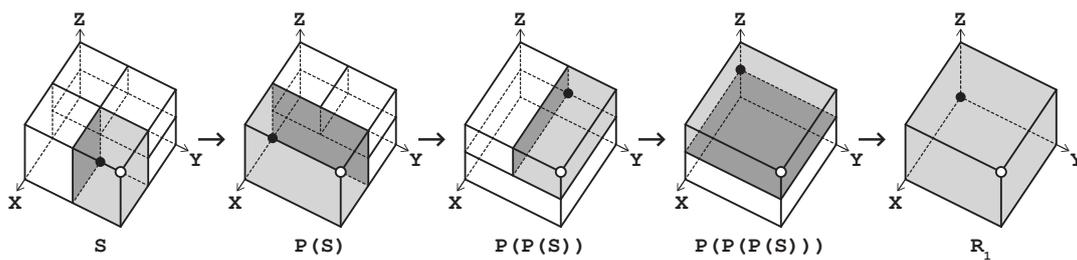


Fig. 10-5 親子関係列の例

10.2.7 全域根付き木の定義

これらすべての親子関係列の列を統合したものを S_n の全域根付き木 T_n と定義する。
Fig. 10-6 に、4 個の小直方体を持つ直方体分割図の全域根付き木 T_4 を示す。

要旨

目次

序章

第1部

第2部

結章

後付

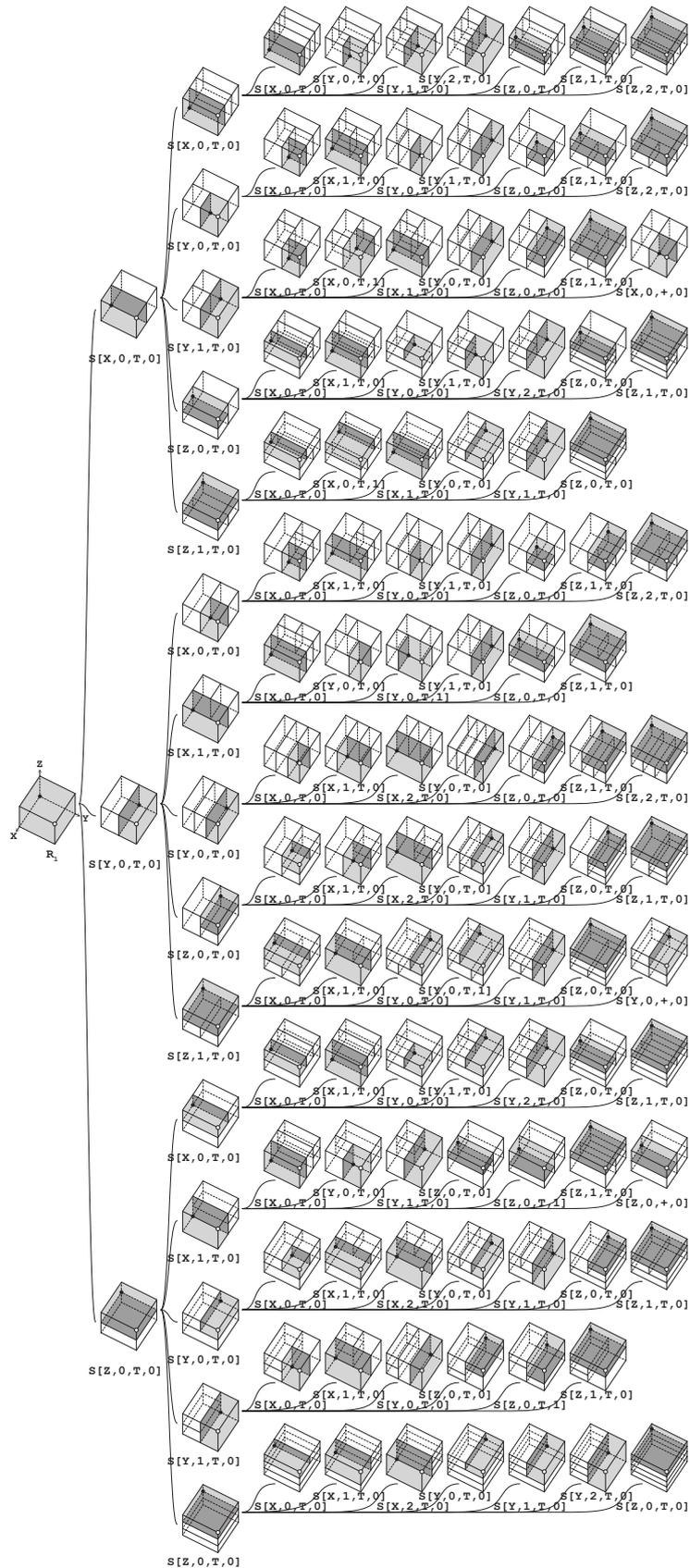


Fig. 10-6 全域根付き木 T_4

要旨

目次

第 1 章

第 2 章

第 3 章

第 4 章

第 5 章

第 6 章

第 7 章

第 8 章

第 9 章

第 10 章

第 11 章

第 12 章

後付

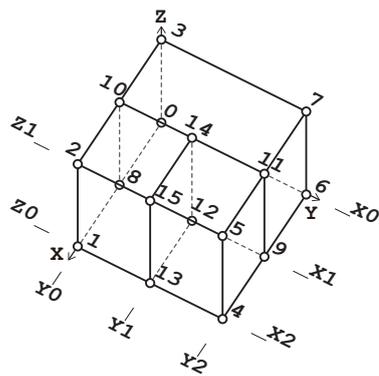
10.3 列挙アルゴリズム

前章では、消去可能な直方体分割図の第 n 小直方体を消去することにより子から親を定義したが、本節では逆に、小直方体を $n-1$ 個持つ任意の親 S を与えたときに、第 n 小直方体を押し込んで追加することにより、親 S から生成可能なすべての子を列挙するアルゴリズムを提案する。このアルゴリズムを R_l から再帰的に適用することにより、 T_n を構成することができる。このことは、列挙される対象が消去可能な直方体分割図の集合 S_n に限られるものの、対象とする直方体分割図を列挙することができることを意味する。

10.3.1 与えられる親のデータ構造

本節では、与えられる任意の親 S が持つべきデータ構造について説明する。親 S の各頂点はユニークな頂点番号 id を持つ。 id がユニークであれば id を打つ順番は問わない。頂点番号 id を持つ頂点を P_{id} と表す。また P_{id} は、 $X-, X+, Y-, Y+, Z-, Z+$ の 6 方向に対する接続関係を保持しておかなければならない。 P_{id} の接続関係を $P_{id}=[X-, X+, Y-, Y+, Z-, Z+]$ と表す。 P_{id} が他の頂点と接続している場合は、その方向に相手の頂点番号を保持する。また P_{id} が外形直方体の外部に対して接続している場合、その方向を保持する。さらに P_{id} が外部と他の頂点のどちらも接続していない場合は、その方向に -1 を保持する。Fig. 10-7 に、与えられる任意の親 S が持つべきデータ構造の例を示す。例えば P_{14} の接続関係は、 $P_{14}=[-1, 15, 10, 11, 12, Z+]$ のように表される。

また与えられる任意の親 S における X 軸、 Y 軸、 Z 軸それぞれの基準座標も保持しておく。Fig. 10-7 では、 X 軸の基準座標は $[X0, X1, X2]$ のように表現しているが、実際には X 軸に垂直な分割面の各 X 座標値を保持しておく。



- | | |
|--------------------------------|----------------------------------|
| $P0 = [X-, 8, Y-, 6, Z-, 3]$ | $P8 = [0, 1, Y-, 12, Z-, 10]$ |
| $P1 = [8, X+, Y-, 13, Z-, 2]$ | $P9 = [6, 4, 12, Y+, Z-, 11]$ |
| $P2 = [10, X+, Y-, 15, 1, Z+]$ | $P10 = [3, 2, Y-, 14, 8, Z+]$ |
| $P3 = [X-, 10, Y-, 7, 0, Z+]$ | $P11 = [7, 5, 14, Y+, 9, Z+]$ |
| $P4 = [9, X+, 13, Y+, Z-, 5]$ | $P12 = [-1, 13, 8, 9, Z-, 14]$ |
| $P5 = [11, X+, 15, Y+, 4, Z+]$ | $P13 = [12, X+, 1, 4, Z-, 15]$ |
| $P6 = [X-, 9, 0, Y+, Z-, 7]$ | $P14 = [-1, 15, 10, 11, 12, Z+]$ |
| $P7 = [X-, 11, 3, Y+, 6, Z+]$ | $P15 = [14, X+, 2, 5, 13, Z+]$ |

Fig. 10-7 与えられる親 S が持つべきデータ構造の例

10.3.2 親から子を列挙する再帰アルゴリズム

具体的なアルゴリズムを示す前に、まずはアルゴリズムの基本的な考え方を説明する。

第 10.2.3 項において、 X 軸に消去可能な直方体分割図 R の例を示した (Fig. 10-3) が、このときは図の左から右へ向かって徐々に第 n 小直方体を消去することで子から親を定義した。本節で提案するアルゴリズムはその逆順をたどる。Fig. 10-3 の最右が与えられた親 S であるとして、右から左へと新しい第 n 小直方体を押し込んでいくことによって子の直方体分割図を生成する。ただし新しい第 n 小直方体を押し込むことができる場合の数は一通りとは限らない。押し込み可能なすべての場合を洗い出し、任意の親 S から生成可能なすべての子を列挙しなければならない。このとき問題となるのは、押し込む方向と、押し込む面と、押し込みを止める位置である。

まずは押し込む方向について考える。新しい第 n 小直方体を押し込む起点は常に親 S の G である。 G を起点に X - 軸方向、 Y - 軸方向、 Z - 軸方向のいずれかの方向に新しい第 n 小直方体を押し込んでいくのだが、押し込んだ後の V は、Fig. 10-2 で示した 7 つの TYPE の内のいずれかの TYPE になるはずである。したがって TYPE1 から 7 までのすべての場合を試す。すなわち TYPE1, 4, 7 の場合は X - 軸方向に、TYPE2, 5 の場合は Y - 軸方向に、TYPE3, 6 の場合は Z - 軸方向に押し込むことになる。

次に押し込む面について考える。新しく押し込む第 n 小直方体が直方体であるためには、押し込む面は G を含む長方形でなくてはならない。したがって、外形直方体の $X+$ 面、 $Y+$ 面、 $Z+$ 面の各面において、面を構成する頂点を用いてできる長方形の内、 G を含む長方形をすべて洗い出さなくてはならない。その押し込み可能な長方形の集合を **押し込み面** と呼び、 I と表す。押し込み面 I を取得するアルゴリズムは第 10.3.3 項で詳しく説明する。

最後に押し込みを止める位置 (**押し込み位置** V_{new} とする) について考える。押し込み面 I を押し込んでいくと、新しい第 n 小直方体は押し広げられると同時に、押し込み面 I を共有するその他の小直方体 C_i は押し縮められる。その際、 C_i とその他の小直方体との隣接関係は変化してもよいが、 C_i 以外の小直方体同士の隣接関係は変化してはならない。そのような条件を保ったまま押し込み面 I を徐々に押し込んでいくと、新しい第 n 小直方体とその他の小直方体との隣接関係が変化する場合がある。その隣接関係が異なれば、異なる直方体分割図となるので、そのすべての場合を試さなくてはならない。このように新しい第 n 小直方体とその他の小直方体との隣接関係が変化するとき、押し込み位置 V_{new} を求めるための基準となる座標を **押し込み基準座標** と呼び、 J と表す。押し込み基準座標 J を取得するアルゴリズムは第 10.3.4 項で詳しく説明する。また押し込み位置 V_{new} を求めるアルゴリズムは第 10.3.5 項で詳しく説明する。

新しい第 n 小直方体を押し込んだ後の V の TYPE が、先述した押し込む方向の TYPE

と一致しているかどうかを調べ、同じならば子の直方体分割図として出力する。

以上のような基本的な考え方を踏まえ、小直方体の数が $n-1$ 個である任意の親 S が与えられたとき、 n 個の小直方体を持つ S の子をすべて列挙するアルゴリズムを Fig. 10-8 のように提案する。

