

階層型最適化方式と デザインへの応用に関する研究

2021年 3月

久富 あすか

概要

最適化問題とは、与えられた制約条件下で、目的関数の値が最大または最小になる設計変数の値の組み合わせを求める問題である。近年、最適化技術は工学分野をはじめとするさまざまな領域で活用され、工場の生産計画や商品の輸送計画などを最適化問題として解決している。

しかし、最適化問題の中には、制約条件が強い、かつ、目的関数が曖昧であるなど、種類の異なる複数の難しさを含むために求解が困難な問題が存在する。

本論文は、上記のような種類の異なる複数の問題の難しさを分けることで対象問題を階層的に表現し、それぞれの難しさに対して異なる解法を適用する階層型最適化方式を提案する。本方式の特徴は、各階層が独立しているため、各難しさに対応する異なる解法を用いることが可能な点にある。例えば、問題の特性に依存しない大域的最適化を行うメタヒューリスティクス、問題依存の効率的な局所探索アルゴリズム、さらに、機械学習に基づく生成モデルなどを、各階層の解法として利用できる。

本論文では、種類の異なる複数の難しさを含む最適化問題の例として、タイリング（平面充填）が可能な図形（タイル）の生成問題に着目する。本問題は、与えられた図形（目標図形）をタイルに変形する難しさ、図形間の類似度を定量化する難しさ、さらに、タイルに適した目標図形を用意することの難しさを含む。これらはそれぞれ、強い制約条件、曖昧な目的関数、および、膨大な探索空間の、3種類の求解を困難にする要因に対応している。本研究では、タイル生成問題における異なる2つの問題に着目し、提案する階層型最適化方式を具体化した手法として、目標図形に類似するタイルを生成する手法、および、映像をもとに対象オブジェクトの目標図形とタイルを生成する手法の2つの手法を提案する。実験結果より、先行研究では目標図形に類似するタイルを生成できない場合でも、提案手法では目標図形に類似したタイルを生成することを確認した。

目次

第 1 章	序論	1
1.1	研究背景と目的	1
1.2	本論文の構成	4
第 2 章	関連研究	6
2.1	最適化問題において求解を困難にする要因	6
2.2	タイル生成問題の定義	10
2.2.1	タイリングの基本	11
2.2.1.1	Isohedral タイリング	11
2.2.1.2	タイリングパターン	12
2.2.1.3	タイリングが可能である制約	15
2.2.2	エッシャー化問題	16
2.3	エッシャー化問題を解決する先行研究	16
2.3.1	解析的にエッシャー化問題を解決する手法	16
2.3.1.1	解析的最適化手法 (AOM)	17
2.3.1.2	重み付き解析的最適化手法	23
2.3.1.3	網羅的な解析的最適化手法	24
2.3.2	メタヒューリスティクスを用いてエッシャー化問題を解決する手法	25
2.3.2.1	焼きなまし法を用いた手法	25
2.3.2.2	遺伝的アルゴリズムを用いた手法	25
2.4	メタヒューリスティクス	26
2.4.1	最適化問題	26

2.4.2	局所探索	27
2.4.2.1	焼きなまし法	27
2.4.2.2	タブー探索	28
2.4.3	進化計算	29
2.4.3.1	自己適応型差分進化	29
2.4.3.2	遺伝的アルゴリズム	31
2.4.3.3	分散推定アルゴリズム	35
(a)	PBIL	37
2.4.3.4	粒子群最適化	37
2.4.4	ハイブリッドアルゴリズム	38
2.4.4.1	遺伝的局所探索	39
2.4.4.2	Parallel recombinative simulated annealing	39
2.4.5	2段階最適化	40
2.5	オートエンコーダ	42
2.5.1	Variational Autoencoder (VAE)	43
2.5.2	Convolutional Variational Autoencoder (ConvVAE)	45
第3章	階層型最適化方式の提案	46
3.1	はじめに	46
3.2	階層型最適化方式の概要	48
3.2.1	基本的なアイデア	48
3.2.2	形状空間における階層型最適化方式の概念的解釈	49
3.2.3	階層型最適化方式と解析的最適化手法の関係	51
3.2.4	階層型最適化方式と DDGA の関係	51
3.3	階層型最適化方式の構成	54
3.3.1	上位最適化層と下位最適化層の関係	54
3.3.2	上位最適化層	55
3.3.2.1	上位最適化層の必要性	55
3.3.2.2	上位最適化層の個体表現	55

3.3.2.3	上位最適化層における遺伝的アルゴリズム	57
3.3.2.4	目的関数	59
(a)	双方向マッピングベース関数	60
(b)	回転関数	61
(c)	重み付き線形和	62
3.4	評価実験	63
3.4.1	実験準備	63
3.4.2	実験 1：階層型最適化方式の構成の有効性の評価	63
3.4.3	実験 2：生成されたタイル形状の従来手法との比較	66
3.4.3.1	実験 2-1：定量評価	68
(a)	制約違反数	68
(b)	目標図形との類似性	68
3.4.3.2	実験 2-2：主観評価	71
3.4.3.3	階層型最適化方式の動作	74
3.4.4	実験 3：漫画のキャラクタへの応用	77
3.5	おわりに	80
第 4 章	階層型最適化方式の拡張	81
4.1	はじめに	82
4.2	アルゴリズムの構成	84
4.2.1	基本的なアイデア	84
4.2.2	処理手順	85
4.2.3	潜在変数の最適化	87
4.3	評価実験	90
4.3.1	実験準備	90
4.3.2	実験結果	91
4.4	おわりに	92
第 5 章	製菓への応用	97
5.1	はじめに	97

目次

5.2	アルゴリズムの構成	98
5.2.1	基本的なアイデア	98
5.2.2	処理手順	99
5.2.3	製菓特有の制約条件	99
5.3	評価実験	101
5.3.1	実験準備	101
5.3.2	実験結果	101
5.4	おわりに	102
第 6 章	処理速度の高速化の試み	106
6.1	はじめに	106
6.2	アルゴリズムの構成	107
6.2.1	基本的なアイデア	107
6.2.2	処理手順	107
6.3	評価実験	109
6.3.1	実験準備	109
6.3.2	実験結果	110
6.4	おわりに	110
第 7 章	結論	115
7.1	研究の結論	115
7.2	今後の展望	117
参考文献		120
付録 A	付録 1：第 3 章の実験 1 の全結果	131
付録 B	付録 2：第 3 章の実験 2-1 の全結果	137
付録 C	付録 3：第 3 章の目標図形の複雑さと ELTHON の性能との関係	154

第 1 章

序論

1.1 研究背景と目的

最適化問題とは、与えられた制約条件を全て満たしつつ、ある目的関数の値が最大または最小になる設計変数の値の組み合わせを求める問題である [1, 2]. 設計変数が離散値をとる場合は組み合わせ最適化問題あるいは離散最適化問題と呼ばれ、連続値をとる場合は連続最適化問題（または単に最適化問題）と呼ばれる.

近年、最適化は、実社会の多くの場面において用いられ、工学分野をはじめとするさまざまな領域で活用されている. 最適化は、解決したい問題を、設計変数、目的関数、制約条件から構成されるモデルとして定義し、汎用的な求解アルゴリズムを用いて解を発見することで問題を解決する. 工場の生産計画 [3] や商品の輸送計画 [4] などを最適化問題として定義し、最良の計画を最適化により求めることができる.

最適化問題はあらゆる分野で用いられているが、最適化問題における求解を困難にするさまざまな要因が存在する. 本論文では、上記の要因を下記の 6 つの要因に大別する.

- (i) 制約条件が強く、実行可能領域が狭いため、解の発見が難しい
- (ii) 目的関数を明確に定義することが困難であり、計算機上に実装することが難しい
- (iii) 設計変数の総数（次元数）が多く、探索空間が膨大となる
- (iv) 目的関数の形状が複雑であり、大域的最適解を発見することが難しい
- (v) 解候補を評価するための目的関数の計算コストが高く、反復回数に制限がある
- (vi) トレードオフ関係を含む複数の目的関数が存在し、単一の目的関数にすることが困

難であるため、単一の大域的最適解を定義することが困難である

(i) の制約条件が強いために解を発見することが難しい問題は、問題の制約条件が強くなることにより、すべての制約条件を満たす解から構成される実行可能領域のサイズが小さくなり、解を発見することが困難になる [5, 6]. (ii) の目的関数の明確化が困難で計算機上に実装が難しい問題は、シミュレーションや機械学習を用いて目的関数の値を近似的に計算する、あるいは、人間を評価器として最適化のループ内に組み入れるなどのアプローチを必要とする [7–9]. (iii) の設計変数の次元数が多い問題では、次元数が増えることにより探索空間の大きさが指数関数的に増大するために、良好な解の発見が困難になる [10]. 最適化問題のなかには、単一の要因のみならず、複数個の求解を困難にする要因を含む最適化問題も存在する。複数の要因が組み合わさることにより、大域的最適解や良好な準最適解の発見が飛躍的に困難になる。

本論文は、求解を困難にする複数の要因を含む最適化問題において、問題に含まれる要因を分けることで問題を階層的に表現し、それぞれの要因に対して異なる解法を適用する階層型最適化方式を提案する。本方式の特徴は、各階層が独立しているため、各要因に対応する異なる解法を用いることが可能な点である。例えば、問題の特性に依存しない大域的最適化を行うメタヒューリスティクス [11], 問題依存の効率的な局所探索アルゴリズム [11], さらに、機械学習に基づく生成モデルなど [12] を、各階層の解法として利用できる。メタヒューリスティクスは、勾配を利用する最適化とは異なり、目的関数の連続性や凸性を前提とせず、目的関数や制約条件をより自由に設計することを許すため、目的関数の明確化が困難な問題においても良好な準最適解の発見を可能にする。一方、問題依存の局所探索アルゴリズムは、対象問題に固有の知識や条件を導入することで強い制約条件を含む問題においても効率的に実行可能解を得ることができる。また、データの生成過程を明示的に記述する生成モデルは、分布を学習することで対象の特徴を含むデータを生成することが可能である。

なお、本論文で提案する階層型最適化方式と関連する手法として、複数の解法を組み合わせるハイブリッドアプローチが挙げられる。提案する階層型最適化方式も広義のハイブリッド方式とみなせるが、最適化問題における一般的なハイブリッド方式は、対象問題を包括的に捉え、複数の最適化アルゴリズムが同一の問題を解く [13–15]. これに対して、

提案する階層型最適化方式は、対象問題を階層的に分割し、各アルゴリズムがそれぞれ異なる階層の部分問題を解決する。

本研究では、求解を困難にする要因のうち (i), (ii) および (iii) に着目し、これらの要因を複合的に含むタイリング（平面充填）が可能な図形（タイル）の生成問題 [16,17] に着目する。タイリングとは、隙間および重複がなく 2 次元平面を有限種類の図形で敷き詰める行為である [16,17]。平面を連続的に覆うことができる図形は、タイル、または、タイリング図形と呼ぶ。

タイル生成問題は 3 つの難しさを含む。第 1 の難しさは、タイリングが可能な制約条件を満たしつつ図形の形状を変形することが難しいことである。同一の形状を持つ 2 つの図形が 1 つの辺を共有するため、ある辺を変形するともう一方の辺も変形を行う必要があるため、接する図形同士のエッジ形状が一致するようにタイルを設計する必要がある（図 1.1）。この難しさは、最適化問題において求解を困難にする要因 (i) に該当する。第 2 の難しさは、生成したタイルと目標図形との類似性を適切に評価する関数が不明である点である。この難しさは、要因 (ii) に該当する。第 3 の難しさは、タイリングに適した目標図形を用意することが難しいことである。この難しさは、要因 (ii) および (iii) に該当する。

本研究では、タイル生成問題における異なる 2 つの問題に着目し、提案する階層型最適化方式を具体化した手法として、目標図形に類似するタイルを生成する手法、および、映像をもとに対象オブジェクトの目標図形とタイルを生成する手法の 2 つの手法を提案する。前者の提案手法 1 は、与えられた目標図形に類似するタイルを生成する問題を二階層で表現し、上位最適化層にはメタヒューリスティクス、下位最適化層にはエッジ形状に関する制約条件を満たす図形を解析的に導出する先行研究 [18,19] を用いる。後者の提案手法 2 は、先行研究 [17–26] とは異なる新たな問題設定を解く手法であり、従来は人間により描画されていた目標図形を、映像から自動生成することを可能にする。提案手法 2 は三階層の構造を持ち、上位最適化層では、対象オブジェクトの目標画像をパラメトリックに生成可能な変分オートエンコーダ [12,27] を用いた生成モデルと、その潜在変数空間における最適化との組み合わせにより、タイリングに適した目標図形を生成する。中位最適化層および下位最適化層は、提案手法 1 と同じ構成をとり、生成された目標図形に類似するタイルを生成する。これらの手法がそれぞれ、先行研究 [17–26] と同一の問題設定においてより優れた解を安定的に発見できること、および、先行研究 [17–26] では取り込まれ

ていない問題設定においてタイルを生成できることを示すことにより，提案する階層型最適化方式が求解を困難にする複数の要因を含むタイル生成問題において有効であることを示す．

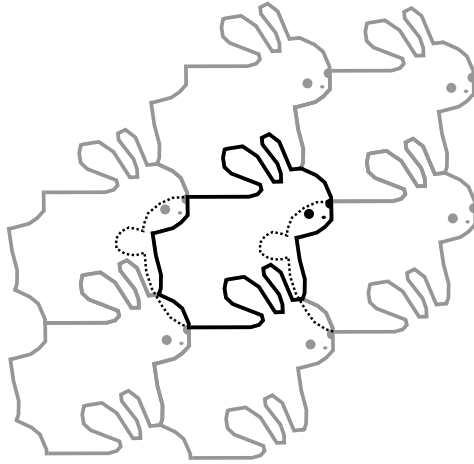


図 1.1. タイルの設計におけるエッジ形状を一致させる難しさ

1.2 本論文の構成

本論文は，全 7 章で構成される．第 1 章は序論とし，第 7 章において結論を述べる．

第 1 章では，本論文の研究対象について説明し，その一例であるタイル生成問題 [16,17] を解決する階層型最適化方式を提案する．また，本方式と最適化問題における一般的なハイブリッド方式 [13–15] との相違点について述べる．

第 2 章では，関連研究について述べる．まず，タイル生成問題の定義について述べ，先行研究，および，本研究で用いる最適化や機械学習の手法について説明する．次に提案方式に関連する先行研究として，タイル生成問題を解析的に解く手法 [18–24]，および，メタヒューリスティクスを用いて解決する手法 [17,25,26] について説明する．

第 3 章では，階層型最適化方式を具体化した第 1 の手法，すなわち，与えられた目標図形に類似したタイルを生成する手法を提案する．本手法は，先行研究 [17–26] と同様の定式化のもとでタイル生成問題を解く．提案する手法はタイル生成問題を二階層で表現し，上位最適化層にはメタヒューリスティクスを用い，下位最適化層にはタイリングに必要な制約条件を満たす図形を解析的に導出する先行研究 [18,19] を用いる．実験結果より，本

第 1 章 序論

手法は、タイリングが可能な条件を満たしつつ、目標図形に類似するタイルを生成できることを確認した。

第 4 章では、階層型最適化方式を具体化した第 2 の手法、すなわち、タイリングに適した目標図形を生成する手法の提案を行う。本手法は、先行研究 [17–26] では考慮されていない新たな問題設定を解く手法である。タイル生成問題においては一般に、目標図形は人間により描画されることを前提としていたが、本手法は映像を用いてタイルに適した目標図形を生成する。本手法は 3 つの階層から構成されており、上位最適化層では、対象オブジェクトの画像をパラメトリックに生成するモデルを用いて、潜在変数の最適化によりタイリングに適した目標図形を生成する。中位最適化層および下位最適化層は、目標図形に類似するタイルを生成する。これにより、目標図形の描画の試行錯誤が不要となることが期待できる。実験結果により、先行研究 [17–26] では考慮されていない問題設定において、目標図形とタイルを生成できることを確認した。

第 5 章では、提案手法が多様な産業に応用可能であることを示す一つの事例として、製菓への応用を試みる。第 3 章で提案した手法の上位最適化層に新たな制約条件を追加することで、成形の際に壊れやすい部位の発生を抑制する。実験結果により、製菓に適したタイリングが可能な図形が生成されることを確認した。

第 6 章では、第 3 章で提案した手法の処理速度の高速化を試みる。第 3 章および第 4 章では、アルゴリズムの挙動の解釈の容易性、および、生成される解の多様性を考慮して単純なメタヒューリスティクスを使用して評価実験を行った。このため、より優れた収束特性を持つメタヒューリスティクスを用いて高速化を試みる。提案する手法が、将来的に導入を検討している対話型最適化の実現に寄与するか検証を行う。実験結果により、第 3 章で提案した手法より収束速度が向上したことを確認した。

第 7 章では、結論として、これまで述べた第 1 章から第 6 章を総括する。

第 2 章

関連研究

本章では，関連研究について述べる．まず，2.1 節において，最適化問題において求解を困難にする要因について述べる．次に，2.2 節では，タイリングの基礎知識，および，タイル生成問題の定義について説明する．はじめに，基礎知識，および，問題の定義を説明することにより，本論文の対象問題について理解を深める．2.3 節では，提案方式に関連する先行研究として，タイル生成問題を解決する手法，および，各手法の問題点について述べる．先行研究の特徴と問題点について述べることで，第 3 章以降で提案する手法との違いを明確にする．2.4 節，および，2.5 節では，2.3 節で説明した先行研究と提案方式にて用いるメタヒューリスティックス，および，提案方式にて用いるオートエンコーダについて説明する．

2.1 最適化問題において求解を困難にする要因

本論文では，最適化問題において求解を困難にする要因を下記の 6 つの要因に大別する．

- (i) 制約条件が強く，実行可能領域が狭いため，解の発見が難しい
- (ii) 目的関数を明確に定義することが困難であり，計算機上に実装することが難しい
- (iii) 設計変数の総数（次元数）が多く，探索空間が膨大となる
- (iv) 目的関数の形状が複雑であり，大域的最適解を発見することが難しい
- (v) 解候補を評価するための目的関数の計算コストが高く，反復回数に制限がある

(vi) トレードオフ関係を含む複数の目的関数が存在し、単一の目的関数にすることが困難であるため、単一の大域的最適解を定義することが困難である

(i) の制約条件が強いことで、解を発見することが難しい最適化問題は、問題の制約条件が強くなることにより、すべての制約条件を満たす解から構成される実行可能領域のサイズが小さくなり、解を発見することが難しくなる [5,6]. 加えて、目的関数の値を改善する探索方向と、制約条件を満たす探索方向とはトレードオフ関係にあることが多い。実行可能領域における大域的最適解を発見する際は、まず、探索空間における初期解に含まれる制約違反を解消する方向に最適化を行い、実行可能領域に到達した後に、実行可能領域を逸脱しないように目的関数の値を改善する方向に最適化を行う必要がある。しかし、制約が強く、実行可能領域が狭い最適化問題では、実行可能領域に到達するまでの探索が困難である。さらに、到達した実行可能領域内に大域的最適解が含まれない場合は、制約違反領域（実行可能領域以外）に再び侵入して他の実行可能領域に移動することが要求される。このため、一旦、制約条件を違反する方向に探索を行う必要があることと、必ずしも他の実行可能領域に到達できる保証はない。これらの理由により、制約条件が強い最適化問題は解を発見することが難しい。

(ii) の目的関数の明確化が困難であるために計算機上に実装が難しい問題は、解の評価基準である目的関数を数式として定義することが困難であることを指している。例えば、人間の感性に依存して解を評価する必要がある場合はこの難しさを含む [28]. このような難しさに対しては、シミュレーションを行ってその結果をもとに目的関数の値を算出する方法や、機械学習を用いて目的関数の値を近似的に計算する、あるいは、人間を評価器として最適化のループ内に組み入れるなどのアプローチが研究されている [7-9].

(iii) の設計変数の次元数が多いため探索空間のサイズが膨大となる問題は、次元数が増えることにより探索空間の大きさが指数関数的に増大するために求解が難しくなる [10]. 深層ニューラルネットワークにおける重みの訓練は、数十万次元を超えることが多く、確率的勾配降下法等の目的関数が微分可能である問題を対象とする解法により解くことは可能であるが、探索空間が膨大なため良好な解を発見することは容易でない。また、対象問題の領域知識を利用しないメタヒューリスティクス等のブラックボックス最適化では、従来は数十次元程度の問題が主に対象とされていたが、近年では共進化アルゴリズムな

どを中心として、数千次元程度の問題の最適化を行う大規模大域的最適化 (Large-Scale Global Optimization: LSGO) 技術の研究 [29] が盛んに行われている。

(iv) の目的関数の形状により求解が困難となる問題は、目的関数の連続性や大域的多峰性により大域的最適解の発見が困難となることを指す [30]。一般に、目的関数が凸関数である場合は、局所最適解が大域的最適解となるため、探索が容易である。一方、目的関数が非連続で微分不可能である場合は、勾配を利用する手法では大域的最適解の発見が困難である。また、メタヒューリスティクスは一般に、多峰性関数や目的関数にノイズを含む関数であっても局所解に陥りにくい特性を持つが、探索空間の大部分においてある局所解に誘導するような地形をもち、かつ、大域的最適化がその局所解の反対方向に位置するような、いわゆるだまし問題のような景観を持つ関数において極端に探索性能が低下する [31]。

(v) の目的関数の計算コストが高い問題は、(ii) のように目的関数の明確化が困難であるためにシミュレーションを行い、その結果に基づいて目的関数の値を計算する問題や、人間を評価器として用いるために処理時間を要し、反復回数が限定される問題を指す [32, 33]。一般に、シミュレーションモデルの精度と処理速度はトレードオフの関係にあり、再現度の高いシミュレーションを行うほど高い計算コストが必要となり、より少ない反復回数で良好な解を発見することが求められる。

(vi) の複数の目的関数を含む問題は多目的最適化問題と呼ばれ、経済学や経営学の分野で古くから研究されている [34, 35]。一般に、問題に複数の目的関数が含まれる場合はそれらの関数の線形和をとることで単一の目的関数を用いる。しかし、性能とコストのように相反する目的関数を含む場合は、両者の適切な重みを設定することが困難となる。このため、それらの目的関数を同時に用いて最適化問題を解く多目的最適化が研究されている [36]。多目的最適化では一般に、単一の大域的最適解を定義することは困難であるため、他の解よりも明確に劣る (全ての目的関数における値が劣る) 解を除くパレート最適解の集合を発見することが目的となる。得られたパレート最適解集合から一つの解を選択する際は、一般に、タスクの目的や用途を考慮してユーザーが行う必要がある。なお、考慮すべき目的関数の総数が 4 以上となる場合はパレート最適解の数が膨大となり、進化計算を用いた多目的最適化アルゴリズムを用いる場合であっても、解集団内の多くの解が他の解に支配されない解となり、解の優劣がつきにくくなるために探索の速度が落ち、最適化

が困難となる。

(i) の強い制約条件を含む最適化問題は一般的に組み合わせ最適化分野で古くから研究されており，ナーススケジューリング問題 [37]，グラフ彩色問題 [38] や n クイーン問題 [39] などが挙げられる．一方，(ii) の目的関数の明確な定義が困難な最適化問題は，旅客機の翼状形状の最適化問題 [40]，モンタージュ作成問題 [28] やライティング問題 [41] などが挙げられる．(iii) の設計変数の次元数が多く，探索空間が膨大となる最適化問題は，配水網の設計問題 [42] や大規模熱交換器ネットワーク設計問題 [43] などが挙げられる．

さらに，最適化の実応用における難しさの例を考えるために，新幹線の車両形状の設計 [44, 45]，および，資産投資のリスク最小化と収益最大化の計算 [46, 47] の 2 つの事例に着目する．

新幹線の車両形状の設計では，新幹線の先頭車両の形状の設計に最適化が用いられている [44, 45]．この最適化問題では，走行時間の短縮，つまり，高速化のための車両設計を行う．特に，高速走行により発生するトンネル微気圧波を低減するための先頭形状の最適化では，先頭形状部の断面積分布が重要となる．新幹線の車両形状の設計 [44, 45] は前述の (ii) および (v) を含み，目的関数の明示的な定義が困難であることから，遺伝的アルゴリズム [11, 48] が用いられた．これにより，車両形状の制約条件を満たしつつ，走行速度の向上，および，トンネル微気圧波も改良以前と同様に維持できる先頭形状を開発した．

次に資産投資のリスク最小化と収益最大化の計算 [46, 47] では，安定した収益を得るために投資者のリスクを減らす一方で，収益は高くなるようにポートフォリオを最適化する．一般的に，リスクと収益はトレードオフ関係にあるため，このトレードオフ関係を考慮して最適なポートフォリオを選択する必要がある．すなわち前記の要因のうち (vi) を含み，多目的最適化問題として定式化することで，トレードオフ関係にある目的関数を同時に用いることが可能となる．ポートフォリオの選択に多目的最適化を用いることにより，多くの選択肢の中から，人手では発見が難しい投資計画を立案することが可能である．

2.2 タイル生成問題の定義

タイル生成問題とは、隙間および重複がなく 2 次元平面を敷き詰める図形（タイル，または，タイリング図形）を生成する問題である．また，平面にタイルを敷き詰める行為をタイリングと呼ぶ [16,17].

オランダの画家 M.C. Escher（エッシャー）は数学や結晶学からタイリングについて研究し，タイリングに関する芸術的な作品を多く残した [49]. タイリングは，芸術的側面と数学的側面を持ち，さまざまな図形を表現でき，装飾性に富み，繰り返しのパターンによって任意の領域を装飾できる．このため，タイリングは，工業的な応用もされており，建造物の壁面，天井等のテクスチャ [50] の他，図 2.1 のようなコースターや緩衝材などにも実利用されている．



(a) 緩衝材 *1



(b) コースター兼鍋敷き

図 2.1. タイリングの実利用の例

本論文では，タイル生成問題のなかでも，目標図形に類似するタイルをコンピュータにより自動設計することを目指すエッシャー化問題 [16,17] (2.2.2 項を参照) を解決する手法について提案する．これは，エッシャー化問題がコンピュータ上でタイルの自動設計を行う問題であること，および，エッシャー化問題を解決する先行研究 [17–26] が存在しており，比較評価を行うためである．

*1 http://www.sanodesign.jp/cushion/Cushion_san.html

2.2.1 タイリングの基本

本論文では、平面を1種類のタイルで敷き詰め、かつ、敷き詰め方が一定である Isohedral タイリングを対象とする。Isohedral タイリングは、数学的に扱いやすく、幅広い表現が可能であり、かつ、単純さと複数の難しさを含む問題である。さらに、2種類以上のタイルを組み合わせ、1つのタイルとして扱うことも可能である [17].

Isohedral タイリングは、子孫集団とトポロジ的に等しいタイル T により、以下のよう に数学的に定義できる [51].

$$\bigcup_i \mathcal{C}_i(T) = \mathbb{R}^2, \quad (2.1)$$

$$\mathcal{I}(\mathcal{C}_i(T)) \cap \mathcal{I}(\mathcal{C}_j(T)) = \phi \quad \forall i, j, i \neq j. \quad (2.2)$$

ここで、 $\mathcal{I}(T)$ は T の内点集合を表し、 $\mathcal{C}_i(T)$ は T の i 番目の複製を表し、 \mathbb{R}^2 はユークリッド平面を表している。上記の定義により、タイル T における、隣接するタイル間の共通部分は、点や曲線または空にすることができる。1つのタイルとのみと隣接している境界部分（辺）をタイリング辺（tiling edge）と呼び、2つ以上のタイルと隣接している境界部分（点）をタイリング頂点（tiling vertex）と呼ぶ。また、タイリング頂点を線分をつないで構成される多角形をタイリング多角形（tiling polygon）と呼ぶ。また、ここでのタイルについては、閉円板と同相であるものとし、他のタイルとの共通部分における連結成分数は1とする。

2.2.1.1 Isohedral タイリング

図 2.3(a) [52] のような1種類のタイルから構成されるタイリングを Monohedral タイリングと呼ばれ、複数種類のタイルからなるタイリングを Multihedral タイリングと呼ばれる [51]. Monohedral タイリングの中でも、図 2.3(b) のように繰り返し構造を持ち、タイルの敷き詰めパターンが一定であるタイリングを Isohedral タイリングと呼ぶ。

Isohedral タイリングは全てのタイルが同じ特性をもつため、数学的に扱いやすく、またさまざまな形を表現できる柔軟性も持っている。さらに、2種類以上のタイルを結合させ、1つのタイルとみなすことで Multihedral タイリングも表現できる [17] ことより、汎用性が高い。

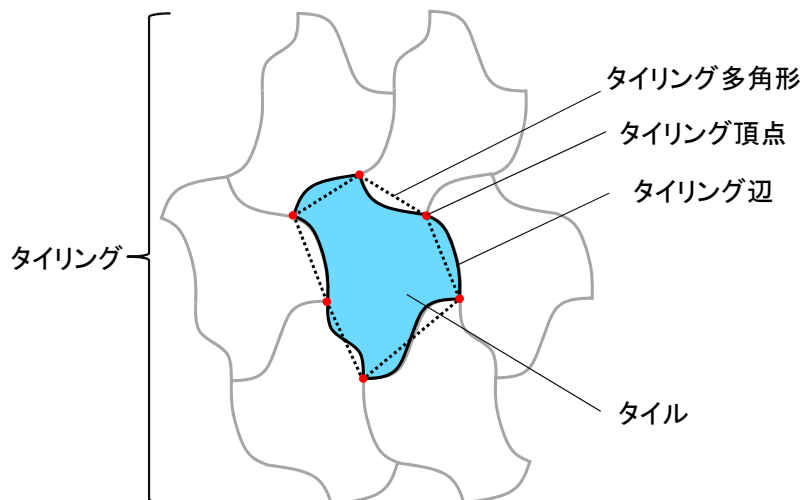


図 2.2. タイリングの定義

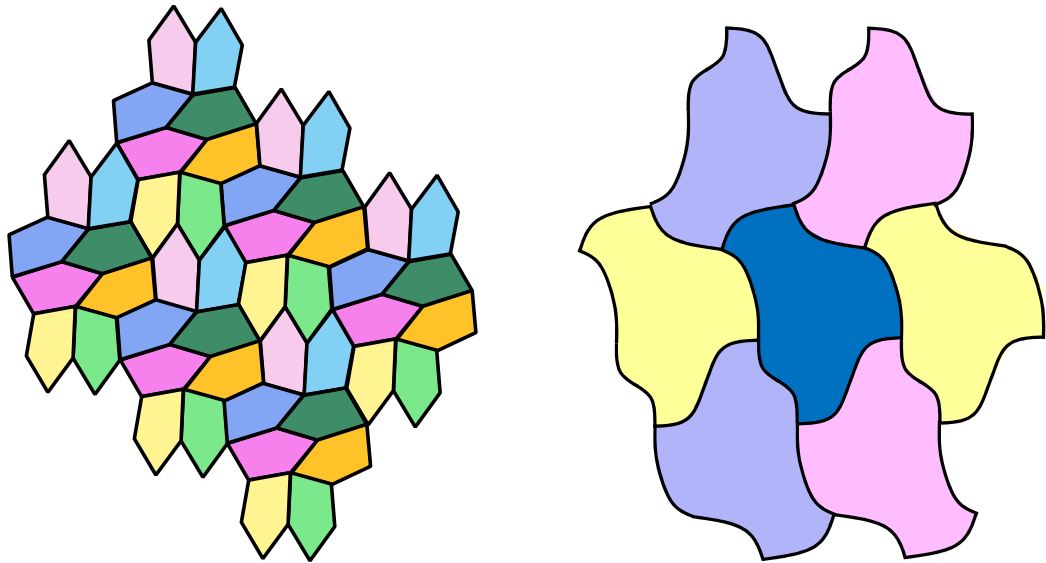
2.2.1.2 タイリングパターン

タイリングパターン (Tiling Pattern : TP) とは, Isohedral タイリングのタイルの敷き詰め方法, および, タイルの辺形状の制約を決めるパターンである [51]. タイリングパターンは, 全 93 パターンあり, incidence symbol の違いにより分類したもので, ここで, incidence symbol とは, 図 2.4 に示すような Isohedral タイリングのタイル間の隣接関係を表すラベルである.

タイリングパターンは, incidence symbol により, 93 パターンに分類することができ, IH01 ... IH93 の記号で表される [51]. この分類は, タイルを複製して, 平行移動, 回転, 鏡映により平面を敷き詰めるパターンのことである. 図 2.5 にタイリングパターン IH01, および, IH16 の例を示す. ここで, ある 2 つのタイリング辺が同じ形状を持つ必要がある場合, incidence symbol に割り当てるラベルは向きを含めて同じラベルを割り当てる. また, 両方向に向きを持ち, 線対称な特徴があるタイリング辺の場合, incidence symbol に割り当てるラベルは向きを持たないとして, 何も付けない.

また, 先行研究 [18,19] (2.3.1.1 項参照) では, この incidence symbol の満たす条件を行列 A で表現し, $\mathbf{A}u = \mathbf{0}$ (式 2.3) を満たす u を求めることでタイルを生成する.

このような行列表現の性質上, 93 パターンから冗長性を排除することにより, 29 パターンのみの行列 A を生成することで, すべてのタイリングパターンを表現することが



(a) Monohedral タイリングではあるが
Isohedral タイリングではない [52]

(b) Monohedral タイリングかつ
Isohedral タイリングである

図 2.3. Monohedral タイリングの例

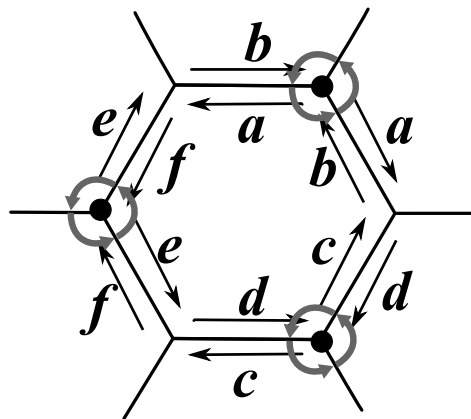


図 2.4. IH07 の incidence symbol

できる [51]. これは、指定したタイリングパターンのタイルを、式 2.3 を満たす解空間として考えるため、type-I (2.2.1.3 項参照) などの強い制約を持つタイリングパターンの実行可能解が他のタイリングパターンに全て含まれる場合があるためである。例えば、IH10 は各辺を自由に変形させることができる六角形をベースとして、1 種類の type-J のタイ

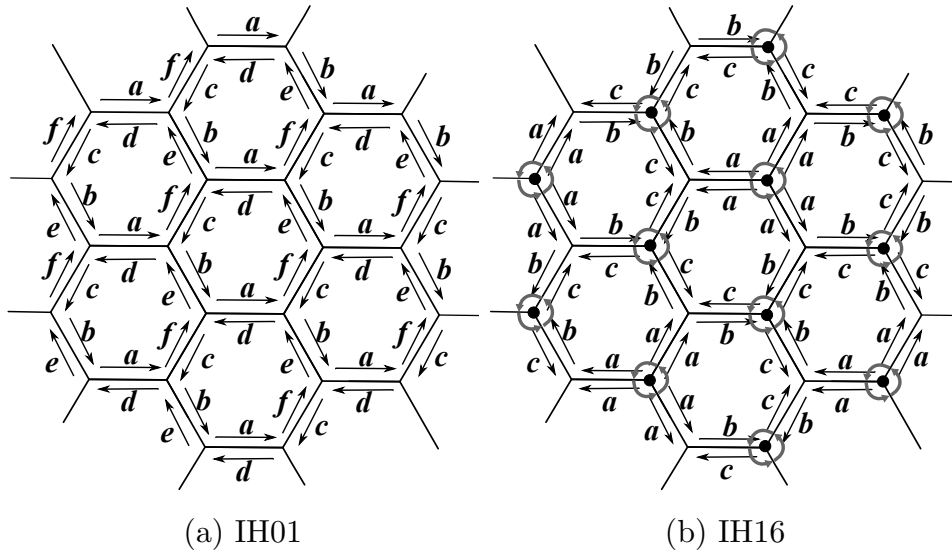


図 2.5. タイリングパターンの例 : IH01, IH16

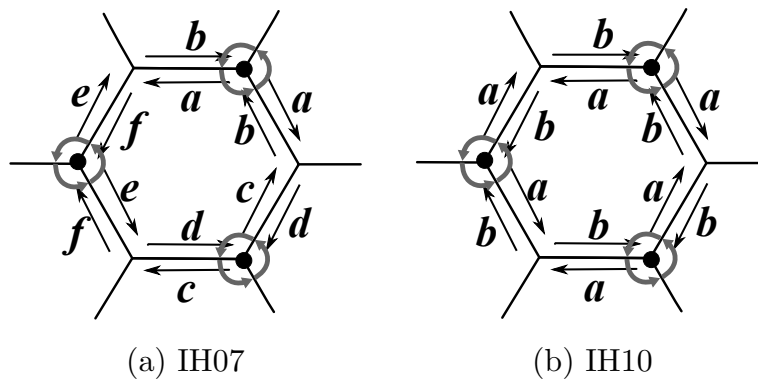


図 2.6. タイリングパターンの冗長性の排除例 (IH10 は IH07 の特殊なケースとして扱う)

リング辺しか存在しない (6 辺すべてが点対称で同じ形状の辺である) が, IH07 は六角形をベースとして, 3 種類の type-J のタイリング辺が存在する (3 組の辺がそれぞれ同一形状の辺である). IH07 を行列 A で表現すると, IH10 でタイルは, IH07 における制約ですべて生成可能となる (図 2.6). このため, IH10 は IH07 の特殊なケースとしてみなすことができる.