以下で Fig. 10-8 のアルゴリズムを実行順に説明する。

- STEP0: 小直方体の数が $n-1$ 個である任意の親 S を引数として渡しながらか、**enumerate_all_children** 関数を呼び出す。
- STEP1: 親 S を出力する。
- STEP2: まだ S に新しい第 n 小直方体が追加されていないのでスキップされる。
- STEP3: 以降は押し込む方向についてすべての TYPE を試す。
- STEP3-1: まずは親 S を引数として渡しながらか、TYPE1 に相当する **enumerate_type1** 関数を呼び出す。
- STEP4-1: 親 S の押し込み面 I を取得する。
- STEP4-2: それぞれの I の場合について、 $i=0$ から I の数 -1 まで for 文を回す。
- STEP4-3: 押し込み基準座標 J を取得する。
- STEP4-4: それぞれの J の場合について、 $j=0$ から J の数 -1 まで for 文を回す。(ただし、**enumerate_type1** 関数のように、T 型 (TYPE1, 2, 3) の関数の場合は J の数 -1 までだが、+ 型 (TYPE4, 5, 6, 7) の関数の場合は J の数 -2 まで for 文を回すので注意を要する。)
- STEP4-5: 押し込み位置 V_{new} を設定する。
- STEP4-6: 新しい第 n 小直方体を V_{new} まで押し込む。
- STEP4-7: 第 n 小直方体を押し込んだ後の V の TYPE が、関数の TYPE (この場合は TYPE1) と一致しているかどうかを調べ、もし異なれば、この関数が呼び出された STEP の後に戻る。
- STEP4-8: もし同じならば **enumerate_all_children** 関数を再帰的に呼び出す。このとき引数で渡されるのは新しく生成された子の直方体分割図であり、その直方体分割図では I のスライド方向、 i の値、 V の型、 j の値を新たに保持しておく。したがって新しく生成された子の直方体分割図 $S[X, i, T, j]$ のように表される。
- STEP1: 子の直方体分割図 $S[X, i, T, j]$ が出力される。
- STEP2: $S[X, i, T, j]$ が n 個の小直方体を持つので、前に **enumerate_type1** 関数が呼び出された STEP3-1 の後に戻る。以降同様に、TYPE2 から TYPE7 までに相当する STEP3-2 から STEP3-7 までの関数が呼び出されるが、STEP3-1 の関数 **enumerate_type1** と比較して、第 n 小直方体を押し込む方向が異なるだけなのでここでは省略する。

以上のアルゴリズムにより、小直方体の数が $n-1$ 個である任意の親 S が与えられたとき、 n 個の小直方体を持つ S の子をすべて列挙することができる。

同様にこのアルゴリズムを用いて、STEP0 で R_i を引数として渡せば、STEP4-8 で `enumerate_all_children` 関数を再帰的に呼び出すことにより、小直方体の数が n 個である T_n を構成することができるので、 S_n をすべて列挙できる。

次節以降で STEP4-1、STEP4-3、STEP4-5 のそれぞれのアルゴリズムの内容を詳しく説明していく。なお Fig. 10-10, 11, 12 においては、Fig. 10-9 に示す親 S が例として与えられたものとする。なお Fig. 10-9 の左は親 S の軸測投影図、右は Z 軸の基準座標階ごとの平面図である。平面図の室内の斜線は吹抜を表し、吹抜がある室は下階の室と同じ小直方体であることを意味する。

```

procedure enumerate_all_children(S) {
STEP1      S を出力 ;
STEP2      if(S が n 個の小直方体を持つ){ return; }
STEP3      // 以下押し込む方向についてすべての TYPE を試す
STEP3-1    enumerate_type1(S);      //Fig. 2 の TYPE1 に相当
STEP3-2    enumerate_type2(S);      //Fig. 2 の TYPE2 に相当
STEP3-3    enumerate_type3(S);      //Fig. 2 の TYPE3 に相当
STEP3-4    enumerate_type4(S);      //Fig. 2 の TYPE4 に相当
STEP3-5    enumerate_type5(S);      //Fig. 2 の TYPE5 に相当
STEP3-6    enumerate_type6(S);      //Fig. 2 の TYPE6 に相当
STEP3-7    enumerate_type7(S);      //Fig. 2 の TYPE7 に相当
}

procedure enumerate_type1(S) {
STEP4-1    押し込み面 I を取得する ;
STEP4-2    for i = 0 to I の数 -1 {
STEP4-3        押し込み基準座標 J を取得する ;
STEP4-4        for j = 0 to J の数 -1 { //T 型 (TYPE1, 2, 3) は J の数 -1 まで、+
// 型 (TYPE4, 5, 6, 7) は J の数 -2 まで
STEP4-5            押し込み位置  $V_{new}$  を設定する ;
STEP4-6            新しい第 n 小直方体を  $V_{new}$  まで押し込む ;
STEP4-7            if(新しい第 n 小直方体の V が TYPE1 である){
STEP4-8                enumerate_all_children(S[X,i,T,j]);
            }
        }
    }
}

STEP0      enumerate_all_children(S);

```

Fig. 10-8 親から子を列挙する再帰アルゴリズム

10.3.3 STEP4-1 : 押し込み面 I を取得するアルゴリズム

例えば、Fig. 10-9 のような S が与えられた場合、第 n 小直方体を X^- 方向に押し込んで追加することを考える。第 10.3.2 項で述べたように、新しく押し込む第 n 小直方体が直方体であるためには、押し込む面は G (Fig. 10-10 では P_{38}) を含む長方形でなくてはならない。したがって、外形直方体の X^+ 面において、面を構成する頂点を用いてできる長方形の内、 G を含む長方形をすべて洗い出さなくてはならない。

今、 S の外部から S の外形直方体の X^+ 面を、 X^- 軸方向に向かってみているものとする。まず外形直方体の X^+ 面を構成する頂点を PI に保持しておき、以降は PI 内にある頂点のみについて考える。

次に、頂点 G からスタートし、接続する頂点を反時計回りに辿りながら、訪問順にその頂点の id を I に保持していく。一度訪れた頂点は PI から削除することで再訪しないようにする。各頂点を辿っていく際に、ひとつ前に訪問した頂点から今訪問している頂点を結ぶベクトルを $VecP$ とし、今訪問している頂点から次に訪問する予定の頂点を結ぶベクトルを $VecN$ とする。もし $VecP$ を基準に $VecN$ と成す角 A が時計回りに 90 度であるならば、次に訪問する予定の頂点は訪問しない。それ以外であれば訪問する。また、もし A が反時計回りに 90 度を成すとき、その角度を A_{sum} に加算して保持しておく。頂点 G に戻ってきた時に A_{sum} が 360 度になっていれば終了し、 I を返す。もし A_{sum} が訪問の途中で 360 度より大きくなった場合、あるいは訪問する頂点がなくなった場合は I を返さずに終了する。

以上のアルゴリズムによってすべての押し込み面 I を取得することができる。Fig. 10-10 にすべての押し込み面 I を示す。

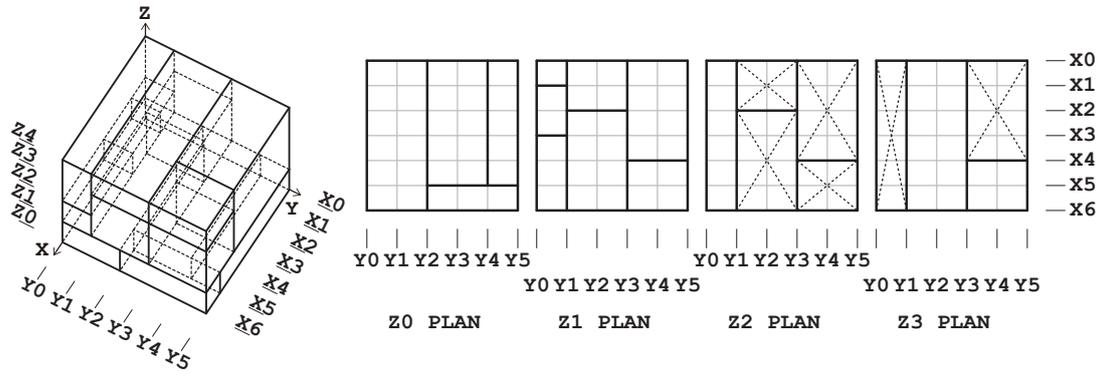


Fig. 10-9 与えられた直方体分割図 \$S\$ の例

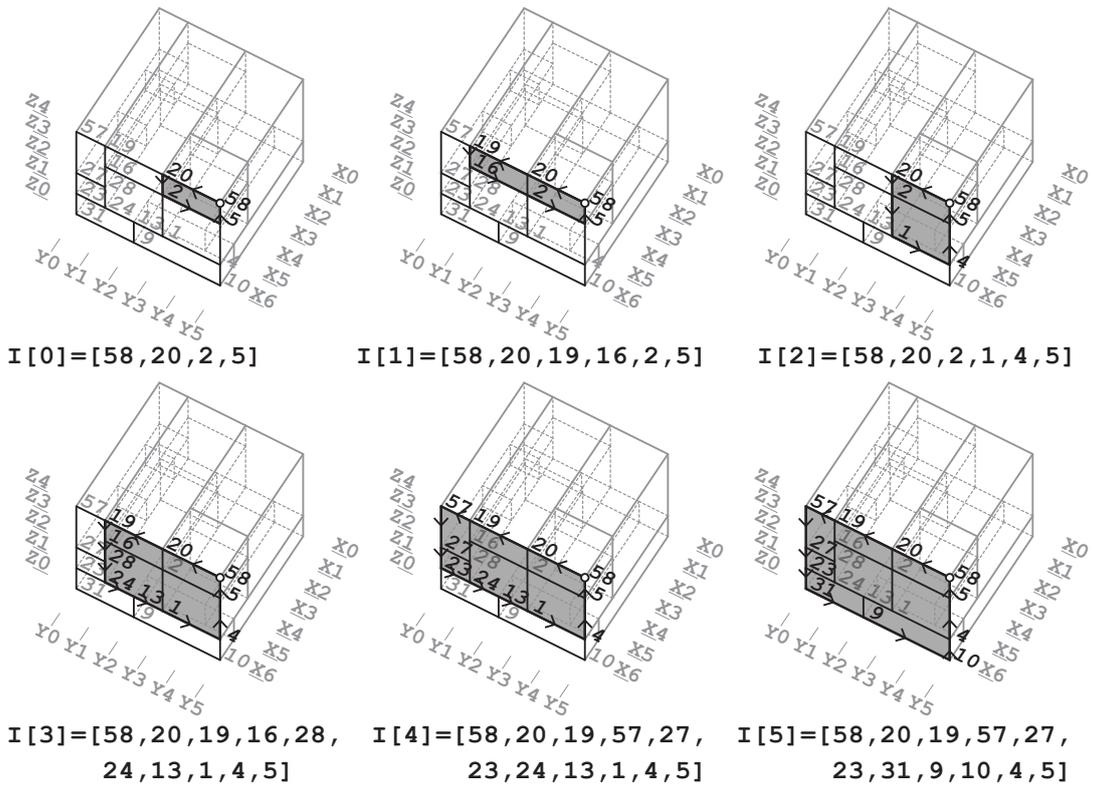


Fig. 10-10 STEP4-1: 押し込み可能な長方形 \$I\$ を取得

10.3.4 STEP4-3: 押し込み基準座標 J を取得するアルゴリズム

第10.3.2項で述べたように、押し込み面 I を X -方向に押し込むことで新しい第 n 小直方体とその他の小直方体との隣接関係が変化する場合がある。その隣接関係が異なれば、異なる直方体分割図となるので、そのすべての場合を試さなくてはならない。このように新しい第 n 小直方体とその他の小直方体との隣接関係が変化するとき、押し込み位置 V_{new} を求めるための基準となる座標が基準座標が J である。本節では、その押しこみ基準座標 J を取得するアルゴリズムを Fig. 10-11 を参照しながら説明する。Fig. 10-11 では Fig. 10-10 で取得した押し込み面 I の中で $i=3$ の場合を例に考える。

- STEP4-3-1: 押し込み面 $I[3]$ を共有するその他の小直方体の集合を C_i とする。 C_i を構成するすべての頂点の中で、最小の Y 座標を $minY$ 、最大を $maxY$ 、最小の Z 座標を $minZ$ 、最大を $maxZ$ とする。
- STEP4-3-2: C_i の中で、 Y -面の Y 座標が $minY$ である小直方体の集合を C_{i_minY} とする。
- STEP4-3-3: C_{i_minY} が Y -方向に隣接している小直方体の集合を $C_{i_minY_linkY}$ とする。
- STEP4-3-4: C_{i_minY} の中で、 X -面の X 座標が最大の小直方体を $C_{i_minY_maxX}$ とする。
- STEP4-3-5: $C_{i_minY_maxX}$ が Y -方向に隣接している小直方体の集合を $C_{i_minY_maxX_linkY}$ とする。
- STEP4-3-6: $C_{i_minY_maxX_linkY}$ の中で、 X -面の X 座標が、 $C_{i_minY_maxX}$ の X -面の X 座標以下で、かつ、 X -面の X 座標が最大の小直方体を $C_{i_minY_maxX_linkY_maxX}$ とする。
- STEP4-3-7: C_i の中で、 Z -面の Z 座標が $minZ$ である小直方体の集合を C_{i_minZ} とする。
- STEP4-3-8: C_{i_minZ} が Z -方向に隣接している小直方体の集合を $C_{i_minZ_linkZ}$ とする。
- STEP4-3-9: C_{i_minZ} の中で、 X -面の X 座標が最大の小直方体を $C_{i_minZ_maxX}$ とする。
- STEP4-3-10: $C_{i_minZ_maxX}$ が Z -方向に隣接している小直方体の集合を $C_{i_minZ_maxX_linkZ}$ とする。
- STEP4-3-11: $C_{i_minZ_maxX_linkZ}$ の中で、 X -面の X 座標が、 $C_{i_minZ_maxX}$ の X -面の X 座標以下で、かつ、 X -面の X 座標が最大の小直方体を $C_{i_minZ_maxX_linkZ_maxX}$ とする。
- STEP4-3-12: $C_{i_minY_maxX_linkY_maxX}$ の X -面の X 座標と、 $C_{i_minZ_maxX_linkZ_$

$maxX$ のX-面のX座標を比較して大きい方の小直方体を *targetC* とする。

STEP4-3-13: *targetC* の X- 面の X 座標を押し込み基準座標 *J* に追加する。さらに、 Ci_minY_linkY と Ci_minZ_linkZ の X+ 面の X 座標が *targetC* の X- 面の X 座標より大きい場合、その X 座標を *J* に追加する。*J* は座標が大きい順にソートしておく。

以上のアルゴリズムにより、押しこみ基準座標 *J* を取得することができる。Fig. 10-11 の例では、押しこみ基準座標 $J=[X6, X5, X3, X1]$ と表記しているが、実際には X 軸の基準座標と対応する各 X 座標値を保持しておく。

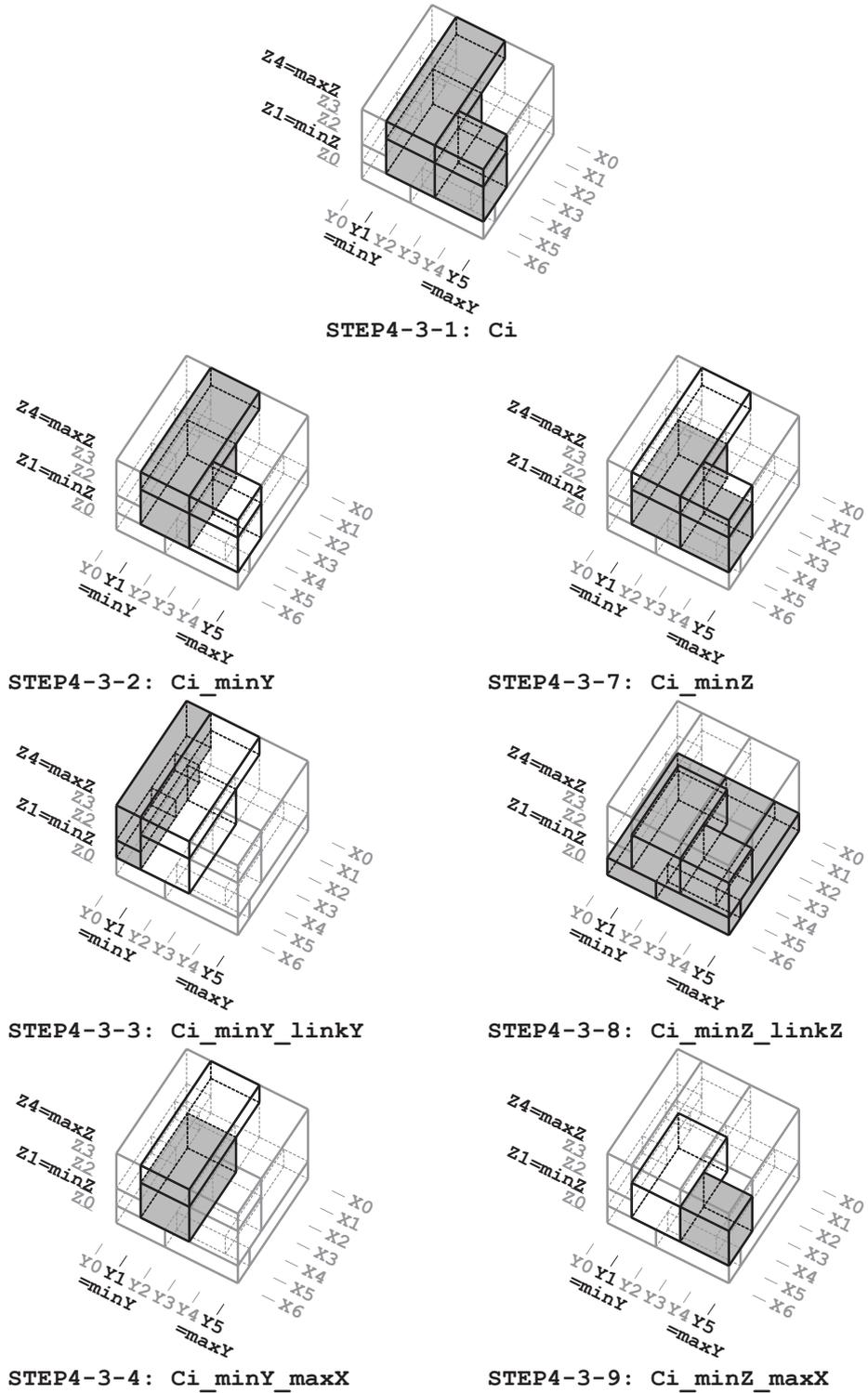


Fig. 10-11 STEP4-3: 押し込み基準座標 J を取得

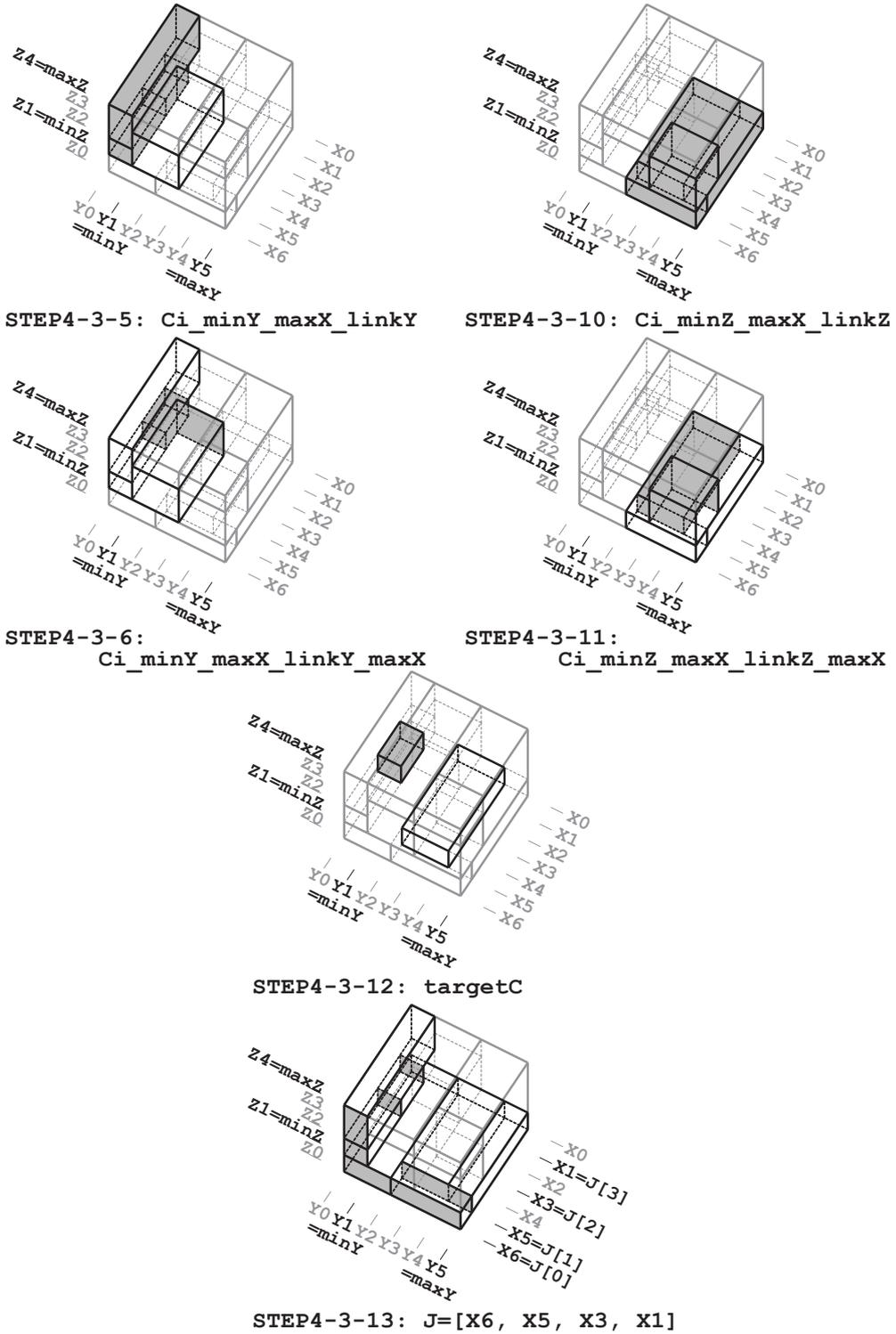


Fig. 10-11 STEP4-3: 押し込み基準座標 J を取得

10.3.5 STEP4-5: 押し込み位置 V_{new} を設定

本節では、STEP4-3 で取得した押し込み基準座標 J を基準として、押し込み位置 V_{new} を求めるアルゴリズムを Fig. 10-12 を参照しながら説明する。Fig. 10-12 における押し込み基準座標 J は、Fig. 10-11 で取得した $J=[X6, X5, X3, X1]$ を例に考える。

- STEP4-5-1: 親 S の X 軸 の 基 準 座 標 を $XList$ と す る と、 $XList=[X0, X1, X2, X3, X4, X5, X6]$ である。 $XList$ の中で、 $J[j+1]$ より小さい座標の集合を $beforeX$ とする。例えば $j=1$ の場合、 $J[2]=X3$ であるから、 $beforeX=[X0, X1, X2]$ である。
- STEP4-5-2: $XList$ の中で、 $J[j+1]$ と等しい座標を jX とする。例えば $j=1$ の場合、 $J[2]=X3$ であるから、 $jX=X3$ である。
- STEP4-5-3: 親 S を構成するすべての頂点 P_{id} の中で、その X 座標 $P_{id,x}$ が $J[0]$ より小さく、かつ、 $J[j+1]$ より大きく、かつ、 P_{id} の Y 座標 $P_{id,y}$ が $minY$ 以上で、かつ、 P_{id} の Z 座標 $P_{id,z}$ が $minZ$ 以上のとき、その X 座標 $P_{id,x}$ の集合を $editX$ とする。例えば $j=1$ の場合、 $editX=[X4]$ である。
- STEP4-5-4: 同様に $P_{id,x}$ が $J[j]$ 以上で、かつ、 $P_{id,y}$ が $minY$ 以下で、かつ、 $P_{id,z}$ が $minZ$ 以下のとき、その X 座標の集合を $afterX$ とする。例えば $j=1$ の場合、 $afterX=[X5, X6]$ である。
- STEP4-5-5: Fig. 10-8 で示したアルゴリズムでは、Fig. 10-2 の TYPE1 に相当する `enumerate_type1` 関数しか記載しなかったが、STEP4-5-5 においては、T 型 (TYPE1, 2, 3) と + 型 (TYPE4, 5, 6, 7) とで若干アルゴリズムが異なるので注意を要する。押し込み位置 V_{new} を加えた新たな X 軸の基準座標を $newXList$ とする。T 型 (TYPE1, 2, 3) の場合、 V_{new} は $editX$ の最後の座標と $afterX$ の最初の座標の midpoint とする。 $newXList$ は、 $beforeX$ 、 jX 、 $editX$ 、 V_{new} 、 $afterX$ の各座標をこの順に追加した配列となる。例えば $j=1$ の場合、 $newXList=[X0, X1, X2, X3, X4, V_{new}, X5, X6]$ となり、押し込み面 I はこの V_{new} まで押し込まれる。そのとき新しく生成される子の直方体分割図は $S[X, 3, T, 1]$ で表される。+ 型 (TYPE4, 5, 6, 7) の場合、 V_{new} は jX である。 $newXList$ は、 $beforeX$ 、 $editX$ 、 X_{new} 、 $afterX$ の各座標をこの順に追加した配列となる。例えば $j=1$ の場合、 $newXList=[X0, X1, X2, X4, X3, X5, X6]$ となり、押し込み面 I は $X3$ まで押し込まれる。そのとき新しく生成される子の直方体分割図は $S[X, 3, +, 1]$ で表される。また先述したように、 J の数を $lenJ$ とすると、STEP4-4 において、T 型 (TYPE1, 2, 3) の場合は $lenJ-1$ まで for 文を繰り返すが、+ 型 (TYPE4, 5, 6, 7) の場合は $lenJ-2$ までであること