2.2.1.3 タイリングが可能である制約

タイリング辺の変形について考える場合，この変形は incidence symbol に依存して制限される．この制限は，次の 4 つに分けられる．

type-S

タイリング辺の両側のラベルが同じ名前でも向きが異なるとき，タイリング辺の中点に関して点対称でなければならない．その条件を満たす範囲でのみ変形できる．

type-U

タイリング辺の両側のラベルの名前が異なり，少なくとも一方が向きを持たないとき，タイリング辺の垂直二等分線に関して線対称でなければならない．その条件を満たす範囲でのみ変形できる．

type-J

タイリング辺の両側のラベルの名前が異なり，ともに向きを持つものの場合，タイリング辺は自由に変形できる．

type-I

タイリング辺の両側のラベルの名前が等しく，ともに向きを持たない，または，同じ向きを持つとき，タイリング辺は直線分でなければならない．これは，タイリング辺の両側のラベル名が等しく，向きなしが同向きを持つときは，type-S と type-U の両方の条件を満たさないといけないため，全く変形できず，直線分となる．

また，これらの制限にくわえ，同じラベルが付いたタイリング辺は，すべて同じ形でなければならない．

タイリング頂点に関する制限も同様に，incidence symbol に依存して得られる．タイリング頂点を動かすことは，タイリング多角形が変形されるということを意味するが，incidence symbol によってタイリング多角形の辺の長さや内角が制限される．

タイリングの制約条件はタイリングパターンの種類によってそれぞれ異なる．例として，IH07 の場合を見る．まず，IH07 の incidence symbol は図 2.4 のようになっている．IH07 は，3 組の type-J で制限されるタイリング辺から構成されており，また，この 3 組は隣り合い，六角形を構成するには 120 度回転を用いてタイリング辺を重ね合わさなければ

ばならない.

2.2.2 エッシャー化問題

Kaplan らは, コンピュータによるタイリングの自動設計を目指し, タイルの生成問題をエッシャー化問題として定義した [16,17].

エッシャー化問題: 閉じた平面図形 (目標図形) W が与えられたとき, 以下のような新たな閉じた平面図形 U を見つける.

条件 1: U は可能な限り W に類似する

条件 2: U は平面を隙間かつ重複なく敷き詰めることができる

つまり, エッシャー化問題を解決することで, タイル U を得ることができる.

さらに本論文では, 先行研究において未解決の問題を具体化する 3 番目の条件を Kaplan らの定義に追加する.

条件 3: U を構成する辺がお互いに交差しない (つまり, 自己交差を含まない)

先行研究 [17–19] では条件 3 を含まないため, 従来手法の出力結果には, 条件 3 に違反する場合がある.

本論文では, エッシャー化問題の 3 つの条件 (Kaplan らが定義した 2 つの条件と本論文で追加した 1 つの条件) を満たす図形 U をタイル U として定義する.

2.3 エッシャー化問題を解決する先行研究

2.3.1 解析的にエッシャー化問題を解決する手法

エッシャー化問題を解く手法の 1 つには, エッシャー化問題を固有値問題としてモデル化し, 解析的に解く手法がある. 本項では解析的にエッシャー化問題を解決する手法について紹介する.

2.3.1.1 解析的最適化手法 (AOM)

小泉らは、固有値問題としてエッシャー化問題をモデル化し、解析的に解決する Isohedral タイリングを設計手法を提案した [18,19]. 本論文では、小泉らの手法を解析的最適化手法 (Analytical optimization method) と呼び、AOM と省略する.

解析的最適化手法には、人手により作成された粗い目標図形 \tilde{W} が必要であり、粗い目標図形 \tilde{W} は、目標図形 W を近似し、 \tilde{N} 点からなる図形である. この粗い目標図形 \tilde{W} からタイル U の頂点座標を計算する. ここで、粗い目標図形 \tilde{W} とは、目標図形 W の形状に類似するが、頂点数が少なく、粗い図形形状のことを指す. 次にタイル U は、 \tilde{W} と U の対応する頂点間のユークリッド距離の二乗和式 (式 2.5) を最小化することにより構築される.

解析的最適化手法の解は、2.2.2 項の条件 2 である、タイルの対応する 2 辺の形状が一致する制約を満たすことが保証されている. また、解析的最適化手法は、低い計算コストでタイルを生成することができる. これは、最適化対象の関数が単純な凸関数であることを仮定して解析的に解を導出するため、高速にタイルを導出できるためである. また解析的最適化手法は、低い計算コストでタイルを生成できるため、他の研究者により採用、および、改善されている [20–24] (2.3.1.2, 2.3.1.3 項参照).

解析的最適化手法は常に条件 2 を満たす図形を生成できるが、以下の欠点がある.

1. 解析的最適化手法は、与えられた粗い目標図形 \tilde{W} の頂点に敏感である. つまり、出力図形の形状は、与えられた頂点のわずかな変化により、大きく変化する可能性がある. 図 4.1 は、耳と後ろ足の形状が微妙に異なるがほぼ同じ形状のウサギの入力頂点から、異なる形状の出力図形が生成されたことを示す.
2. 図 2.8 に示すように、出力図形に自己交差を含む場合がある. ほぼ同じ形状のウサギを入力しているにもかかわらず、図 2.8(a) の出力図形は自己交差を含まないが、図 2.8(b) の出力図形は耳部分に自己交差が発生している. また、目標図形 W の形状から自己交差の発生を事前に知ることはできない.
3. 図 2.9 に示すように、同じ粗い目標図形 \tilde{W} を与えても、選択するタイリングパターンが異なると出力図形の形状が大きく異なる. 同じ形状のウサギを入力していた場合でも、選択するタイリングパターンが異なると、図 2.9(a) の出力図形は粗い

目標図形 \tilde{W} の形状に類似するが、図 2.9(b) の出力図形は粗い目標図形 \tilde{W} の形状とは大幅に異なる形状である。

4. 解析的最適化手法は、与えられた粗い目標図形 \tilde{W} と生成した出力図形の類似性を評価する目的関数の自由度が低い。解析的最適化手法は、解析的に出力図形を導出するために目的関数の形状が連続かつ凸関数である必要がある。このため、解析的最適化手法に用いる目的関数は、2 図形間の頂点数が等しい、かつ、2 図形の頂点間に対応付けを行う必要がある。解析的最適化手法に用いる目的関数は、この 2 つの条件を満たした関数である必要があるため、目的関数の自由度が低い。

上記の欠点より、自己交差を含まず粗い目標図形 \tilde{W} に類似するタイルを得るためには、粗い目標図形 \tilde{W} の形状を試行錯誤しながら、解析的最適化手法を繰り返し適用する必要がある。つまり、適切なタイリングパターンの探索と粗い目標図形 \tilde{W} の形状の微調整を人手で行う必要がある。さらに、試行錯誤中にタイリングパターンを変更すると、再度、粗い目標図形 \tilde{W} の形状の微調整を行う必要がある場合がある。また、目標図形 W の頂点をすべて使用しても、粗い目標図形 \tilde{W} よりも必ず良い出力図形が得られるとは限らない。このため、目標図形 W の頂点は、粗い目標図形 \tilde{W} の形状、頂点数とタイリングパターンに応じて選択する必要がある。

以降、解析的最適化手法の計算方法の詳細を示す。粗い目標図形を \tilde{W} とし、 \tilde{W} に類似するタイルを U とする。 \tilde{W} の境界上に反時計回りに並んだ \tilde{N} 個の点の列 $(Q_1, Q_2, \dots, Q_{\tilde{N}})$ を考え、 \tilde{W} を \tilde{N} 角形で近似する。 U も同様に $(P_1, P_2, \dots, P_{\tilde{N}})$ と考える。

$i = 1, 2, \dots, \tilde{N}$ に対して、 P_i の座標を (x_i, y_i) とする。これらの座標を一系列に並べてできる $2n$ 次元の縦ベクトルを

$$\mathbf{u} = (x_1, y_1, x_2, y_2, \dots, x_{\tilde{N}}, y_{\tilde{N}})^t$$

とする。ただし、 t は転置を表す。図形 \tilde{W} に対しても同様に $(Q_1, Q_2, \dots, Q_{\tilde{N}})$ の縦ベクトルを $\tilde{\mathbf{w}}$ とする。

IH07 のタイルを生成する方法を例にあげ、解析的最適化手法の説明を行う。IH07 の場合の U を構成する \tilde{N} 点を図 2.10 のように表す。

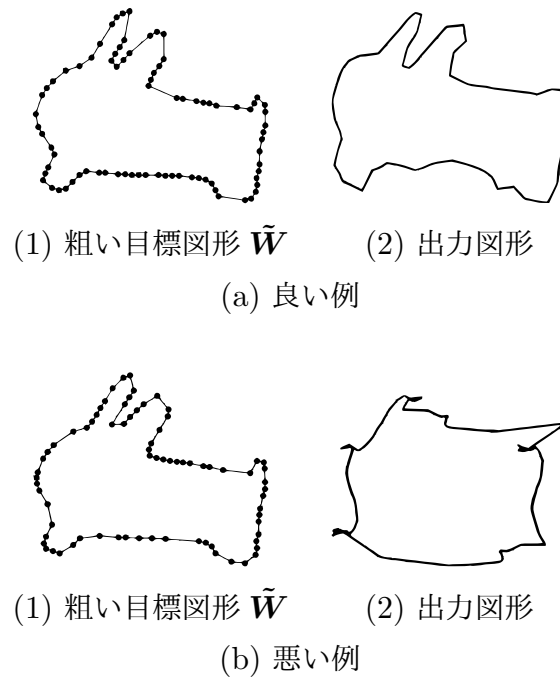


図 2.7. 粗い目標図形 \tilde{W} の形状に敏感な例

IH07 のタイリング多角形は六角形であるため、 \tilde{N} を 6 の倍数で $\tilde{N} = 6m$ としている。図 2.10 に示すように、正六角形の左下の頂点が $P_{\tilde{N}} = P_{6m}$ になるように反時計回りに点を並べると、 P_m, P_{3m}, P_{5m} が 120 度回転の中心となる。

点 (x, y) を原点として反時計回りに角度 θ ラジアンだけ回転した結果、点 (x', y') に移ったとすると、

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

が成り立つ。回転角度が d のとき、ここの 2×2 行列を $R(d)$ とすると

$$R(d) = \begin{pmatrix} \cos \frac{d\pi}{180} & -\sin \frac{d\pi}{180} \\ \sin \frac{d\pi}{180} & \cos \frac{d\pi}{180} \end{pmatrix}$$

P_1, P_2, \dots, P_{m-1} は、 P_m を中心とする時計回り 120 度回転（反時計回り 240 度回転）によって、それぞれ $P_{2m-1}, P_{2m-2}, \dots, P_{m+1}$ に移る。したがって、

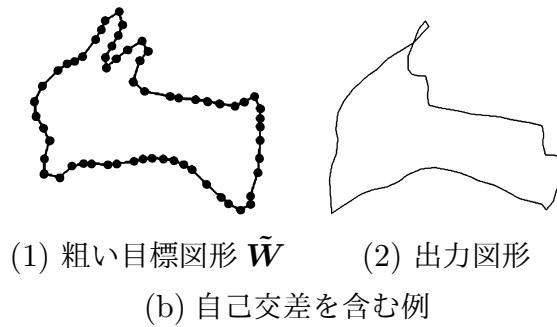
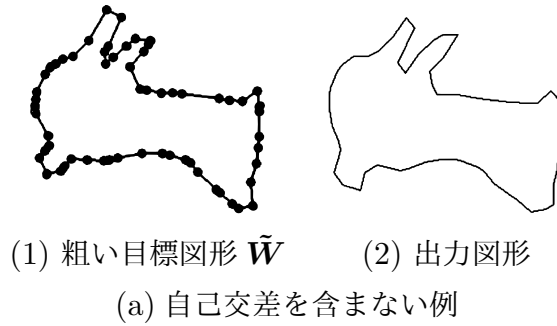


図 2.8. 似た粗い目標図形 \tilde{W} の形状から自己交差が生じる例

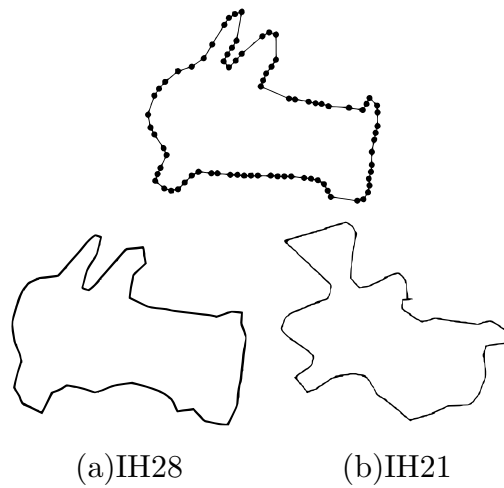


図 2.9. タイリングパターン選択により出力図形が異なる例

$$\begin{pmatrix} x_{2m-i} - x_m \\ y_{2m-i} - y_m \end{pmatrix} = R(240) \begin{pmatrix} x_i - x_m \\ y_i - y_m \end{pmatrix}, i = 1, 2, \dots, m-1$$

また、同じ回転によって P_{6m} は P_{2m} へ移るので、

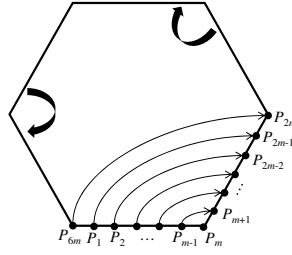


図 2.10. 120 度回転による辺の対応

$$\begin{pmatrix} x_{2m} - x_m \\ y_{2m} - y_m \end{pmatrix} = R(240) \begin{pmatrix} x_{6m} - x_m \\ y_{6m} - y_m \end{pmatrix}$$

も成り立つ.

これらの 2 つの方程式が, ひとつのタイリング辺に対して m セットの $2m$ 個あり, タイリングが可能な制約として行列で表現できる. 残りの 2 辺についても同様の制約が得られるため, 合計 $6m$ 個の方程式で表される. $\tilde{N} = 6m$ より, この連立方程式は $2m$ 個の変数に対する \tilde{N} 個の方程式から成っている. これらをまとめて,

$$\mathbf{A} \mathbf{u} = \mathbf{0} \tag{2.3}$$

で表すことができる. \mathbf{A} はタイリングが可能であるための制約であり, タイリングパターンによって行列の要素が変化する. IH07 の場合, \mathbf{A} は \tilde{N} 行, $2\tilde{N}$ 列の定数ベクトルとなる.

次に, タイル \mathbf{U} と粗い目標図形 $\tilde{\mathbf{W}}$ との類似度を定量化するため, \mathbf{U} と $\tilde{\mathbf{W}}$ の点を一対一に対応付ける ($i = 1, 2, \dots, \tilde{N}$ に対して P_i が Q_i に対応する). 各対応点間のユークリッド距離を 2 つの図形の形状差 F_L とする. つまり, 形状差 F_L が 2 図形間の類似性を表す指標となる. 2 つの図形の形状差 F_L を P_i と Q_i のユークリッド距離の 2 乗和 $d(P_i, Q_i)$ で表すと次のようになる.

$$F_L = \sum_{i=1}^{\tilde{N}} d^2(P_i, Q_i)$$

この F_L を目的関数とし、最小化問題として定式化する。形状差 F_L は2図形間の変化量を示すため、形状差 F_L が小さくなるほど2図形間の類似性は高いことを示す。ここで、 Q_i の座標を (x_i, y_i) と置く。したがって、

$$\tilde{\mathbf{w}} = (\bar{x}_1, \bar{y}_1, \dots, \bar{x}_{\tilde{N}}, \bar{y}_{\tilde{N}})^t$$

で表す。このとき、 F_L は次のように変形できる。

$$F_L = \sum_{i=1}^{\tilde{N}} \{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2\} \quad (2.4)$$

$$= (\mathbf{u} - \tilde{\mathbf{w}})^t \cdot (\mathbf{u} - \tilde{\mathbf{w}}) \quad (2.5)$$

ここで、タイリングが可能な制約を示す方程式 (2.3) の解空間の次元を k とする。この解空間を張る正規直交基底を1つ選び、それを $(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k)$ とする。 \mathbf{g}_i は $2\tilde{N}$ 次元の縦ベクトルである。これらを並べてできる行列を

$$\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k)$$

と表す。このとき、 \mathbf{u} は k 次元の変数ベクトル \mathbf{z} を用いて

$$\mathbf{u} = \mathbf{G}\mathbf{z} \quad (2.6)$$

と表せる。これは、タイリングが可能な制約 (式 2.3) を満たす \mathbf{u} として、一般解を表現している。これを式 (2.5) に代入し、

$$F_L = (\mathbf{G}\mathbf{z} - \tilde{\mathbf{w}})^t \cdot (\mathbf{G}\mathbf{z} - \tilde{\mathbf{w}}) \quad (2.7)$$

となる。これより、1つの関数による最適化問題としての定式化が実現する。ここで、式 (2.7) は \mathbf{z} に関して連続であるため、式 (2.7) が最小値をとるのは、 \mathbf{z} のどの成分についても最小値を達成していなければならない。また、 F_L は形状差であるため、非負であることが明らかである。すなわち最小値が1つであり、極値も変曲点も持たない。以上より、 F_L が最小をとる \mathbf{z} は、

$$\begin{aligned} \frac{\partial F_L}{\partial z_i} &= \frac{\partial [(\mathbf{G}\mathbf{z} - \tilde{\mathbf{w}})^t \cdot (\mathbf{G}\mathbf{z} - \tilde{\mathbf{w}})]}{\partial z_i} \\ &= 2(\mathbf{G}\mathbf{z} - \tilde{\mathbf{w}})^t \cdot \mathbf{g}_i = 0, \quad i = 1, 2, \dots, k \end{aligned} \quad (2.8)$$

を満たす.

k 個の方程式からなる (2.8) をまとめ, 変形すると

$$\begin{aligned} (\mathbf{G}z - \tilde{\mathbf{w}})^t \cdot \mathbf{G} &= 0 \\ \mathbf{G}^t(\mathbf{G}z - \tilde{\mathbf{w}}) &= \mathbf{G}^t\mathbf{G}z - \mathbf{G}^t\tilde{\mathbf{w}} = 0 \end{aligned}$$

さらに, G は正規直交基底行列により, G^tG は単位行列となるため

$$z = \mathbf{G}^t\tilde{\mathbf{w}} \quad (2.9)$$

となり, G の式で z が求まる. (2.6) に (2.9) を代入し,

$$\mathbf{u} = \mathbf{G}\mathbf{G}^t\tilde{\mathbf{w}} \quad (2.10)$$

以上より, タイリングが可能な制約を示す方程式 (2.3) の一般解を解空間における正規直交基底 \mathbf{G} と変数ベクトル z で表現することで, 2つの形状差 F_L を最小とする z を求める式へと帰着でき, 粗い目標図形 $\tilde{\mathbf{W}}$ からタイル \mathbf{U} の解析的な生成が可能な式 (2.10) が導出できる.

ここで, P_i と Q_i を一対一の対応を定めたが, この第一頂点 (開始点) の違いにより, \mathbf{G} が変わること注意到する. つまり, $\tilde{\mathbf{W}}$ に最も近い \mathbf{U} を求めるためには, \tilde{N} 通り第一頂点を 1 つずつズラして F_L を算出し, その中で最小となる \mathbf{U} が出力図形となる.

このように小泉らは, 行列計算を用いた解析的最適化手法により, エッシャー化問題の条件 1 を必ず満たしつつ, 高速かつ効率的にタイルの自動生成を行うことが可能である. 解析的最適化手法を用いて良質な解 (タイル) も確認された.

2.3.1.2 重み付き解析的最適化手法

今堀らは, 解析的最適化手法 [18,19] の目標図形形状に重みを用いることで, 形状を崩したくない部分など, 重みにより形状に優先度を導入した [20,21].

解析的最適化手法 [18,19] では, 粗い目標図形 $\tilde{\mathbf{W}}$ を構成する頂点に優先度がないため, タイルに変形する際, ユーザが優先的に保持したい形状を崩して変形する場合がある. 一方, 今堀らが提案した手法 [20,21] は, 粗い目標図形 $\tilde{\mathbf{W}}$ を構成する頂点に重みを導入することで, ユーザが優先的に保持したい形状部分の変化量を少なくタイルに変形すること

が可能となる。さらに、重みの値を調整することで、粗い目標図形 \tilde{W} の形状の優先度を細かくつけることも可能である。この手法は、解析的最適化手法と同様に固有値問題としてモデル化しているため、解析的に解を導出しつつ、タイルにユーザの優先度も反映することができる。

しかし、この手法は解析的最適化手法と同様に、タイルに自己交差を含む場合があること、および、適切なタイリングパターンを選択する必要があるという問題を解決できていない。さらに、重みはユーザが与えるため、調整するパラメータが解析的最適化手法より増加しているため、適切な粗い目標図形 \tilde{W} の形状、および、適切なタイリングパターン、適切な重みをユーザが決定する必要がある。

2.3.1.3 網羅的な解析的最適化手法

永田らは、タイルの各辺に割り当てる頂点数を網羅的に変更する手法を提案している [22–24]。

解析的最適化手法 [18, 19] は、タイルの全辺に同数の頂点を配置した場合のみ考慮しており、このモデル化では、表現できるタイルの形状の自由度が低く、目標図形 W を単純化した \tilde{N} 角形の粗い目標図形 \tilde{W} の頂点を適切に選択しないと良好な解は得られないという問題がある (2.3.1.1 項参照)。一方、永田らが提案した手法 [22–24] は、 \tilde{N} 点の頂点をタイルの辺に割り当てる全割り当て方を考慮することで、表現可能なタイルの形状の自由度を向上させた。これにより、解析的最適化手法と比較して、粗い目標図形 \tilde{W} に類似するタイルを生成することが可能となった。

また、解析的最適化手法では、 \tilde{N} 角形の粗い目標図形 \tilde{W} とタイル U の形状差 F_L を求めるために、開始点を1つずつズラしながら最小となる F_L を求めるため、 F_L を \tilde{N} 回計算する必要があり、これに、タイルの辺への全割り当て方を考慮すると計算量は $O(\tilde{N}^3)$ 以上となる。一方、この手法では、開始点と割り当て方を考慮した際の計算量を $O(\tilde{N})$ まで削減することができる。さらに、分岐限定法の考え方に類似した方法で計算を削減することで、高速に解を導出することが可能となった。

しかし、適切なタイリングパターンを事前に推定する必要がある。また、重み付き解析的最適化手法 [20, 21] と同様に重みを用いているため、ユーザによる重みの調整が必要である。重み付き解析的最適化手法と同様に、事前にユーザが決定するパラメータが多いた

め、ユーザの試行錯誤は必要である。

2.3.2 メタヒューリスティクスを用いてエッシャー化問題を解決する手法

エッシャー化問題を解く手法の 1 つには、メタヒューリスティクスを用いて、タイルの辺形状を直接変形する手法がある。本項ではメタヒューリスティクスを用いてエッシャー化問題を解決する手法について紹介する。

2.3.2.1 焼きなまし法を用いた手法

Kaplan は焼きなまし法 (Simulated Annealing : SA) [11, 53] (2.4.2.1 項参照) を用いてエッシャー化問題を解く手法を提案した [17]。

この手法は、Isohedral タイリングの 93 パターンをもとに基本図形を与え、これを初期図形として、図形の辺形状をランダムな値により変化させ、評価する。評価が上がる場合は変化を受け入れ、評価が下がる場合は、温度パラメータ T による確率によって変化を受け入れるか受け入れないかを判別する。この処理を繰り返し、与えられた図形に類似する形状のタイルを生成する。この手法により凸に近い図形に対して、良質な解の生成が数値実験により示されている。

しかし、辺変形の探索空間が膨大であることや、焼きなまし法の特性上、図形間のデータ交換が不可能なため、与えられた図形形状によっては、良質な解に収束しない場合や探索時間が膨大になるという問題がある。

2.3.2.2 遺伝的アルゴリズムを用いた手法

小野らは実数値最適化によりタイル形状を直接生成する手法 (Direct Deformation using Genetic Algorithm: DDGA) を提案した [25, 26]。本論文では、小野らの手法を DDGA と省略する。

この手法は、遺伝的アルゴリズムを用いることで、目的関数と制約条件を柔軟に用いることが可能な利点により、エッシャー化問題の条件 3 (生成されたタイルに自己交差を含まない) に違反する解を淘汰することができる。さらに、この手法の解は、条件 2 (対応する 2 辺の形状が一致する制約) を満たすことが保証される。DDGA は、解析的最適化

手法 [18, 19] により初期図形を作り，辺の形状を設計変数として，辺形状の変形を繰り返すことで，タイルを生成する．また DDGA は，良好なタイルが見つかるまで，ユーザが入力する図形を再描画できるインタラクティブシステムとして実装している．

ただし，設計変数が実数値のため探索空間が膨大であるため，解の品質や処理時間に問題がある．さらに，設計変数が互いに強く依存しているため，図形形状を適切に変形するためには，複数の設計変数を同時に調整する必要がある．このため，良好なタイルを生成できない場合がある．

2.4 メタヒューリスティックス

最適化アルゴリズムには，問題に特化した手法と，問題に特化しない手法とが存在する [11]．典型的な最適化問題においては，問題に特化した最適化アルゴリズムが提案されていることが多く，問題に関する情報を用いることで，効率的に最適解を得ることができる．しかし，問題に特化した最適化アルゴリズムでは，問題に関する情報が十分に得られない場合，対象問題を限定しない汎用的な手法が必要となる．特に目的関数の入出力関係のみを用いて最適化を行うブラックボックス最適化であることが望ましい．一般に，汎用的な最適化アルゴリズムをメタヒューリスティックスと呼ぶ．本節では本論文で用いるメタヒューリスティックスについて述べる．

2.4.1 最適化問題

最適化問題とは，与えられた制約条件を全て満たしつつ，ある目的関数の値が最大または最小になる変数の値の組み合わせを求める問題である [1, 2]．最適化問題を解くことを最適化という．最適化とは，ある基準に従ってもっとも適切な状態を探索することである．

最適化問題は一般的に以下のように表される． $f(x)$ を目的関数， F を実行可能領域と呼ぶ．目的関数 $f(x)$ は実数値，あるいは整数値をとる関数 $f : F \rightarrow X$ である． $f(x)$ を

最小または最大にする実行可能解を最適解と呼ぶ。

$$\begin{aligned} & \text{Minimize } f(x) \\ & \text{Subject to } x \in F \\ & \quad F \subseteq X \end{aligned}$$

設計変数が離散値をとる場合は組み合わせ最適化問題あるいは離散最適化問題と呼ばれ、連続値をとる場合は連続最適化問題と呼ばれる。

2.4.2 局所探索

2.4.2.1 焼きなまし法

焼きなまし法 (Simulated Annealing : SA) は、大域的最適化問題への汎用の確率的メタアルゴリズムであり、遷移確立に金属加工の焼きなまし (annealing) を模している [11, 53]。金属の焼きなましとは、金属材料を高温に熱した後で徐々に冷やして、結晶を成長させて内部のひずみなどの欠陥を減らし、より良い金属を作る作業である。広大な探索空間内の与えられた関数の大域的最適解に対して良好な近似を与える。近傍内の各解に解の良さに応じた遷移確率を設定し、それに従って次の解をランダムに選ぶ。温度 (temperature) と呼ばれるパラメータ T により調整し、以下 3 つのステップを繰り返すことによって解を得る。なお、以下では最小化問題を考える。

ステップ 1 $i = 0, k = 0$ と置く。

ステップ 2 初期解 x_0 を生成し、初期温度 T_0 を設定する。

ステップ 3 以下のステップ a, b および c を、終了条件を満たすまで繰り返す。

a) 次の解候補 s を、 x_i の近傍 $\epsilon(x_i)$ からランダムに選択する。

b) $f(s) \leq f(x_i)$ ならば、受理確率を 1 とし、それ以外の場合は、確率を $\exp[-\{f(s) - f(x_i)\} / T_k]$ として、確率に従い解候補 s を受理し、 $x_{i+1} = s, i = i + 1$ とする

c) 冷却スケジュールに従い、 T_k 値の更新を行い、 $T_{k+1} = T_k, k = k + 1$ とする。

ステップ 4 終了条件が満たされれば探索を終了。

このとき、初期温度 T_0 を高温に設定することでランダムな移動を生じやすくし、大域的な探索が可能となる。探索が進むにつれて徐々に小さくすることでランダムに移動する

確率を減少させる。

評価値が $f(s) \leq f(x_i)$ の場合は $x_{i+1} = s$ とし, $f(s) > f(x_i)$ の場合は T_k の温度が高いほど $x_{i+1} = s$ となる確率が高くなり, T_k の温度が低いほど置き換える確率が低くなる。

焼きなまし法では, 初期温度 T_0 の定め方と温度の更新方法を定める必要がある。以下に基本的な手法を示す。

初期温度 T_0 の定め方

探索初期で, 解候補の中から受理される解候補の割合が SAP_{init} ($0 < SAP_{init} < 1$) になるように初期温度 T_0 を設定する。

温度の更新方法

温度 T_k の更新方法を冷却スケジュール (cooling schedule) と呼び, よく用いられる方法は以下の 2 つである。

幾何冷却法 (geometric cooling) T_k 値を $T_{k+1} = rT_k$, ($0 < r < 1$) と等比級数的に減少させる

対数冷却法 (logarithmic cooling) T_k 値を $T_{k+1} = c/(\log(k+1))$, ($0 < c$) に従って徐々に減少させる。

焼きなまし法の動作は, T_k 値が高い場合, ランダムサーチ, 低い場合, 局所探索となり, 中間値の場合, 再アニーリングが適用される。再アニーリングとは, 暫定解が見つかった時点の温度 T_{found} を記憶し, 解が更新されない場合, 温度を $T := T_{found}$ と一旦高くすることである。

2.4.2.2 タブー探索

タブー探索は過去の一定期間内に見つけた解候補をタブーリストに登録し, 近傍点の中から次の解候補を生成する際にはリストに登録されていない解候補を選択する最適化アルゴリズムである [11, 54]。タブー探索は, 局所解への収束を避けるため, 解が悪化する方向への移動も許容するが, タブーリストにより直前に探索した点への移動を禁止することで, 同じ点を何度も探索することを避けつつ局所最適化を乗り越え, 大域的最適解を求めると探索を行う。

以下にタブー探索のアルゴリズムを示す．なお，ここでは最小化問題を考える．

1. 初期探索点 x_0 をランダムに選択し， $i = 0$, $t = 0$ とする．
2. タブーリスト TL を初期化する．
3. 次の探索点の候補集団 STS を， x_i の近傍 $\epsilon(x_i)$ から x_i と TL に属する点を除いたものとする． STS が空集合でない場合は近傍探索で優れた解を求め， STS が空集合の場合は 5 へ進む．
4. STS が空集合でない場合， STS 中で最も優れた解 sts を次の探索点として受理し， $x_{i+1} = sts$, $t = 0$, $i = i + 1$ として，現在の探索点を TL に追加し， TL を更新して 6 へ進む．
5. STS が空集合の場合， $t = t + 1$ とし，もし， t が規定値 t^* に達した場合は終了．それ以外の場合は TL を更新して 6 へ進む．
6. 終了条件を満たさなければ，3 に戻る．

2.4.3 進化計算

進化計算は，生物の進化のメカニズムを模倣し，データ構造を変形，合成，選択する手法である [55]．本項では代表的な進化計算について紹介する．

2.4.3.1 自己適応型差分進化

2006 年に Brest らによって提案された自己適応型差分進化 (Self-Adaptive DE: jDE) [56] は，差分進化 [57] の制御パラメータであるスケールファクタ SF と交叉率 CR を個体ごとに保持し，一定の確率でそれぞれの値をランダムに変化させる．これにより， SF と CR の組み合わせで，個体ごとに異なる役割を持たせた探索が可能となり，さらにユーザによるパラメータの調整が不要となる．

各世代における τ_1 と τ_2 の確率により，jDE は指定された範囲内の制御パラメータ値として $SF_{i,g+1}$ と $CR_{i,g+1}$ は，以下の式に従いランダムに変更する．

$$SF_{i,g+1} = SF_l + SF_u \cdot r_1 \quad (2.11)$$

$$CR_{i,g+1} = r_2 \quad (2.12)$$

ここで， r_j は $[0, 1]$ を範囲とした一様乱数値であり， τ_1 と τ_2 は，それぞれ SF_i と CR_i を

変更する確率である。\$SF_L\$ と \$SF_u\$ は、スケールファクタの範囲を決定する。\$SF_L = 0.1\$, \$SF_u = 0.9\$, \$\tau_1 = 0.1\$, \$\tau_2 = 0.1\$ を構成することにより、jDE は良い収束特性を示した。つまり、個体集団の多様性を維持することにより局所最適を脱する [58]。ただし、収束速度は、JADE [59], SHADE [60,61], CMA-ES [62] などの他の自己適応型メタヒューリスティクスほど高速ではない。

図 2.11 に jDE の擬似コードを示す。2-3 行目で初期化を行う。まず、\$N_P\$ 個の個体 \$\mathbf{x}_{1,g}, \dots, \mathbf{x}_{N_P,g}\$ から成る個体集団 \$\mathbf{p}_g\$ を生成する。各個体は \$D\$ 次元の実数値ベクトルにより構成され、定義範囲内においてランダムに数値を生成する。生成した個体集団 \$\mathbf{p}_g\$ を目的関数 \$f(\cdot)\$ を用いて評価し、個体 \$\mathbf{x}_{i,g}\$ における評価値は \$f(\mathbf{x}_{i,g})\$ となる。終了条件を満たすまで、5 行目以降の処理を繰り返す。個体集団 \$\mathbf{p}_g\$ 内の全 \$N_P\$ 個の個体 \$\mathbf{x}_{1,g}, \dots, \mathbf{x}_{N_P,g}\$ に対して、6 行目以降の処理を行う。\$\mathbf{x}_{i,g}\$ の制御パラメータである \$SF_i\$ と \$CR_i\$ を式 (2.12), 式 (2.12) に従い更新を行う。次に \$\mathbf{x}_{i,g}\$ に遺伝子操作を適用し、探索に用いる子個体を生成する。まず、\$\mathbf{x}_{i,g}\$ に突然変異を適用し、次に \$\mathbf{x}_{i,g}\$ をターゲットベクトルとし、\$\mathbf{p}_g\$ 内からベースベクトル \$\mathbf{x}_{base,g}\$ を選択する。\$\mathbf{x}_{base,g}\$ に摂動を加え、変異ベクトル \$\boldsymbol{\nu}_{i,g}\$ を生成する。ここで、\$\boldsymbol{\nu}_{i,g}\$ を生成する操作を差分突然変異と言ひ、\$\mathbf{p}_g\$ 内からランダムに \$2Y\$ 個の個体を選び、\$Y\$ 個の対の個体 \$\mathbf{x}_{r_{2y-1,g}}, \mathbf{x}_{r_{2y,g}}\$ の差分ベクトルに \$SF_i\$ を掛け合わせ、\$\mathbf{x}_{base,g}\$ を加え、\$\boldsymbol{\nu}_{i,g}\$ を生成する。\$\boldsymbol{\nu}_{i,g}\$ を生成する式を以下に示す。

$$\boldsymbol{\nu}_{i,g} = \mathbf{x}_{base,g} + \sum_{y=1}^Y SF_i (\mathbf{x}_{r_{2y-1,g}} - \mathbf{x}_{r_{2y,g}}) \quad (2.13)$$

ただし、\$base \neq r_1 \neq r_2 \neq \dots \neq r_{2y}\$ である。次に、\$\mathbf{x}_{i,g}\$ に交叉を適用し、\$\mathbf{x}_{i,g}\$ と \$\boldsymbol{\nu}_{i,g}\$ の交叉によりトライアルベクトル \$\mathbf{u}_{i,g}\$ を生成する。本手法では、二項交叉を用いた。二項交叉では、\$CR_i\$ とランダムに選択した次元 \$j_{rand} \in [1, D]\$ に基づき、以下の式により、\$\mathbf{u}_{i,g}\$ の各要素 \$\mathbf{u}_{i,j,g} \in \mathbf{u}_{i,g} (j = 1, 2, \dots, D)\$ を決定する。

$$\mathbf{u}_{i,j,g} = \begin{cases} \boldsymbol{\nu}_{i,j,g} & \text{if } (rand_{i,j}[0, 1] \leq CR_i \text{ or } j = j_{rand}) \\ \mathbf{x}_{i,j,g} & \text{(otherwise)} \end{cases} \quad (2.14)$$

ここで、\$rand_{i,j}[0, 1]\$ は範囲 \$[0, 1]\$ の一様乱数である。上記の突然変異と交叉をまとめると以下のようなになる。

$$\mathbf{u}_{i,j,g} = \begin{cases} \mathbf{x}_{base,g} + SF_i (\mathbf{x}_{r_{2y-1,j,g}} - \mathbf{x}_{r_{2y,j,g}}) & \text{if } (rand_{i,j}[0, 1] \leq CR_i \text{ or } j = j_{rand}) \\ \mathbf{x}_{i,j,g} & \text{(otherwise)} \end{cases} \quad (2.15)$$

最後に生成された $\mathbf{u}_{i,g}$ を用いて評価する. $\mathbf{x}_{i,g}$ におけるトライアルベクトル $\mathbf{u}_{i,g}$ の評価値は $f(\mathbf{u}_{i,g})$ となる. $f(\mathbf{x}_{i,g})$ と $f(\mathbf{u}_{i,g})$ を比較し, 評価値が優れているベクトルを次世代の個体とする. 世代数を更新し, 5 行目の処理に戻る.

Algorithm 1 自己適応型差分進化のアルゴリズム

```

1:  $g \leftarrow 0$ 
2: 個体  $\mathbf{x}_1^{(g)}, \dots, \mathbf{x}_{N_P}^{(g)}$  からなる初期個体集団  $\mathbf{P}^{(g)}$  を生成する.
3: 個体集団  $\mathbf{P}^{(g)}$  内の全開口保  $\mathbf{x}_i^{(g)}$  ( $i \in \{1, \dots, N_P\}$ ) を評価する.
4: while 終了条件を満たさない do
5:   for  $i = 1$  から  $N_P$  まで実行する do
6:      $\mathbf{x}_i^{(g)}$  の制御パラメータ  $SF_i$  と  $CR_i$  を更新する.
7:      $\mathbf{x}_i^{(g)}$  に交叉と突然変異を適用する.
8:     すべてのトライアルベクトル  $\boldsymbol{\tau}_i^{(g)}$  ( $i \in \{1, \dots, N_P\}$ ) を評価する.
9:      $\mathbf{z}_i^{(g)}$  の適応度と  $\boldsymbol{\tau}_i^{(g)}$  の適応度を比較して, 優れたベクトルを  $\mathbf{P}^{(g+1)}$  に追加する.
10:  end for
11:    $g \leftarrow g + 1$ 
12: end while

```

図 2.11. 自己適応型差分進化の疑似コード

2.4.3.2 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm : GA) [11, 48] は 1970 年代 John Holland らにより提案された, 進化的計算法に代表される近似最適化アルゴリズムである. 探索する解を個体とみなし, 個体集団に対し, 選択, 交叉, 突然変異といった生物の進化過程を模倣した操作を行うことで解探索を行う. GA の原理は, ダーウィンの「種の起源」に端を発する進化論に基づく遺伝子による進化を参考にしている. 生物は細胞の中に遺伝子の組み合わせである染色体をもっており, 生物のすべての先天的特徴は, 染色体がどのような遺伝子の組み合わせで成り立っているかによって決定される. 生物の進化には, それぞれ異なる染色体を持つ個体の集団が必要である. 集団内では 2 つの個体を親とする交配によって, 新たな世代の個体が生み出される. この新たな世代の個体は, 双方の親の染色体を交叉させた遺伝子を受け継ぐ. 一方で, 双方の親とも異なる遺伝子の組み合わせとなるため, 親の世代から進化した個体となる. さらに, 低い確率で発生する突然変異によって

遺伝子には変化が生じる。この変化は前世代の遺伝子とは無関係であり、交配とは異なる進化をする。これらの進化は親個体からのランダムな変化であり、進化後の個体が親の個体より優れたものであるとは限らない。しかし、生物は進化の結果、その生物を取り巻く環境に適した特徴を持っている。つまり、個体集団内で環境に適応できない個体が死滅し、環境に適応する特徴を持つ優秀な個体が生き残り、次世代の親となる。これにより、環境に適した特徴を発現する遺伝子を持つ個体が集団内に増えるため、生物は集団として環境に適した方向へと進化していくことができる。

GA の基本的な処理手順は図 2.12 の疑似コードで示すように、初期集団の生成に始まり、終了条件を満たすまで、評価、複製選択、交叉、突然変異、生存選択の処理が繰り返し行われるという手順になっている。図 2.12 の疑似コードにおける各手順についての詳細を下記に示す。

Algorithm 2 遺伝的アルゴリズム

```
1:  $g \leftarrow 0$ 
2: 個体  $\mathbf{x}_{1,g}, \dots, \mathbf{x}_{N_P,g}$  からなる初期個体集団  $\mathbf{p}_g$  を生成する
3: while 停止条件を満たさない間 do
4:   個体集団  $\mathbf{p}_g$  内の全個体  $\mathbf{x}_{i,g}$  ( $i \in \{1, \dots, N_P\}$ ) を評価する
5:    $\mathbf{p}_g$  から  $N_{pr}$  個の親個体  $\mathbf{x}_{j,g}$  ( $j \in \{1, \dots, N_{pr}\}$ ) を選択する
6:   親個体に遺伝子操作を行い、 $N_{ch}$  個の子個体  $\mathbf{x}_{k,g+1}$  ( $k \in \{1, \dots, N_{ch}\}$ ) を得る
7:   子個体集団  $\mathbf{x}_{k,g+1}$  ( $k \in \{1, \dots, N_{ch}\}$ ) の中から、親個体と入れ替える個体を選択する
8:   選択した子個体と親個体を入れ替えた集団を個体集団  $\mathbf{p}_{g+1}$  とする
9:    $g \leftarrow g + 1$ 
10: end while
```

図 2.12. 遺伝的アルゴリズムの疑似コード

初期集団生成 (Initialization)

疑似コードの 2 行目では、初期個体集団の生成を行う。生成される個体数は集団サイズによって定められており、各個体は解候補となりうる個体を持っている。一般

的に初期集団として生成される個体はランダムに遺伝子を組み合わせる構成される。大域的な解の探索を行うためにも、初期集団の解候補は多様性を持たせて生成することが重要である。

一般に、集団サイズが小さい場合、1 世代あたりの計算量は小さくなるが、解に収束するまでの世代数が大きくなるとともに、局所解へ陥る可能性が高くなる。一方、集団サイズが大きい場合は、計算量が大きくなるが、多様な解候補を得ることができる。このため、対象とする問題によって最適な集団サイズは異なる。

評価 (Evaluation)

疑似コードの 3 行目では、個体集団が終了条件を満たしているか判定する。条件としては一般的に次のようなものが挙げられる。

- 進化世代が規定数を超える
- 個体集団の適応度の平均値が規定値を超える
- 個体集団の最大または最小適応度が規定値を超える

ここで、適応度 (fitness) とは、どれだけその個体が優秀であることを示したもので、対象とする問題の最適解に近いほど良い値を示す。

疑似コードの 4 行目では、解候補の評価を行う。あらかじめ問題に応じた目的関数 (評価関数) を定めておき、これを用いて目的関数値、つまり、適応度を求める。目的関数は解の品質や性能を表す関数である。

選択 (Selection)

疑似コードの 5 行目の複製選択では、個体集団の中から交叉、突然変異などの遺伝的操作の対象となる親個体を選択する。また、疑似コードの 7 行目の生存選択では、評価によって得られた適応度に基づいて、個体集団の中から次世代に子孫を残すことができる個体を選択する。一方で、死滅させる個体も選択する。複製選択および、生存選択の手法は世代交代モデルによって決まる。以下に各世代交代モデルについて記載する [63]。

1. SGA (SimpleGA) 複製選択では適応度に比例した選択確率を用いたルーレット選択 [11] によって、集団から個体を復元抽出する。すなわち、親個体においては重複する可能性も起こる。生存選択では親の個体を新たに生成された子個体で置き換える。

2. IGS (Iterated Genetic Search) 複製選択では適応度を無視して、母集団から個体をランダムに非復元抽出する。すなわち、母集団から重複無く全ての親個体を用いて組み合わせを作る。生存選択では親集団から平均適応度以下の個体をランダムに選択し、生成された子個体と入れ換える。
3. SS (Steady State) 複製選択ではランキング選択 [11] によって母集団から個体を復元抽出する。生存選択では親集団から最悪個体を選び、生成された子個体と入れ換える。
4. CHC (C: Cross generational elitist selection, H: Heterogeneous recombination, C: Cataclysmic mutation) 複製選択では適応度を無視して、母集団から個体をランダムに非復元抽出する。親集団と子集団を合わせた 2 世代の中から、適応度の高い順に母集団サイズ分の個体を次世代に残す。
5. ER (Elite Recombination) 複製選択では適応度を無視して、母集団から個体をランダムに非復元抽出する。生存選択では各家族 (2 つの親個体と 2 つの子個体) の中から、最良 2 個体を次世代に残す。
6. MGG (Minimal Generation Gap) [64] 複製選択では個体集団からあらかじめ定められた数 (一般的には 2 つ) の個体をランダムに選択し、親個体とする。生存選択では家族の中で最良の適応度をもつ解候補の保存と、ルーレット選択 [11] を行う。家族から最良の適応度を持つ個体と、その個体を除く家族集団からルーレット選択によって選ばれた 1 個体の計 2 個体を次世代に残す。以下に、MGG のアルゴリズムを示す。
 - 1) 初期化: 個体集団 p_0 の初期化。
 - 2) 複製選択: 個体集団 p_g から親個体 $x_{pr,g}$ をランダムに 2 個体選択 (g は世代数)。
 - 3) 遺伝子操作: 親個体 $x_{pr1,g}, x_{pr2,g}$ から子個体 $x_{of,g}$ を生成。
 - 4) 生存選択: 家族集団 $x_{pr1,g}, x_{pr2,g}, x_{of,g}$ から 2 個体選択する。1 個体目は、エリート選択, 2 個体目はルーレット選択により選び、親個体 $x_{pr1,g}, x_{pr2,g}$ と置き換え、次世代の個体集団 p_{g+1} とする。
 - 5) 終了判定: 終了条件を満たさなければ, 2 に戻る。

交叉 (Crossover)

疑似コードの 6 行目の交叉は, 2 つの個体を親として互いの個体の一部を交換した新たな遺伝子をもつ個体を生成するものである. GA における一般的な交叉方法として, 1 点交叉, 多点交叉, 一様交叉が挙げられる.

1. 1 点交叉 (One-point Crossover) は, 親となる個体の染色体の切断箇所をランダムに一カ所指定し, 指定された箇所を交叉点として 2 つの親の遺伝子を交叉させる (図 2.13).
2. 多点交叉 (N-point Crossover) 親となる個体の染色体の切断箇所をランダムに数カ所 (図の例では 2 カ所) 指定し, それらの箇所を交叉点として 2 つの親から交互に遺伝子を受け継ぐ (図 2.14).
3. 一様交叉 (Uniform Crossover) は, 交叉時にランダムなマスクパターンを作成し, そのマスクパターンをもとに, 二つの親個体の持つ染色体の入れ替えを行い, 子個体の染色体を決定する交叉手法である. 図 2.15 に交叉の例を示す. ビットストリング型の遺伝子を持つ二つの親個体から, 二つの子個体を生成する. マスク配列の長さは, 染色体長に設定し, マスクのバイナリ情報をもとに子個体に親の遺伝子を複製する.

突然変異 (Mutation)

疑似コードの 6 行目の突然変異は, ある一定の確率に基づいて個体の一部の遺伝子を他の遺伝子に置き換える操作である. 交叉による探索は親の遺伝子に依存するため, 限られた範囲しか探索できず局所解に陥ってしまう場合がある. そこで, まったく異なる遺伝子に変更する突然変異を用いることによって, より広い範囲を探索することができる. 突然変異は局所解への収束を防ぐことができるが, 突然変異率 (発生率) を高くすると, ランダムな探索になり, 交叉による進化的な探索を阻害してしまう. このため一般に突然変異率は低い値に設定される.

2.4.3.3 分散推定アルゴリズム

分散推定アルゴリズム (Estimation of Distribution Algorithm: EDA) [65,66] とは, μ 個の個体により構成される個体集団の中から, 優れている個体を M 個選択し, 選択された優れた個体に関する何らかの確率モデルを構築し, この確率モデルに従い, 次世代の個体集団を生成するアルゴリズムである.

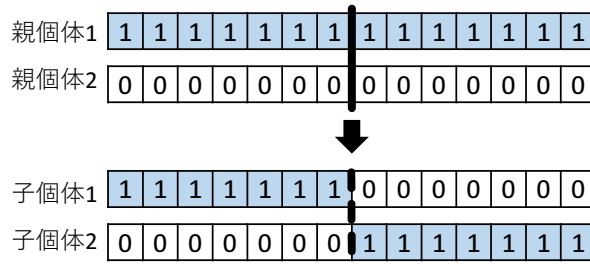


図 2.13. 1 点交叉

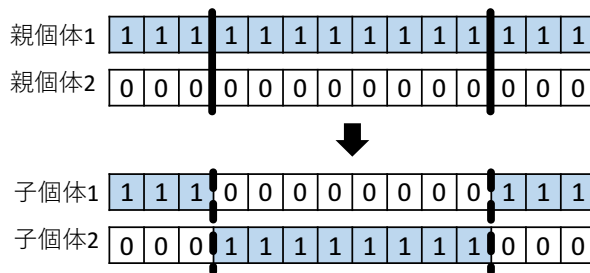


図 2.14. 多点交叉

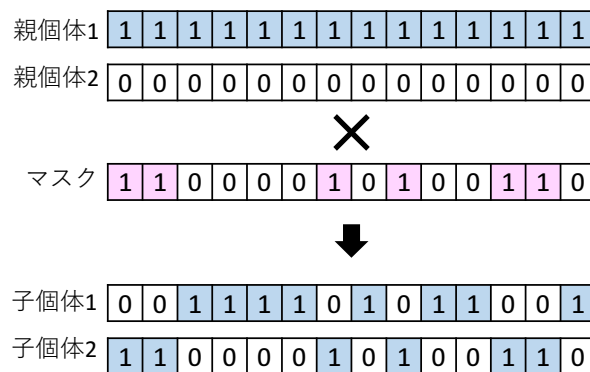


図 2.15. 一様交叉

一般化した EDA のアルゴリズムは、以下の通りである。

1. 初期化： 個体集団 p_0 の初期化 (個体数 μ)。
2. 評価： 個体集団 p_g の全個体 $x_{i,g} (i \in \{1, \dots, \mu\})$ の適応度の評価を行う (g は世代数, i は個体の識別番号)。
3. 選択： 適応度をもとに優れた個体を M 個選択し, 個体集団 m_g をつくる。
4. 確率モデル構築： 選択された個体集団 m_g をもとに確率分布を推定する。

5. 個体生成： 推定された確率分布をもとに次世代の個体集団 p_{g+1} を生成する.
6. 終了判定： 終了条件を満たさなければ, 2 に戻る.

EDA は GA と異なり, 適応度の高い順に個体を選択し, 確率分布を推定するモデルを構築して, そのモデルに基づいて次世代の個体集団を生成することである.

■(a) **PBIL** PBIL(Population-based incremental learning) は, GA の個体集団 p_g 中に含まれる個体 $x_{i,g}$ の遺伝子の分布を学習して最適化を行う手法である [65,66].

以下に PBIL のアルゴリズムを示す.

1. $p_c = 0.5(c \in \{1, \dots, L\}, L: \text{個体の遺伝子長})$ として初期化する.
2. 確率 p_c に基づいて, μ 個の個体 $x_{i,g}(i \in \{1, \dots, \mu\})$ を生成する.
3. 全個体 $x_{i,g}$ の評価を行う.
4. 個体 $x_{i,g}$ を適応度が優れている順にソートし, 上位 M 個 ($M < \mu$) の個体 $x_{pro,g}(pro \in \{1, \dots, M\})$ を用いて確率 p_c を更新する.
5. 終了条件を満たせば終了し, それ以外は 2 に戻る.

ここで, 確率 p_c は c 番目の遺伝子が 1 となる確率を示している. また, 確率 p_c の更新式は以下の通りである.

$$p_c = p_c(1.0 - R) + x_{pro,g}^c \cdot R \quad (2.16)$$

ここで, $x_{pro,g}^c$ は $x_{pro,g}$ の c 番目の遺伝子を表す. また, R ($0.0 < R < 1.0$) は, 学習の速度を決定するパラメータであり, 通常, 0.0 に近い値を用いる. PBIL で用いられる確率モデルは, 確率変数間の相互依存性がなく, 独立して学習を進める. PBIL は優れた個体 $x_{pro,g}(pro \in \{1, \dots, M\})$ の遺伝子の確率分布を学習することで最適解を求める手法であり, EDA として最も単純なアルゴリズムの 1 つである.

2.4.3.4 粒子群最適化

粒子群最適化 (Particle Swarm Optimization) は群知能の 1 つであり, 社会的振る舞いを参考にして考えられたアルゴリズムである [55,67]. 遺伝子操作を行う GA とは異なり, 粒子群最適化では自分と仲間の動きにより次の行動が決定される.

粒子群最適化の個体は, 位置ベクトル loc_i , 速度ベクトル v_i と, 個体が最良の適応度を

獲得した場所 pos_i を記憶している。そして、個体全体における最も良い適応度を獲得した場所 pos_g も各個体が共有している。

速度ベクトル v_i は、以下の式により更新される。

$$v_i = \chi(\omega v_i + \kappa_1 \cdot (pos_i - loc_i) + \kappa_2(pos_g - loc_i)) \quad (2.17)$$

ここで、 χ は収束係数を示し、 $0.9 \leq \chi \leq 1.0$ の範囲からランダムに選択され、 ω は減衰係数を示す。また、 κ_1, κ_2 は各個体と次元固有の乱数であり、その上限値は 2.0 である。もし、速度ベクトル v_i がある制限を超えた場合、最大速度 V_{max} が代わりに使用される。こうすることで、探索領域内に個体を保ちつつ、探索を行うことが可能となる。

さらに各個体の位置ベクトル loc_i は世代ごとに $loc_i \leftarrow loc_i + v_i$ により更新される。

以下に粒子群最適化のアルゴリズムを示す。

1. 初期化する。
2. 各次元において、過去に獲得した位置よりも適応度が良い場合、現在の位置を pos_i に保存する。
3. 各次元において、 pos_i のうち、最も適応度が良いものを pos_g に保存する。
4. 各次元において、上記で探索した pos_i, pos_g と以前の速度 v_i から、次の速度 $v_{i+1}(t)$ を決定する。
5. $v_{i+1}(t)$ が制限速度内なら次の位置 $loc_{i+1}(t)$ を求める。制限速度を超えた場合、最大速度 V_{max} に置き換える。
6. 個体数分だけ繰り返す。

2.4.4 ハイブリッドアルゴリズム

最適化問題を対象とするもっとも一般的なハイブリッドアルゴリズムは、メタヒューリスティクスによる大域探索と、問題領域の知識を利用した局所探索を組み合わせた探索手法である [13–15]。GA は高い大域的探索能力を発揮する一方で、局所探索能力の欠如することが多くの研究で示されている。この点を補うために、GA と局所探索を組み合わせた枠組みの開発が盛んに行われている [2, 68, 69]。最適化問題を対象とするハイブリッドアルゴリズムの中で、メタヒューリスティクスの欠点を補うために、メタヒューリスティ

クスと局所探索を組み合わせた遺伝的局所探索 [68] と、2 つのメタヒューリスティクスを組み合わせた Parallel recombinative simulated annealing [2, 69] について説明する。

2.4.4.1 遺伝的局所探索

遺伝的局所探索 (Genetic Local Search) は、個体を評価する際に、個体が表す解をそのまま評価するのではなく、その個体を初期値として局所探索を行う手法である [68]。

遺伝的局所探索では、GA の各個体に局所探索を適用し、GA による大域的な探索を、局所探索による解の近傍の系統的な探索を行う手法である。用いる局所探索は種々のアルゴリズムが考えられ、問題の領域に応じた固有の工夫も導入しやすい。この局所探索と GA を組み合わせることにより、両者の欠点を克服することが可能である。このようなハイブリッド化を可能としているのは、GA が個体集合を対象に目的関数のみを頼りに探索を進めているため、適用範囲が広い手法であることによる。

遺伝的局所探索のアルゴリズムを以下に示す。

1. GA により個体集団を初期化する。
2. 各個体を初期値として局所探索を実行する。
3. 局所探索により得られた解と適応度をもとに、GA が遺伝的操作を行う。
4. GA の終了条件を満たさない場合は、2 へ戻る。

2.4.4.2 Parallel recombinative simulated annealing

Parallel recombinative simulated annealing (PRSA) は、焼きなまし法による探索に GA の特徴である個体集合と遺伝子操作を組み合わせた手法である [2, 69]。

PRSA は、並列に実行している焼きなまし法の間で、探索で得た情報を交叉により交換するものである。一方、GA の選択演算は親子に限定されており、良い個体を増殖させることによる探索の重点化は生じない。

以下に PRSA のアルゴリズムを示す。なお、ここでは最小化問題を考える。

1. ランダムに M 個の個体を生成し初期個体集団 p_0 とし、初期温度 T_0 を設定する。
2. 個体をランダムにペアリングし、各ペアに対して以下の処理を行う。
 - a) 2 個の親個体 $x_{pr1,g}, x_{pr2,g}$ に遺伝子操作を適用し、2 つの子個体 $x_{of1,g}, x_{of2,g}$

を生成する.

- b) 2 組の親子のペアのそれぞれについてトーナメント選択を行う. 確率は $par = 1/[1 + \exp\{(f(x_{pr,g}) - f(x_{of,g})) / T_k\}]$ により求める. 次世代に残る確率は, 親個体が par であり, 子個体が $(1 - par)$ である.

3. 終了条件を満たせば終了し, 満たさない場合は冷却スケジュールに従って, T_k 値を更新し, 2 に戻る.

2.4.5 2 段階最適化

2 段階最適化 (Bilevel optimization) は, 1 つの最適化がもう一方の最適化の制約の役割を持つ 2 段階の最適化を含む手法である [70, 71]. 上位レベルの最適化がリーダーとなり, 下位レベルの最適化がフォロワーとして, 組織的階層により問題の解決を行う. 2 段階最適化は, 入れ子構造のため, 問題を解くことが難しい. 特に, 2 段階線形計画問題のような簡単なインスタンスに対してさえ, 非凸や切断性の問題がある. 2 段階線形計画問題は NP 困難であり, 最適化の解を評価するだけでも NP 困難なタスクであることが証明されている.

2 段階最適化の一般的な定式化は以下の数式で表される.

$$\min_{x \in X, y} F(x, y) \tag{2.18a}$$

$$s.t. G(x, y) \leq 0 \tag{2.18b}$$

$$\min_y f(x, y) \tag{2.18c}$$

$$s.t. g(x, y) \leq 0 \tag{2.18d}$$

ここで, $x \in \mathbb{R}^{n_1}$, $y \in \mathbb{R}^{n_2}$ である. 上記の式の変数は, 上位変数 $x \in \mathbb{R}^{n_1}$ と下位変数 $y \in \mathbb{R}^{n_2}$ の 2 つのクラスに分けられる. 同様に目的関数は, 上位目的関数 $F: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}$ と, 下位目的関数 $f: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}$ から成る. また, ベクトル値関数 $G: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{m_1}$ は上位制約を示し, $g: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{m_2}$ は下位制約を示す. また, 上位制約 G は, 両方のレベルの変数が含まれており, 特殊な役割を果たす. 上位レベル最適化は, 下位レベル最適化の意思決定者を拘束しないため, 間接的に最適化を行う必要がある.

式 (2.18) に関する緩和問題は, 以下の数式で表され, この式の最適値が式 (2.18) の

最適値の下界となる.

$$\min_{x \in X, y} F(x, y) \quad (2.19a)$$

$$s.t. G(x, y) \leq 0 \quad (2.19b)$$

$$s.t. g(x, y) \leq 0 \quad (2.19c)$$

ここで, 緩和された実行可能領域は, 以下の式で表される.

$$\Omega = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, G(x, y) \leq 0, g(x, y) \leq 0\} \quad (2.20)$$

与えられた固定ベクトル $\bar{x} \in X$ に対する下位レベル最適化の実行可能領域は, 以下のよう
に定義される.

$$\Omega(\bar{x}) = \{y \in \mathbb{R}^{n_2} : g(\bar{x}, y) \leq 0\} \quad (2.21)$$

一方, 下位レベル最適化の反応ベクトルは, 以下の数式で表される.

$$R(\bar{x}) = \{y \in \mathbb{R}^{n_2} : y \in \operatorname{argmin}\{f(\bar{x}, \hat{y}) : \hat{y} \in \Omega(\bar{x})\}\} \quad (2.22)$$

ここで, 2 段階最適化の定義に従うと, 下位レベル最適化は, 大域的最小化が解ける問題である必要がある. つまり, 下位レベル最適化は凸である必要がある. すべての $y \in R(\bar{x})$ は, 論理的反応である. 与えられた x に対して, $R(x)$ は暗黙的に定義された x の多価関数であり, この引数値によっては, 空の場合もある. 最後に, 2 段階最適化の実行可能点を再グループ化した集合 \mathcal{IR} は, 以下の数式で表せる.

$$\mathcal{IR} = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, G(x, y) \leq 0, y \in R(x)\} \quad (2.23)$$

\mathcal{IR} は, リーダーの実行可能領域に対応して, 誘導領域と呼ぶ. \mathcal{IR} は, 一般的に非凸であり, 上位制約のもと切断される場合, 空の場合がある.

また, 2 段階最適化には, 2 つのモデルアプローチがある. 1 つ目は, 楽観的 2 段階最適化と呼ばれ, 反応集合 $R(x)$ がシングルトンでない場合, リーダーは $\Omega(x)$ の中から自分に最も適した要素を選択することを仮定する. このとき, 点 $(x^*, y^*) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ は以下のように定義され, 式 (2.18) の局所最適解と呼ぶ.

$$x^* \in X, \quad (2.24a)$$

$$G(x^*, y^*) \leq 0, \quad (2.24b)$$

$$y^* \in R(x^*), \quad (2.24c)$$

$$F(x^*, y^*) \leq F(x^*, y) \text{ for all } y \in R(x^*) \quad (2.24d)$$

さらに、次のような x^* (半径 $\delta > 0$) の開近傍 $V(x^*; \delta)$ が存在する.

$$\phi_0(x^*) \leq \phi_0(x) \text{ for all } x \in V(x^*; \delta) \cap X \quad (2.25)$$

ここで、 $\phi_0(x) = \min_y \{F(x, y) : y \in R(x)\}$ となる. $V(x^*) = X$ に対応して、 $\delta = \infty$ が選択できる場合、大域的楽観解と呼ぶ.

一方、リーダーとフォロワーの協力が許されない場合やリーダーがリスク回避のため、フォロワーが望まない選択を行った場合のダメージを制限した場合を悲観的局所解と言う. 点 $(x^*, y^*) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ は式 (2.18) の悲観的局所解となり、以下で表現できる.

$$x^* \in X, \quad (2.26a)$$

$$G(x^*, y^*) \leq 0, \quad (2.26b)$$

$$y^* \in R(x^*), \quad (2.26c)$$

$$F(x^*, y^*) \geq F(x^*, y) \text{ for all } y \in R(x^*) \quad (2.26d)$$

さらに、次のような x^* ($\delta > 0$) の開近傍 $V(x^*; \delta)$ が存在する.

$$\phi_p(x^*) \leq \phi_p(x) \text{ for all } x \in V(x^*; \delta) \cap X \quad (2.27)$$

ここで、 $\phi_p(x) = \max_y \{F(x, y) : y \in R(x)\}$ となる. $\delta = \infty$ が選択できる場合、大域的悲観解と呼ぶ.

これらのモデルアプローチは、フォロワーの視点から説明できる. 楽観的解は、協調的な行動から得られる解であり、悲観的解は、攻撃的なフォロワーにより得られる解である.

2.5 オートエンコーダ

オートエンコーダとは、教師なし学習の1種であり、特徴抽出手法の1つである [12]. オートエンコーダは、順伝搬でループのないニューラルネットワークを用いて表現学習を行う. 表現学習は、ニューラルネットワークを含む機械学習の技術である. ニューラルネットワークの各層はデータの特徴量を学習する. つまり、 $t+1$ 番目の層は、 t 番目の層が学習した表現を用いて学習する. 層が進むごとに単純な表現から複雑な表現を学習し、概念の階層と呼ばれる構造を構築する. この階層は、上位層では抽象的になり、下位層では具体的になる.

オートエンコーダは、エンコーダとデコーダから構成される. エンコーダは、表現学習を行うことで入力されたデータを別の表現に変換し、デコーダは、変換された表現を元の

形式に再変換する。つまり、オートエンコーダは生成モデルの 1 種である。エンコーダは、入力データの新しい表現を学習する役割を担う。ここで、新しい表現は、元の特徴量と入力データの集合から得られた新しい特徴量集合を示す。エンコーダ関数は以下の式で表される。

$$\mathbf{h} = f(\mathbf{x}) \quad (2.28)$$

ここで、入力データ \mathbf{x} を学習済のエンコーダに与えることにより、新しい表現 \mathbf{h} を用いて、より高解像度な出力画像を生成する予定である。

一方、エンコーダ関数の出力を用いて、入力データに近似する \mathbf{r} を再構成するデコーダ関数は、以下の式で表される。

$$\mathbf{r} = g(\mathbf{h}) \quad (2.29)$$

デコーダ関数は、エンコーダの出力 \mathbf{h} を入力とし、再構成関数 g を用いて \mathbf{r} を出力する。

オートエンコーダは、損失関数を最小化することにより、学習を行う。損失関数は、再構成誤差が可能な限り小さくなるように定義される。ここで、再構成誤差とは、デコーダが \mathbf{h} を元に再構成した \mathbf{r} と \mathbf{x} の差のことである。

2.5.1 Variational Autoencoder (VAE)

変分オートエンコーダ (Variational Autoencoder : VAE) とは、平均値ベクトル $\boldsymbol{\mu}$ と分散ベクトル $\boldsymbol{\sigma}$ が出力されるオートエンコーダである [12, 27]。この 2 つのベクトルは、一連の確率変数を構成する。つまり、 i 番目の μ_i と i 番目の σ_i は、 i 番目の確率変数の平均値と標準偏差を表す。このように確率的な出力をエンコーダが構成することで、VAE は入力データから学習した情報に従って連続空間からサンプリングすることが可能である。

VAE の学習方法について説明する [27]。エンコーダは、平均値 $\boldsymbol{\mu}$ と分散 $\boldsymbol{\sigma}$ を出力し、入力データポイント \mathbf{x} のための潜在変数 \mathbf{z} を以下のように生成する。

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (2.30)$$

ここで、 \odot は要素ごとの積を表す。

各パラメータを求めるために、入力データポイント \mathbf{x} の N 個の互いに独立で同一の分布に従うサンプルからなるデータセット $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ について考える。 \mathbf{X} は観測され

ていない潜在変数 \mathbf{z} を含む、何らかのランダムな処理により生成されたと想定する。この処理は 2 つのステップから構成される。(1) 事前分布 $p_{\theta^*}(\mathbf{z})$ から $z^{(i)}$ が生成される。(2) 条件付き分布 $p_{\theta^*}(\mathbf{x}|\mathbf{z})$ から $x^{(i)}$ が生成される。ここでは、事前分布 $p_{\theta^*}(\mathbf{z})$ と尤度 $p_{\theta^*}(\mathbf{x}|\mathbf{z})$ は、分布 $p_{\theta}(\mathbf{z})$ と $p_{\theta}(\mathbf{x}|\mathbf{z})$ のパラメトリックの集合から来ており、これらの確率密度関数は、 θ と \mathbf{z} の両方に関してほぼ全ての場所で微分可能であることを仮定する。しかし、真のパラメータ θ^* と潜在変数 $z^{(i)}$ の値は未知である。また、真の事後分布 $p_{\theta}(\mathbf{z}|\mathbf{x})$ は、解決が困難であるため、近似した認識モデル $q_{\phi}(\mathbf{z}|\mathbf{x})$ を導入する。ここで、 $q_{\phi}(\mathbf{z}|\mathbf{x})$ をエンコーダ、 $p_{\theta}(\mathbf{x}|\mathbf{z})$ をデコーダと呼ぶ。

限界尤度 $p_{\theta}(\mathbf{x})$ は、個々のデータポイントの限界尤度の和 $\log p_{\theta}(x^{(1)}, \dots, x^{(N)}) = \sum_{i=1}^N \log p_{\theta}(x^{(i)})$ から構成され、それぞれ以下のように書き換えることができる。

$$\log p_{\theta}(x^{(i)}) = D_{KL}(q_{\phi}(\mathbf{z}|x^{(i)})||p_{\theta}(\mathbf{z}|x^{(i)})) + \mathcal{L}(\theta, \phi; x^{(i)}) \quad (2.31)$$

最初の右辺は、真の事後分布からの近似の KL ダイバージェンスである。この KL ダイバージェンスは正であり、2 番目の右辺の項 $\mathcal{L}(\theta, \phi; x^{(i)})$ は、データポイント i の限界尤度の変分下限と呼ばれ、次のように書くことができる。

$$\log p_{\theta}(x^{(i)}) \geq \mathcal{L}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log q_{\phi}(\mathbf{z}|\mathbf{x}) + \log p_{\theta}(\mathbf{x}, \mathbf{z})] \quad (2.32)$$

ここで、 $\mathcal{L}(\theta, \phi; x^{(i)})$ は以下のように書くことができる。

$$\mathcal{L}(\theta, \phi; x^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|x^{(i)})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|x^{(i)})} [\log p_{\theta}(x^{(i)}|\mathbf{z})] \quad (2.33)$$

変分パラメータ ϕ と生成パラメータ θ の両方に関して、微分して下限 $\mathcal{L}(\theta, \phi; x^{(i)})$ を最適化したい。しかし、 ϕ を基準とした下限の勾配は少し問題がある。このタイプの問題に対する通常の（ナイーブな）モンテカルロ勾配推定は、次の通りである。

$$\nabla_{\phi} \mathbb{E}_{q_{\phi}(\mathbf{z})} [f(\mathbf{z})] = \mathbb{E}_{q_{\phi}(\mathbf{z})} [f(\mathbf{z}) \nabla_{q_{\phi}(\mathbf{z})} \log q_{\phi}(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^L f(\mathbf{z}^{(l)}) \nabla_{q_{\phi}(\mathbf{z}^{(l)})} \log q_{\phi}(\mathbf{z}^{(l)}) \quad (2.34)$$

ここで、 $\mathbf{z}^{(l)} \sim q_{\phi}(\mathbf{z}|x^{(i)})$ である。

しかし、この勾配推定は非常に高い分散を示すため、VAE には不向きである。

この問題を解決するために、 $q_{\phi}(\mathbf{z}|\mathbf{x})$ からサンプルを生成する別の方法を導入する。 $q_{\phi}(\mathbf{z}|\mathbf{x})$ は、以下の対角共分散構造を持つ多変量の正規分布として仮定できる。

$$\log q_{\phi}(\mathbf{z}|\mathbf{x}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}) \quad (2.35)$$

さらに, [27] にて提案されている推定器では, 以下のように KL ダイバージェンスを計算し, 推定を行わずに微分することが可能である.

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) \simeq \frac{1}{2} \sum_{d=1}^{D_l} (1 + \log((\sigma_d)^2) - \mu_d^2 - \sigma_d^2) + \frac{1}{L} \sum_{l=1}^L \log p_{\boldsymbol{\theta}}(\mathbf{x} | \mathbf{z}^{(l)}) \quad (2.36)$$

ここで, μ_d と σ_d は, $\boldsymbol{\mu}$ と $\boldsymbol{\sigma}$ の d 番目の要素を示し, $\boldsymbol{\theta}$ と $\boldsymbol{\phi}$ は, 式 (2.36) の勾配を使用して同時に最適化される.