に留意したい。

以上のアルゴリズムにより、すべての押し込み位置 V_{new} を設定することができる。

要
旨

目
次

第
1
章

第
2
章

第
3
章

第
4
章

第
5
章

第
6
章

第
7
章

第
8
章

第
9
章

第
1
0
章

第
1
1
章

第
1
2
章

後
付

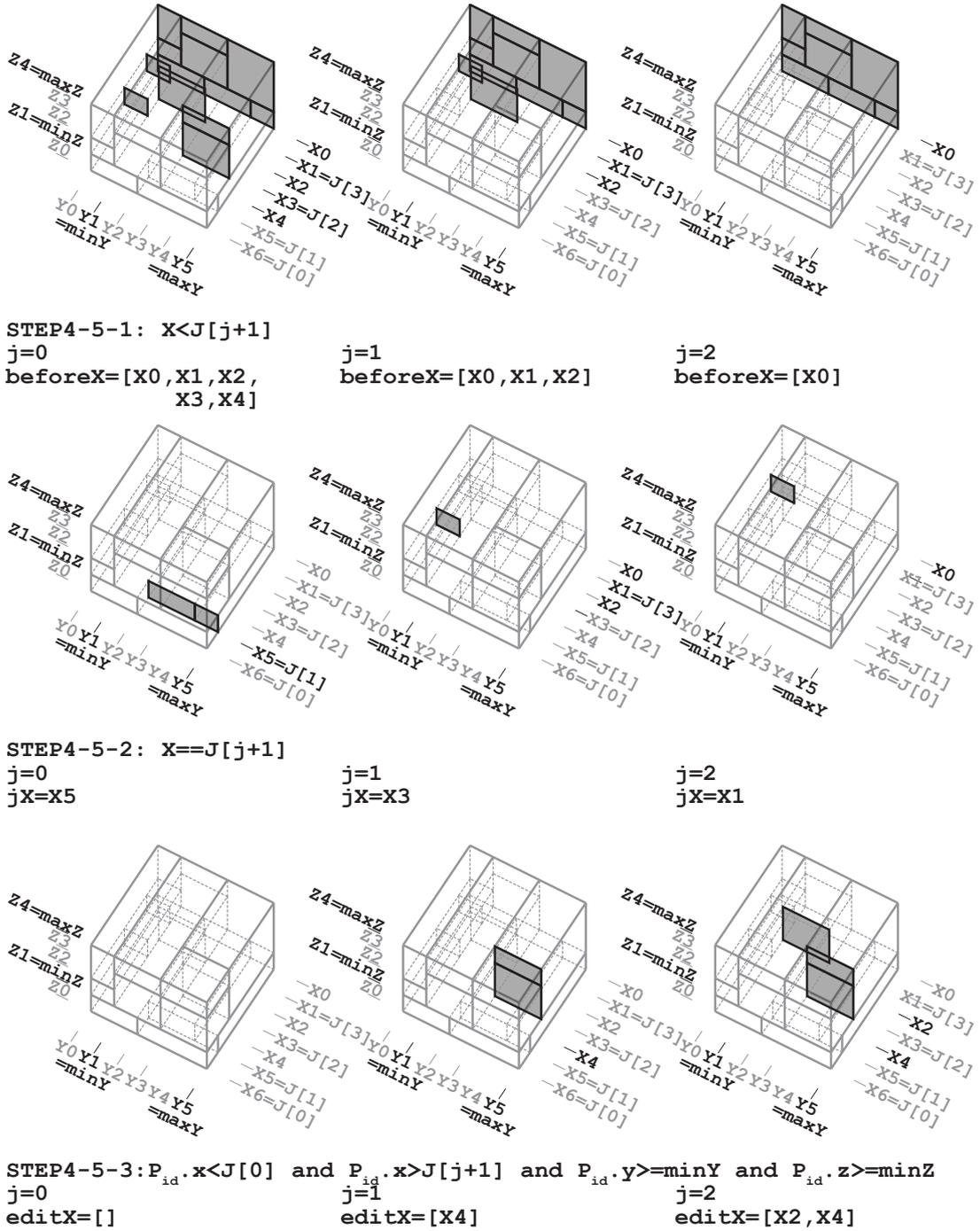
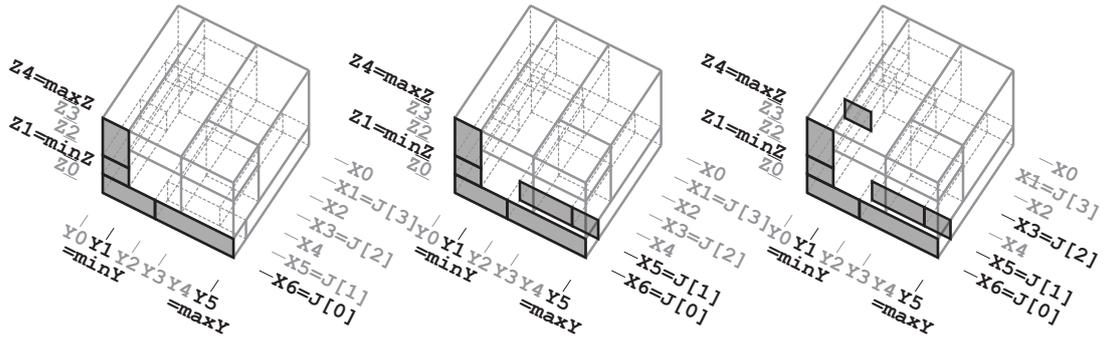
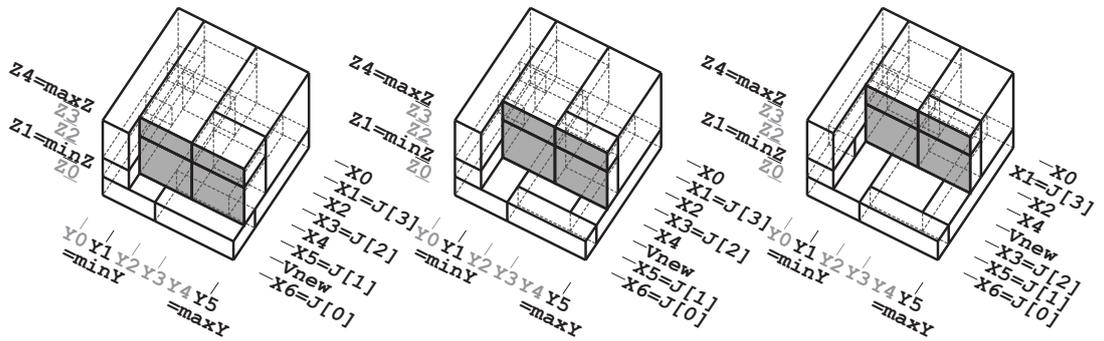


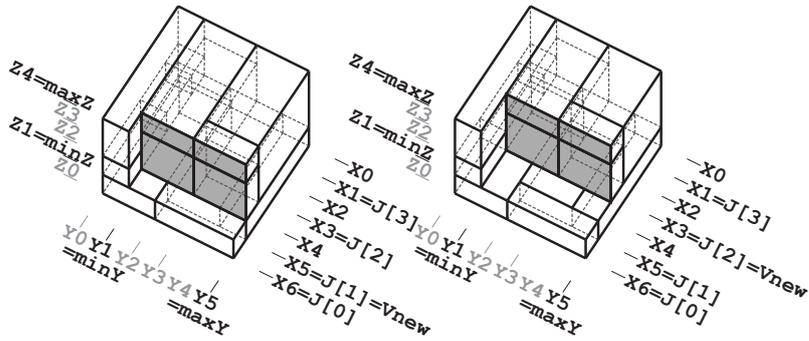
Fig. 10-12 STEP4-3: 押しこみ位置 V_{new} を設定



STEP4-5-4: $P_{id}.x > J[j]$ and $P_{id}.y \leq \min Y$ and $P_{id}.z \leq \min Z$
 j=0 afterX=[X6] j=1 afterX=[X5, X6] j=2 afterX=[X3, X5, X6]



STEP4-5-5: TYPE1の場合
 $newXList = beforeX + jX + editX + Vnew + afterX$
 j=0 newXList=[X0, X1, X2, X3, X4, X5, Vnew, X6] j=1 newXList=[X0, X1, X2, X3, X4, Vnew, X5, X6] j=2 newXList=[X0, X1, X2, X4, Vnew, X3, X5, X6]
 S[X, 3, T, 0] S[X, 3, T, 1] S[X, 3, T, 2]



STEP4-5-5: TYPE4, 7の場合
 $Xnew = jX$
 $newXList = beforeX + editX + Vnew + afterX$
 j=0 newXList=[X0, X1, X2, X3, X4, X5, X6] j=1 newXList=[X0, X1, X2, X4, X3, X5, X6]
 S[X, 3, +, 0] S[X, 3, +, 1]

Fig. 10-12 STEP4-3: 押しこみ位置 V_{new} を設定

10.4 検証

本節では、前章で提案したアルゴリズムを実装し、その実行結果を元にアルゴリズムの妥当性を検証する。

10.4.1 中野論文との比較

本節では、本研究で提案したアルゴリズムと中野ら [10-4][10-5] のアルゴリズムを比較する。

任意の直方体分割図 R の中で、階が1層 (Z 軸の基準座標が $[Z_0, Z_1]$) のみである任意の直方体分割図を U とし、そのすべて集合を U_n とする。 U_n を上からみれば2次元長方形分割図と同じであるので、 U_n をすべて列挙することができれば、漏れも重複もない2次元長方形分割図が列挙できることを意味する。

Fig. 10-2 で示した任意の直方体分割図 R の TYPE の中で、TYPE3, 5, 6, 7, 8 は Ev の中に必ず Z -軸方向への線分が含まれてしまうので、 U は必ず TYPE1, 2, 4 のいずれかの TYPE となる。またこのとき、分割面 F_{ci} が I と一致するため、TYPE1, 2, 4 の第 n 小直方体は必ず消去することができる。したがって Fig. 10-8 で示したアルゴリズムの中で STEP3-3, 5, 6, 7 を実行しなければ、 U_n を列挙することができると考えられる。

その U_n の出力結果と中野らの論文で発表されている2次元の長方形分割図の数を比較したものを表 10-1 に示す。Fig. 10-8 のアルゴリズムでは、1 から n 個までの小直方体を持つ直方体分割図がまとめて列挙されるが、この表では比較しやすいように、直方体分割図が持つ小直方体の数ごとに、出力された直方体分割図の数を分けて示す。

小直方体の数が7までの結果をみる限り、両者はよく一致している。したがって、小直方体の数が7までではあるが、本研究で提案したアルゴリズムによって列挙される直方体分割図 S_n は、漏れも重複もない2次元長方形分割図を包含していることが実証できた。

表 10-1 中野論文との比較

小直方体の数 n	中野論文の長方形分割図の数	本論文の1層の直方体分割図の数
1	1	1
2	3	3
3	9	9
4	34	34
5	162	162
6	920	920
7	5934	5934

10.4.2 吉田論文との比較

本節では、本研究で提案したアルゴリズムと吉田 [10-1] のアルゴリズムを比較する。任意の直方体分割図 R の中で、直方体分割図を構成するすべての頂点の次数が 4 以下である T 字型の任意の直方体分割図を W とし、そのすべて集合を W_n とする。Fig. 10-2 で示した任意の直方体分割図 R の TYPE の中で、TYPE1, 2, 3 は V の型が T 型であり、TYPE4, 5, 6, 7 は + 型であるので、 W は必ず TYPE1, 2, 3 のいずれかの TYPE となる。またこのとき、分割面 Fci が I と一致するため、TYPE1, 2, 3 の第 n 小直方体は必ず消去することができる。したがって Fig. 10-8 で示したアルゴリズムの中で、STEP3-4, 5, 6, 7 を実行しなければ、 W_n を列挙することができると考えられる。

その W_n の出力結果と吉田論文で発表されている T 字型のみの直方体分割図の数を比較したいのだが、吉田論文では回転および鏡面反転による置換によって同じとみなせる直方体分割図を取り除いているので単純な比較はできない。そこで吉田のアルゴリズムを元に、置換によって同じとみなせる直方体分割図を取り省かないプログラムを自作して比較した。その結果を表 10-2 に示す。ここでも表 10-1 と同様に、直方体分割図が持つ小直方体の数ごとに、出力された直方体分割図の数を分けて示している。