2.5.2 Convolutional Variational Autoencoder (ConvVAE)

畳み込み変分オートエンコーダ (Convolutional VAE: ConvVAE) は, VAE のエンコーダとデコーダに畳み込み層と逆畳み込み層を追加したオートエンコーダである [72].

VAE のエンコーダとデコーダに畳み込み層と逆畳み込み層を追加することにより, 画像から特徴を抽出しやすくなる [73].

逆畳み込み層は, 入力 of 空間的なアップサンプリングを行い, 画像の潜在変数の生成モデルと意味的セグメンテーションアルゴリズムには重要な要素である. この逆畳み込み層の目的は, 逆畳み込みを行うことで特徴量の数を減らしながら入力の空間サイズを増加させることである. 大域的表現に基づく生成モデルは, まず, 空間分解能が小さく, 特徴量の数が多い他の表現 (例えば, 潜在変数のような数値) に線形的にマッピングされる. 次に逆畳み込み層により, 逆畳み込み演算が潜在変数のような表現に適用されて, 空間分解能を徐々に増加させ, 特徴量を減少させる. 最後の出力層では, 画像などの形で出力する. 畳み込み層と逆畳み込み層を導入することで, 画像から特徴を抽出することができるため, 画像を処理する際に ConvVAE は用いられる場合がある.

詳細な説明は, 2.5.1 項を参照されたい.

第 3 章

階層型最適化方式の提案

本章では，提案する階層型最適化方式を具体化した第 1 の手法，すなわち，与えられた目標図形に類似したタイルを生成する手法を提案する．本手法は，先行研究と同様の定式化のもとでエッシャー化問題を解く．提案手法 1 はエッシャー化問題を二階層で表現し，上位最適化層にはメタヒューリスティクスを用い，下位最適化層にはタイリングに必要な制約条件を満たす図形を解析的に導出する解析的最適化手法を用いる．提案手法 1 は，エッシャー化問題が含む 2 つの難しさを解決することで，目標図形に類似したタイルを生成する．提案手法 1 の上位最適化層において目的関数が曖昧な難しさを解決し，下位最適化層において制約条件が強い難しさを解決する．実験結果より，本手法は，タイリングが可能な条件を満たしつつ，目標図形に類似するタイルが生成できることを確認した．

まず，3.1 節では，先行研究の問題点，および，本章の要点について述べる．次に，3.2 節では，提案手法の主な考え方と形状空間における解釈を述べる．3.3 節では，階層型最適化方式や双方向マッピングベース関数など，提案手法の詳細を示す．3.4 節では，実験により提案手法の有効性を先行研究と比較した．3.5 節では，本章の結論を述べる．

3.1 はじめに

本章では，階層型最適化方式を具体化し，エッシャー化問題に適用した方式の提案を行う．この方式は，先行研究 (第 2 章を参照) と同様の定式化のもとでエッシャー化問題を解く．

最適化問題の中には，制約条件が強く，かつ，解の評価基準が曖昧であるなど，種類の異

なる複数の難しさを含むために求解が困難な問題が存在する。このような問題の一例として、エッシャー化問題 [16,17] (タイル生成問題) がある。このエッシャー化問題を解く手法として、これまでに、小泉らの解析的最適化手法 [18,19](2.3.1.1 項を参照) や Kaplan, 小野らのメタヒューリスティクスを用いてエッシャー化問題を解決する手法 [17, 25, 26] (2.3.2 項を参照) が提案されている。

しかし、これらの先行研究にはいくつかの問題点がある。解析的最適化手法 [18,19] は、生成されたタイル U に自己交叉を含んでしまう問題や解析的最適化手法の入力となる粗い目標図形 \tilde{W} の形状に非常に敏感である問題がある。これらの問題から、解析的最適化手法は \tilde{W} の形状を調整しつつ、適切なタイリングパターンを選択する必要があるため、人手による試行錯誤が必要となる。

メタヒューリスティクスを用いてエッシャー化問題を解決する手法 [17, 25, 26] は、目的関数や制約条件を柔軟に設定できる一方、タイル U の形状を直接最適化しているため、探索空間が膨大であり、変数間の依存が強いため、十分に目標図形 W に類似する U を生成することが難しい。Kaplan は、ほぼ凸形状に近い W に対してのみ、 W に類似する U が生成できことを実証しており [17]、GA をベースとした DDGA [25,26] では、最適化の際にユーザのサポート、および、初期集団の生成に解析的最適化手法を用いるという2つの支援が必要であることがわかっている。

本章では、階層型最適化手法を用いたエッシャー風タイルを生成する手法を提案する。またこの手法を本論文では、Escher-Like Tile design method using Hierarchical Optimization と命名し、以降この手法を指す際は、ELTHON と呼ぶ。本手法は、入力された粗い目標図形 \tilde{W} から高速に解を導出する解析的最適化手法 [18,19] と、さまざまなメタヒューリスティクスを組み合わせた手法である。エッシャー化問題は、入力される目標図形 W の形状や解きたいタスクにより、適切な図形間の類似度の評価指標が異なるため、メタヒューリスティクスの柔軟性が必要である。しかし、DDGA [25,26] で示されているように、同じ問題を組み合わせで解決する単純なハイブリッドアルゴリズムは、タイルを設計することに適していない。そこで、ELTHON では、問題の難しさを分割する階層的なアプローチを採用する。階層的に組み合わせる最適化手法は、それぞれの長所を活かして分割した問題を解決する。

ELTHON が解決する問題 (エッシャー化問題) は、強い制約条件を持つため、解の発

見が容易でない難しさ、および、図形間の類似性を評価する万能な関数は発見されていない難しさを含む。これらの難しさは、タイルの設計における、接する図形同士のエッジ形状が一致する制約を満たす難しさ、および、目標図形との類似性を適切に評価する関数が発見されていない難しさに該当する。エッシャー化問題が含む難しさを解決するために、ELTHON は、下位最適化層に解析的最適化手法を用いて、エッジ形状の制約を満たす図形を解析的に導出する。しかし、解析的最適化手法には問題点があり、この問題点を解決するために、上位最適化層にメタヒューリスティクスを用いて、図形間の類似性をより適切に評価し、かつ、目標図形 W の形状の調整作業を自動化する。上記のように階層的に最適化を行うことで、エッシャー化問題の難しさを解くことが期待される。

本章の要点は以下の通りである。

- ELTHON は、階層構造を持つ最適化アプローチを採用し、種類の異なる最適化アルゴリズムにより分解した難しさを解決することでエッシャー化問題を解決する。具体的には、上位最適化層が与えられた問題（目標図形 W ）を下位最適化層で解決できる問題（粗い目標図形 \tilde{W} ）に修正する。
- 本章では、エッシャー化問題の定式化を変更し、探索空間サイズを縮小する。
- 本章では、メタヒューリスティクスの利点である目的関数の柔軟性を利用して、頂点数が大きく異なる2つの多角形間の、双方向のマッピングに基づく図形形状間の類似度を評価する関数を利用する。

3.2 階層型最適化方式の概要

3.2.1 基本的なアイデア

本章で提案する ELTHON は、エッシャー化問題の難しさを分割し、先行研究 [17–26] の問題点を解決する階層型最適化方式を採用している。2.2.2 項で述べたように、エッシャー化問題には3つの条件がある。これらの条件を解決するために ELTHON では、2層の異なる解法を用いている。上位最適化層に用いるメタヒューリスティクスは、さまざまな目的関数や制約条件を柔軟に利用することが可能であるため、エッシャー化問題の条件1と条件3を解決することを保証し、解析的最適化手法 [18, 19] は高速に条件2を満た

した解を導出することを保証する。ELTHON のメタヒューリスティクスは、問題を直接解決するものではなく、解析的最適化手法の入力を最適化する役割を持つ。なぜなら、メタヒューリスティクスにより問題を直接解決すると、巨大な設計変数空間内から良いタイルを探索する必要があるため、良い解を探索することが難しいことが分かっているためである [17, 25, 26].

上記の理由から ELTHON では、下位最適化層に粗い目標図形 $\tilde{\mathbf{W}}$ をタイルに変形する解析的最適化手法を採用し、上位最適化層にメタヒューリスティクスを採用する。上位最適化層は、下位最適化層への入力を最適化するために、下位最適化層の入力に当たる粗い目標図形 $\tilde{\mathbf{W}}$ の適切な集合を目標図形 \mathbf{W} から求めることと、適切なタイリングパターンの選択を行う。下位最適化層は、 $\tilde{\mathbf{W}}$ と選択されたタイリングパターンを入力として、条件 2 を満たす、かつ、 $\tilde{\mathbf{W}}$ に類似するタイル \mathbf{U} に変形する。次に上位最適化層は、下位最適化層の出力を評価する。上位最適化層で \mathbf{U} を評価することにより、下位最適化層では考慮できない目的関数や制約条件を用いることができる。特に本章では、頂点数が大きく異なる 2 つの多角形間の双方向のマッピングに基づく図形形状間の類似性距離関数を導入する。これにより、2 つの多角形を解析的最適化手法で用いる関数 F_L (式 2.4) よりも厳密に比較することが可能である。

ELTHON は、先行研究の欠点である解析的最適化手法の $\tilde{\mathbf{W}}$ の調整の試行錯誤が必要であることや、メタヒューリスティクスに基づく手法の解の品質の低さを解消することができる。

3.2.2 形状空間における階層型最適化方式の概念的解釈

本項では、有限の頂点を持つ全ての 2 次元多角形 (図 3.1 を参照) から構成される概念空間 \mathcal{S} における ELTHON の解釈について説明する*¹。入力された目標図形 \mathbf{W} (図 3.1 の星印) は、多数の頂点からなる滑らかな形状の表現を可能とする多角形である。 \mathcal{F}_{C_2} と \mathcal{F}_{C_3} はそれぞれ、条件 2 と条件 3 を満たす領域を示し、その 2 つの領域が重なる領域 $\mathcal{F}_{C_2} \cap \mathcal{F}_{C_3}$ が実行可能領域である。 \mathcal{S} 上の距離は、図形間の類似度合いを示し、距離が近いほど類似度が高く、距離が遠いほど類似度が低い。エッシャー化問題は実行可能領域

*¹ ここでは、議論を簡単にするために、タイリングパターンを省略しており、図 3.2 では最適化中に ELTHON が適切なタイリングパターンを選択している。

$(\mathcal{F}_{C_2} \cap \mathcal{F}_{C_3})$ の中で, \mathbf{W} に最も近い最適解 \mathbf{U}^* を求める問題である. 最も近い最適解 \mathbf{U}^* は, 図 3.1 のひし形マークで示されている.

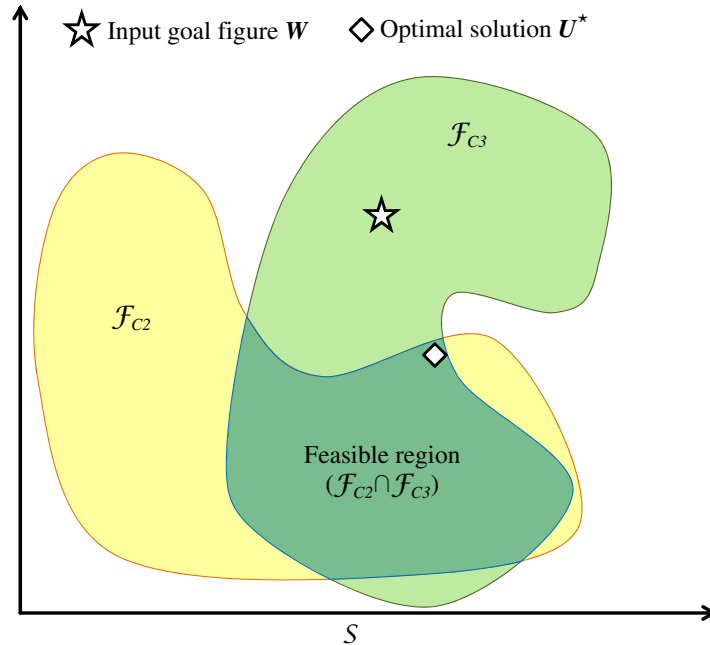


図 3.1. 入力された目標図形 \mathbf{W} と概念空間 S 上の最適解

図 3.2(a) に ELTHON の視覚的な解釈を示す. 上位最適化層は, 粗い目標図形 $\tilde{\mathbf{W}}_{i,g}$ ($i \in \{1, \dots, N_P\}$) を生成する. ここで, i と g はそれぞれ, 解候補の個体集団内の番号と, 現在の世代数を示し, N_P は個体数を示す. $\tilde{\mathbf{W}}_{i,g}$ は, \mathbf{W} の輪郭上から \tilde{N} 個の頂点を選択して生成されているため, \mathbf{W} と似た形状を持つ. この $\tilde{\mathbf{W}}_{i,g}$ (図 3.2(a) の黒丸のマーク) は, GA などのメタヒューリスティクスにおける遺伝子型である. ELTHON の下位最適化層では, 解析的最適化手法 [18, 19] を採用し, 条件 2 を満たすことがほとんどない $\tilde{\mathbf{W}}_{i,g}$ を条件 2 を満たす図形 $\mathbf{U}_{i,g}$ に変形している. 解析的最適化手法によって得られたタイル $\mathbf{U}_{i,g}$ は, $\tilde{\mathbf{W}}_{i,g}$ の表現型とみなされる (図 3.2(a)ii)). 解析的最適化手法は, S 上の任意の点を \mathcal{F}_{C_2} 上の点に変換する (図 3.2(a)ii)). 変換された点は, 必ず \mathcal{F}_{C_2} 上に存在するが, 必ずしも $\mathcal{F}_{C_2} \cap \mathcal{F}_{C_3}$ 上に存在するとは限らない. 次に上位最適化層は, 条件 1 と条件 3 (つまり, 図 3.2(a)iii) に示す $\mathbf{U}_{i,g}$ と \mathbf{W} との形状の類似距離, および, $\mathbf{U}_{i,g}$ に自己交差を含まないこと) から $\mathbf{U}_{i,g}$ を評価し, この結果に基づいて新たな解候補 $\tilde{\mathbf{W}}_{i,g+1}$ を生成する.

ELTHON は解候補の生成と評価を繰り返すことで、エッシャー化問題の 3 つの条件をすべて満たす解を探索する。

3.2.3 階層型最適化方式と解析的最適化手法の関係

本項では、解析的最適化手法 [18, 19] と ELTHON の S 上での挙動の違いを探ることで、解析的最適化手法と ELTHON の関係を議論する。3.2.2 節で述べたように、解析的最適化手法は S 上の任意の点を \mathcal{F}_{C_2} 上の点に変化するが、 $\mathcal{F}_{C_2} \cap \mathcal{F}_{C_3}$ 上の点に位置することは保証しない。また、変換前の点がわずかに変化するだけで、変換後の点の位置が大きく変化する可能性がある (2.3.1.1 項を参照)。このため、解析的最適化手法は目標図形 W の輪郭上で頂点座標の位置を試行錯誤して、粗い目標図形 \tilde{W} (図 3.2(b)i) の黒丸のマーク) を構築する必要がある。ELTHON の上位最適化層は、上記の解析的最適化手法の人手による試行錯誤に代わる役割を持つ (図 3.2(a)i)。

一方、解析的最適化手法の目的関数は図 3.2(b)ii) に示すように、 U と \tilde{W} の対応する点間の距離を計算する。しかし、この計算方法には 2 点改良する余地がある。1 点目は、粗い目標図形 \tilde{W} ではなく、目標図形 W とタイル U の類似度を計算する必要がある。2 点目は、 U 上の頂点は、必ずしも \tilde{W} 上の対応する頂点と最も近いとは限らないため、 U の頂点に加えて U の辺にたいしても、それぞれ最も近い点を探索する必要がある。

これらの問題を解決するために、3.3.2.4 (a) 項にて述べている双方向マッピングベース関数を導入している。

3.2.4 階層型最適化方式と DDGA の関係

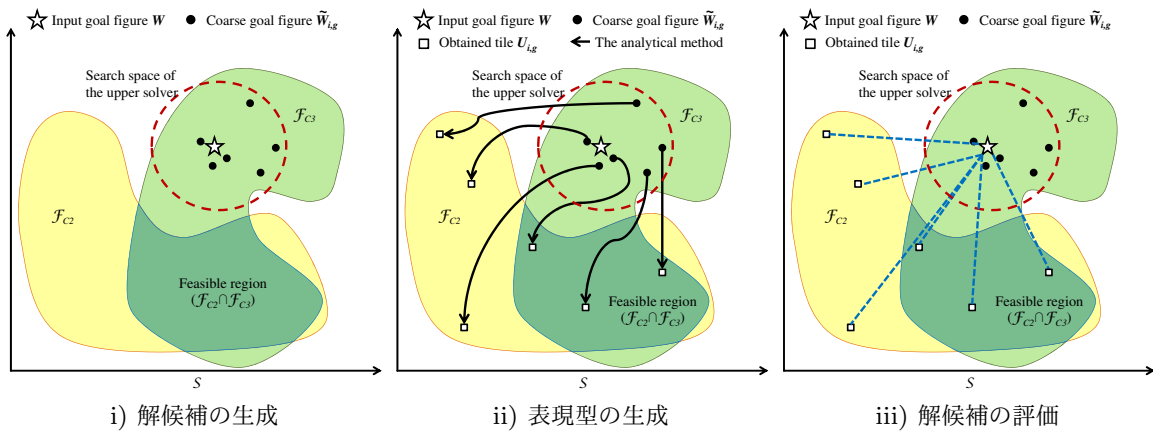
本項では、GA ベースの先行研究 (DDGA) [25, 26] と ELTHON との関係について述べる。両手法とも、メタヒューリスティクスと解析的最適化手法 [18, 19] を採用しているが、後述するようにアーキテクチャが異なる。

まず、両手法の違いは、両手法のメタヒューリスティクスが異なる問題を解決している点である。DDGA の GA は図 3.2(c)ii) に示すように、 $U_{i,g}$ の形状を直接最適化し、 $U_{i,g}$ から $U_{i,g+1}$ を生成する。一方、ELTHON のメタヒューリスティクスを用いている上位最適化層では、 $U_{i,g}$ の形状を直接最適化するのではなく、図 3.2(a)i) に示すように $\tilde{W}_{i,g}$

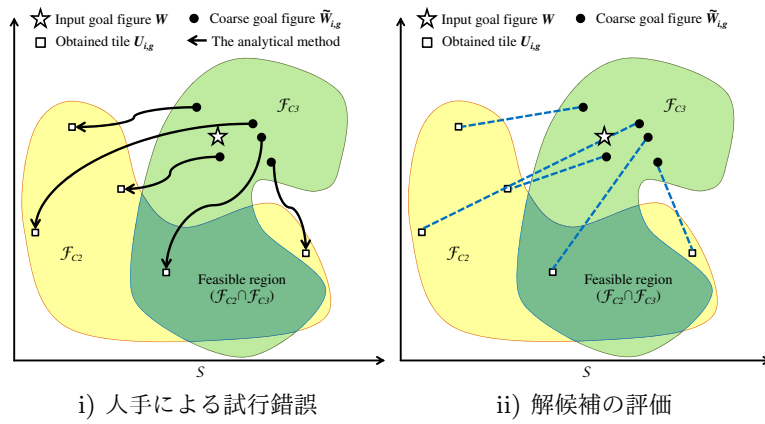
を最適化している。DDGA では、GA と解析的最適化手法が同じ問題（タイル U に変形する）を解くため、DDGA は単純なハイブリッド最適化アルゴリズムであるのに対して、ELTHON は階層的なアプローチを採用し、メタヒューリスティクスと解析的最適化手法が異なる問題と解く。

次に、両手法の違いは解析的最適化手法の適用方法が異なる点である。DDGA は、図 3.2(c)i) に示すように初期個体の生成にのみ解析的最適化手法を適用している。一方、ELTHON では、遺伝子型と表現型の対応付けを行うために解析的最適化手法を適用している。つまり、ELTHON は図 3.2(a)ii) に示すように、すべての解候補に対して解析的最適化手法を適用している。

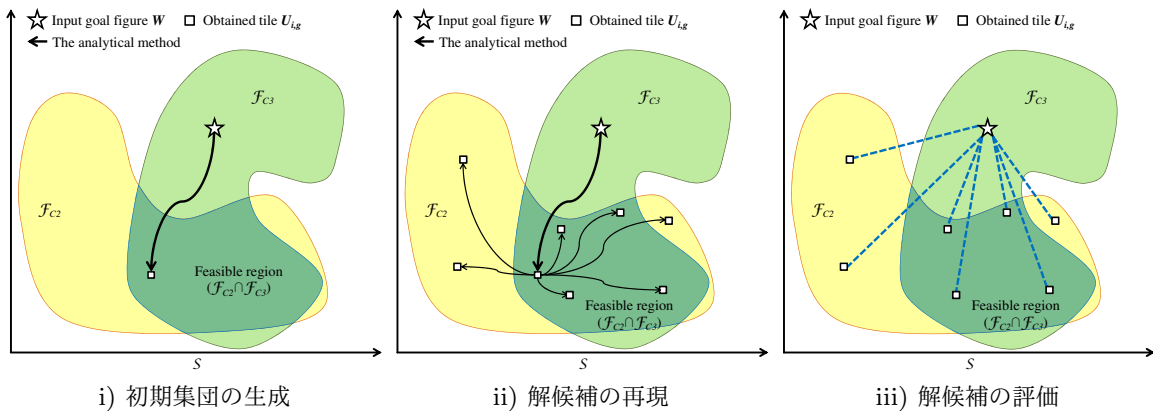
第 3 章 階層型最適化方式の提案



(a) ELTHON



(b) AOM [18, 19]



(c) DDGA [25, 26]

図 3.2. 各手法のタイル U の生成・評価手順に関する概念的解釈

3.3 階層型最適化方式の構成

3.3.1 上位最適化層と下位最適化層の関係

ELTHON は，上位最適化層の 2 値最適化に GA [11, 48] や分散推定アルゴリズム (EDA) [65, 66] のようなメタヒューリスティクスを採用することができる．同様に，ELTHON の下位最適化層に解析的最適化手法 [18, 19] とその拡張手法 [20–24] (2.3.1.2, 2.3.1.3 項を参照) を採用することができる．本章では，GA と解析的最適化手法を採用する．これは，アルゴリズムの挙動の解釈の容易性，および，生成される解の多様性を考慮しているため，単純なメタヒューリスティクスを使用する．

図 3.3 に ELTHON のアーキテクチャの概要を示す．GA に基づく上位最適化層は，粗い目標図形 \tilde{W}^* を構成する頂点の最適な集合と， \tilde{W}^* の適切なタイリングパターン p^* を求める問題である． $\mathbf{x}_{i,g}$ とで表される解候補（個体）は，選択された頂点の集合 $\tilde{W}_{i,g}$ とタイリングパターン $p_{i,g}$ から構成される．

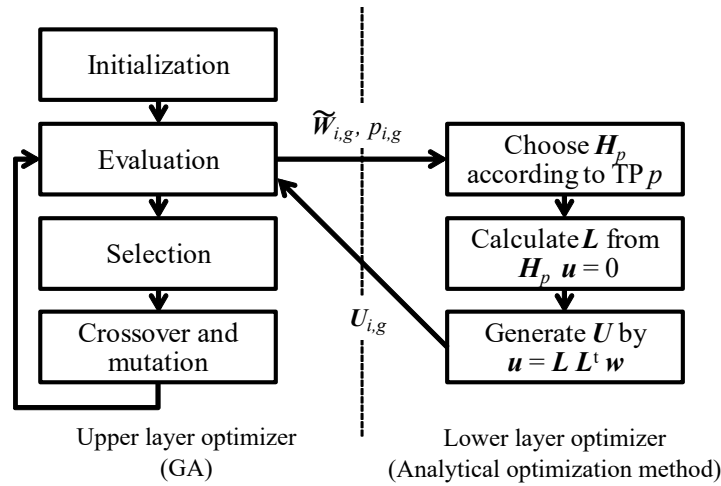


図 3.3. ELTHON のアーキテクチャの概要

アルゴリズムはまず，上位最適化層の解候補 $\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \dots, \mathbf{x}_{N_p,1}$ をランダムに初期化する．上位最適化層は，個体を下位最適化層に渡し，その個体の表現型を形成する．下位最適化層は個体を受け取り，個体を 1 つずつ最適化し，各 $\mathbf{x}_{i,g}$ の表現型としてタイル $U_{i,g}$ を生成する．この表現型は $\tilde{W}_{i,g}$ に類似しつつ条件 2 を満たす．

生成された個体の表現型 $U_{i,g}$ は、上位最適化層に戻され、条件 1 と条件 3 を満たしているかを確認するために、目的関数と制約条件により評価される。ここで、上位最適化層は $U_{i,g}$ と目標図形 W と比較するのに対して、下位最適化層は 3.2.3 項で説明したように、 $U_{i,g}$ と $\tilde{W}_{i,g}$ とで比較することに注意いただきたい。

適応度に基づいて、上位最適化層は、選択、交叉、突然変異などの遺伝子操作により新しい個体 $x_{i,g+1}$ を生成する。以上の処理を、停止条件に達するまで繰り返す。

3.3.2 上位最適化層

3.3.2.1 上位最適化層の必要性

上位最適化層は、粗い目標図形 \tilde{W} を構成する頂点の集合と適切なタイリングパターン p を探索する。探索された \tilde{W} と p は、両方とも下位最適化層に入力される。

2.3.1.1 項で述べたように、解析的最適化手法 [18, 19] では、入力形状を少し変えるだけでも出力される形状が大きく変化することがある。このため、 \tilde{W} を構成する適切な頂点の集合を見つけることは難しい。しかし、入力された目標図形 W の変化と解析的最適化手法により変換されたタイル U との間に全く関連がなければ、ランダムサーチでも良い \tilde{W} を探索することができ、メタヒューリスティクスは不要となる。

メタヒューリスティクスを問題に適用することの妥当性は、適応度距離相関 (Fitness Distance Correlation : FDC) [74] によって評価することができる。最良解までの距離が減少するにつれて適応度が増加する場合は、メタヒューリスティクスはランダムサーチよりも高速に解を見つけることができる。図 3.4 は、問題インスタンスの FDC の一例を示す図である。この図は、最適解からハミング距離が異なる場合の双方向マッピングベース関数 F_{BM} (3.3.2.4 (a) 項を参照) の適応度をプロットした図である。各ハミング距離において、既知の最良解から 1,000 個の解をランダムにサンプリングした結果である。平均適応度は、既知の最良解からの距離が長くなるにつれて低下する傾向があり、メタヒューリスティクスの方がランダムサーチよりも効果的であることが示唆された。

3.3.2.2 上位最適化層の個体表現

3.2.4 項で述べたように、上位最適化層は、与えられた目標図形 W のダウンサンプリング方法 (ダウンサンプリング後の図が粗い目標図形 \tilde{W} に当たる) と、適切なタイリン

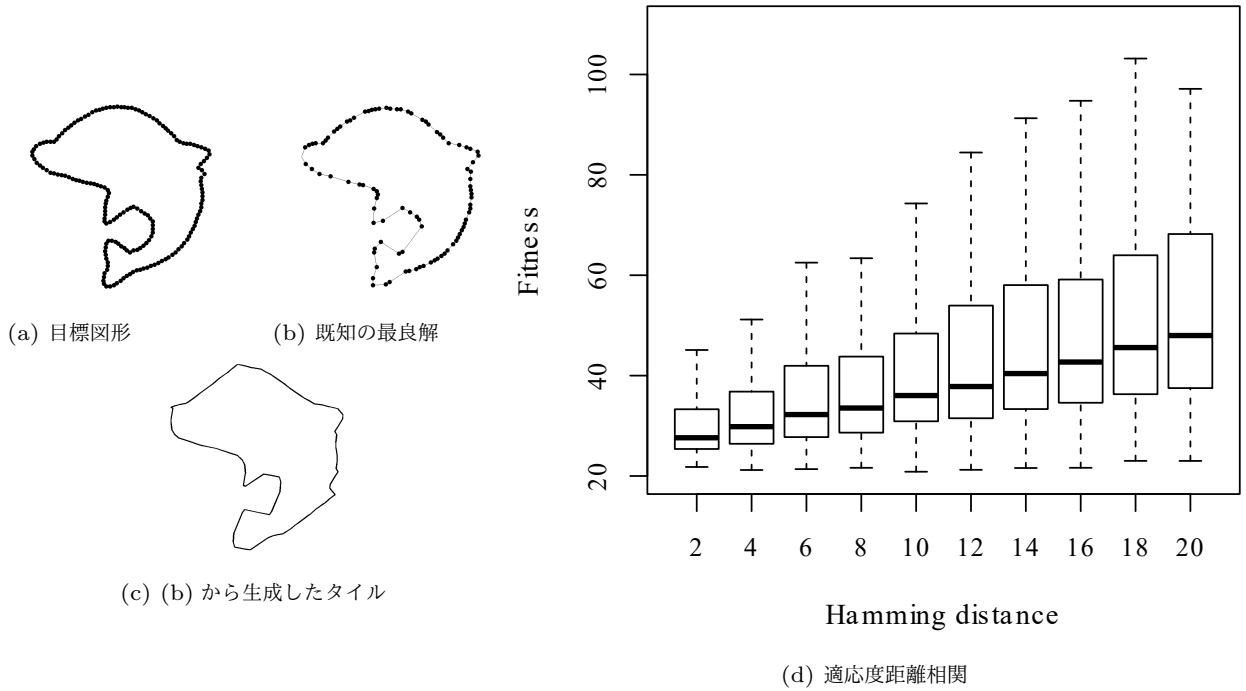


図 3.4. 上位最適化層が解く問題の適応度距離相関

グパターン p の組み合わせを探索するのに対して, DDGA の GA は直接タイル U を探索する. そのため, DDGA の GA が \tilde{N} 個の実数値*2 からなる膨大な探索空間を探索しなければならないのに対して, ELTHON は探索空間を N 個の 2 値*3 に縮小している.

図 3.5 に上位最適化層の遺伝子型と表現型の個体表現を示す. 遺伝子型は長さ $N + 5$ の 2 値の文字列で表され, N ビットはマスクパターン, 5 ビットはタイリングパターン p を表す. マスクパターンは, W の輪郭に沿って選択された頂点の集合を表し, このマスクパターンを基に $\tilde{W}_{i,g}$ を形成する. タイリングパターン $p_{i,g}$ の 5 ビットの 2 値文字列は, 29 個のタイリングパターンのうち 1 つのタイリングパターンを表す. 29 個のタイリングパターンは 93 個のタイリングパターンから冗長性を排除して選ばれた [51] (2.2.1.2 を参照). 29 個のタイリングパターンとバイナリ列との対応は, 表 3.1 に示す.

上位最適化層の表現型は, 下位最適化層を $\tilde{W}_{i,g}$ と $p_{i,g}$ に適用して得られたタイル $U_{i,g}$

*2 より正確には, 変数の数はタイリングパターンに依存している. 例えば, IH01 では辺の半分を任意に変更することができるため, $\tilde{N}/2$ 点の頂点座標が必要となる. IH16 では辺の 3 分の 1 を変更するため, $\tilde{N}/3$ 点の頂点座標が必要となる.

*3 タイリングパターンを表す変数を除く

である。 $U_{i,g}$ は常に辺の一致制約（条件 2）を満たすが、必ずしも自己交差の制約（条件 3）を満たすとは限らない。

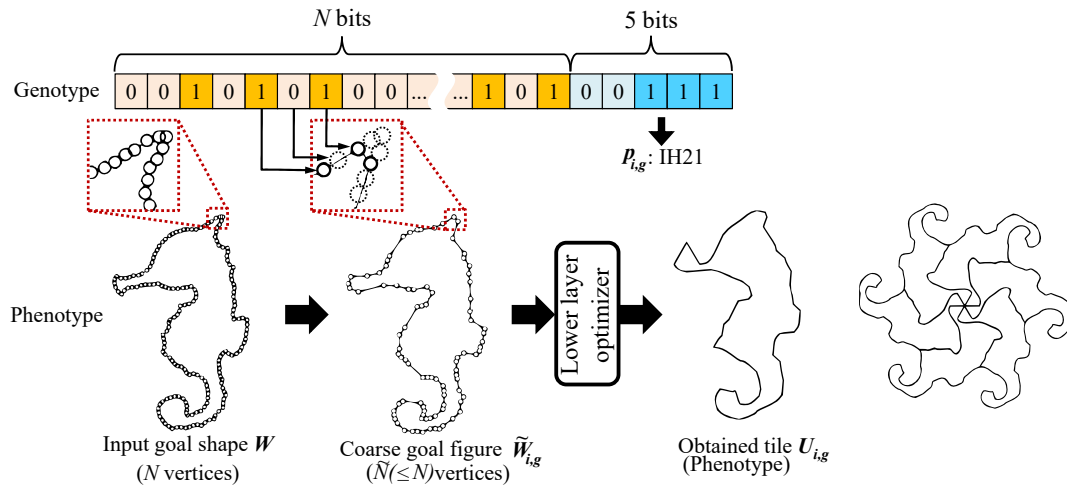


図 3.5. 個体表現

表 3.1. バイナリ列とタイリングパターン p の対応

バイナリ列	TP p	バイナリ列	TP p	バイナリ列	TP p
00000	IH01	01010	IH27	10100	IH51
00001	IH02	01011	IH28	10101	IH52
00010	IH03	01100	IH30	10110	IH53
00011	IH04	01101	IH31	10111	IH55
00100	IH05	01110	IH33	11000	IH39
00101	IH06	01111	IH41	11001	IH79
00110	IH07	10000	IH43	11010	IH84
00111	IH21	10001	IH44	11011	IH86
01000	IH23	10010	IH46	11100	IH88
01001	IH25	10011	IH47		

3.3.2.3 上位最適化層における遺伝的アルゴリズム

ELTHON の上位最適化層は、2 値最適化のための任意のメタヒューリスティクスを採用することができる。階層型最適化方式の有効性を検証するために、本章では EDA [65,66]

のような最新のメタヒューリスティクスではなく、MGG (Minimal Generation Gap) モデル [64] を持つ単純な GA を採用している。MGG は連続的な世代交代モデルであり、個体集団の一部が 1 世代で入れ替わることを意味し、その挙動は容易に理解できる (2.4.3.2 項を参照)。MGG を用いた GA は個体集団の多様性を維持することができるため、上位最適化層が最適化の際に適切なタイリングパターンを選択することに役立つ。このためには、良いタイリングパターンを持つ個体を徐々に増やし、特定のタイリングパターンへの早期収束を避ける必要がある。

図 3.6 に GA に基づく上位最適化層のアルゴリズムを示す。初期集団 $\mathbf{p}_1 = \{\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, \dots, \mathbf{x}_{N_p,1}\}$ は、ランダムに生成された個体 $\mathbf{x}_{i,1}$ から形成され、ほぼ等数の個体が各タイリングパターンを保有している。

初期化後、GA のメインループ (図 3.6 の 4 行目から 11 行目) は、個体の評価と生成を繰り返し行う。個体 $\mathbf{x}_{i,g}$ を評価する場合、下位最適化層は対応する表現型 $U_{i,g}$ を生成する (2.3.1.1 を参照)。そして、得られた $U_{i,g}$ の適応度を目的関数により計算する (3.3.2.4 (a) 項を参照)。ELTHON では、ランキング [75] やスケーリング [76, 77] などの手法により評価することも可能だが、目的関数値を直接適応度として用いる。

評価後、上位最適化層は、得られた適応度を基に個体を再現する。MGG では、個体集団の一部を新しい個体に置き換える。再現された個体の割合は、世代間ギャップ R_G によって決定される。次世代の個体数が $N_p \times R_G$ に達するまで、アルゴリズムは選択、交叉、突然変異を繰り返す。まず、個体集団 \mathbf{p}_g から 2 つの個体 $\mathbf{x}_{pr1,g}$, $\mathbf{x}_{pr2,g}$ をランダムに非復元抽出で選択する。親個体 $\mathbf{x}_{pr1,g}$ と $\mathbf{x}_{pr2,g}$ との一樣交叉により、2 つの子個体 $\mathbf{x}_{of1,g+1}$ と $\mathbf{x}_{of2,g+1}$ が生成される。交叉に続いて、 $\mathbf{x}_{of1,g+1}$ と $\mathbf{x}_{of2,g+1}$ のビット反転による突然変異が m_p の確率で行われる。ここで、 m_p は突然変異率と呼ばれる。 $\mathbf{x}_{pr1,g}$, $\mathbf{x}_{pr2,g}$, $\mathbf{x}_{of1,g+1}$, $\mathbf{x}_{of2,g+1}$ の中で最も優れた個体と、残りの 3 個体の中からルーレット選択により選択されたもう 1 つの個体が、次世代の個体集団 \mathbf{p}_{g+1} に格納される。個体の再現後、 \mathbf{p}_g 内の親個体として選択されなかった個体は \mathbf{p}_{g+1} に格納される。

上記の処理は、アルゴリズムが予め設定した停止条件に達すると終了する。

Algorithm 3 遺伝的アルゴリズム

- 1: $g \leftarrow 0$
 - 2: 個体 $\mathbf{x}_{1,g}, \dots, \mathbf{x}_{N_P,g}$ からなる初期個体集団 \mathbf{p}_g を生成する
 - 3: **while** 停止条件を満たさない間 **do**
 - 4: 個体集団 \mathbf{p}_g 内の全個体 $\mathbf{x}_{i,g}$ ($i \in \{1, \dots, N_P\}$) を評価する
 - 5: **while** $|\mathbf{p}_{g+1}| < N_P \times R_G$ **do**
 - 6: \mathbf{p}_g から 2 個体 $\mathbf{x}_{pr1,g}$ と $\mathbf{x}_{pr2,g}$ を親個体としてランダムに非復元抽出で選択する
 - 7: 親個体を一様交差し, 子個体 $\mathbf{x}_{of1,g+1}$ と $\mathbf{x}_{of2,g+1}$ を得る
 - 8: $\mathbf{x}_{pr1,g}, \mathbf{x}_{pr2,g}, \mathbf{x}_{of1,g+1}, \mathbf{x}_{of2,g+1}$ の中から適応度が最も良い個体と, ルーレット選択により選択された 1 個体を \mathbf{p}_{g+1} に追加する.
 - 9: **end while**
 - 10: \mathbf{p}_g の中で選択されていない残りの個体を \mathbf{p}_{g+1} に追加する
 - 11: $g \leftarrow g + 1$
 - 12: **end while**
-

図 3.6. 上位最適化層で用いる GA のアルゴリズム

3.3.2.4 目的関数

条件 2 は下位最適化層によって保証されているため, 上位最適化層の目的関数と制約条件は条件 1 と条件 3 を検証するだけで良い. このため目的関数は, $\mathbf{U}_{i,g}$ と \mathbf{W} との類似度と $\mathbf{U}_{i,g}$ の自己交差のペナルティにより決定します. 全体的な目的関数 $F(\cdot)$ は次の式で与えられる.

$$F(\mathbf{U}_{i,g}) = F_{\mathcal{T}}(\mathbf{U}_{i,g}, \mathbf{W}) + \rho(\mathbf{U}_{i,g}), \quad \mathcal{T} \in \{BM, TF, BM + TF\} \quad (3.1)$$

ここで, 関数 $F_{\mathcal{T}}(\mathbf{U}_{i,g}, \mathbf{W})$ は, $\mathbf{U}_{i,g}$ と \mathbf{W} との間の形状の類似距離を評価し, $\rho(\mathbf{U}_{i,g})$ は, $\mathbf{U}_{i,g}$ の自己交差の数に依存するペナルティである.

本章では, GA における目的関数と制約条件の柔軟性を利用して, 図形形状の類似距離を計算する 3 つの関数を用いた. 例えば, 解析的最適化手法 [18, 19] では, 図形の輪郭を捉えた最も近い頂点間のユークリッド距離を用いている. しかし, 先行研究 [17, 25, 26] にて用いられている回転関数の方が局所的な凹凸を評価するのに適している. 本章ではま

ず、解析的最適化手法に用いられた目的関数に似ているが、より厳密に最近傍点を求められ、得られたタイル U と与えられた目標図形 W を比較する距離ベース関数 $F_{BM}(\cdot)$ を提案する。次に先行研究 [17, 25, 26] にて用いられた回転関数 $F_{TF}(\cdot)$ を用いて評価する。最後に、上記の2つの関数の重み付き線形和 $F_{BM+TF}(\cdot)$ を用いて評価する。

■(a) 双方向マッピングベース関数 本章では、解析的最適化手法 [18, 19] で使用されている関数 F_L (式 2.3) と F_L を改良した双方向マッピングベース関数 F_{BM} を紹介する。 F_{BM} と F_L はどちらも、2つの多角形の最近傍点間のユークリッド距離の2乗和を計算するが、 F_{BM} は、対応する最近傍点自体を求めるのに対し、 F_L は、 $\tilde{W}_{i,g}$ と $U_{i,g}$ の対応する点間の距離を計算するのみである。解析的最適化手法では、 F_L が連続的な凸関数であることが要求されたため、対応する点間の距離のみ計算する。しかし、変換されたタイル $U_{i,g}$ に対応する $\tilde{W}_{i,g}$ の頂点は、 $U_{i,g}$ の輪郭上の別の点に近い可能性がある。また、与えられた目標図形 W と粗い目標図形 \tilde{W} は、全く異なる座標の場合があるため、 $U_{i,g}$ は $\tilde{W}_{i,g}$ ではなく、 W と比較しなければならない。このためには、2つの図形 W と $U_{i,g}$ との間の最近傍点を対にしなければならないが、多くの場合、 W と $U_{i,g}$ の頂点数は異なる。

$U_{i,g}$ 上の各点について、 W 上の最も近い頂点を探索する一方向マッピングでは、凸形状を正しく評価できない。例えば、図 3.7(a) の一方向マッピングでは、 $U_{i,g}$ の突出した形状を正しく評価することができない。 $U_{i,g}$ の $u_{12}, u_{13}, \dots, u_{15}$ に最も近い点は、それぞれ、 $\nu_W(u_{12}), \nu_W(u_{13}), \dots, \nu_W(u_{15})$ であり、ここで、 $\nu_W(u_j)$ は W の輪郭上で u_j に最も近い点を表す。 $U_{i,g}$ と W との対応点 (対になっている u_{12} と $\nu_W(u_{12}), u_{13}$ と $\nu_W(u_{13}), \dots$) 間の距離が小さいので、図形形状の類似性も高くなる。しかし、 u_{13} と u_{14} との間には、 $U_{i,g}$ 上に頂点が存在しないため、突起の先端付近の距離が計算されず、 U における凸形状の再構成が乏しい。なお、突起部分を延長しても関数値に変化はない。また、逆方向の一方向マッピング、つまり、 W 上の各点について $U_{i,g}$ 上の最も近い点を求める場合も同様の問題がある (図 3.7(b) を参照)。

そのため、最も近い点を探索し、双方向の $U_{i,g}$ と W との形状の類似距離を計算する必要がある。提案された双方向マッピングベース関数 F_{BM} は、次のように定義される。

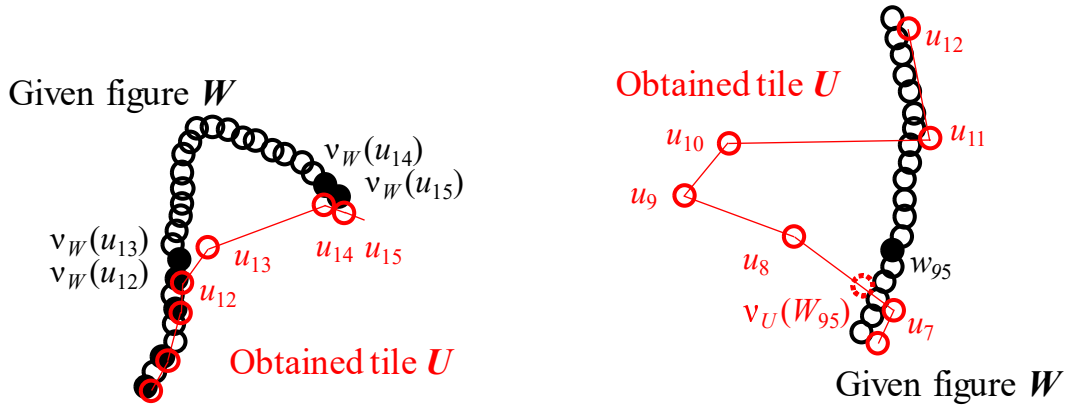
$$F_{BM}(U_{i,g}; W) = \min_{\theta} \{f_{ED}(r(U_{i,g}; \theta), W) + f_{ED}(W, r(U_{i,g}; \theta))\}, \quad (3.2)$$

ここで、 $r(\mathbf{U}_{i,g}; \theta)$ は角度 θ を介して $\mathbf{U}_{i,g}$ を回転させる回転演算子である。角度オフセット θ は、 $\mathbf{U}_{i,g}$ と \mathbf{W} との類似度が最も高くなるように $\mathbf{U}_{i,g}$ を回転させる。このとき、 $\mathbf{U}_{i,g}$ の重心位置は、 \mathbf{W} の重心位置と一致すると仮定する。以下の議論では、議論を簡単にするため、 $r(\mathbf{U}_{i,g}; \theta)$ を \mathbf{U} として表記する。

$f_{ED}(\cdot)$ は、2つの図形間の形状の類似距離を2つの図形形状の輪郭上の最も近い点間のユークリッド距離の和として計算する。

$$f_{ED}(\mathbf{A}, \mathbf{B}) = \frac{1}{|\mathbf{A}|} \sum_{i=1}^{|\mathbf{A}|} \left\{ \left(x_{a_i} - x_{\nu_{\mathbf{B}}(a_i)} \right)^2 + \left(y_{a_i} - y_{\nu_{\mathbf{B}}(a_i)} \right)^2 \right\}, \quad (3.3)$$

ここで、 \mathbf{A} と \mathbf{B} は2つの多角形を表す。多角形 \mathbf{A} は $|\mathbf{A}|$ 頂点 $\{a_1, a_2, \dots\}$ を含み、 (x_{a_i}, y_{a_i}) は a_i の座標を示し、 $(x_{\nu_{\mathbf{B}}(a_i)}, y_{\nu_{\mathbf{B}}(a_i)})$ は a_i に最も近い \mathbf{B} の輪郭上の点の座標である。また、 $(x_{\nu_{\mathbf{B}}(a_i)}, y_{\nu_{\mathbf{B}}(a_i)})$ は必ずしも \mathbf{B} の頂点に対応するとは限らず、 \mathbf{B} の辺上に存在する場合もあることに注意されたい。



(a) $f_{ED}(\mathbf{U}_{i,g}, r(\mathbf{W}; \theta))$ では評価できない突出した形状の欠損 (b) $f_{ED}(r(\mathbf{W}; \theta), \mathbf{U}_{i,g})$ では評価できない不要な凹みの出現

図 3.7. 一方向マッピングでは評価できない場合の例

■(b) 回転関数 回転関数は、辺の長さに対する角度を表した多角形の1次元表現である [78, 79]。2つの多角形の回転関数の差は、次のような類似度の目的関数 F_{TF} を提供する。

$$F_{TF}(\mathbf{U}_{i,g}; \mathbf{W}) = \min_{\theta} f_{TF}(r(\mathbf{U}_{i,g}; \theta), \mathbf{W}), \quad (3.4)$$

$$f_{TF}(\mathbf{A}, \mathbf{B}) = \int_0^1 |\varphi_{\mathbf{A}}(s) - \varphi_{\mathbf{B}}(s)| ds, \quad (3.5)$$

ここで、 \mathbf{A} と \mathbf{B} は多角形であり、 s は正規化された円弧の長さであり、 $\varphi_{\mathbf{A}}(s)$ は s に対する反時計回りの接線の角度を示す回転関数である。 \mathbf{A} と \mathbf{B} の間の回転関数の類似度は、 $\varphi_{\mathbf{A}}(s)$ と $\varphi_{\mathbf{B}}(s)$ の差分の積分として定義され（図 3.8 の灰色部分の合計）、差分の積分が 0 に近いほど \mathbf{A} と \mathbf{B} の間の類似性が高いことを示す。 $f_{TF}(r(\mathbf{U}_{i,g}; \theta), \mathbf{W})$ は回転に敏感であるため、すべての回転角度 θ に対して計算され、最小値が F_{TF} に割り当てられる。

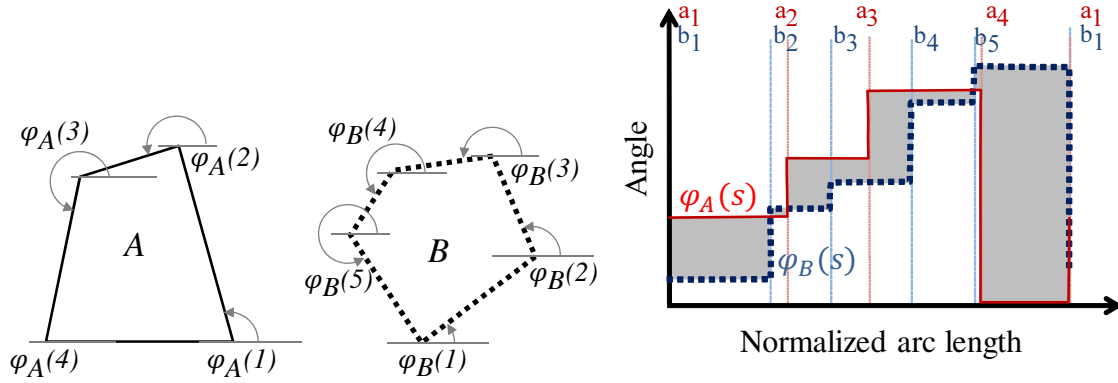


図 3.8. 回転関数 F_{TF} の計算例.

■(c) 重み付き線形和 3つ目の関数である $F_{BM+TF}(\cdot)$ は、双方向マッピングベース関数と回転関数の重み付き線形和を取った関数であり、これまでの両関数の利点を活かした関数である。具体的には、2つの多角形の輪郭を F_{BM} により簡単に評価し、多角形の局所的な形状を F_{TF} で評価する。

$$F_{BM+TF}(\mathbf{U}_{i,g}; \mathbf{W}) = F_{BM}(\mathbf{U}_{i,g}; \mathbf{W}) + \beta F_{TF}(\mathbf{U}_{i,g}; \mathbf{W}), \quad (3.6)$$

ここで、 β は、各目標図形 \mathbf{W} に対して自動的に決定される重みパラメータである。重み付け後、初期個体集団 \mathbf{p}_1 における、 F_{BM} と βF_{TF} の平均値は同一である。

$$\beta = \frac{\sum_{i=1}^{N_p} F_{BM}(\mathbf{U}_{i,1}; \mathbf{W})}{\sum_{i=1}^{N_p} F_{TF}(\mathbf{U}_{i,1}; \mathbf{W})} \quad (3.7)$$

3.4 評価実験

3.4.1 実験準備

本章では、ELTHON の有効性を検証するために3つの実験を行った。最初の実験では、先行研究の DDGA [25, 26] との比較を行った。この実験により、ELTHON の階層構造の有効性、つまり、探索空間の縮小効果を明らかにする（実験1）。2つ目の実験では、ELTHON, DDGA, 解析的最適化手法 [18, 19] で生成したタイルの品質の比較を行う（実験2）。この実験では、生成されたタイルをいくつかの基準（制約違反数と図形形状の類似度の指標：実験2-1）と、主観的な評価（実験2-2）により評価した。また、実験2では、ELTHON の挙動を明らかにした（3.4.3.3を参照）。3つ目の実験では、ELTHON が漫画のキャラクタなど複雑な形状の図形から、どれだけ効果的にタイルを生成できるか評価する（実験3）。

図3.9は、評価実験で用いる32枚の目標図形 \mathbf{W} を示したものである。各目標図形 \mathbf{W} は、用いる32図形の形状を具体的に表現できる180点 ($N = 180$) で構成されている。

3.4.2 実験1：階層型最適化方式の構成の有効性の評価

実験1では、DDGA [25, 26] との比較を行った。

本実験では、ELTHON の構成を次のようにした。上位最適化層には、一様交叉、ビット反転突然変異、世代交代モデルには MGG [64] を採用した（2.4.3.2項を参照）。停止条件を指定する関数評価数（FEs）の上限は、50,000FEsとする。個体集団サイズ N_P 、交叉率、個体あたりの突然変異率 m_p 、世代交代ギャップはそれぞれ、100, 0.96, 0.03, 0.5とする双方向マッピングベース関数と回転関数により得られる評価値はオーダーが異なるため、ペナルティ関数 ρ は F_{BM} では100, F_{TF} では1.0を自己交差の発生回数に乗じた値を加算する。階層構造方式の有効性を明確にするために、ELTHON, DDGA ともに目的関数は F_{TF} とした。また、DDGA [26] の著者は2つのタイリングパターン（IH07, IH28）しか実装していないため、実験1のELTHONは同じ2つのタイリングパターンに限定した。

DDGAのパラメータは参考文献 [26] の値に設定した。個体集団サイズ、交叉率、個体

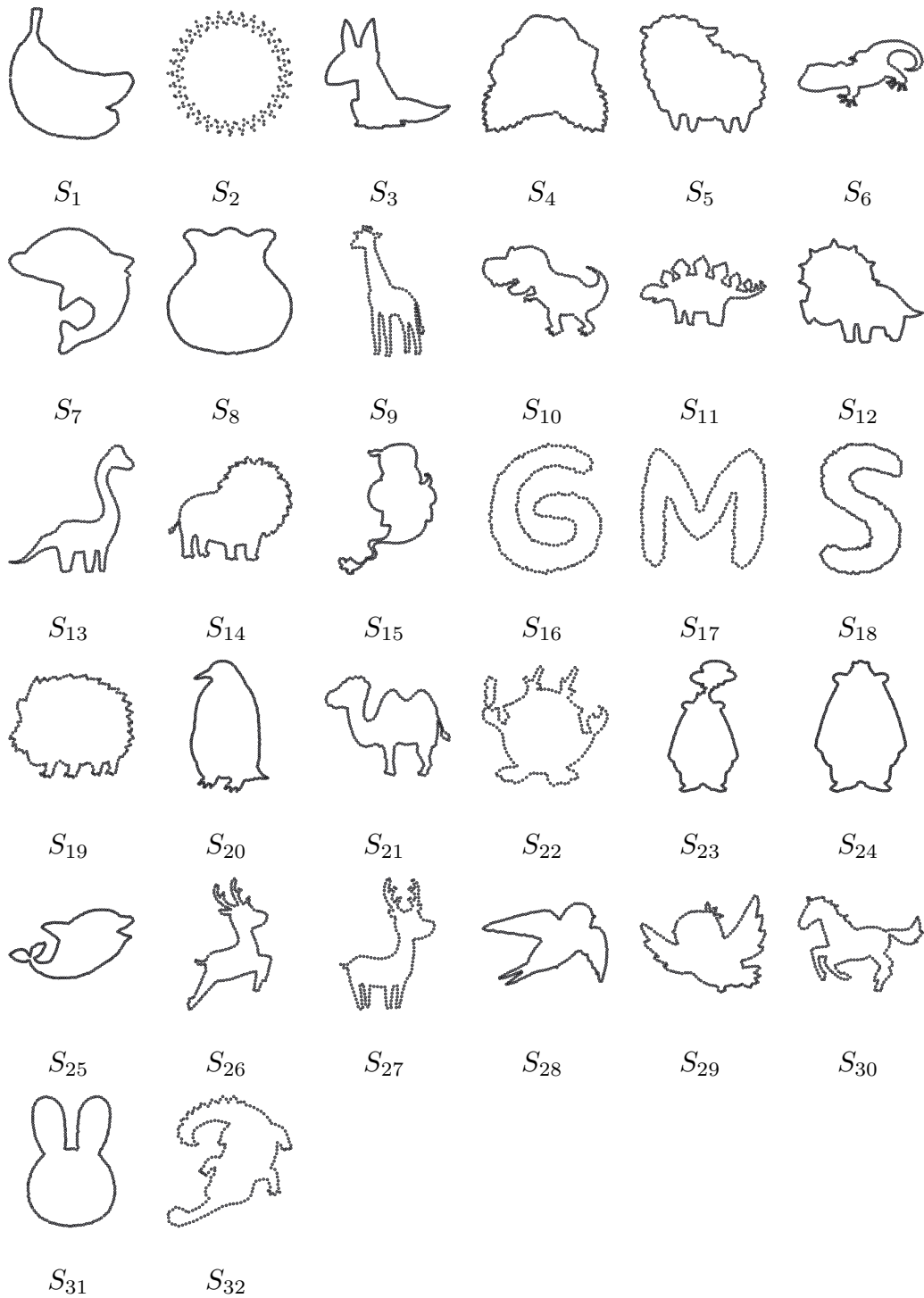


図 3.9. 目標図形 W

あたりの突然変異率, 次世代に保持されるエリート数をそれぞれ, 100, 0.98, 0.05, 2 とした. 停止条件は ELTHON との整合性を考慮して, 50,000FEs とした. DDGA における対話処理は省略した. 各手法について 5 試行実験を行った.

図 3.10 は, 実験の全体的な結果, つまり, 個体集団内で最も優れた個体の適応度の推移を示している. この結果は, 32 個の目標図形を 5 回実行して得られた適応度を平均化した結果である. 図 3.11 は実験 1 の各結果を示す. 図 3.11 は, 目標図形 S_7 , S_{12} , S_{29} の 5 試行の結果を平均化した最良解の適応度の遷移を示す. また, 表示されている図形は, 5 試行中最も適応度が良い最良解を見つけた試行において, 表示されているタイミン グに生成された最良解である (全図形の結果は付録 A に示す). テストした 32 図形のう ち, 以下のような傾向が見られた.

Observation 1 (2 手法による F_{TF} 値の比較) ELTHON は 32 図形中 21 図形 (図 3.11 (a) S_7 , (c) S_{29} など) において, DDGA よりも優れた F_{TF} 値を示したのに対して, DDGA は図 3.11 (b) S_{12} のみ ELTHON を上回った*4.

Observation 2 (収束速度) 同一品質の解を見つけるためには, DDGA と比較して ELTHON が約 33 倍の速さで発見できることがわかる.

(Observation 2) の高速化の程度は, 2 手法により得られた最良解が同じ F_{TF} 値に達した際の FEs を比較して算出した. 図 3.10 より, 最適化終了時 (50,000 FEs) の DDGA による最良解の適応度の平均値は 0.97 である. ELTHON は約 1,500 FEs の最適化により, 同品質の解を発見したため, $50,000/1,500 \simeq 33$ 倍の高速化を実現した.

ELTHON は, \tilde{N} 次元の連続 (実数値) 探索空間 (実験では 60 次元) から N 次元のバイナリ探索空間 (実験では 180 次元) に探索空間を縮小した*5. このため, ELTHON は DDGA よりも短い処理時間で優れた解を発見することができた. 探索空間の大きさを大まかに比較するため, 60 次元の実数値を 100 個の離散値と仮定する. この場合, ELTHON と DDGA の探索空間サイズはそれぞれ, $2^{180} \simeq 10^{54}$ と $100^{60} = 10^{120}$ として計算される.

ELTHON の計算コストの低さは, 全世代の個体ごとに反復処理を行った解析的最適化

*4 有意水準 5% の Welch 検定によるもの

*5 180 次元はタイリングパターンを選択するための変数を覗いた次元数である.

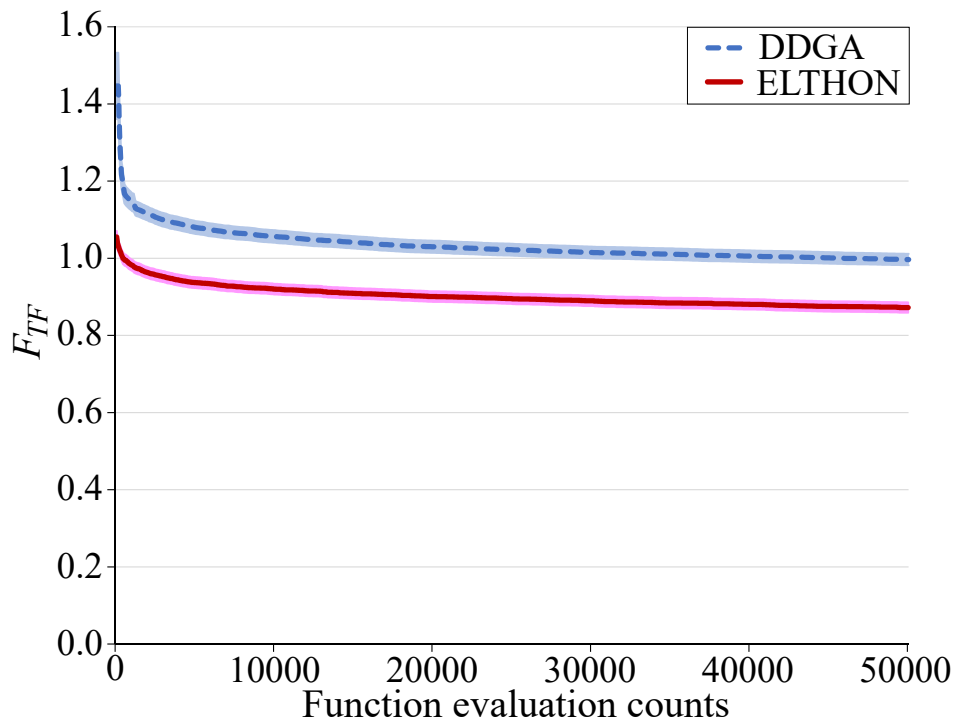


図 3.10. 実験 1 の結果 : 32 個のすべての図形における, 最良解の適応度の遷移の比較

手法にも起因しているを考える. これに対して, DDGA では, 初期個体集団の生成時に解析的最適化手法を適用するのみである.

3.4.3 実験 2 : 生成されたタイル形状の従来手法との比較

実験 2 では, ELTHON のタイルの生成性能を解析的最適化手法 [18, 19], および, DDGA [25, 26] と比較した. 本実験では, 生成されたタイルの品質を評価するために, ELTHON の上位最適化層の目的関数に F_{BM} , F_{TF} , F_{BM+TF} の 3 つの関数で試行し, それぞれの組み合わせを個別にテストした. また, 解析的最適化手法と DDGA の目的関数はそれぞれ F_L (式 (2.4)) と F_{TF} (式 (3.4)) を用いた. これらの 5 つの手法の性能比較を図 3.9 に示す 32 図形について行った.

DDGA と ELTHON については 5 試行行い, 解析的最適化手法は粗い目標図形 \tilde{W} とする入力を 174 回実行し, \tilde{W} の形状を微妙に変更するユーザの試行錯誤を再現した. \tilde{W} の形状の変更を再現するために, 与えられた目標図形 W の輪郭を構成する点群から頂点をサンプリングした. サンプリングは 2 種類の方法で行った.

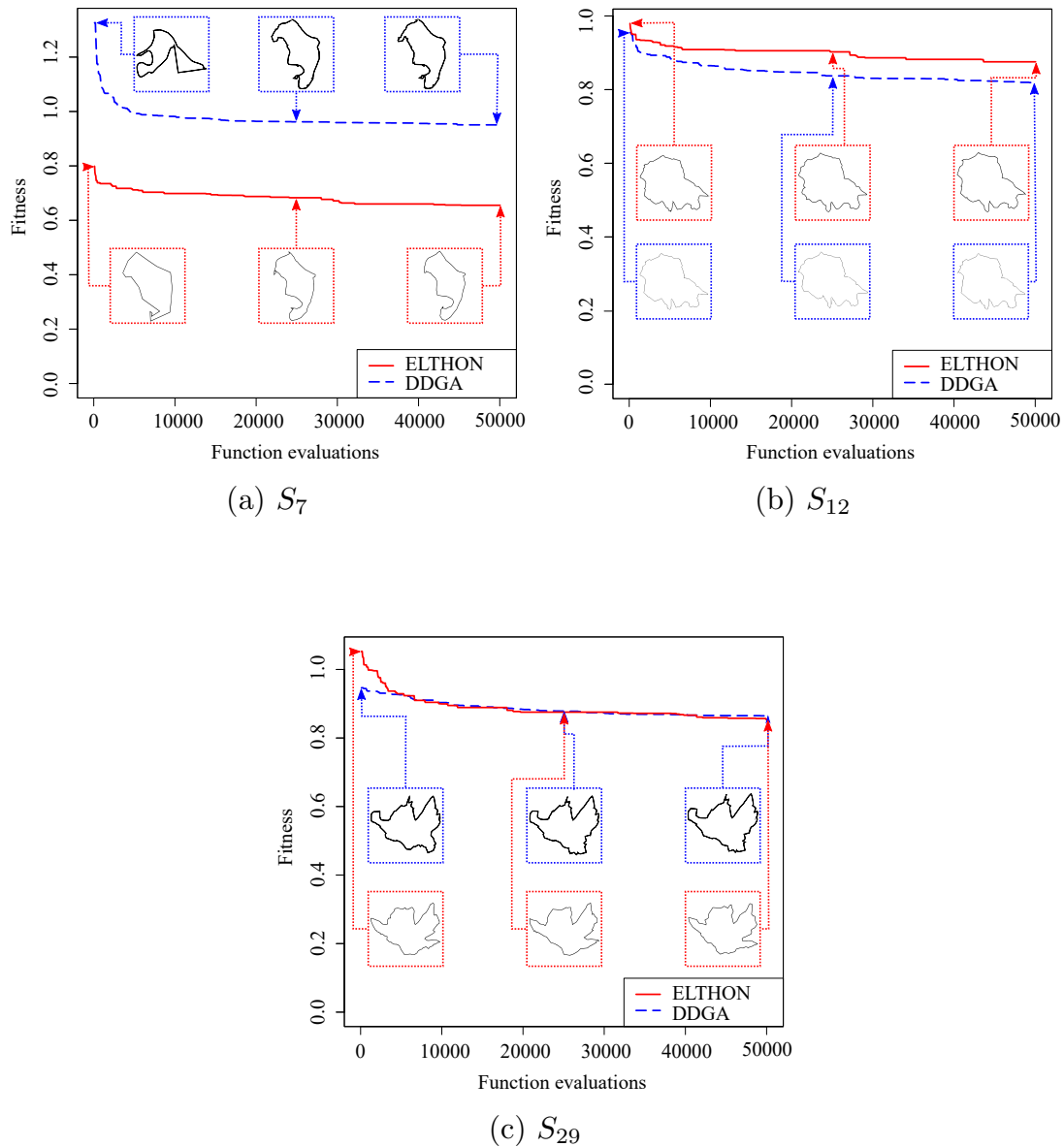


図 3.11. 実験 1 の結果：最良解の適応度の遷移の比較

- 等間隔サンプリング (Σ^{EI}): このサンプリング手法は、一定の点数ごとに点を選択する。選択された点は、ほぼ等間隔で区切られている。
- 三角形分割によるサンプリング (Σ^{TM}): 入力された図形を三角形の集合に分割し、三角形の全角度の中で最小角度を最大にする Delaunay 三角法 [80] を適用して、三角形の角に対応する点を求めるサンプリング手法である。

選択された点の総数は、60, 120, 180 点のように変化させた。この3種類の点数と冗長性を排除した29種類のタイリングパターンと上述した2種類のサンプリング方法を組み合わせ、1図形に対して174回の解析的最適化手法を適用してテストを行い、各ケースで F_L が最も低い出力を最良解とする。

ELTHON と DDGA のパラメータは実験1と同様であるが、ELTHON はタイリングパターンの全29種類を用いて評価し、DDGA には ([25,26] の実装を踏襲して)、シルエットの類似度を補助的に用いる F_{TF} ベースの関数を導入する*6。

3.4.3.1 実験 2-1 : 定量評価

■(a) 制約違反数 実験2に用いたすべての手法は、条件2を満たすことを保証するが、解析的最適化手法は条件3を考慮していない。DDGA も ELTHON も自己交差の完全な排除を保証していないが、いずれの手法も自己交差制約に違反する解にはペナルティを与える。そこでそれぞれの手法において、自己交差制約に違反した解の数を調べた。

自己交差の制約に違反する最良解の数を表3.2に示す。表3.2より、以下のことが観測された。

Observation 3 (制約違反の比較) 解析的最適化手法では58.9%、DDGA では38.8%の最良解においてタイルに自己交差を含んでいるのに対して、ELTHON は全32図形で制約違反の解消に成功した。

DDGA がペナルティ機構を有するにも関わらず自己交差を除去できなかった理由は、DDGA の設計変数間に強い依存性があることに起因している。そのため、局所的な最適条件ができ、辺形状を改善しながら自己交差の制約違反を除去することが難しい。つまり、ELTHON は DDGA のように辺形状を直接修正する手法から、解析的最適化手法のための適切な粗い目標図形 \tilde{W} の頂点を目標図形 W からサンプリングする手法へと、エッシャー化問題を再構築することに成功した。

■(b) 目標図形との類似性 図3.12から図3.17は5つの方法によって生成された S_1 , S_7 , S_{16} , S_{20} , S_{26} , S_{28} の代表的な結果を示す(全図形の結果は付録Bに示す)。結果の

*6 実験2のDDGAの目的関数は、 F_{TF} とシルエット画像の類似距離の重み付き線形和であり、2つの図形を 32×32 画素に縮小したときの画素の一致率を定義する。

表 3.2. 32 図形の目標図形に関する自己交差の制約に違反する最良解の数

目標図形	AM	DDGA	ELTHON		
			F_{BM}	F_{TF}	F_{BM+TF}
S_1	5/6	0/5	0/5	0/5	0/5
S_2	3/6	3/5	0/5	0/5	0/5
S_3	3/6	2/5	0/5	0/5	0/5
S_4	0/6	3/5	0/5	0/5	0/5
S_5	1/6	0/5	0/5	0/5	0/5
S_6	6/6	4/5	0/5	0/5	0/5
S_7	2/6	1/5	0/5	0/5	0/5
S_8	0/6	0/5	0/5	0/5	0/5
S_9	5/6	5/5	0/5	0/5	0/5
S_{10}	4/6	4/5	0/5	0/5	0/5
S_{11}	5/6	4/5	0/5	0/5	0/5
S_{12}	0/6	0/5	0/5	0/5	0/5
S_{13}	6/6	0/5	0/5	0/5	0/5
S_{14}	4/6	3/5	0/5	0/5	0/5
S_{15}	6/6	2/5	0/5	0/5	0/5
S_{16}	6/6	2/5	0/5	0/5	0/5
S_{17}	6/6	1/5	0/5	0/5	0/5
S_{18}	1/6	3/5	0/5	0/5	0/5
S_{19}	2/6	1/5	0/5	0/5	0/5
S_{20}	3/6	2/5	0/5	0/5	0/5
S_{21}	5/6	4/5	0/5	0/5	0/5
S_{22}	3/6	0/5	0/5	0/5	0/5
S_{23}	3/6	2/5	0/5	0/5	0/5
S_{24}	2/6	0/5	0/5	0/5	0/5
S_{25}	5/6	0/5	0/5	0/5	0/5
S_{26}	4/6	3/5	0/5	0/5	0/5
S_{27}	6/6	2/5	0/5	0/5	0/5
S_{28}	6/6	2/5	0/5	0/5	0/5
S_{29}	2/6	1/5	0/5	0/5	0/5
S_{30}	3/6	5/5	0/5	0/5	0/5
S_{31}	2/6	1/5	0/5	0/5	0/5
S_{32}	4/6	2/5	0/5	0/5	0/5
合計	58.9%	38.8%	00.0%	00.0%	00.0%

図は、目標図形、得られたタイル、 F_{BM} , F_{TF} , F_{BM+TF} の値、選択されたタイリングパターン、最良解の頂点数を示している。最良解は各手法の最小の目的関数値であり、平均目的関数値は DDGA と ELTHON の 5 試行によって得られた最良解と解析的最適化手法の 6 回の最良解（3 種類の頂点数と 2 種類のサンプリング方法の組み合わせで得られた）により計算された。最適化中に使用しなかった形状の類似距離（DDGA は F_{BM} 、解析的最適化手法は F_{TF} など）の値は、最適化後に得られた最良解をもとに計算した。平均値の欄では、タイリングパターンは 5 試行中最も頻繁に選択されたタイリングパターンを示し、解析的最適化手法の場合は最も頻繁に選択されたサンプリング方法も示す。

従来手法と ELTHON の類似度を比較すると、以下のようなことが分かった。

Observation 4（目的関数値の比較） F_{BM} を目的関数として用いた ELTHON は、32 図形中 19 図形（ S_7 , S_{16} など）で解析的最適化手法よりも良い F_{BM} 値を持つ解が得られたが、解析的最適化手法は ELTHON よりも良い解は得られなかった。また、 F_{TF} を目的関数として用いた ELTHON は、32 図形中 22 図形（ S_{26} , S_{28} など）で DDGA よりも良い F_{TF} 値を持つ解が得られたが、DDGA は ELTHON よりも良い解は得られなかった*7。