小直方体の数が 6 までの結果をみると、本論文で生成されて吉田論文で生成されない直方体分割図はなかったが、吉田論文で生成されるが本論文では生成されないものがいくつか認められた。そこで吉田論文のみで生成される直方体分割図をすべて調べた。その中で小直方体の数が 5 である直方体分割図を Fig. 10-13 に示す。また STEP3-4, 5, 6, 7 を実行しない Fig. 10-8 のアルゴリズムから生成した直方体分割図の一部を Fig. 10-14 に示す。この Fig. 10-14 で示した直方体分割図は、吉田論文のアルゴリズムで列挙された直方体分割図の中にも同一のものが存在する。Fig. 10-13 の (イ) と Fig. 10-14 の (イ) を比較すると、分割面の位置座標は異なるものの、隣接グラフはどちらも同じであることがわかる。それは他の (ロ)(ハ)(ニ)(ホ)(ヘ) においても同様である。さらに小直方体の数が 6 である 210 個の直方体分割図につい

表 10-2 吉田論文との比較

小直方体の数 n	吉田論文の T 字型直方体分割図の数	本論文の T 字型直方体分割図の数	吉田論文のみで生成される数	本論文のみで生成される数
1	1	1	0	0
2	3	3	0	0
3	15	15	0	0
4	99	99	0	0
5	789	783	6	0
6	7275	7065	210	0

でも調べたところ、同様の結果であった。第10.1.2項で述べたように、吉田論文の列挙アルゴリズムは、立体格子から不必要な分割面を抜き取るため、隣接グラフが同じでも、分割面の位置座標が異なれば、異なる直方体分割図として生成する。吉田論文で生成されるが本論文では生成されない直方体分割図がいくつか認められた原因は、吉田論文が対象とする直方体分割図の中に、隣接グラフが同じものが含まれるからであると考えられる。

以上より、吉田論文で生成される直方体分割図を隣接グラフが異なるものに限れば、小直方体の数が6までの結果をみる限り、両者はよく一致している。したがって、小直方体の数が6までではあるが、本研究で提案したアルゴリズムによって列挙される直方体分割図 S_n は、漏れも重複もないT字型の長方形分割図 W_n を包含していることが実証できた。

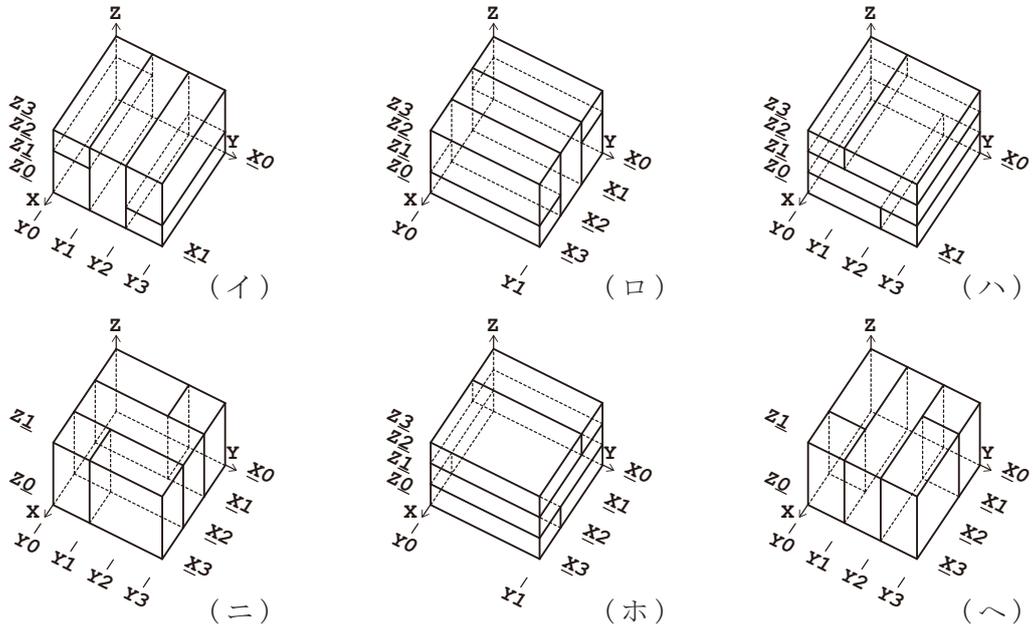


Fig. 10-13 吉田論文のみで生成される直方体分割図 (n=5)

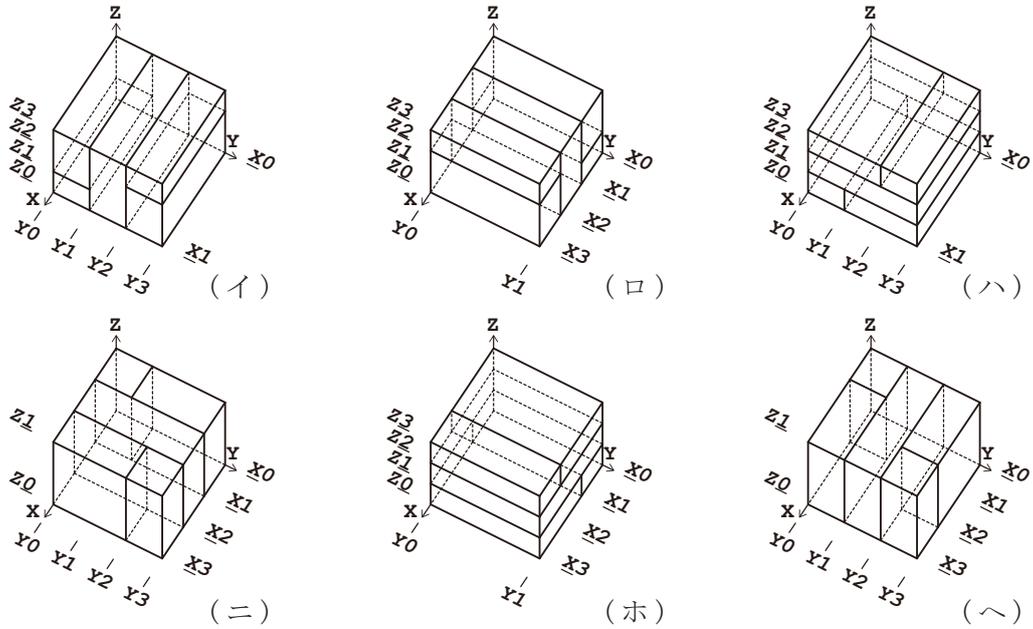


Fig. 10-14 本研究のアルゴリズムから生成された直方体分割図 (n=5)

10.4.3 列挙結果

Fig. 10-8 で示したアルゴリズムを実行し、T 字型だけでなく十字型も含めた消去可能な直方体分割図 S_n を小直方体の数が 7 まで生成した。その結果を表 10-3 に示す。この表では 1 から n 個までの小直方体を持つ直方体分割図 S_n の総数を示している。実行環境は、CPU:MacBookPro Core i7 2.8GHz、OS:Windows10、Memory:16.0GB、Software:RhinoCeros 5.0、Programming language:IronPython 2.7 で、小直方体の数が 7 までの直方体分割図の列挙に約 2 時間半を要した。

T 字型だけでなく十字型も含めた直方体分割図の列挙アルゴリズムは本研究以外では未だ提案されていないため既往研究との比較はできない。今後の研究が待たれる。

以上より、逆探索法を用いることができる範囲（すなわち消去可能な直方体分割図 S_n を対象とする限り）において、小直方体の数が 7 までではあるが、T 字型だけでなく十字型も含めた直方体分割図を求められることが実証できた。

表 10-3 列挙結果

小直方体の数 n	本論文の T 字型および十字型の直方体分割図 S_n の総数	実行時間 h:mm:ss
1	1	0:00:00
2	4	0:00:01
3	19	0:00:02
4	121	0:00:06
5	970	0:00:57
6	9164	0:10:42
7	97841	2:33:36

10.5 結論と今後の課題

本研究の結論を以下にまとめる。

- ・ 中野らの提案した 2 次元長方形分割図の列挙アルゴリズムを 3 次元へ拡張し、逆探索法を用いることができる範囲（すなわち消去可能な直方体分割図 S_n を対象とする限り）において、T 字型だけではなく十字型も含めた 3 次元直方体分割図を列挙するアルゴリズムを提案した。
- ・ 中野らの提案したアルゴリズムとの比較によって、小直方体の数が 7 までではあるが、本研究で提案したアルゴリズムによって列挙される直方体分割図 S_n は、漏れも重複もない 2 次元長方形分割図を包含していることが実証できた。
- ・ 吉田の提案したアルゴリズムとの比較によって、小直方体の数が 6 までではあるが、本研究で提案したアルゴリズムによって列挙される直方体分割図 S_n は、漏れも重複もない T 字型の長方形分割図を包含していることが実証できた。
- ・ 本論文で提案したアルゴリズムを実装し、小直方体の数が 7 までではあるが、逆探索法を用いることができる範囲（すなわち消去可能な直方体分割図 S_n を対象とする限り）において、T 字型だけではなく十字型も含めた直方体分割図を求められることが実証できた。

今後の課題を以下にまとめると、

- ・ 生成される直方体分割図に漏れや重複がないかを演繹的に証明すること。
- ・ 第 n 小直方体が消去不可能な直方体分割図も含めたすべての直方体分割図を列挙すること。

などの課題が考えられる。

要旨

目次

序章

第1部

第2部

結章

後付

第11章

第2部の小結

本章では、第7章で構築したアルゴリズムック・デザインのプロトタイプ、第8章で構築した椅子への応用 STOOLOME、そして第9章で構築した住宅への応用 ARKHITEKTOME の3つのシステムに関して、生成のフェイズ、評価のフェイズ、最適化のフェイズそれぞれの考察を述べる。また最後に第2部の小結を述べる。

11.1 生成のフェイズに関する考察

第7章で構築したアルゴリズムック・デザインのプロトタイプでは、2次元の構造体の生成だったのに対して、第8章の椅子では3次元の接合部とビームによる構造体を生成できた。またプロトタイプと椅子に関しては、接合部を母点としたドロネー四面体を用いてビームを生成したが、第9章の住宅では、隣接グラフの双対グラフとして、直方体分割図を生成することができた。

11.2 評価のフェイズに関する考察

第7章で構築したアルゴリズムック・デザインのプロトタイプ、第8章で構築した椅子への応用 STOOLOME、そして第9章で構築した住宅への応用 ARKHITEKTOME の3つのシステムにおいて、4つの層のうちどの層の〈かち〉を扱っているのかを以下にまとめる。

- (1) 第1層の〈かち〉＝物理的評価の層
 - ・ プロトタイプ
 - ・ STOOLOME
 - ・ ARKHITEKTOME
- (2) 第2層の〈かち〉＝エネルギー的評価の層
 - ・ なし
- (3) 第3層の〈かち〉＝機能的評価の層
 - ・ ARKHITEKTOME
- (4) 第4層の〈かち〉＝記号的評価の層
 - ・ プロトタイプ

以上より、第2層の〈かち〉であるエネルギー的評価に関しては本論では扱っていないことがわかる。今後の課題である。また住宅への応用において、すべての層の評価関数を実装することも今後の課題である。

11.3 高適化のフェイズに関する考察

第7章で構築したアルゴリズムック・デザインのプロトタイプは、生成個体の接合部を交配させるアルゴリズムは実装した。しかしながら、プロトタイプでは多様な個体の生態系がWEBシステム上に存在することに重きをおいて実装したため、世代を経るごとに適応度が高くなっているわけではない。

第8章で構築した椅子への応用STOOLOMEに関しては、高適化のフェイズを実装し、世代を経るに従って大局的には適応度が上がっていることは確認できた。しかしながら、一世代の生成個体数や世代数などの変数は十分にファインチューニングされているとはいえない。今後の課題である。

第9章で構築した住宅への応用ARKHITEKTOMEの3つのシステムに関しては、高適化のフェイズそのものが未実装のままである。今後の課題である。

11.4 第2部の小結

第2部の小結を以下にまとめる。

- (1) 第6章では、第4.9節で作成したアルゴリズムック・デザインのフローグラフをもとに、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインの定式化を行った。
- (2) 第7章では、第6章で得たアルゴリズムック・デザインの定式を基いて、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインのプロトタイプを構築した。
- (3) 第8章では、第7章で構築したアルゴリズムック・デザインのプロトタイプを、椅子のデザインへと帰納的に適用した。「生成のフェイズ」「評価のフェイズ」「高適化のフェイズ」に分けて、STOOLOMEというシステムを実装した。情報環境上で人工的に環境を構築し、その環境に適応する椅子を作成した。各世代の個体数や総世代数など調整の余地は残るが、全体として環境に適応する椅子を作成できたと考える。よって、アルゴリズムック・デザインのプロトタイプを、椅子のデザインへと帰納的に適用することで、その応用可能性を示すことができたと考える。
- (4) 第9章では、同じく第7章で構築したアルゴリズムック・デザインのプロトタイプを、住宅のデザインへと帰納的に適用した。「生成のフェイズ」「評価のフェイズ」に分けて、ARKHITEKTOMEというシステムを実装した。特に生成のフェイズでは「逆探索法による列挙アルゴリズム」を拡張し、与条件を満たす室の隣接関係を生成