F_{BM} を用いた ELTHON は、3.3.2.4 項で述べたように、 F_{BM} 値を直接適応度として使用するため、解析的最適化手法よりも F_{BM} 値が良い解が得られる。しかし、目的関数の数値比較は、エッシャー化問題において必ずしも適切なものではない。図 3.12 から図 3.17 の目的関数値と得られたタイル形状について、さらに観察した結果をまとめると以下のようなになる。

Observation 5（人間の主観を関数で表現することの難しさ） 実験された関数は、人間の主観と同じように図形形状の類似度を評価することはできない。

例えば、 S_1 を解析した場合、 F_{BM} と F_{BM+TF} を目的関数とした ELTHON により生成された最良解の F_{BM} の値はそれぞれ、12.1 と 19.4 であるが、タイルは F_{BM} の場合より F_{BM+TF} の場合の方が目標図形に類似している。次に、 S_{20} を解析した場合、 F_{BM} を用いた解析的最適化手法と ELTHON で生成した最良解の F_{BM} 値はそれぞれ、29.9 と 16.0

*7 有意水準 < 0.05 の Welch の t-検定により決定される

第3章 階層型最適化方式の提案

であったが、両手法により得られたタイル形状は人間の目から見てかなり類似している。

従って、Observation 5 より、 F_{BM} は多くの場合、2 図形形状間の全体的な類似度を適切に評価するが、実験に用いた3つの目的関数は、先行研究と ELTHON の公平な性能評価比較を提供しない。



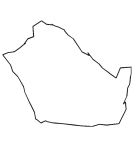
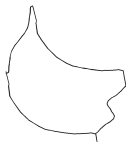




Goal figure W	AOM [18,19]		DDGA [25,26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L		F_{TF}				F_{TF}			
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	45.7	198	94.2	215	12.1	16.8	24	83.5	19.4	21.6
F_{TF}	0.93	3.00	0.67	0.85	0.90	0.94	0.57	0.58	0.60	0.61
F_{BM+TF}	222	766	249	318	184	194	134	194	116	138
p	IH04	IH41	IH07	IH07	IH47	IH23	IH23	IH23	IH23	IH23
Vertices	120	120	60	60	84	89.8	100	97	100	90.6
downsampling way	Σ^{TM}									

図 3.12. 実験 2-1 の結果 (S_1)




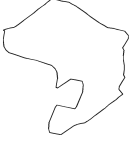

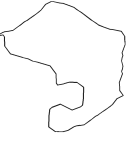


Goal figure W	AOM [18,19]		DDGA [25,26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L		F_{TF}				F_{TF}			
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	103	147	146	199	24.3	25.7	71.9	82.9	26	35.3
F_{TF}	0.50	0.678	0.74	0.85	0.88	0.91	0.56	0.58	0.53	0.61
F_{BM+TF}	201	233	239	306	135	139	142	155	99	111
p	IH23	IH23	IH07	IH07	IH04	IH04	IH04	IH23	IH04	IH04
Vertices	180	120	60	60	96	98.4	90	95.0	90	94.8
downsampling way	Σ^{TM}									

図 3.13. 実験 2-1 の結果 (S_7)

3.4.3.2 実験 2-2 : 主観評価

先行研究と ELTHON によるタイルの品質を比較するために、20名の被験者を募り、主観評価を行った (実験 2-2)。

第 3 章 階層型最適化方式の提案

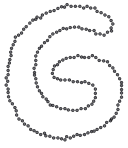



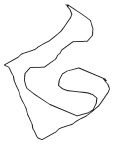



Goal figure W	AOM [18, 19]		DDGA [25, 26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L	F_{TF}	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{TF}	F_{TF}	F_{BM+TF}	F_{BM+TF}
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	570	639	235	1010	129	170	295	325	120	135
F_{TF}	5.34	5.79	0.97	1.18	0.94	1.10	0.71	0.73	1.00	1.06
F_{BM+TF}	1090	1198	329	1120	234	277	365	395	220	238
p	IH21	IH21	IH07	IH07	IH23	IH23	IH23	IH23	IH23	IH23
Vertices	120	120	60	60	85	89	95	92	100	91
downsampling way	Σ^{TM}									

図 3.14. 実験 2-1 の結果 (S_{16})






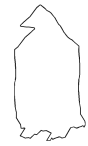


Goal figure W	AOM [18, 19]		DDGA [25, 26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L	F_{TF}	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{TF}	F_{TF}	F_{BM+TF}	F_{BM+TF}
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	29.9	68.4	105	170	16.0	16.2	32.0	43.7	18.4	20.6
F_{TF}	0.98	1.00	0.93	1.36	0.75	0.87	0.74	0.77	0.75	0.76
F_{BM+TF}	129	169	201	306	91.0	103	109	121	87.5	96.9
p	IH04	IH04	IH07	IH07	IH04	IH04	IH04	IH04	IH04	IH04
Vertices	180	120	60	60	84	101	90	92.4	102	99.6
downsampling way	Σ^{TM}									

図 3.15. 実験 2-1 の結果 (S_{20})

被験者には、先行研究と ELTHON により生成された 5 つの出力を、対応する目標図形と比較して、類似度により順位付けを行ってもらった。また、同率順位も許容している。実験した 5 つの方法間の統計的有意差は、Friedman 検定により評価する。Friedman 検定が有意であった場合、Wilcoxon の符号付き順位検定により評価した。

表 3.3 は、5 つの方法により生成されたタイルと、被験者により割れ当てられた順位の平均と標準偏差を示す。最良解と 2 番目に良い結果に有意差がある場合、最良の結果が太字で強調表示される。最良解と 2 番目に良い結果に有意差は無いが、2 番目に良い結果と 3 番目に良い結果に有意差がある場合、上位 2 つの良い結果の順位が強調表示される。

第 3 章 階層型最適化方式の提案









Goal figure W	AOM [18, 19]		DDGA [25, 26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L		F_{TF}				F_{TF}			
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	303	351	161	288	66.4	73.0	205	226	77.4	81.8
F_{TF}	2.87	3.25	0.95	1.02	1.13	1.18	0.86	0.88	0.90	0.98
F_{BM+TF}	644	738	287	410	201	214	312	330	186	199
p	IH04	IH04	IH28	IH28	IH04	IH04	IH07	IH07	IH04	IH04
Vertices	60	120	60	60	102	99.6	90	91.6	96	96
downsampling way	Σ^{EI}									

図 3.16. 実験 2-1 の結果 (S_{26})









Goal figure W	AOM [18, 19]		DDGA [25, 26]		F_{BM}		ELTHON		F_{BM+TF}	
	F_L		F_{TF}				F_{TF}			
										
	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.	Best	Ave.
F_{BM}	310	498	122	195	53.4	57.6	93.2	125	61.1	71.0
F_{TF}	5.09	6.53	0.878	1.02	0.70	0.76	0.54	0.57	0.56	0.61
F_{BM+TF}	991	1372	256	332	150	160	165	202	140	153
p	IH47	IH47	IH07	IH07	IH04	IH04	IH04	IH04	IH04	IH04
Vertices	180	120	60	60	96	98.4	90	90.6	90	93.6
downsampling way	Σ^{EI}									

図 3.17. 実験 2-1 の結果 (S_{28})

Friedman 検定が有意でない場合，または上位 3 つの結果が Wilcoxon の符号付き順位検定で有意差がない場合，結果は強調表示されない。

所見をまとめると以下ようになる。

Observation 6 (目標図形との類似度の比較) 全体的に， F_{BM} と F_{BM+TF} を用いた ELTHON は，32 図形の目標図形のうち，それぞれ 16 図形，11 図形の目標図形に対して，最も視覚的に類似した結果を示した。解析的最適化手法は 2 図形において最も目標図形に類似するタイルを生成したが，DDGA はほとんどの場合において最も類似していな結果となった。

F_{TF} を用いた ELTHON では、2 図形 (S_1, S_{28}) のみが上位の結果となった。いずれの場合も、 F_{TF} の結果は F_{BM+TF} と同等の結果を示した。そこで、次のような観察を行った

Observation 7 (回転関数の寄与) 回転関数 F_{TF} は、 F_{BM} が局所的で微妙な形状の改善に役立つこともあるが、通常では、 F_{BM} が F_{TF} よりも人間の主観に近い結果を示した。

3.4.3.3 階層型最適化方式の動作

ELTHON の挙動を調べるために、ELTHON の 96 回の試行 (3 つの目的関数と目標図形 32 図形、および、5 試行のうち適応度が最も優れた最良解が得られた 1 試行の組み合わせ) において、選択されたタイリングパターンの遷移と選択された頂点数の遷移を観察した。

図 3.18 は、最適化が進むにつれて、全個体で選択されるタイリングパターンの割合の遷移をプロットした図である。選択されたタイリングパターンの割合は色で区別されて、最終世代の最良解が選択したタイリングパターンには丸印を付ける。

96 試行のすべてのケースで、初期個体集団では、すべてのタイリングパターンが等しく選択されている。そして、最適化が進むにつれて、タイリングパターンの割合には 2 つの傾向が見られた。

1 つ目の傾向は、85% のケースで見られた (図 3.18(a)) が、最終世代で最も優れた個体が選択したタイリングパターンの割合が単調に増加する。例えば、図 3.18 (a) では、IH01 の割合は、200 世代目までに急激に増加し、その後、最終世代まで一定の割合を保った。

2 つ目の傾向は、残りの 15% のケースで見られた (図 3.18(b)) では、最適化の途中までは、最良解が選択するタイリングパターンとは異なるタイリングパターンの割合が増加していたが、その後、他のタイリングパターン (最良解が選択するタイリングパターンを含む) の割合が増加し、最終的には、その割合が逆転した。

上記の結果から、ELTHON は多点探索の明確な優位性が明らかになった。

Observation 8 (局所解からの脱出) ELTHON は、タイリングパターンにより局所最適解から脱出する。局所最適解は、最終的には最適ではない中間的に良いタイルを生成する。

表 3.3. 実験 2-2 : 主観評価の結果

Fig	AM		DDGA		ELTHON					
	Ave.	S.D.	Ave.	S.D.	F_{BM}		F_{TF}		F_{BM+TF}	
					Ave.	S.D.	Ave.	S.D.	Ave.	S.D.
S_1	4.00 (± 0.00)		4.95 (± 0.22)		2.80 (± 0.51)		1.70 (± 0.56)		1.25 (± 0.62)	
S_2	2.35 (± 0.57)		5.00 (± 0.00)		1.15 (± 0.65)		3.65 (± 0.57)		2.45 (± 0.80)	
S_3	3.55 (± 1.07)		5.00 (± 0.00)		2.20 (± 0.93)		1.85 (± 0.85)		2.05 (± 0.97)	
S_4	1.75 (± 0.94)		5.00 (± 0.00)		3.40 (± 1.11)		2.10 (± 0.77)		2.20 (± 0.87)	
S_5	2.50 (± 1.07)		5.00 (± 0.00)		1.55 (± 0.97)		3.25 (± 0.77)		2.55 (± 0.86)	
S_6	3.05 (± 0.59)		4.95 (± 0.22)		1.70 (± 0.64)		3.75 (± 0.62)		1.35 (± 0.48)	
S_7	3.70 (± 0.56)		5.00 (± 0.00)		1.05 (± 0.22)		2.80 (± 0.68)		2.25 (± 0.77)	
S_8	1.65 (± 0.85)		4.85 (± 0.65)		1.60 (± 0.80)		4.05 (± 0.22)		1.75 (± 0.94)	
S_9	3.05 (± 0.74)		4.90 (± 0.30)		1.20 (± 0.51)		3.65 (± 0.85)		2.00 (± 0.55)	
S_{10}	3.55 (± 0.80)		4.95 (± 0.22)		2.00 (± 0.45)		3.25 (± 0.54)		1.20 (± 0.40)	
S_{11}	2.80 (± 1.12)		5.00 (± 0.00)		1.90 (± 0.89)		2.40 (± 0.97)		2.65 (± 1.15)	
S_{12}	2.05 (± 0.74)		3.25 (± 1.22)		2.70 (± 1.10)		4.85 (± 0.48)		1.70 (± 0.95)	
S_{13}	1.95 (± 1.07)		4.95 (± 0.22)		3.00 (± 1.00)		2.60 (± 1.32)		2.20 (± 0.87)	
S_{14}	1.90 (± 0.94)		5.00 (± 0.00)		1.75 (± 0.89)		3.20 (± 0.98)		3.05 (± 0.74)	
S_{15}	3.55 (± 0.80)		5.00 (± 0.00)		1.80 (± 1.08)		2.05 (± 1.02)		2.15 (± 0.91)	
S_{16}	3.95 (± 0.22)		5.00 (± 0.00)		1.40 (± 0.58)		2.85 (± 0.57)		1.50 (± 0.50)	
S_{17}	2.75 (± 0.83)		4.80 (± 0.87)		1.20 (± 0.87)		3.50 (± 0.81)		2.35 (± 0.73)	
S_{18}	3.90 (± 0.44)		5.00 (± 0.00)		1.25 (± 0.54)		2.70 (± 0.71)		1.60 (± 0.58)	
S_{19}	3.50 (± 0.97)		4.75 (± 0.62)		1.15 (± 0.36)		3.30 (± 0.84)		2.00 (± 0.71)	
S_{20}	1.90 (± 0.99)		5.00 (± 0.00)		1.80 (± 0.93)		3.45 (± 1.07)		1.85 (± 0.85)	
S_{21}	3.85 (± 0.36)		5.00 (± 0.00)		1.55 (± 0.50)		3.15 (± 0.36)		1.25 (± 0.43)	
S_{22}	2.80 (± 0.68)		4.15 (± 0.36)		1.30 (± 0.56)		4.75 (± 0.43)		1.90 (± 0.44)	
S_{23}	4.55 (± 0.92)		3.65 (± 1.15)		1.60 (± 0.92)		2.25 (± 0.89)		2.40 (± 0.92)	
S_{24}	1.30 (± 0.78)		5.00 (± 0.00)		2.00 (± 0.95)		3.35 (± 1.06)		2.50 (± 1.07)	
S_{25}	3.05 (± 0.59)		4.20 (± 0.60)		1.45 (± 0.59)		4.40 (± 0.73)		1.65 (± 0.57)	
S_{26}	3.95 (± 0.50)		4.75 (± 0.43)		1.65 (± 0.48)		3.15 (± 0.48)		1.25 (± 0.43)	
S_{27}	2.90 (± 1.04)		4.80 (± 0.87)		2.05 (± 1.02)		2.90 (± 1.30)		1.90 (± 0.94)	
S_{28}	4.10 (± 0.30)		4.90 (± 0.30)		2.65 (± 0.73)		1.65 (± 0.73)		1.65 (± 0.57)	
S_{29}	3.75 (± 0.70)		4.95 (± 0.22)		1.40 (± 0.58)		2.65 (± 0.79)		1.90 (± 0.99)	
S_{30}	3.00 (± 0.63)		4.85 (± 0.48)		1.80 (± 0.68)		3.85 (± 0.57)		1.30 (± 0.56)	
S_{31}	4.00 (± 0.00)		5.00 (± 0.00)		1.75 (± 0.94)		1.90 (± 0.77)		1.75 (± 0.83)	
S_{32}	2.70 (± 0.95)		4.90 (± 0.30)		1.35 (± 0.79)		3.55 (± 0.86)		2.35 (± 0.91)	

第 3 章 階層型最適化方式の提案

図 3.18(b) では、200 世代目に IH07 を持つ個体が個体集団内の 4 分の 1 を占めているが、IH07 を用いた最良解の F_{TF} 値は 0.81 で止まっている（図 3.19(a) に IH07 を選択した個体のタイルを示す）。探索の中盤以降、IH04 を選択する個体が増加する。最終的には、個体群の中で最も優れた個体の F_{TF} 値は 0.56 となり、タイリングパターンは IH04 を選択した（図 3.19(b) に IH04 を選択した個体のタイルを示す）。

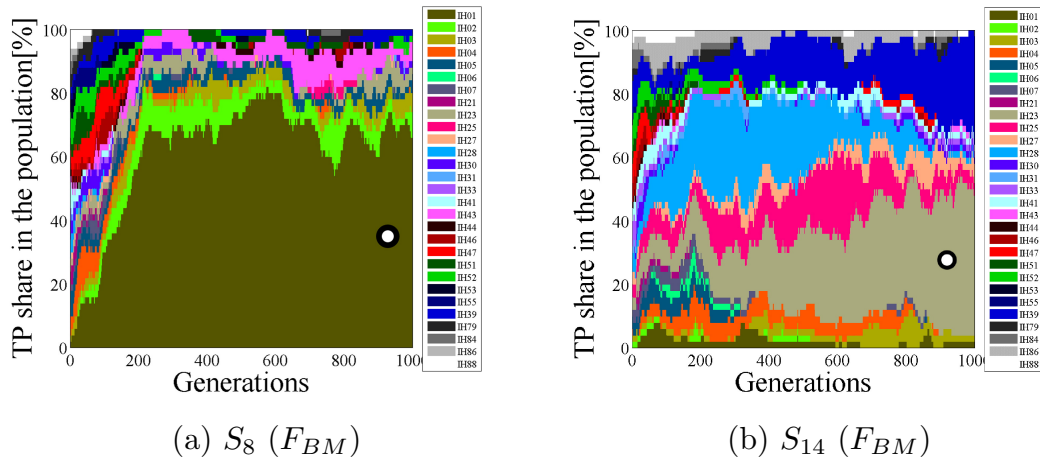
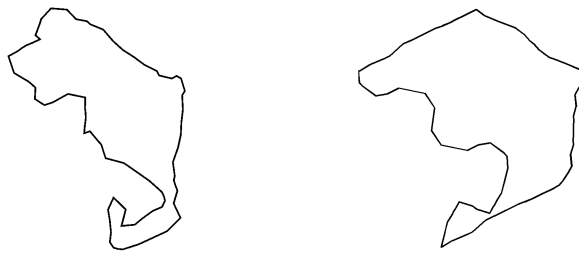


図 3.18. 選択された TP の割合の遷移例



(a) IH07 を選択した個体 (b) IH04 を選択した個体

図 3.19. タイリングパターンによる S_7 のタイルの違い

また、全頂点の中で選択された頂点数の割合は、選択されたタイリングパターンよりも単純に変化した（ S_7 と S_{13} の代表例を図 3.20 に示す）。図 3.20 では、最終世代で最適解が選択した頂点数の総数を、それぞれのケースで囲んでいる。最適解が選択した頂点数の割合は、探索開始から中盤にかけて増加し、それ以降は減少しなかった。

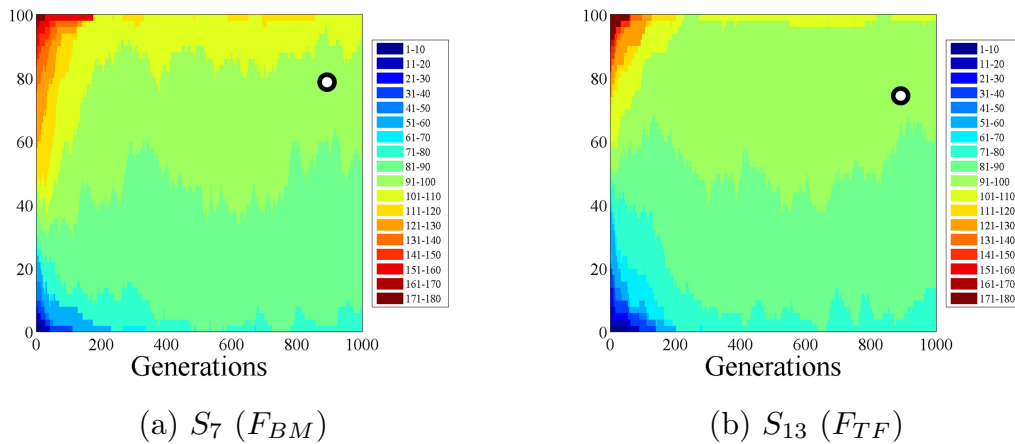


図 3.20. 選択された頂点数の遷移例

3.4.4 実験 3：漫画のキャラクタへの応用

最後の実験では、ELTHON が複雑な漫画のキャラクタをタイルに変形できることを実証する。

目標図形は図 3.21(a) と (b), 3.22(a) と (b) に示すキャラクタ (目標図形 S_{33} と S_{34} , S_{35} , S_{36}) である. 図 3.21(c) と (e), 3.22(c) と (e) はそれぞれ, F_{BM} と F_{BM+TF} , F_{TF} , F_{BM} を用いた ELTHON により選択された頂点を示し, 図 3.21(d) と (f), 3.22(d) と (f) は, ELTHON が生成したタイルを示す. 選択されたタイリングパターンは S_{33} では IH28, S_{34} では IH04, S_{35} では IH46, S_{36} では IH21 である. 図 3.21(g) は図 3.21(d), (f) の輪郭を, 図 3.22(g) は図 3.22(f) の輪郭を, 人手により少し修正したタイルである.

以上の結果から, ELTHON によって生成されたタイルは, 特徴的な凹凸を維持しつつ, 所定の形状に類似することが分かった.

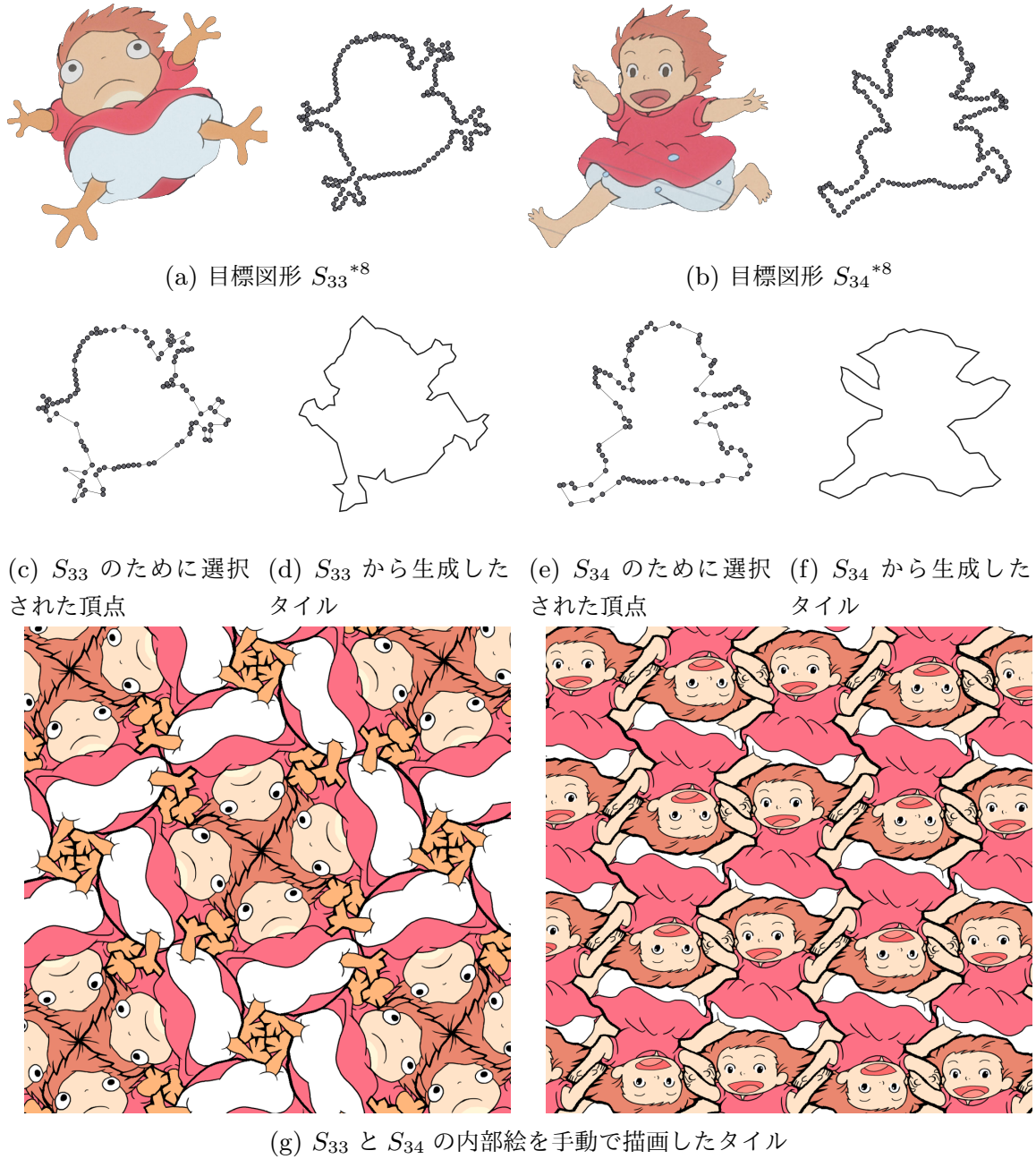


図 3.21. 実例に適用した結果 (S_{33} , S_{34})

*8 ©2008 Studio Ghibli · NDHDMT <https://www.ghibli.jp/works/ponyo/#&gid=1&pid=30>

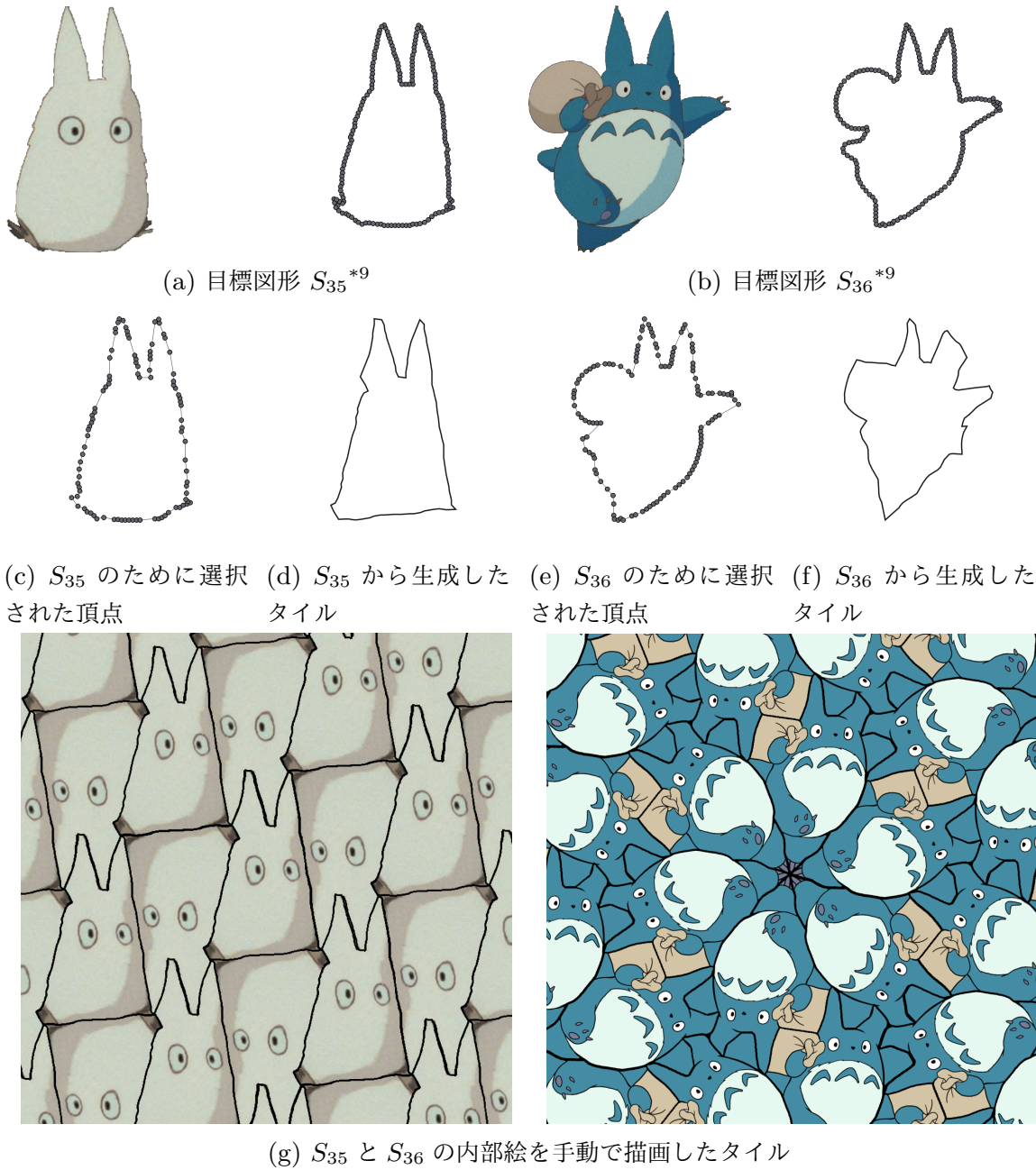


図 3.22. 実例に適用した結果 (S_{35} , S_{36})

*9 ©1988 Studio Ghibli <https://www.ghibli.jp/works/totoro/#&gid=1&pid=19>

3.5 おわりに

本章では，階層型最適化方式に基づくエッシャー風タイルを生成する手法（Escher-Like Tile design method using Hierarchical Optimization : ELTHON）を提案した．上位最適化層は下位最適化層への適切な入力を探査し，固有値問題を解くことでタイルを生成する．つまり，上位最適化層は解析的最適化手法で必要とされる試行錯誤処理を代替するものである．ELTHON は，下位最適化層に解析的最適化手法を用いてエッジ形状の制約を満たす図形を解析的に導出し，上位最適化層にメタヒューリスティクスを用いて図形間の類似性をより適切に評価し，かつ，目標図形 W の形状の調整作業を自動化する．実験の結果より，ELTHON で生成したタイルは，解析的最適化手法の手作業を省きつつ，与えられた目標図形形状に近似することを確認した．また，階層的に最適化を行うことで，エッシャー化問題の難しさを分割して解くことが可能なことを確認した．さらに，3つの目的関数の性能を評価，比較し，その有効性が与えられた目標図形形状に依存することを確認した．

第 4 章

階層型最適化方式の拡張

本章では，階層型最適化方式を具体化した第 2 の手法，すなわち，タイリングに適した目標図形を生成することで，目標図形に類似するタイルを生成する手法の提案を行う．本手法は，先行研究 [17–26] では考慮されていない新たな問題設定を解く手法である．エッシャー化問題においては一般に，目標図形は人間により描画されることを前提としていたが，提案手法 2 は映像や CG モデルを用いてタイルに適した目標図形を生成する．提案手法 2 は 3 つの階層から構成されており，上位最適化層では，対象オブジェクトの画像をパラメトリックに生成するモデルを用いて，潜在変数の最適化によりタイリングに適した目標図形を生成する．中位および下位は，目標図形に類似するタイルを生成する．これにより，目標図形の描画の試行錯誤が不要となることが期待できる．

提案手法 2 を提案するために，エッシャー化問題をもとに新たな問題の定義を行う．提案手法 2 は，エッシャー化問題に 4 番目の条件を追加した問題を解決する．

条件 4： 目標図形 \mathbf{W} は対象オブジェクトの特徴を有する

先行研究 [17–26] では，目標図形 \mathbf{W} は人間により描画または準備されることを前提しているため，対象オブジェクトの特徴を有し，タイリングに適する目標図形であるかは，タイルをせいせいするまで不明である．そこで，提案手法 2 が解決する問題は，対象オブジェクトの特徴を有し，タイリングに適する目標図形 \mathbf{W} ，および，タイルの生成を目指す．

提案手法 2 は，上記の問題（タイル生成問題）が含む 2 つの難しさを解決することで，

目標図形を自動生成し、生成した目標図形からタイルを生成する。提案手法 2 は、上位最適化層において目的関数が曖昧な難しさ、および、探索空間が膨大である難しさを解決し、中位・下位最適化層において制約条件が強い難しさ、および、目的関数が曖昧な難しさを解決する。実験により、本手法は、タイリングに適した目標図形を生成することを確認した。

まず、4.1 節では、第 3 章にて提案した ELTHON の問題点、および、本章の要点について述べる。次に、4.2 節では、処理手順や目的関数など、提案手法 2 の詳細を示す。4.3 節では、実験により提案手法 2 の有効性を示す。4.4 節では、本章の結論を述べる。

4.1 はじめに

本章では、先行研究 [17–26] とは異なる問題設定とし、タイリングに適した目標図形とタイルを同時に生成する問題を解決する。

先行研究、および、第 3 章にて提案した ELTHON は、入力された目標図形に類似するタイルを得ることを目的としている。しかし、エッシャー化問題には、目標図形を用意することが難しいという問題点がある。これは、目標図形の形状によって、タイルへの変換が難しい図形が存在することが要因である。この難しさは、問題自体に起因するものであり、タイルに変形が困難な図形、および、解は存在するが最適な解を発見することが難しい図形が存在する。例えば、円をタイルに変形すると六角形にしか変換できず、これ以上円に似せることはできない。これは、前者の問題の一例である。一方、後者の問題の一例は、図 4.1 に示すように、同じオブジェクトであっても、ポーズが異なるとタイルに変形することが難しい場合がある。上記のことより、タイリングに適した目標図形を探索することが重要であることがわかる。このため、先行研究、および、第 3 章にて提案した ELTHON では、入力する目標図形を何度も描き直す必要がある。

本章では、タイリングに適した目標図形を生成する問題を解決することを目的として、対象オブジェクトの画像をパラメトリックに生成するモデルを構築し、潜在変数の最適化によりタイリングに適した目標図形を生成する手法を提案する。これにより、目標図形の描画の試行錯誤が不要となり、訓練データに含まない画像の生成も期待される。またこの手法を本論文では、Tiling design from Movies using Hierarchical Optimization と命

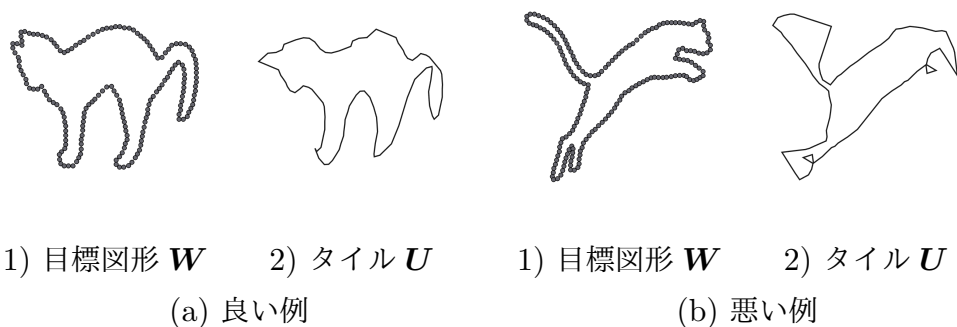


図 4.1. 目標図形 W から影響を受ける例

名し、以降この手法を指す際は、TIMHON と呼ぶ。

本章で提案する手法は、三階層からなる最適化手法となっている。まず、ConvVAE [72] (2.5.2 項参照) により訓練データの低次元特徴量空間を学習する。次に、学習した ConvVAE の潜在変数を最適化することにより、タイリングに適した目標図形を生成する。提案手法は、上位最適化層にて潜在変数の最適化をメタヒューリスティクスにより行い、中間・下位最適化層には第 3 章にて提案した手法を用いて、タイルを生成する。

TIMHON が解決する問題は、設計変数の次元数が多いより問題を解くことが難しい、および、図形間の類似性を評価する万能な関数は発見されていない難しさ、さらに、強い制約条件を持つため、解の発見が容易でない難しさを含む。これらの難しさは、タイル生成における、目標図形を用意することの難しさ、および、図形同士のエッジ形状が一致する制約を満たしつつ、目標図形に類似するタイルを生成する難しさに該当する。これらの難しさを解決するために、TIMHON は、上位最適化層に機械学習の生成モデルとメタヒューリスティクスを用いて、目標図形をパラメトリックに生成する。これにより、図形を生成するという膨大な探索空間を生成モデルの潜在変数空間に圧縮することができる。探索空間を圧縮することで、図形を生成するという自由度が高い問題を解くことが可能となる。さらに、中位・下位最適化層に ELTHON を用いて、目標図形に類似するタイルを生成する。上記のように階層的に最適化を行うことで、目標図形とタイルを同時に設計する問題の難しさを解くことが期待される。

4.2 アルゴリズムの構成

4.2.1 基本的なアイデア

本章では、映像よりキャラクター画像が生成可能なモデルを学習し、タイリングに適したキャラクターのポーズを選択、および、生成する手法を提案する。本手法は、単に映像からタイリングに適したキャラクターのポーズを選択するのみではない。映像から訓練画像を作成し、この訓練画像からパラメトリックにタイリングに適したキャラクターのポーズを生成する。これにより、訓練データ内に含まれないキャラクターのポーズが生成されることが期待できる。以下に、TIMHON の基本アイデアを示す。