し、室配置問題を解いている。また評価のフェイズでは、内田らのビルディング・エレメント論やBill Hillierらによる「Space Syntax 理論」を参照しながら独自の評価関数を実装し、住宅の室相互の隣接関係を定量的に評価した。「高適化のフェイズ」は、未実装のままであるものの、アルゴリズムック・デザインのプロトタイプを、住宅のデザインへと帰納的に適用することで、その応用可能性の一旦を示すことができたと考える。

- (5) 第10章では、アルゴリズムック・デザインのプロトタイプを住宅へ応用する際に、要となる基礎的なアルゴリズムである「逆探索法を用いた直方体分割図の列挙アルゴリズム」を提案し、その有効性を詳細に検証した。

以上から、アルゴリズムック・デザインのプロトタイプは、椅子や住宅へと工学的に应用可能であるであることが明らかとなった。またアルゴリズムック・デザインの住宅への応用可能性を示す上で要となる3次元直方体分割図を列挙するアルゴリズムの妥当性を詳細に検討した点に本研究の大きな成果がある。

第12章 結章

本章では、研究の結論、および今後の課題と展望を述べる。

12.1 結論

本研究の結論を以下にまとめる。

- (1) 第2章において、各種建築設計プロセスを位置づけるための「進化論的枠組み」を仮説的に提案した。
- (2) 第3章では、その「進化論的枠組み」から演繹的に予想される帰結としてグラフ理論を用いた建築設計プロセスのフローグラフを導出した。
- (3) 第4章では、各種設計プロセスを「進化論的枠組み」の中に帰納的に位置づけた。また各種設計プロセスのフローグラフを作成することにより、様々な環境に適合した建築物がどのように生成されているのかそのメカニズムを明らかにした。
- (4) 第5章では、各種設計プロセスのフローグラフをもとに分類を行い、それらの共通点や相違点を考察した。

以上から、アルゴリズムック・デザインとその他の各種設計プロセスを「進化論的枠組み」の中に位置づけることを通して、生成のフェイズ、評価のフェイズ、高適化のフェイズのいずれのフェイズにもアルゴリズムを用い、それらのフェイズを繰り返すことによって情報環境内で建築の試行錯誤を行っているという点が、他の設計プロセスにはないアルゴリズムック・デザインの特異性であることが明らかとなった。

したがって第1部の目的である「アルゴリズムック・デザインとその他の各種設計プロセスを同じ枠組みの中に位置づけることを通して、それらの共通点や相違点を議論するための理論的な枠組みを構築すること」は、基礎的な概論に留まるものの、達成されたと考える。

- (5) 第6章では、第4.9節で作成したアルゴリズムック・デザインのフローグラフをもとに、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインの定式化を行った。
- (6) 第7章では、第6章で得たアルゴリズムック・デザインの定式を基いて、生成のフェイズ、評価のフェイズ、高適化のフェイズという3つのフェイズに分けて、アルゴリズムック・デザインのプロトタイプを構築した。
- (7) 第8章では、第7章で構築したアルゴリズムック・デザインのプロトタイプを、椅子のデザインへと帰納的に適用した。「生成のフェイズ」「評価のフェイズ」「高適化のフェイズ」に分けて、STOOLOMEというシステムを実装した。情報環境上で人工的に環境を構築し、その環境に適応する椅子を作成した。各世代の個体数や総世代数など調整の余地は残るが、全体として環境に適応する椅子を作成できたと考える。よって、アルゴリズムック・デザインのプロトタイプを、椅子のデ

ザインへと帰納的に適用することで、その応用可能性を示すことができたと考える。

- (8) 第9章では、同じく第7章で構築したアルゴリズムック・デザインのプロトタイプを、住宅のデザインへと帰納的に適用した。「生成のフェイズ」「評価のフェイズ」に分けて、ARKHITEKTOME というシステムを実装した。特に生成のフェイズでは「逆探索法による列挙アルゴリズム」を拡張し、与条件を満たす室の隣接関係を生成し、室配置問題を解いている。また評価のフェイズでは、内田らのビルディング・エレメント論やBill Hillierらによる「Space Syntax 理論」を参照しながら独自の評価関数を実装し、住宅の室相互の隣接関係を定量的に評価した。「高適化のフェイズ」は、未実装のままであるものの、アルゴリズムック・デザインのプロトタイプを、住宅のデザインへと帰納的に適用することで、その応用可能性の一旦を示すことができたと考える。
- (9) 第10章では、アルゴリズムック・デザインのプロトタイプを住宅へ応用する際に、要となる基礎的なアルゴリズムである「逆探索法を用いた直方体分割図の列挙アルゴリズム」を提案し、その有効性を詳細に検証した。

以上から、アルゴリズムック・デザインのプロトタイプは、椅子や住宅へと工学的に応用可能であるであることが明らかとなった。またアルゴリズムック・デザインの住宅への応用可能性を示す上で要となる3次元直方体分割図を列挙するアルゴリズムの妥当性を詳細に検討した点に本研究の大きな成果がある。

したがって、第2部の目的である「アルゴリズムック・デザインを定式化し、その基礎的なシステムモデルであるプロトタイプを構築することで、椅子や住宅のデザインへの工学的な応用可能性を示すこと」は、本論文において設定した限定された条件下ではあるものの、達成されたと考える。

12.2 今後の課題と展望

本研究の今後の課題と展望を下記にまとめる。

- (1) 第4章では、帰納的に各種設計プロセスを位置づけたため、本研究で扱わなかった設計プロセスによっては、「建築設計プロセスの進化論的枠組み」の中に位置づけられない可能性は残る。しかしながらそれは本研究に限らず、帰納的論証の宿命であるので、今後、本研究では扱わなかった設計プロセスについても同じ枠組みの中に位置づけられるかどうかを検証したい。
- (2) 「建築設計プロセスの進化論的枠組み」を用いて、次世代の新しい設計プロセスを生成することも今後の課題である。
- (3) 本研究では、第7～9章の評価のフェイズにおいて、第2層の〈かち〉であるエネルギー的評価を扱っていない。今後の課題である。また第9章の住宅への応用において、すべての層の評価関数を実装することも今後の課題である。
- (4) 本研究では、第7～9章高適化のプロセスにGAを用いた。しかしながら、適応地形を登ることができるならば、必ずしもGAである必然性はない。今後、その他のヒューリスティックを用いた手法も検討する必要がある。
- (5) 第8章の椅子への応用では、各世代の個体数や、総世代数などの各パラメータをファインチューニングするところまでは至っていない。今後の課題である。
- (6) 第9章の住宅への応用では、高適化のプロセスを実装することも、今後の課題である。

以上。

参考文献

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

1. 第1章

- [1-1] 日本建築学会編『アルゴリズムック・デザイン——建築・都市の新しい設計手法』（鹿島出版会、2009、pp8）
- [1-2] 菊竹清訓『代謝建築論——か・かた・かたち』（彰国社、第1版1969、復刻版2008）。
- [1-3] 設計方法に関する国際会議は、1962年ロンドン、1965年バーミンガム、1967年ポーツマスで開催された。
- [1-4] 日本建築学会の設計方法小委員会からは下記5冊の書籍が彰国社より発表されている。設計方法パンフレットⅠ『設計方法』（1968）、設計方法パンフレットⅡ『設計プロセス／ケーススタディ』（1971）、設計方法パンフレットⅢ『設計プロセス／道具の提案』（1974）、設計方法パンフレットⅤ『設計方法と設計主体』（1989）、『人間——環境系のデザイン』（1997）。
- [1-5] 磯崎新「プロセス・プランニング論」（『空間へ』鹿島出版会、1997）。
- [1-6] C. アレグザンダー『Note on the Synthesis of Form』（原著は1964年出版）。邦訳は『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978）。
- [1-7] 設計方法パンフレットⅠ『設計方法』（彰国社、1968）序文より。
- [1-8] 難波和彦『戦後モダニズムの極北——池辺陽試論』（彰国社、1999、pp186）
- [1-9] 浜田邦裕（編集協力）、メディア・デザイン研究所（編集制作）『10+1 NO.6 特集＝サイバーアーキテクチャー』（INAX出版、1996）
- [1-10] 渡辺誠『建築は、柔らかい科学に近づく——INDUCTION DESIGN／進化設計論』（建築資料研究社、2002）
- [1-11] 三井和男、大森博司、田川浩、本間俊雄、大崎純『発見的最適化手法による構造のフォルムとシステム』（コロナ社、2004）
- [1-12] 佐々木睦朗『フラックス・ストラクチャー』（TOTO出版、2005）
- [1-13] セシル・バルモンド『インフォーマル』（TOTO出版、2005）
- [1-14] コスタス・テルジディス『アルゴリズムック・アーキテクチュア』（彰国社、2010）
- [1-15] 渡辺誠『アルゴリズムック・デザイン実行系 建築・都市設計の方法と理論 ALGOrithmic Design EXecution and logic』（丸善出版、2012）
- [1-16] Achim Menges、Sean Ahlquist『Computational Design Thinking: Computation Design Thinking』（Wiley、2011）
- [1-17] Wassim Jabi『Parametric Design for Architecture』（Laurence King Publishing、2013）
- [1-18] Ilaria Mazzoleni『Architecture Follows Nature-Biomimetic Principles for Innovative Design』（CRC Press、2013）
- [1-19] 米盛裕二『アブダクション——仮説と発見の論理』（勁草書房、2007）
- [1-20] 山口純、門内輝行：設計プロセスにおける論理的、倫理的、美学的次元の関係、C.S. パーソンの規範学に基づく探求としての設計プロセスのモデル、日本建築学会計画系論文集、Vol. 79、No. 703、pp1881-1890、2014. 09

2. 第2章

- [2-1] C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp65）
- [2-2] J.C. ジョーンズ『デザインの手法——人間未来への手がかり』（池辺陽訳、丸善、1973、pp82）
- [2-3] 日本建築学会建築計画委員会『設計方法Ⅱ 設計プロセス／ケーススタディ』（彰国社、1973、pp4）
- [2-4] 吉川弘之：設計とは何か：一般設計学の試み（〈特集〉設計論特集号）、日本機械学会誌、Vol. 84、No. 749、pp. 328-335、1981. 04、を元に筆者作成
- [2-5] ユクスキュル／クリサート『生物から見た世界』（日高敏隆＋羽田節子訳、岩波書店、2005）。原著は1934年。
- [2-6] 菊竹清訓『代謝建築論——か・かた・かたち』（彰国社、復刻版2008、ppi）
- [2-7] 武谷三男『弁証法の諸問題』（勁草書房、1968、新装版初版2010）。
- [2-8] 「著書の解題——5 『代謝建築論』 菊竹清訓」（『INAX REPORT』No. 171、INAX 出版、2007）。
- [2-9] 吉村靖孝「トライアングル・モデル」（『建築文化』2003年12月号、彰国社、2003）。
- [2-10] 武谷三男『弁証法の諸問題』（勁草書房、1968、新装版初版2010、pp109）
- [2-11] 菊竹清訓『代謝建築論——か・かた・かたち』（彰国社、復刻版2008、pp14）
- [2-12] 菊竹清訓『代謝建築論——か・かた・かたち』（彰国社、復刻版2008、pp7）
- [2-13] 菊竹清訓『代謝建築論——か・かた・かたち』（彰国社、復刻版2008、pp13）
- [2-14] 米盛裕二『アブダクション——仮説と発見の論理』（勁草書房、2007、pp54）
- [2-15] 米盛裕二『アブダクション——仮説と発見の論理』（勁草書房、2007、pp54）
- [2-16] 米盛裕二『アブダクション——仮説と発見の論理』（勁草書房、2007、pp68）
- [2-17] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、p. 146）
- [2-18] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、p. 161）
- [2-19] 日本建築学会設計方法小委員会：設計方法パンフレットⅠ『設計方法』（彰国社、1968）
- [2-20] 日本建築学会設計方法小委員会：設計方法パンフレットⅡ『設計プロセス／ケーススタディ』（彰国社、1971）
- [2-21] 日本建築学会設計方法小委員会：設計方法パンフレットⅢ『設計プロセス／道具の提案』（彰国社、1974）
- [2-22] 日本建築学会設計方法小委員会：設計方法パンフレットⅣ『設計方法論』（彰国社、1981）
- [2-23] 日本建築学会設計方法小委員会：設計方法パンフレットⅤ『設計方法と設計主体』（彰国社、1989）
- [2-24] 日本建築学会編『人間——環境系のデザイン』（彰国社、1997）
- [2-25] 門内輝行『人間—環境系のデザインの展望——21世紀のデザインビジョン』（日本機械学会、No. 06-5、Design シンポジウム2006 講演論文集）

3. 第3章

- [3-1] C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp12）
- [3-2] C・アレグザンダー『パターン・ランゲージ——環境設計の手引』（平田翰那訳、鹿島出版会、1984、原著は1977年発表）

- [3-3] J.C. ジョーンズ『デザインの手法——人間未来への手がかり』（池辺陽訳、丸善、1973、pp82）
- [3-4] 藤村龍至『批判的工学主義の建築：ソーシャル・アーキテクチャをめざして』（エヌティティ出版、2014）
- [3-5] 山梨知彦『業界が一変する BIM 建設革命』（日本実業出版社、2009）
- [3-6] Wassim Jabi『Parametric Design for Architecture』（Laurence King Publishing、2013）
- [3-7] Ilaria Mazzoleni『Architecture Follows Nature-Biomimetic Principles for Innovative Design』（CRC Press、2013）
- [3-8] 日本建築学会編『アルゴリズム・デザイン——建築・都市の新しい設計手法』（鹿島出版会、2009）
- [3-9] Achim Menges、Sean Ahlquist『Computational Design Thinking: Computation Design Thinking』（Wiley、2011）
- [3-10] ローレンス・レッシング『CODE——インターネットの合法・違法・プライバシー』（山形浩生＋柏木亮二訳、翔泳社、2001、pp157）

4. 第4章

- [4-1] C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp44）
- [4-2] C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp17）
- [4-3] C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp31）
- [4-4] 中谷礼仁『セヴェラルネス——事物連鎖と人間』（鹿島出版会、2005）
- [4-5] 「無自覚なシステムでは、個人は一つの代行機関であった」C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp49）
- [4-6] B・ルドフスキー『建築家なしの建築』（渡辺武信訳、SD 選書、1984）
- [4-7] J.C. ジョーンズ『デザインの手法——人間未来への手がかり』（池辺陽訳、丸善、1973、pp20）
- [4-8] 「材料は、もはや手元にはない。建物は、耐久性のあるものとなり、補修とか調整を繰り返すことは、以前にくらべると少なくなる。建設はもはや住む人の手から離れ、不都合が起これると、専門家がそれを確認し永久的な処置をする前に、幾度か報告され説明されねばならない。（…中略…）不都合が修正を誘導するにはかなりの時間が必要になる。伝統の固さもまた溶けてしまう。任意的な変更に対する抵抗力は弱まり、変化のための変化が受け入れられるようになる。（…中略…）以前はゆっくりと動いていて、適応の時間も十分にあった文化が、今は、あまりにも早く変化するので、適応がそれについていけなくなる。（…中略…）プロセスの根本的な条件、それが平衡にいたるまでに時間を要するという事実、が破られる」C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp47）
- [4-9] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp110）
- [4-10] 日埜直彦「ポストモダニズムと建築」（『10 + 1』No. 31、pp18-19）<http://tenplusone-db.inax.co.jp/backnumber/article/articleid/1119/>
- [4-11] その他の原因としては、中谷礼仁が『セヴェラルネス』で明らかにしたように、メタポリズム建築のスケルトンが「強い技術」で作られていたために転用できなかったという「技術的・実的な問題」

- が挙げられる。中谷礼仁『セヴェラルネス——事物連鎖と人間』（鹿島出版会、2005、pp99-103）
- [4-12] 磯崎新『空間へ』（鹿島出版会、1997、pp66）
- [4-13] 米盛裕二『パースの記号学』（勁草書房、1981、pp110）
- [4-14] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp225）
- [4-15] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp160）
- [4-16] 磯崎新『手法が——カウンター・アーキテクチュア』（鹿島出版会、1997、pp62）
- [4-17] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp136）この論考ではまだ第2層のエネルギー性は分離されていないが、ここではこの論考に基づきながら、私なりに関係主義を定式化している。
- [4-18] 難波和彦「箱の家の展開」<http://www.kai-workshop.com/archives/002.html>
- [4-19] 磯崎新『手法が——カウンター・アーキテクチュア』（鹿島出版会、1997、pp9）
- [4-20] 米盛裕二『パースの記号学』（勁草書房、1981、pp143-159）(1) 類似記号 (icon, iconic sign) とは、その記号の性質がその対象の性質と類似している記号のことをいう。写真や地図など。(2) 指標記号 (index, indexical sign) とは、その指示対象とのあいだに特に重要な類似関係をもっていないが、物理的な対応関係をもつ記号のことをいう。風見鶏や温度計など。(3) 象徴記号 (symbol, symbolic sign) とは、その指示対象とは特別な類似性も物理的な対応関係もないが、一般観念や習慣がその対象との結びつきを作り出した記号のことをいう。家紋やブランドロゴなど。
- [4-21] ジャン ボードリヤール『消費社会の神話と構造 普及版』（紀伊國屋書店、1995、pp67）
- [4-22] ローレンス・レッシング『CODE ——インターネットの合法・違法・プライバシー』（山形浩生+柏木亮二訳、翔泳社、2001、pp157）
- [4-23] C・アレグザンダー『パタン・ランゲージ——環境設計の手引』（平田翰那訳、鹿島出版会、1984、原著は1977年発表）
- [4-24] 藤村龍至『批判的工学主義の建築：ソーシャル・アーキテクチャをめざして』（エヌティティ出版、2014）
- [4-25] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp210）
- [4-26] 藤村龍至「グーグル的建築家像をめざして——「批判的工学主義」の可能性」（『思想地図』vol. 3、NHK 出版、2009、pp91）
- [4-27] 江渡浩一郎は『パターン、Wiki、XP ——時を超えた創造の原則』（技術評論社、2009）の中で、パターン、Wiki、XP といった情報環境を代表する技術の起源が、アレグザンダーのパタンランゲージにあることを明らかにしている。
- [4-28] 藤村龍至「グーグル的建築家像をめざして——「批判的工学主義」の可能性」（『思想地図』vol. 3、NHK 出版、2009、pp105）
- [4-29] 濱野智史『アーキテクチャの生態系——情報環境はいかに設計されてきたか』（NTT 出版、2008）p. 18
- [4-30] この限界のほかにも、たとえば、読み込んだコンテキストの変数が10個あり、それらの変数に適合する部分的な〈かたち〉の可能性がそれぞれ3パターンあるとしたら、本来すべての〈かたち〉の可能性は、 $3^{10}=59049$ パターンあるはずである。「超線形設計プロセス」はそのパターン探索のうちで、ひとつの経路しか探索していないことになる。それで「濃密な」建物が出力できるのかと

以前質問したら、藤村氏からは「他者と政治的な合意形成さえできれば問題はない」という答えが返ってきた。とすれば「超線形設計プロセス」はコンテキストによりよく適合する形を出力するための方法というよりは、政治的な合意形成の手法となるのではないだろうか。

- [4-31] 山梨知彦『業界が一変する BIM 建設革命』（日本実業出版社、2009）
- [4-32] 東浩紀『動物化するポストモダン——オタクから見た日本社会』（講談社現代新書、2001）において指摘され、建築分野にもすでに多くの兆候があるように思う。
- [4-33] コスタス・テルジディス『アルゴリズムック・アーキテクチャ』（田中浩也監訳、荒岡紀子+重村珠穂+松川昌平訳、彰国社、2010）。コスタスは「ツール・ユーザー」と「ツール・メーカー」あるいは「コンピュータライゼーション」と「コンピューテーション」を対比させ、現状の建築分野におけるコンピュータの利用方法を批判し、本来コンピュータが持つ計算という概念を用いた建築デザインのあり方を問い直している。
- [4-34] ローレンス・レッシング『CODE——インターネットの合法・違法・プライバシー』（山形浩生+柏木亮二訳、翔泳社、2001）。レッシングは、人の行動や社会秩序をコントロールするための方法を「規範」「法律」「市場」「アーキテクチャ」の4つに分類し、デジタル・コンテンツの不正コピー防止技術を例に挙げ、情報環境の「アーキテクチャ」によってユーザーが無意識のうちにコントロールされる危険性を指摘する。
- [4-35] 前掲載書『アーキテクチャの生態系』において、濱野智史は「アーキテクチャ（環境管理型権力）」という概念を次のように要約する。(1) 任意の行為の可能性を「物理的に」封じてしまうため、ルールや価値観を被規制者の側に内面化させるプロセスと必要としない。(2) その規制（者）の存在を気づかせることなく、被規制者が「無意識」のうちに規制を働きかけることが可能。
- [4-36] ニール・ガーシェンフェルド『ものづくり革命——パーソナル・ファブ리케이션の夜明け』（ソフトバンククリエイティブ、2006）
- [4-37] 「すべての可能な不適合の選言である。」（選言とは、英語でいうと「OR」、日本語では「あるいは」「または」の意。論理和。）C. アレグザンダー『形の合成に関するノート』（稲葉武司訳、鹿島出版会、1978、pp19）
- [4-38] 難波和彦『戦後モダニズムの極北——池辺陽試論』（彰国社、1999、pp186）
- [4-39] C・アレグザンダー『パタン・ランゲージ——環境設計の手引』（平田翰那訳、鹿島出版会、1984、原著は1977年発表）
- [4-40] 難波和彦『建築の四層構造——サステイナブル・デザインをめぐる思考』（INAX 出版、2009、pp203）
- [4-41] 第六世代には、セシル・バルモンドの『informal』（TOTO 出版、2005）を、第八世代には、佐々木睦朗の『フラックス・ストラクチャー』（TOTO 出版、2005）を、それぞれ対応させると、その違いを感覚的にイメージしやすいのではないだろうか。
- [4-42] 一例として、『JA77 建築と都市のアルゴリズム』（新建築社、2010）に掲載されている拙作「Topological Grid」を参照のこと。
- [4-43] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009）

5. 第5章

- [5-1] コスタス・テルジディス『アルゴリズムック・アーキテクチャ』（田中浩也監訳、荒岡紀子+重村珠穂+松川昌平訳、彰国社、2010）

6. 第6章

- [6-1] スチュアート・カウフマン『自己組織化と進化の論理』（米沢富美子監訳、日本経済新聞社、1999、pp271）
- [6-2] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009、pp3）
- [6-3] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009、pp19）
- [6-4] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009、pp25）
- [6-5] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009、pp61）

7. 第7章

- [7-1] リチャード・ドーキンス『盲目の時計職人—自然淘汰は偶然か？』（早川書房、2004、pp111）の「バイオモルフ」のコンピュータ・プログラムにインスピレーションを得ている。
- [7-2] 《ARKHITEKTOME 2013》、16th DOMANI 明日展、出展、@国立新美術館、東京、2013.12.14 - 01.26
- [7-3] 久保幹雄+J・P・ペドロソ『メタヒューリスティクスの数理』（共立出版、2009、pp61）
- [7-4] 伊庭斉志『遺伝的アルゴリズムの基礎—GAの謎を解く』（オーム社、1994）
- [7-5] 平野廣美『応用事例でわかる遺伝的アルゴリズムプログラミング』（パーソナルメディア、1995）
- [7-6] 古田均、杉本博之『遺伝的アルゴリズムの構造工学への応用』（森北出版、1997）
- [7-7] <http://arkhitektome.com>

8. 第8章

- [8-1] 《stoolome》、ORF2014、出展、@六本木ミッドタウン、東京、2014.11.12 - 11.16

9. 第9章

- [9-1] 《ARKHITEKTOME 2015》、ORF2015、出展、@六本木ミッドタウン、東京、2015.11.21-22
- [9-2] 《ARKHITEKTOME 2015》、ARKHITEKTOME / アルキテクトーム 2015 展、個展、@プリズミックギャラリー、東京、2015.10.23-11.17