アイデア 1: **ConvVAE** によるキャラクター画像の潜在変数の学習：ConvVAE [72] (2.5.2 項参照) は、低次元の潜在変数によりキャラクターのポーズの特徴を表現することができる。また、潜在変数を変更することでキャラクターのポーズを連続的に変形、および、補間することが可能である。

アイデア 2: **潜在変数の最適化**：タイリングに適したキャラクターのポーズを探索するために、潜在変数の最適化を行う。TIMHON ではさまざまなポーズを取得するために、個体集団の多様性を維持しつつ、徐々に収束することが可能な自己適応型差分進化 (jDE) [56] (2.4.3.1 項参照) を採用する。デコードしたキャラクターのポーズの輪郭形状が ELTHON の入力となる。ELTHON により、タイルへの変形、および、元のキャラクターのポーズとタイルの類似度を評価する。

図 4.2 に TIMHON の構造と ELTHON と解析的最適化手法 [18, 19] の関係を示す。TIMHON の入力は、タイルに変形したい対象キャラクターを含む映像であるのに対して、ELTHON の入力は、キャラクターの輪郭形状である。このため、TIMHON は、キャラクターのポーズを生成するための上位最適化層と、目標図形 \mathbf{W} の輪郭形状を修正して $\tilde{\mathbf{W}}$ を探索するための中間最適化層、および、 $\tilde{\mathbf{W}}$ をタイルに変形するための下位最適化層の 3 層からなる階層型最適化方式となっている。

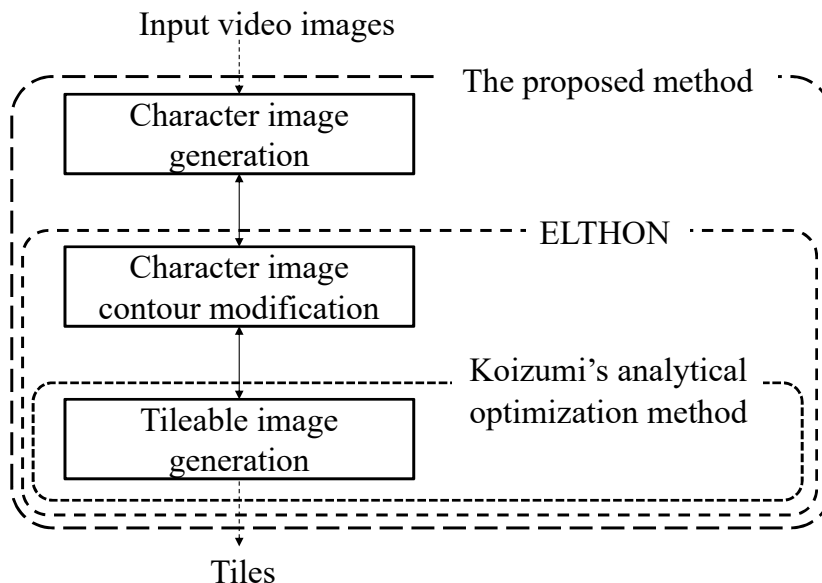


図 4.2. TIMHON の構造と先行研究との関係

ここで、TIMHON は、対象キャラクターのシルエット（輪郭）を生成する手法であり、輪郭内部の画像の生成は行わないことに留意されたい。エッシャー化問題の最も難しい点は、タイルの輪郭を求めることであり、タイルの輪郭に従って内部の絵を描くことは難しくないため、生成の対象としない。

4.2.2 処理手順

TIMHON は、図 4.3 に示すように学習フェーズとタイル生成フェーズの 2 つのフェーズから構成される。学習フェーズでは、入力された映像から抽出した対象キャラクターのシルエット画像を用いて ConvVAE [72] の学習を行う。TIMHON では、対象キャラクターのポーズを得ることを目的としているため、対象キャラクターの輪郭内部の情報は無視する。ConvVAE を学習させたあと、生成フェーズではタイリングに適したキャラクターのポーズを生成し、生成されたキャラクターの輪郭形状をタイルに変形する。タイル生成フェーズは図 4.2 に示す 3 階層の TIMHON を用いて目標図形とタイルを生成する役割を持つ。

学習フェーズとして、映像から ConvVAE の訓練データを作成し、ConvVAE の学習を行う。まず訓練データの作成では、映像をフレームに分け、セマンティックソフトセグメンテーション [81] により各フレームを対象キャラクター（前景）とその他（背景）に分離

第 4 章 階層型最適化方式の拡張

され、 T_f フレームに対して、前景と背景のラベル付けを手動で行う。その後、残りのセグメンテーションされたフレームは、HSV ヒストグラムを特徴量とした、単純な k-近傍法により自動的に前景と背景のラベル付けが行われる。抽出された対象キャラクタは白黒のシルエット画像に変換され、対象キャラクタの大きさにトリミングされて、 $P_t \times P_t$ ピクセルにリサイズする。図 4.4 に訓練データの一例を示す。上記の処理により得られた $P_t \times P_t$ ピクセルのシルエット画像が ConvVAE の訓練データとなる。ConvVAE は教師なし学習であるため、TIMHON の学習フェーズで行う手作業は、 T_f フレームに対して行うラベル付けのみである。

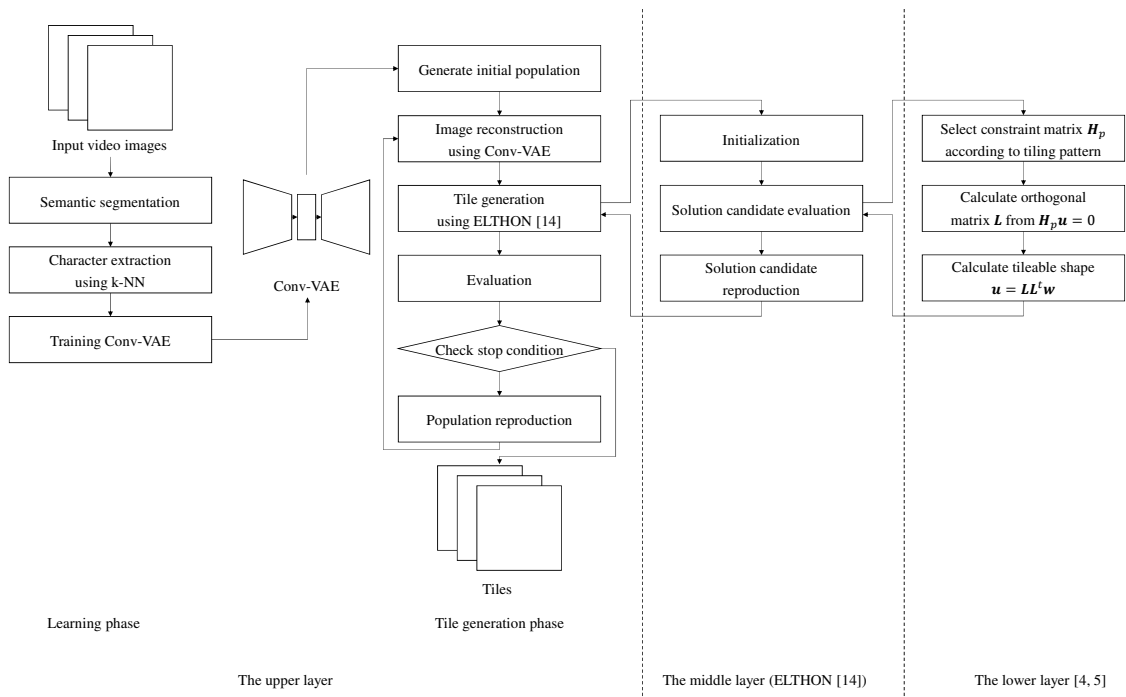


図 4.3. TIMHON の処理手順



図 4.4. ConvVAE の訓練データの例

タイル生成フェーズでは、タイリングに適したキャラクタのポーズを探索するために適

切な潜在変数値の組み合わせを求める。そして、生成されたキャラクタの輪郭形状をタイルに変形する。図 4.3 に示すように、タイル生成フェーズでは、メタヒューリスティクスを用いて、解候補の生成と評価を繰り返し行う。これは、最適化の設計変数が ConvVAE が学習した潜在変数値の組み合わせであり、キャラクタのポーズを ConvVAE のデコーダを用いて復号しているためである。潜在変数の最適化により生成されたキャラクタの輪郭形状をタイルに変形後、上位最適化層にてタイルの評価を行い、新たな解候補を生成する。ここで、TIMHON では、生成されたキャラクタの輪郭形状をタイルに変形するために ELTHON (第 3 章を参考) を使用しているが、エッシャー化問題を解決する手法であれば、どの手法でも適用可能である。TIMHON で ELTHON を用いる理由は、目標図形に類似するタイルを生成できるためである。上記の処理を停止条件が成立するまで繰り返し実行する。

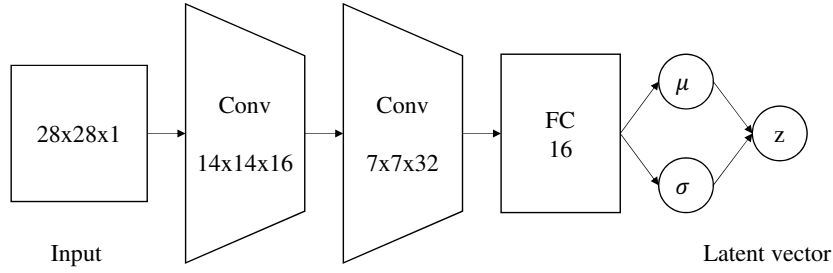
図 4.5 に TIMHON が用いる ConvVAE のネットワーク構造を示す。エンコーダは、2 層の畳み込み層と全結合層から構成され、全ての層の活性化関数として ReLU 関数を用いる。全結合層は平均 μ と分散 σ を出力して、入力データポイント \mathbf{x} の潜在変数 \mathbf{z} を生成する (詳細は 2.5.2 項を参照)。一方、デコーダは、全結合層と 2 層の逆畳み込み層から構成される。上記のエンコーダ・デコーダネットワークにより、入力データポイント \mathbf{x} の次元を減らすことができる。ここで、潜在変数 \mathbf{z} の次元数は D_l 次元とする。

4.2.3 潜在変数の最適化

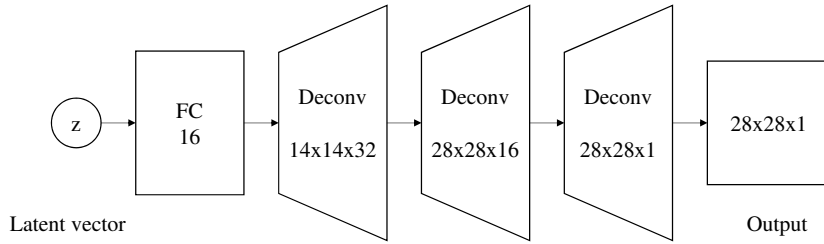
TIMHON では、潜在変数の最適化により、タイリングに適し、かつ、映像に含まれないキャラクタのポーズを生成する。

最適化を行う設計変数は、 D_l 次元の潜在変数 \mathbf{z} (図 4.6) であり、最小化する目的関数 F_{lo} は、ConvVAE のデコーダを用いて生成されたキャラクタの輪郭形状 $\mathbf{W}(\mathbf{z})$ と ELTHON により変形されたタイル形状 $\mathbf{U}(\mathbf{z})$ との形状の類似度の距離、および、 $\mathbf{U}(\mathbf{z})$ の形状の複雑度の重み付き線形和である。 $\mathbf{W}(\mathbf{z})$ と $\mathbf{U}(\mathbf{z})$ の形状の類似度の距離は小さいほど類似性が高いことを示し、 $\mathbf{U}(\mathbf{z})$ の形状の複雑度は大きいほど $\mathbf{U}(\mathbf{z})$ の形状が単純でないことを示す。

$$F_{lo}(\mathbf{z}) = \gamma F_{BM}(\mathbf{U}(\mathbf{z}), \mathbf{W}(\mathbf{z})) + (1 - \gamma) F_C(\mathbf{U}(\mathbf{z})) + \rho(\mathbf{W}(\mathbf{z})). \quad (4.1)$$



(a) エンコーダネットワーク



(b) デコーダネットワーク

図 4.5. TIMHON が用いる ConvVAE のネットワーク構造

ここで、 $\mathbf{U}(z)$ は潜在変数 z から復元したキャラクタの輪郭形状 $\mathbf{W}(z)$ を ELTHON により変換したタイルを表す。また、 F_{BM} は、双方向マッピングベース関数 (3.3.2.4(a) 項を参照) であり、 $F_C(\mathbf{U}(z))$ は形状の複雑さを評価する関数であり、以下の数式で表現される。

$$F_C(\mathbf{U}(z)) = \frac{l(\mathbf{U}(z))}{2\sqrt{S(\mathbf{U}(z))}\pi} \quad (4.2)$$

ここで、 $l(\mathbf{U}(z))$ と $S(\mathbf{U}(z))$ はそれぞれ、 $\mathbf{U}(z)$ の輪郭の長さ と $\mathbf{U}(z)$ の面積を示す。また、 $\mathbf{U}(z)$ と同じ面積を持つ円の円周と $\mathbf{U}(z)$ の円周の長さの比を示す $F_C(\mathbf{U}(z))$ を導入することで単純すぎる形状を持つタイルが生成されることを抑制する。

$\rho(\mathbf{W}(z))$ は品質の悪い $\mathbf{W}(z)$ を抑制するために、 $\mathbf{W}(z)$ が「 $\mathbf{W}(z)$ は、大きなボケを持つ領域を含まない」と「 $\mathbf{W}(z)$ の隣接する領域が 1 つである」という 2 つの制約条件に違反した場合、ペナルティ値を返す関数である。 $\rho(\mathbf{W}(z))$ は、以下の式により計算される。

$$\rho(\mathbf{W}(z)) = \rho_{blur}(\mathbf{W}(z)) + \rho_{cont}(\mathbf{W}(z)) \quad (4.3)$$

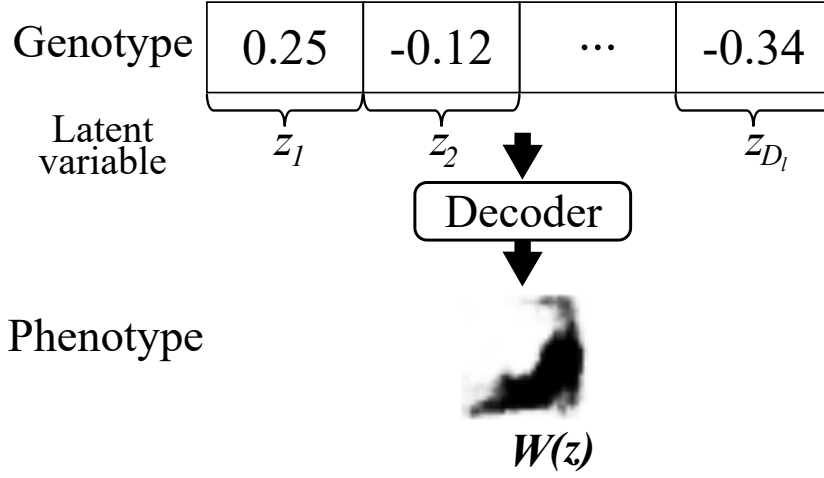


図 4.6. TIMHON の上位最適化層の個体表現

$$\rho_{blur}(\mathbf{W}(z)) = \begin{cases} V_{blur} & \text{if } var(\mathcal{L}(\mathbf{W}(z))) < R_{blur} \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

$$\rho_{cont}(\mathbf{W}(z)) = \begin{cases} V_{cont} & \text{if } N_{area}(\mathbf{W}(z)) > 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

ここで、 $var(\mathcal{L}(\mathbf{W}(z)))$ は復元された $\mathbf{W}(z)$ のラプラシアンフィルタ [82, 83] の分散を表す。 $N_{area}(\mathbf{W}(z))$ は、 $\mathbf{W}(z)$ の隣接する領域の数がを表す。

1 つ目の制約条件である $\mathbf{W}(z)$ は、大きなボケを持つ領域を含まないは、 $\mathbf{W}(z)$ のボケ度合いを数値化し確認する。 $\mathbf{W}(z)$ のボケ度合いは、 $\mathbf{W}(z)$ にラプラシアンフィルタを適用して、適用後に得られた画像 $\mathcal{L}(W)$ の分散 $var(\mathcal{L}(\mathbf{W}(z)))$ を計算する [82, 83]。 $var(\mathcal{L}(\mathbf{W}(z)))$ が閾値 R_{blur} を下回った場合、 $\mathbf{W}(z)$ はボケ度合いが強いとしてペナルティ値を与える。ラプラシアン画像の分散 $var(\mathcal{L}(\mathbf{W}(z)))$ を用いて画像のボケ具合を表す式は以下のように示せる。

$$var(\mathcal{L}(\mathbf{W}(z))) = \sum_{(i,j) \in \Omega(x,y)} (\Delta I(i,j) - \bar{\Delta I})^2 \quad (4.6)$$

ここで、 $\Omega(x,y)$ はピクセルの局所近傍を示し、 ΔI はラプラシアンフィルタで畳み込んだラプラシアン画像を示し、 $\bar{\Delta I}$ は $\Omega(x,y)$ 内のラプラシアン画像の平均値を示し、 i, j は画像内の座標を示す。上記の式を $\mathbf{W}(z)$ 全体に適用することで、 $\mathbf{W}(z)$ のボケ度合いを表すことができ、 $var(\mathcal{L}(\mathbf{W}(z)))$ が閾値 R_{blur} を下回った場合は、 $\mathbf{W}(z)$ のボケ度合いが強いと判断する。また、ラプラシアンフィルタを適用後の画像の分散により、画像のボケ度合いを評価できる理由は以下の通りである。一般的にラプラシアンフィルタは、エッジ

検出に使用されており、焦点があった画像であれば、エッジ部分と非エッジ部分（背景）との差が大きくなり、画像の分散が大きいことを示す。一方、画像の分散が小さい場合、エッジ部分と非エッジ部分（背景）との差が小さくなり、画像中のエッジ部分が少なくなり、ボケた画像であると判断ができる。このため、ラプラシアンフィルタを適用した画像の分散により、画像のボケ度合いを評価できる。

2つ目の制約条件である $\mathbf{W}(z)$ の隣接する領域が1つであるは、 $\mathbf{W}(z)$ 内の隣接する領域が1つであるか確認する。 $\mathbf{W}(z)$ を2値画像に変換し、 $\mathbf{W}(z)$ 内の領域数 $N_{area}(\mathbf{W}(z))$ を検出し、 $N_{area}(\mathbf{W}(z))$ が2以上存在する場合、ペナルティ値を与える。つまり、 $N_{area}(\mathbf{W}(z))$ は、 $\mathbf{W}(z)$ 内の連結成分の総数である。ただし、 $N_{area}(\mathbf{W}(z))$ が2以上の場合でも、隣接する領域サイズが小さい（領域を構成する画素数が少ない）場合は、領域として扱わない。

TIMHON は、各解候補 x_i に自身の制御パラメータ（スケールファクタ F_i と交叉率 CR_i ）を持つ単純な自己適応機構を持つ jDE [56] を用いる。詳細な説明は、2.4.3.1 項を参照されたい。

4.3 評価実験

4.3.1 実験準備

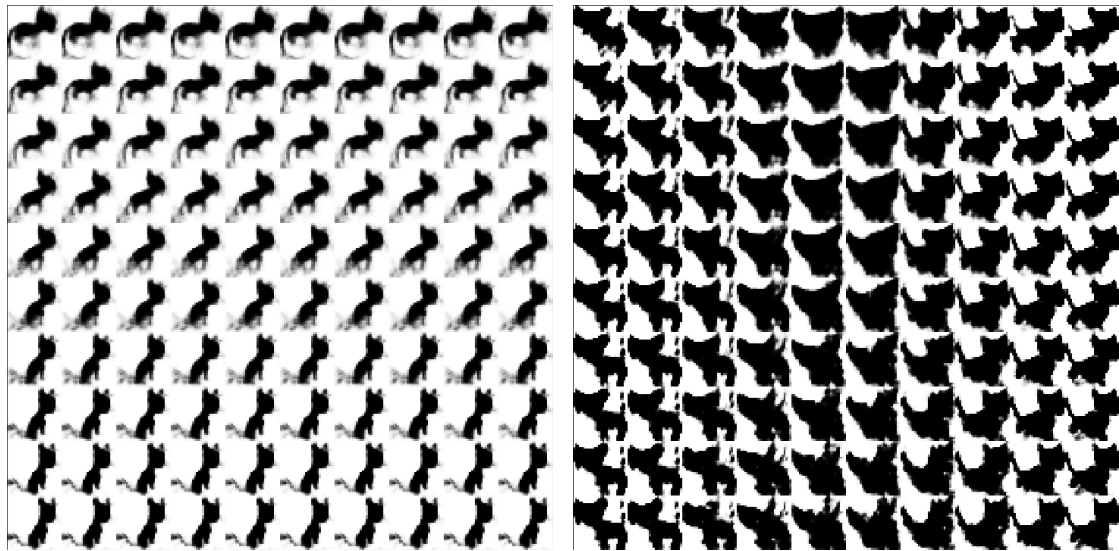
本章では、TIMHON の有効性を検証するために、映像からタイルを生成する実験を行った。この実験では、猫の映像 69 本（映像 1 本あたり平均 2,673 フレーム）を含むデータセットを作成した。上記のデータセットを作成するために、セグメンテーションを行うのために 150 フレーム、すなわち、1つの映像に対して $T_f = 2$ または 3 フレームに手動でラベル付けを行った。

TIMHON の ConvVAE に関するパラメータは以下のように設定した。入力画像サイズ P_t を 28 ピクセル、潜在変数空間の次元数 D_l を 6 から 10 次元、バッチサイズを 128、エポック数を 200 とした。jDE の制御パラメータ F_L, F_U, R_{SF}, R_{CR} は、参考文献 [58] をもとにそれぞれ、0.1, 0.9, 0.1, 0.1 とした。jDE における停止条件、個体集団サイズはそれぞれ、500, 10 とした。残りのパラメータ $V_{blur}, V_{cont}, R_{blur}, \gamma$ はそれぞれ、 $10^3, 10^3, 3.0 \times 10^3, 0.01$ と設定した。ELTHON は、一様交差、ビット反転による突然変異、MGG

世代交代モデル [64] を用いた。停止条件, 個体集団サイズ, 交叉率, 突然変異率, 試行回数は, 25,000, 50, 0.4, 0.03, 1 とした。

4.3.2 実験結果

図 4.7 は, 訓練済の ConvVAE の潜在変数空間を可視化した例を示す。図 4.7(a) に示す $D_l = 7$ の結果では, z_1 と z_6 をそれぞれ $[-4.0, 0.0]$, $[-4.0, 4.0]$ の範囲で変化させ, z_2, z_3, z_4, z_5, z_7 をそれぞれ 2.0, 0.0, 0.0, 0.0, 0.0 に固定した結果である。図 4.7(a) より, 縦軸の値が小さくなるにつれて, 立っている猫が徐々に座るポーズに変化している。同様に, 図 4.7(b) に示す $D_l = 10$ の結果では, z_3 と z_5 をそれぞれ, $[-4.0, 4.0]$, $[-4.0, 0.0]$ の範囲で変化させ, $z_1, z_2, z_4, z_6, z_7, z_8, z_9, z_{10}$ をそれぞれ, 0.0, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 2.0 に固定した結果である。図 4.7(b) より, 横軸の値が大きくなるにつれて, 左を向いていた猫が徐々に右を向くように変化している。



(a) $D_l = 7$ の結果 (横軸と縦軸はそれぞれ, $-4.0 \leq z_1 \leq 0.0$, $-4.0 \leq z_6 \leq 4.0$. z_2, z_3, z_4, z_5, z_7 はそれぞれ, 2.0, 0.0, 0.0, 0.0, 0.0 に固定した結果).

(b) $D_l = 10$ の結果 (横軸と縦軸はそれぞれ, $-4.0 \leq z_3 \leq 4.0$, $-4.0 \leq z_5 \leq 0.0$. $z_1, z_2, z_4, z_6, z_7, z_8, z_9, z_{10}$ はそれぞれ, 0.0, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 2.0 に固定した結果).

図 4.7. 訓練済み ConvVAE の可視化結果の例.

図 4.8 から 4.12 は、jDE を用いて潜在変数を最適化して生成したタイリングに適したポーズを持つキャラクタ画像と、ELTHON を用いてタイルに変形した結果を示している。また、目的関数 $F_{l_0}(z)$ の値も示す。図は潜在変数空間の次元を 6 から 10 次元に変更した場合の結果を示している。

提案手法は、多点探索の利点を活かして、タイリングに適したさまざまなキャラクタのポーズを探索している。また、各復元画像 $\mathbf{W}(z)$ に最も類似する訓練画像も図 4.8 から 4.12 に示している。図 4.8 の 3 番目、4 番目の結果、図 4.10 の 1 番目の結果、図 4.12 の 3 番目、5 番目の結果のように、尻尾の位置や足の位置が異なるなど、訓練画像に似ていない復元画像 $\mathbf{W}(z)$ の出力も発見された。このことから、ConvVAE は訓練画像の中から中間的なポーズを生成することに成功して、jDE を用いた潜在変数の最適化により、タイリングに適したキャラクタのポーズを補間できていることがわかった。

TIMHON は、訓練画像の部分的な形状を組み合わせることで、訓練画像に似ていない復元画像 $\mathbf{W}(z)$ を出力したと考える。また本実験では、復元画像 $\mathbf{W}(z)$ に最も類似する訓練画像の探索には、単純なピクセル間のハミング距離を用いている。このため、キャラクタのポーズの類似性までは考慮できていないと考える。この理由により、訓練データ内に復元画像 $\mathbf{W}(z)$ により類似する訓練画像が存在する可能性がある。

図 4.13 に本評価実験で得られたタイルを用いたタイリングアートを示す。内部のイラストは、人間により描かれたものである。図 4.13 より、TIMHON はさまざまなポーズの猫のタイルの生成に成功していることがわかる。

4.4 おわりに

本章では、映像からタイリングに適したキャラクタ画像を生成し、タイルに変形する手法を提案した。TIMHON は、ConvVAE を用いてキャラクタの特徴量空間を学習し、タイリングに適したキャラクタのポーズを探索し、映像に含まれていない新たなポーズを生成することができる。実験結果より、TIMHON は、タイリングに適したさまざまなポーズのキャラクタ画像を探索することに成功し、訓練データに含まれないポーズも生成することができた。また、階層的に最適化を行うことで、目標図形とタイルを同時に生成する問題の難しさを分割して解くことが可能なことを確認した。一方、定量的な評価実験や復

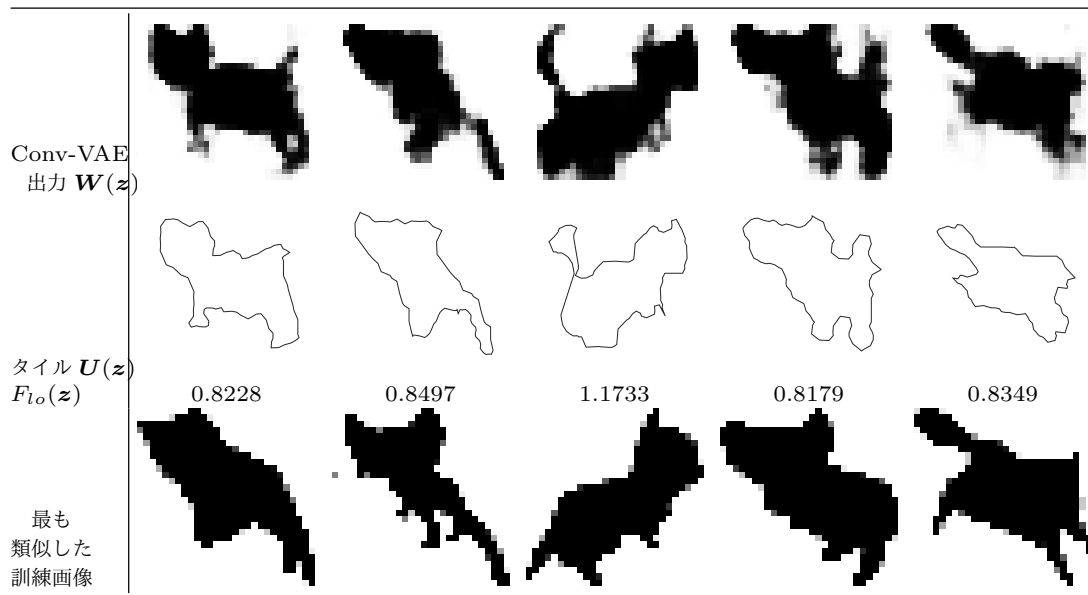


図 4.8. $D_l = 6$ の実験結果.

元画像 $W(z)$ と訓練画像の適切な比較を行えていないため、今後はこれらの点に関して追加実験や改善が必要である。

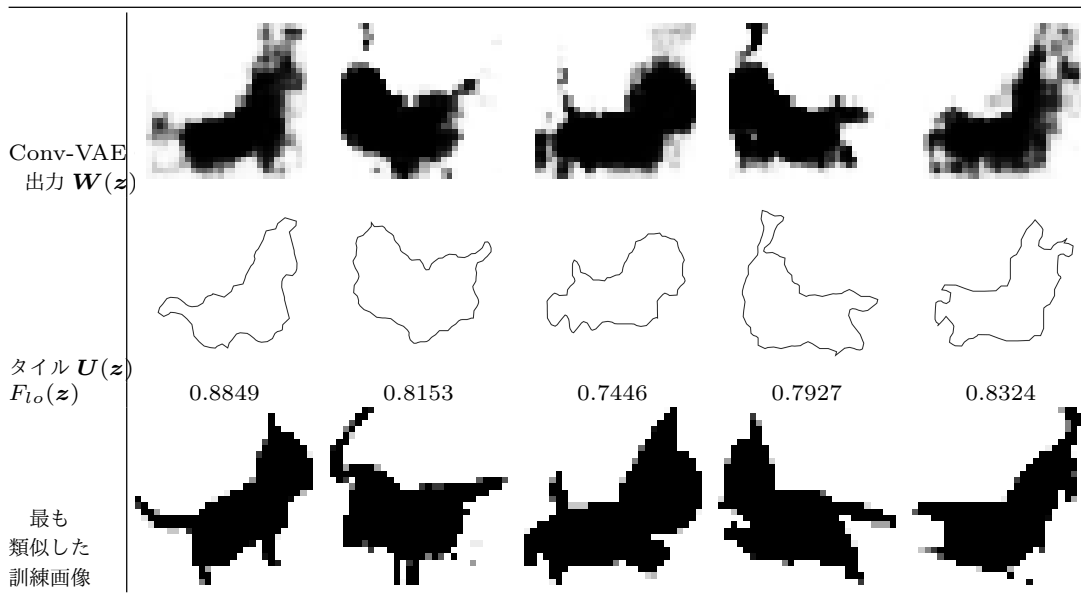


図 4.9. $D_l = 7$ の実験結果.

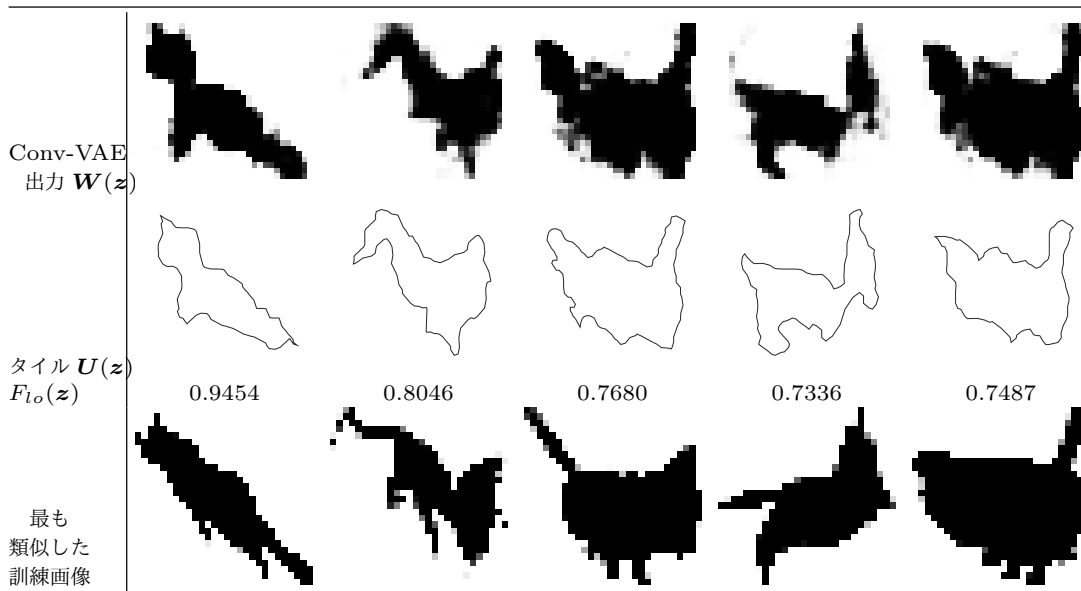


図 4.10. $D_l = 8$ の実験結果.

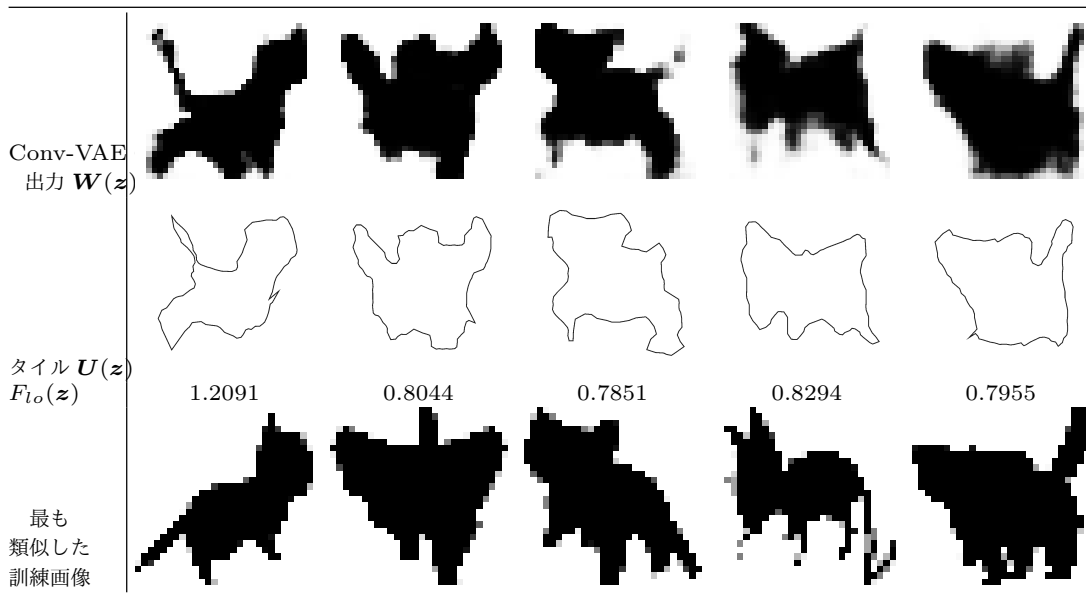


図 4.11. $D_l = 9$ の実験結果.