- [9-3] 吉田 勝行、直方体分割図の電算機による作成法と設計計画への適用性に関する研究 (1) : 直方体分割図の機械的な作成法とそのプログラム化、日本建築学会論文報告集、第 293 号、pp. 105-115、1980. 7
- [9-4] 神山 直之、瀧澤 重志、加藤 直樹：フロアプランの列挙に基づく 2 階建て住宅の室配置アルゴリズム、日本建築学会環境系論文集、第 601 号、pp. 65-72、2006. 3
- [9-5] NAKANO Sin-ichi:Enumerating Floorplans with n Rooms, IEICE transactions on fundamentals of electronics, communications and computer sciences, Vol. 85, No. 7, pp. 1746-1750, 2002. 7
- [9-6] David Avis, Komei Fukuda:Reverse search for enumeration, Discrete Applied Mathematics, Vol. 65, pp. 21-46, 1996
- [9-7] 松川昌平：逆探索法を用いた直方体分割図の列挙アルゴリズム、日本建築学会計画系論文集、第 82 巻、第 735 号、2017. 5
- [9-8] 内田祥哉 宇野英隆 井口洋佑：BuildingElement の定義に就いて、1959. 6
- [9-9] 真鍋恒博：図解 建築構法計画講義 「ものしぐみ」から建築を考える、彰国社、1999
- [9-10] 内田祥哉 宇野英隆 井口洋佑：BuildingElement の性格の種類に就て、1959. 6
- [9-11] 内田祥哉 宇野英隆 井口洋佑：BuildingElement の性格に就て、1959. 10
- [9-12] 内田祥哉 他：BuildingElement の考え方に就て、1960. 10
- [9-13] 井口洋佑 内田祥哉 剣持吟：BuildingElement の位置による分類、1961. 1
- [9-14] 内田祥哉：BuildingElement の性質の単位について、1961. 1
- [9-15] 原広司 内田祥哉：熱・音と BuildingElement の関係について、1961. 6
- [9-16] 内田祥哉：建築構法、市ヶ谷出版社、1981
- [9-17] Bill Hillier、Julienne Hanson 『The Social Logic of Space』(Cambridge University Press、1989)
- [9-18] 佐々木雅宏、松川昌平「建築の空間構成における、視線、動線、空気の関係性を定量的に評価する手法の提案」、日本建築学会大会学術講演梗概集(九州)、2016

10. 第 10 章

- [10-1] 吉田勝行：直方体分割図の電算機による作成法と設計計画への適用性に関する研究 (1) 直方体分割図の機械的な作成法とそのプログラム化、日本建築学会論文報告集、第 293 号、pp. 105-115、1980. 7
- [10-2] 吉田勝行：直方体分割図の電算機による作成法と設計計画への適用性に関する研究 (2) 直方体分割図の種類と、それを母体としたラインプランに納め得る設計計画上の条件、日本建築学会論文報告集、第 295 号、pp. 103-113、1980. 9
- [10-3] Avis, D. and Fukuda, K. : Reverse search for enumeration, Discrete Applied Mathematics, Vol. 65, pp. 21-46, 1996
- [10-4] Nakano, S. : Enumerating Floorplans with n Rooms, IEICE transactions on fundamentals of electronics, communications and computer sciences, Vol. 85, No. 7, pp. 1746-1750, 2002. 7
- [10-5] 吉井訓史、中野真一：フロアプラン列挙アルゴリズムの実装、電子情報通信学会技術研究報告 COMP コンピューテーション、Vol. 103, No. 119, pp. 7-14, 2003. 6

- [10-6] 神山直之, 瀧澤重志, 加藤直樹: フロアプランの列挙に基づく2階建て住宅の室配置アルゴリズム, 日本建築学会環境系論文集, 第601号, pp.65-72, 2006.3
- [10-7] 藤本壮介建築設計事務所: House H, 新建築, Vol.84, No.13, pp.148-155, 2009.12
- [10-8] 藤本壮介建築設計事務所: House NA, JA, No.83, pp.32-41, 2011
- [10-9] 岡崎甚幸, 伊藤明宏: 室・通路・出入口の最適配置モデルの研究, 日本建築学会論文報告集, 第311号, pp.75-81, 1982.1
- [10-10] 岩田伸一郎, 宗本順三, 吉田哲, 阪野明文: 移動コストを評価関数とした室配置へのGA適用と発想支援「An approach to the optimum layout of single-storey buildings」における病院手術棟を事例として, 日本建築学会計画系論文集, 第519号, pp.341-347, 1999.5
- [10-11] 岩田伸一郎, 水沼靖昭, 宗本順三: 大学の組織構成要素のまとまりと要求面積充足度に基づいた既存校舎への室再配置計画, 日本建築学会計画系論文集, 第565号, pp.167-173, 2003.3
- [10-12] 岩田伸一郎, 宗本順三: 自律的な大学組織の相制関係に基づいた施設配置計画法, 日本建築学会計画系論文集, 第572号, pp.99-106, 2003.10
- [10-13] 岡崎甚幸, 伊藤明宏: 逐次近似型室配置・通路モデルの研究, 日本建築学会論文報告集, 第339号, pp.90-100, 1984.5
- [10-14] 青木義次: プラン作成と遺伝進化とのアナロジー 室配置問題の遺伝進化アルゴリズムによる解法, 日本建築学会計画系論文集, 第481号, pp.151-156, 1996.3
- [10-15] 村岡直人, 青木義次: 遺伝的アルゴリズムによる平面形状の最適化と設計ノウハウの獲得, 日本建築学会計画系論文集, 第497号, pp.111-115, 1997.7
- [10-16] 岩田伸一郎, 宗本順三, 吉田哲, 阪野明文: 移動コストを評価関数とした廊下パターンと室配置へのGA適用「An approach to the optimum layout of single-storey buildings」における病院手術棟を事例として, 日本建築学会計画系論文集, 第518号, pp.329-333, 1999.4
- [10-17] 寺田秀夫: 室空間の隣接関係により定義された長方形分割図を求める方法について 室空間の配置計画のための分析と総合の方法に関する研究(その1), 日本建築学会計画系論文報告集, 第414号, pp.69-80, 1990.8
- [10-18] 辻正矩, 川窪広明: 計画と条件に適合する長方形分割図の作成方法について グラフ理論的アプローチによる平面計画の方法 その1, 日本建築学会計画系論文集, 第494号, pp.129-136, 1997.4
- [10-19] 川窪広明, 辻正矩: 長方形分割図の室寸法を決定する方法について グラフ理論的アプローチによる平面計画の方法 その2, 日本建築学会計画系論文集, 第513号, pp.159-166, 1998.11
- [10-20] 太田利彦: 動線計画の数学的解析 第1報, 日本建築学会論文報告集, 第69号, pp.293-296, 1961.10
- [10-21] W. J. Mitchell, J. P. Steadman and R. S. Liggett: Synthesis and optimization of small rectangular floor plans, Environment and Planning B, Vol.3, pp.37-70, 1976
- [10-22] L. Cheng, L. Deng and D. F. Weng: Floorplan Design for 3-D ICs, The 12th Workshop on SASIMI, pp.395-401, 2004
- [10-23] Ohta, H. et al.: The 0-Sequence Representation of 3D-Dissection, IEICE transactions on fundamentals of electronics, communications and computer sciences, Vol.91, No.8, pp.2111-2119, 2008.08
- [10-24] グラフ理論において、あるグラフの全てのノードとそのグラフを構成するエッジの一部分のみで構

要
旨

成される木（連結で閉路を持たないグラフ）のことを全域木という。またあるノードを選んで、それが一番最初の親であると考え、そのノードを基準として2つのノードの間に親子関係を考えることが出来る。このとき、その一番最初の親のノードを根といい、根を持つ木を単なる木と区別して根付き木という。したがって、根を持つ全域木は、全域根付き木と呼ばれる。

目
次

[10-25] 深さ優先探索とは、探索対象となる木の根から、目的のノードが見つかるまで、深く伸びていく探索である。その後はバックトラックして、最も近くの探索の終わっていないノードまで戻る。

序
章第
1
部第
2
部結
章後
付

本研究に関連する論文、書籍、展覧会

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

1 査読付き論文

- [1-1] 松川昌平：逆探索法を用いた直方体分割図の列挙アルゴリズム、日本建築学会計画系論文集、第82巻、第735号、2017.5
- [1-2] 松川昌平：設計プロセスの進化論的枠組みの提案とアルゴリズムック・デザインの位置づけに関する研究、日本建築学会計画系論文集、査読中

2 講演論文

- [2-1] 松川昌平「逆探索法を用いた直方体分割図の列挙アルゴリズム」、日本建築学会大会学術講演梗概集（九州）、2016
- [2-2] 佐々木雅宏、松川昌平「建築の空間構成における、視線、動線、空気の関係性を定量的に評価する手法の提案」、日本建築学会大会学術講演梗概集（九州）、2016
- [2-3] 松川昌平「アルキテクトーム／ARKHITEKTOME—アルゴリズムックデザイン原理モデル」、日本建築学会大会（近畿）情報システム技術部門 研究協議会資料、2014
- [2-4] 松川昌平「Topological Grid によるデータベース化とその検索手法に関する研究」、第35回情報・システム・利用・技術シンポジウム論文集、2012
- [2-5] 松川昌平「不合理な建築の設計」、建築学会誌『建築雑誌2011年8月号』、p16-17

3 書籍

- [3-1] 論考：松川昌平「設計プロセス進化論」（柄沢祐輔（著）、田中浩也（著）、藤村龍至（著）、ドミニク・チェン（著）、松川昌平（著）、メディア・デザイン研究所（編集）『設計の設計』所収、INAX出版、2011年09月）所収、p213-286）
- [3-2] 作品：松川昌平「Topological Grid」（『JA77 建築と都市のアルゴリズム』（新建築社、2010））

4 展覧会

- [4-1] 《ARKHITEKTOME 2015》、ORF2015、出展、@六本木ミッドタウン、東京、2015.11.21 - 11.22
- [4-2] 《ARKHITEKTOME 2015》、ARKHITEKTOME／アルキテクトーム2015展、個展、@プリズミックギャラリー、東京、2015.10.23 - 11.17
- [4-3] 《stoolome》、ORF2014、出展、@六本木ミッドタウン、東京、2014.11.12 - 11.16
- [4-4] 《ARKHITEKTOME 2014》、七夕祭、出品、@慶應SFC、神奈川、2014.07.05 - 07.06
- [4-5] 《Topological Grid+ARKHITEKTOME》CAADRIA WorkShop 2014、出展、@京都工芸繊維大学、京都、2014.06.12 - 06.16
- [4-6] 《polytopia》、磯崎新 都市ソラリス展、出展、@ICC、東京、2014.02.11 - 03.02
- [4-7] 《ARKHITEKTOME 2013》、16th DOMANI 明日展、出展、@国立新美術館、東京、2013.12.14 - 01.26

謝辭

要旨

目次

第1章

第2章

第3章

第4章

第5章

第6章

第7章

第8章

第9章

第10章

第11章

第12章

後付

本論文は、私が1998年に独立してから現在（2017年2月）までのアルゴリズムック・デザインに関する研究を取りまとめたものであり、多くの方々の御指導と御協力により完成することができました。ここに改めて皆様に感謝申し上げます。

私の博士課程の指導教授であり本論文の主査を引き受けて下さった宇野求教授には深く感謝申し上げます。2016年8月23日の夜、宇野先生と打ち合わせさせて頂いたときにすべてが吹っ切れました。それから怒涛の半年間でした。宇野先生からの叱咤激励、ご指導ご鞭撻の数々、いくら感謝してもしきれません。本当にどうもありがとうございました。これから少しずつでも恩返しできればと思っております。

本論文の副査を引き受けて下さった郷田桃代教授、今本啓一教授、倉渕隆教授、高橋治教授、伊藤香織教授、岩岡竜夫教授、岸本達也准教授による重要かつ的確なフィードバックによって、少しずつ論が進化していく様が非常に刺激的でした。お忙しい中何度もご指導頂きましたこと深く感謝申し上げます。

第10章で提案した列挙アルゴリズムを、中野眞一教授（群馬大学工学部）に見て頂いたところ、第10.2.4項で示した消去不可能な直方体分割図が存在することをご指摘頂きました。ここに感謝申し上げます。

第1部は、『設計の設計』に所収された拙論「設計プロセス進化論」が元になっています。貴重な機会を頂いた編集者の飯尾次郎さん、そして様々な議論を戦わせた共著者の藤村龍至さん、田中浩也さん、ドミニク・チェンさん、柄沢祐輔さんに感謝申し上げます。

第2部のケーススタディの数々は、慶應義塾大学SFC松川研究室のメンバーとともに、5年間にわたって少しずつ積み上げてきたプロジェクトです。皆さんなくしては本論文は存在していません。本当にどうもありがとうございます。そしてこれからも一緒に建築の計算（不）可能性を探求していきましょう。

松川研の卒業生であり研究員でもある佐々木雅宏さん、亀岡千花さん、藤平祐輔さんに深く感謝いたします。3人が側にいてくれたことがどれだけ心強かったことか。本当にありがとうございます。これからも3人が起業したarchiroidと松川研で、研究と実践の両輪を回していきましょう。

無償の愛を注いでくれた父と母に、そして姉と弟に深く感謝いたします。他界した父はきっとこの論文を誰よりも読みたかったに違いありません。本論文を父に捧げます。

執筆期間中、3人の子供たちにはとても寂しい思いをさせていたと思います。あまり遊んであげられなくてごめんね。これからいっぱい遊ぼう！

最後に親愛なる妻へ。心より感謝します。ありがとうございます。そしてこれからもよろしく。

2017年02月14日

慶應義塾大学SFC松川研究室にて

松川 昌平