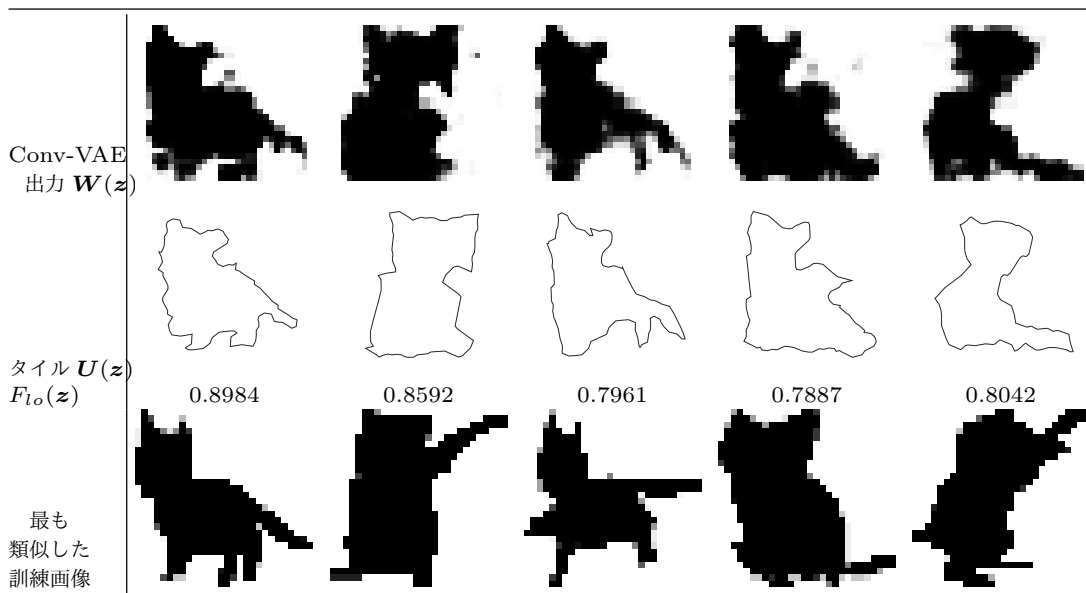
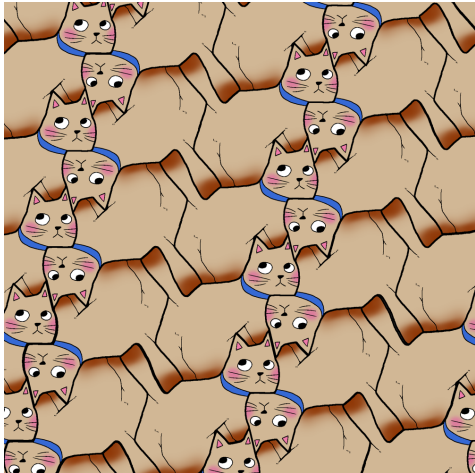
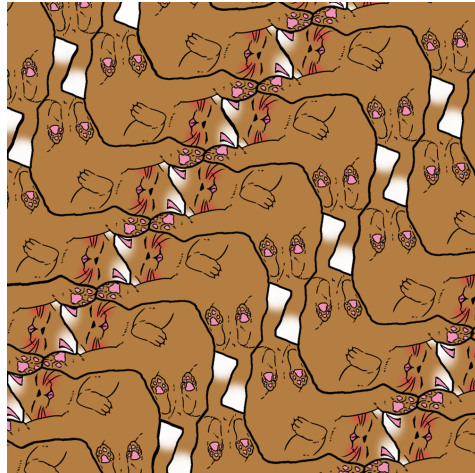


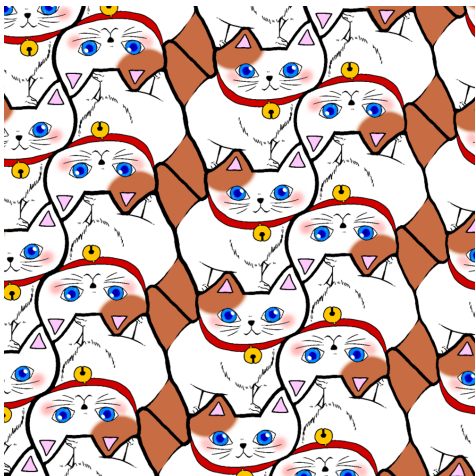
図 4.12. $D_l = 10$ の実験結果.



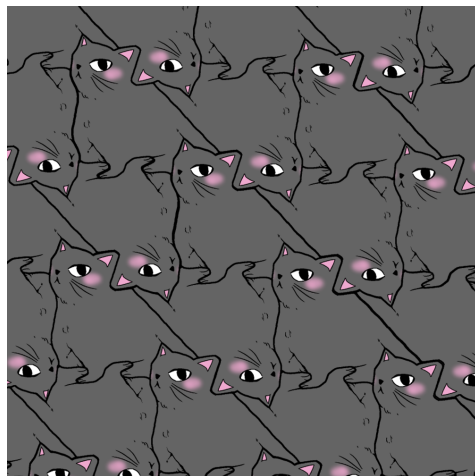
(a) 図 4.8 の 3 番目のタイル



(b) 図 4.9 の 4 番目のタイル



(c) 図 4.11 の 2 番目のタイル



(d) 図 4.12 の 1 番目のタイル

図 4.13. タイリングアードの例

第5章

製菓への応用

本章では，エッシャー化問題を解決する手法が多様な産業に応用可能であることを示す一つの事例として，製菓への応用を試みる．第3章で提案した ELTHON の上位最適化層に新たな制約条件を追加することで，成形の際に壊れやすい部位の発生を抑制する．実験により，製菓に適したタイリングが可能な図形が生成されることを確認した．

まず，5.1 節では，タイリングを製菓に応用する利点を述べる．次に，5.2 節では，処理手順や製菓特有の制約条件など，ELTHON に新たな制約を追加した手法の詳細を示す．5.3 節では，実験により ELTHON に新たな制約を追加した手法の有効性を第3章にて提案した ELTHON と比較する．5.4 節では，本章の結論を述べる．

5.1 はじめに

本章では，第3章，第4章にて提案した ELTHON と TIMHON の多様な産業に応用可能であることを示す一つの事例として，製菓への応用を試みる．

最近，タイリングはグラフィックデザインや建物のテクスチャのみではなく，緩衝材やコースターなどの物理的な物体の形状デザインにも用いられている（2.2 項を参照）．このように，タイリングを物理的な対象物に導入することは，芸術分野のみならず，工学や産業などの分野に新たな価値をもたらすものである．

そこで本章では，新たな分野にタイリングを導入する試みの一つとして，製菓への応用を行う．チョコレートやクッキーなどのお菓子の形状を工夫することで，お菓子の付加価値や魅力を高めることができる．また，実際に，お菓子の形状を工夫して，付加価値を高

める実例も存在する*1。例えば、映画やアニメの登場人物の形状をしたクッキーは、遊園地などのお土産として人気があります。クッキーの形状をタイルにすることで、パズルのように遊べたり、クッキーの型崩れが生じないように隙間がない包装が可能であったりと、お菓子に付加価値を持たせることができる。

本章では、第3章にて提案した ELTHON に製菓特有の制約条件を導入し、実際に製菓に使用する型を作成する。ここで、第4章にて提案した TIMHON を用いることも可能であることに留意されたい。本手法は、ELTHON の利点である目的関数や制約条件の柔軟性が高いことを活かし、上位最適化層に製菓特有の新たな制約条件を追加する。新たな制約条件を導入することで、出力されるタイルの形状に壊れやすい部分の発生を抑制しつつ、製菓に適したタイリングが可能な図形が生成できる。

5.2 アルゴリズムの構成

5.2.1 基本的なアイデア

本章では、ELTHON を製菓に適用し、タイルのお菓子を作成する抜き型や焼き型を設計する手法を提案する。本章の提案手法は、基本的な枠組みは ELTHON と同じであり（詳細は第3章を参照）、エッシャー化問題の難しさを分解して解を探索しているが、上位最適化層に製菓特有の制約条件を追加して、製菓に応用する。

ELTHON の目的関数や制約条件の柔軟性を利用して、生成されるタイルの形状に狭くて脆い部分が発生しないようにするための新たな制約条件を導入する。これは、型を用いて生地を抜き出し、オーブンで焼くと、狭くて脆い部分が割れやすくなるためである。つまり、動物などのタイル形状の型を設計する際に、首や足が細くなりすぎないように必要な幅を保った型を設計することができる。新しい制約条件を導入することで、狭く脆い部分の発生を抑制しながら、製菓に適したタイルの型を設計することを目指す。

*1 チョコレートの形状をパズルピースにすることで、自由に繋げられ、チョコレートにパズルという付加価値を追加したお菓子 [84] や、チョコレートに型抜きの要素を追加することで、板チョコを割り、指定の形状に型抜きをするという付加価値を追加したお菓子 [85] などが存在する。

5.2.2 処理手順

ELTHON は第 3 章で述べた通り，2 つの最適化手法が階層的な関係になるように構成しており，図 5.1 にフローチャートを示す。

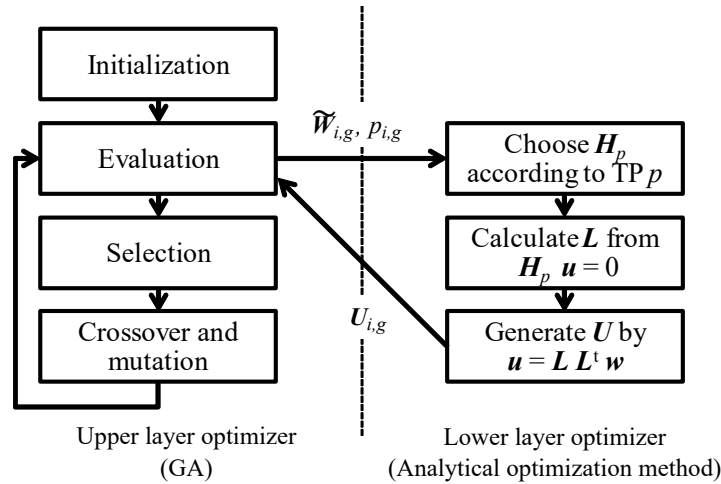


図 5.1. ELTHON の処理手順（再掲）

上位最適化層の GA は，最適なタイリングパターンの選択，および，最適な目標図形 W からダウンサンプリングされた形状 \tilde{W} を形成する頂点を選択する（3.3.2.2 項を参照）。そして，GA の各個体に対して，毎世代，下位最適化層の解析的最適化手法を適用し，エッシャー化問題の条件 2（タイルの対応する 2 辺の形状が一致する制約）を満たすように \tilde{W} からタイル U に変形する．解析的最適化手法は，GA により生成された \tilde{W} に可能な限り類似する U を導出する．下位最適化層の解析的最適化手法が変形した U は，上位最適化層の GA により評価される．つまり，GA は U の自己交差の制約違反と，新たな制約条件として対向する辺間の幅の制約違反の確認を行い，適応度を計算する．ここで，目的関数には，双方向マッピングベース関数を用いる（3.3.2.4 (a) 項を参照）。

5.2.3 製菓特有の制約条件

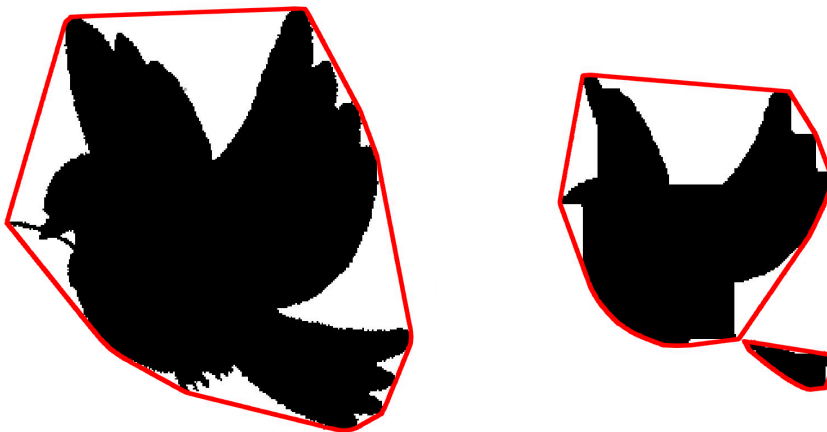
本章で導入する新たな制約条件は，目的関数にペナルティ関数として加算することで，生成されたタイルに狭く脆い部分が形成されることを抑制する役割がある．本章では，製

菓特有の制約条件を以下の不等式で定義する.

$$w_n(\mathbf{U}) \geq \max(w_n(\mathbf{W}), T_s \cdot w_w(\mathbf{W})) \quad (5.1)$$

ここで, $w_n(\cdot)$ と $w_w(\cdot)$ は, それぞれ図形の最も狭い部分の幅, および, 図形の最も広い部分の幅を示している. また, T_s は $0 < T_s < 1$ の範囲のスケールレートを示す. この制約は, タイル \mathbf{U} の最も狭い部分 $w_n(\mathbf{U})$ が, 目標図形 \mathbf{W} の最も狭い部分 $w_n(\mathbf{W})$ より小さくならないように維持するものである. しかし, 目標図形 \mathbf{W} には元々狭い部分が含まれる場合 (動物の尾など) があるため, この場合は, 目標図形 \mathbf{W} の最も広い部分 $w_w(\mathbf{W})$ の一定の割合と $w_n(\mathbf{U})$ を比較する. 一定の割合を表す T_s は, お菓子の材料や作成するお菓子型のサイズに応じて調整する. 上記の制約条件を違反した場合は, 目的関数にペナルティ値 $\rho(\mathbf{U})$ を加算する.

制約条件に用いる $w_n(\cdot)$ と $w_w(\cdot)$ の計算方法は, 図 5.2(a) に示すように, 図形をシルエット画像に変換後, モルフォロジー画像処理 [86] の収縮演算を繰り返し適用する. $w_n(\cdot)$ は, 図 5.2(b) に示すように, 対象のシルエット画像が複数の領域に分割されるか, または, 対象のシルエット画像の凸包の面積が急激に変化するまで収縮演算を適用した回数として定義する. $w_w(\cdot)$ は, 対象のシルエット画像が消えるまで収縮演算を適用した回数として定義する.



(a) 図形の凸包

(b) モルフォロジー画像処理の収縮演算を適用

図 5.2. $w_n(\cdot)$ と $w_w(\cdot)$ の算出方法

5.3 評価実験

5.3.1 実験準備

新たに導入した製菓特有の制約条件の有効性を検証するために、ELTHON（第3章を参照）に新たな制約条件を導入した手法と、導入していない手法との比較を行った。

ELTHON は、一様交差、ビット反転による突然変異、MGG 世代交代モデル [64] を用いた。停止条件、個体集団サイズ、交叉率、突然変異率は、50,000, 50, 0.4, 0.03 とした。新たな制約条件のパラメータであるスケールレート T_s 、ペナルティ値 $\rho(\mathbf{U})$ はそれぞれ、0.1, 1.0×10^4 に設定した。図 5.3 に評価実験に用いた目標図形を示す。また、目標図形の頂点数 N は 180 点として、各目標図形 \mathbf{W} に 5 試行実行した。

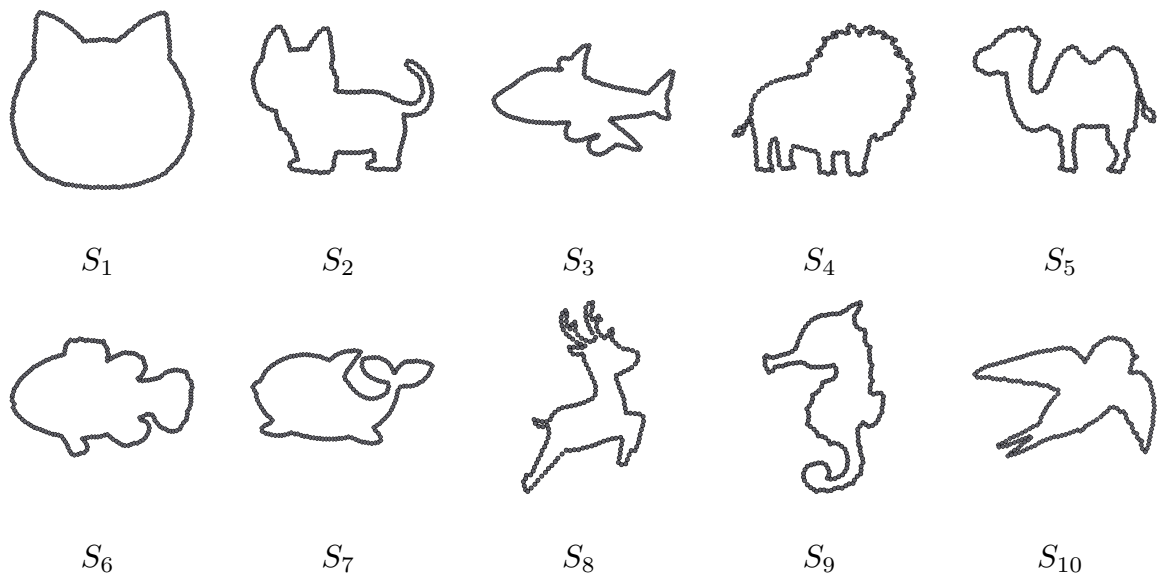


図 5.3. 目標図形 \mathbf{W}

5.3.2 実験結果

図 5.4 から図 5.6 は、評価した 2 つの手法から得られた最良解について比較した結果を示す。グラフの横軸は生成されたタイルの最も狭い部分の幅 $w_n(\mathbf{U})$ を示し、縦軸は適応度を示す。各グラフは、得られた 5 つの最適解を青丸（制約条件を導入していない手法）,

赤矩形（制約条件を導入した手法）で示し、それらのタイル形状を表している。

これらのグラフから、制約条件を導入した手法は、制約条件を導入していない手法と比較して、タイルの最も狭い部分の幅 $w_n(\mathbf{U})$ を広く維持することがわかった。また、10 図形中 4 図形では、新たな制約条件を導入した手法と、導入していない手法で生成したタイルの $w_n(\mathbf{U})$ の平均値に有意差があり、t 検定の結果、 S_1, S_2, S_8, S_9 の p 値はそれぞれ、0.0298, 0.0204, 9.46×10^{-5} , 2.44×10^{-5} であった。一方、 S_4 の結果を見ると、 $w_n(\cdot)$ の計算に不備があることがわかった。 S_4 の動物の尾の幅は、尾の形状によって $w_n(\mathbf{U})$ の計算が正しくされず、一定の幅を維持できていないことがわかる。

次に、目標図形 S_8 について、制約条件を導入した手法により得られたタイル形状に基づいて 3D プリンターを用いてお菓子型を作成して、このお菓子型を用いてパンを作成した。図 5.7 は、お菓子型と焼き上げたパンを示している。

5.4 おわりに

本章では、第 3 章にて提案した ELTHON に、製菓特有の制約条件を導入することで出力されるタイルの形状に壊れやすい部分の発生を抑制し、製菓に適したタイルを生成できることを示した。本手法は、ELTHON の利点である目的関数や制約条件の柔軟性が高いことを活かし、上位最適化層に製菓特有の新たな制約条件を追加することで、タイルに狭くて脆い部分を生成しないようにした。実験結果より、制約条件を導入した手法は、狭い部分であっても一定の幅を維持したままタイルを生成することを確認した。

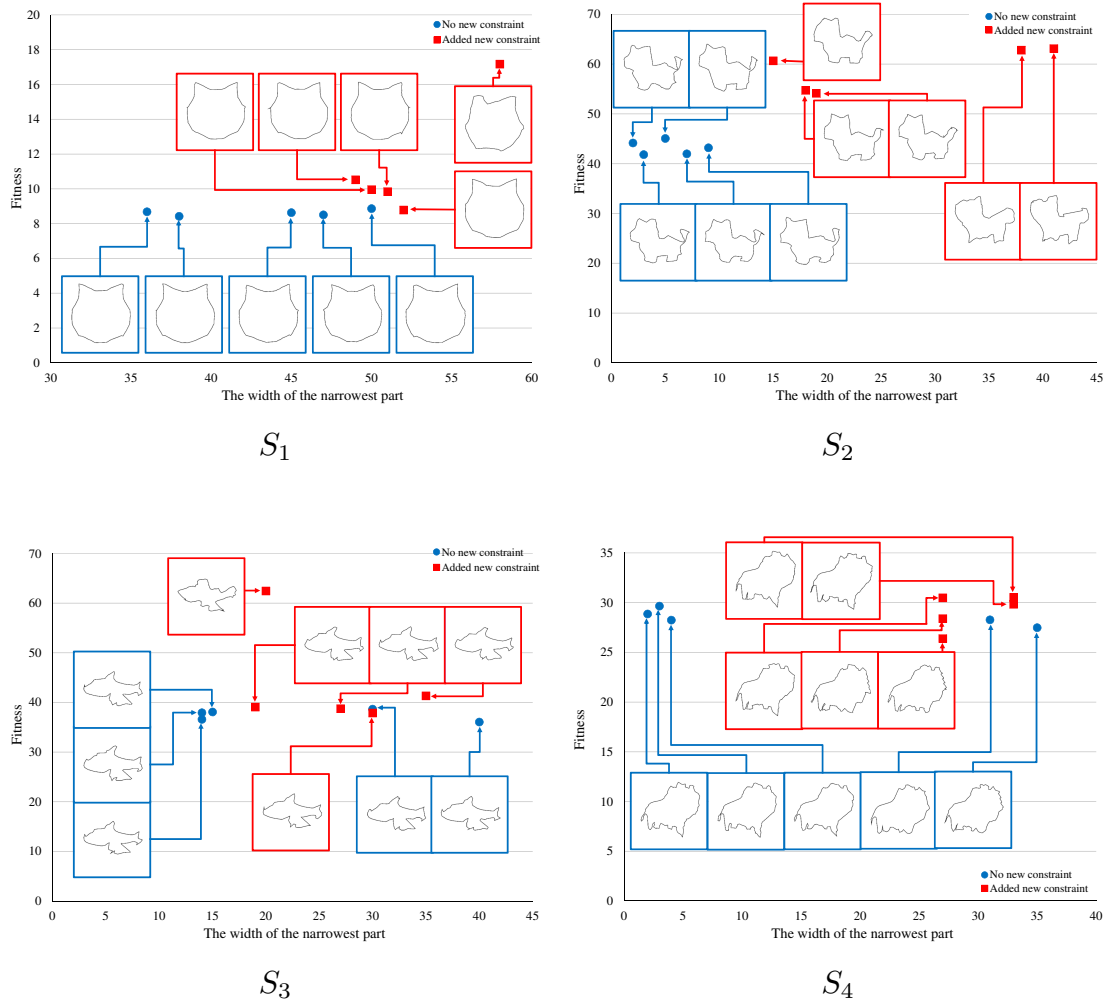
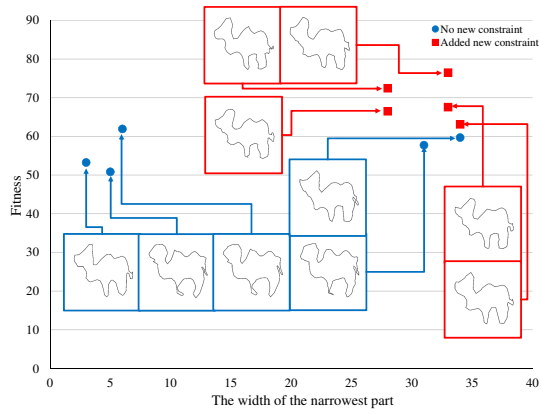
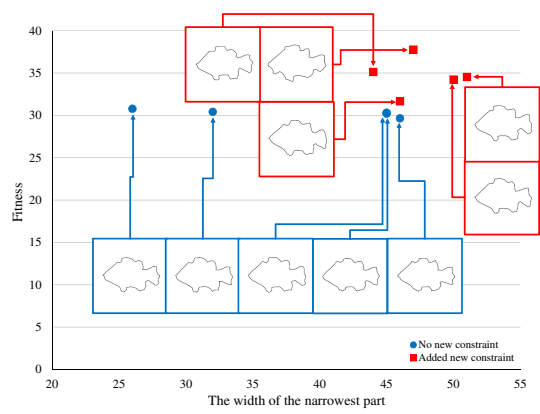


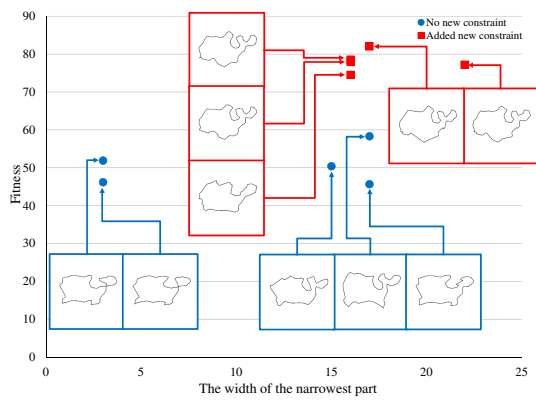
図 5.4. $w_n(\mathbf{U})$ と適応度に関する比較 ($S_1 - S_4$)



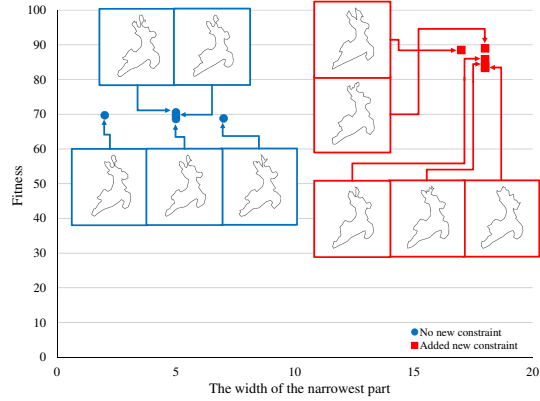
S_5



S_6



S_7



S_8

図 5.5. $w_n(\mathbf{U})$ と適応度に関する比較 ($S_5 - S_8$)

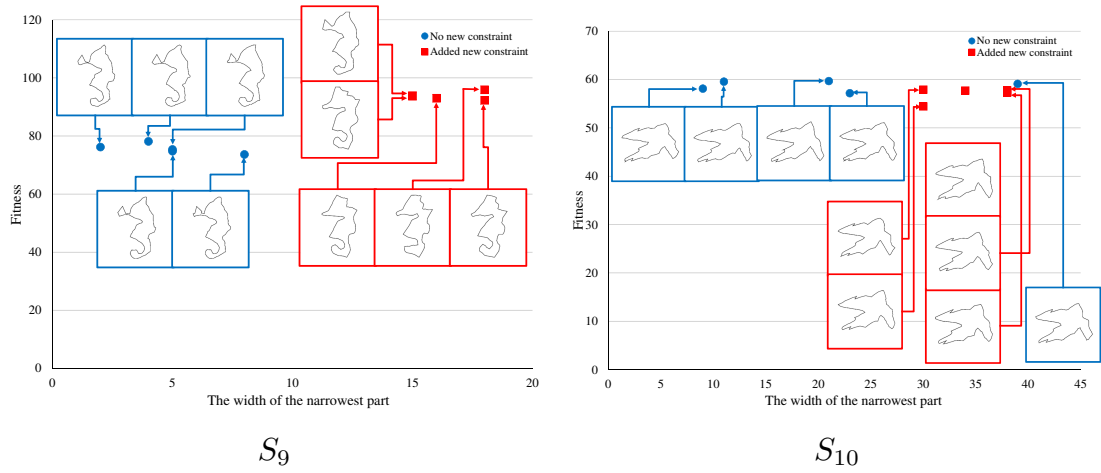


図 5.6. $w_n(\mathbf{U})$ と適応度に関する比較 ($S_9 - S_{10}$)



図 5.7. S_8 のタイルのお菓子型

第 6 章

処理速度の高速化の試み

本章では、ELTHON の処理速度の高速化を試みる。第 3 章および第 4 章では、アルゴリズムの挙動の解釈の容易性、および、生成される解の多様性を考慮して単純なメタヒューリスティクスを使用して評価実験を行った。このため、より優れた収束特性を持つメタヒューリスティクスを用いて高速化を試みる。また本章では、第 3 章にて提案した ELTHON を $\text{ELTHON}^{\text{GA}}$ と呼び、本章にて提案する手法を $\text{ELTHON}^{\text{EDA}}$ と呼ぶ。実験を行い、将来的に導入を検討している対話型最適化の実現に寄与するか検証を行い、 $\text{ELTHON}^{\text{GA}}$ より収束速度が向上したことを確認した。

まず、6.1 節では、処理速度の高速化を行う理由について述べる。次に、6.2 節では、処理手順や目的関数など、 $\text{ELTHON}^{\text{EDA}}$ の詳細を示す。6.3 節では、実験により $\text{ELTHON}^{\text{EDA}}$ の有効性を $\text{ELTHON}^{\text{GA}}$ と比較して示す。6.4 節では、本章の結論を述べる。

6.1 はじめに

第 3 章では、階層型最適化方式を具体化し、エッシャー化問題を解く手法 ($\text{ELTHON}^{\text{GA}}$) を提案した。 $\text{ELTHON}^{\text{GA}}$ は自己交差を含まず目標図形に類似するタイルを生成できるが、上位最適化層は単純 GA [48] を用いているため、計算コストが高いという問題点がある。

本章では、上位最適化層で用いていた単純な GA を分散推定アルゴリズム (Estimation of Distribution Algorithm:EDA) [65,66] に置き換えることで、収束速度の高速化を目指す $\text{ELTHON}^{\text{EDA}}$ を提案する。

本章で収束速度の高速化を目指す理由は、将来的に対話型評価の導入を検討しているためである。対話型評価を導入する場合、評価過程以外の処理時間が長いとユーザの負担が大きくなるため、評価過程以外の処理時間の短縮を目指す必要がある。また、上位最適化層を変更する理由は、下位最適化層の解析的最適化手法 [18, 19] を高速化するのは容易でないが、上位最適化層は単純 GA より収束速度が高速な最適化手法に置き換えるのみのため変更が容易である。また、解析的最適化手法は目的関数 F_L (式 2.3) が単純な凸関数であると仮定し、解析的に解を導出しているため、高速に解を求めることが可能である。この目的関数 F_L を単純な凸関数に保ったまま、新たな目的関数や制約条件を導入することは容易ではないため、解析的最適化手法の高速化は容易でない。

6.2 アルゴリズムの構成

6.2.1 基本的なアイデア

本章では、ELTHON^{GA} の改良を行い、収束速度の高速化を目指す。本章の提案手法 (ELTHON^{EDA}) は、基本的な枠組みは ELTHON^{GA} と同じであり (詳細は第 3 章を参照)、エッシャー化問題の難しさを分解して解を探索しているが、上位最適化層に用いている単純 GA を EDA に置き換えることで、ELTHON^{GA} を改良している。

EDA [65, 66] (2.4.3.3 項を参照) は、一般的な進化計算アルゴリズムと異なり、設計変数ではなく設計変数の確率分布パラメータを最適化し、この確率分布に基づいて個体を生成している。

6.2.2 処理手順

ELTHON^{EDA} は 2 つの最適化手法が階層的な関係になるように構成しており、図 6.1 にフローチャートを示す。

上位最適化層の EDA は、確率分布に従って μ 個の個体 (解候補) をサンプリングしている。ここで、EDA は、最適なタイリングパターンの選択、および、最適な目標図形 \mathbf{W} からダウンサンプリングされた形状 $\tilde{\mathbf{W}}$ を形成する頂点を選択する (3.3.2.2 項を参照)。つまり、ELTHON^{EDA} と ELTHON^{GA} の個体表現は同じである。そして、EDA がサンプリングした各個体に対して、毎世代、下位最適化層の解析的最適化手法を適用し、エッ

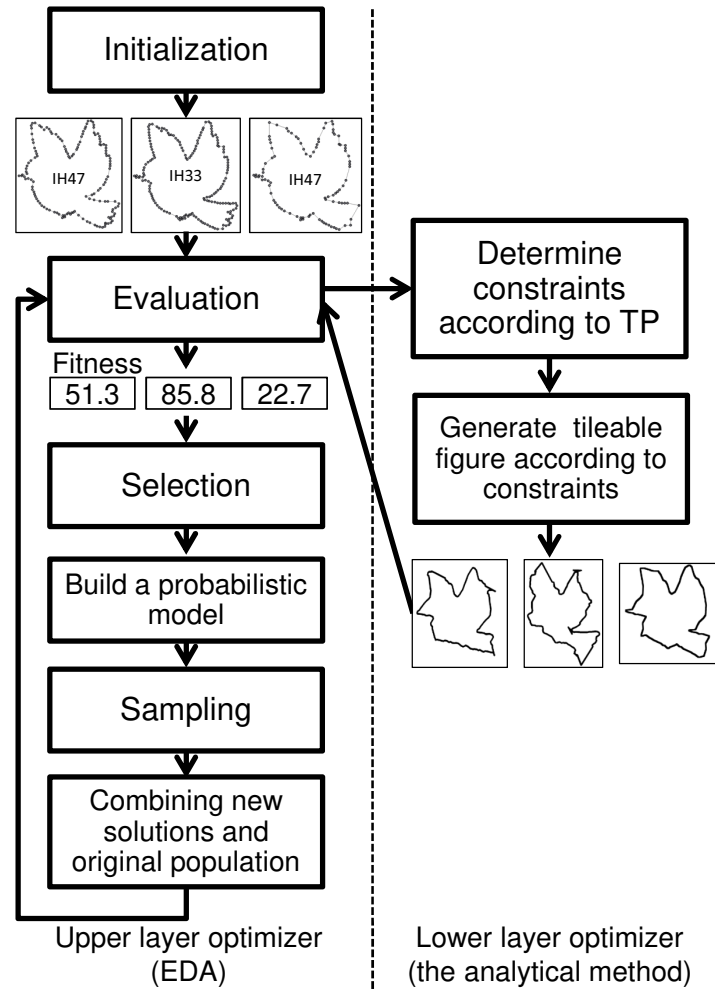


図 6.1. ELTHON^{EDA} の処理手順

シャー化問題の条件 2 (タイルの対応する 2 辺の形状が一致する制約) を満たすように \tilde{W} からタイル U に変形する。解析的最適化手法は、EDA により生成された \tilde{W} に可能な限り類似する U を導出する。下位最適化層の解析的最適化手法が変形した U は、上位最適化層の EDA により評価される。つまり、EDA は U の自己交差の制約違反の確認を行い、適応度を計算し、確率分布パラメータを更新する。ここで、目的関数には、双方向マッピングベース関数を用いる (3.3.2.4 (a) 項を参照)。また、EDA は、解が収束した場合、確率分布を初期化する。

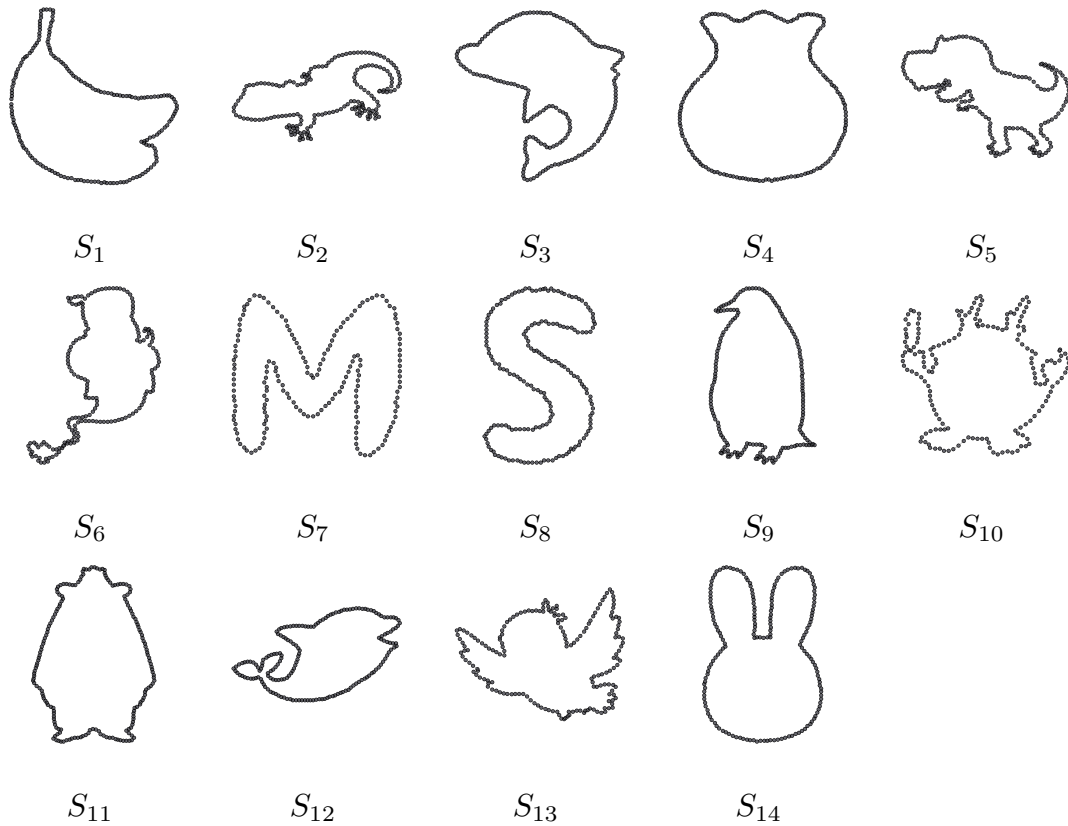


図 6.2. 目標図形 W

6.3 評価実験

6.3.1 実験準備

ELTHON^{EDA} の有効性を検証するために、14 個の目標図形を用いて、ELTHON^{GA} (第 3 章を参照) との比較を行った。

ELTHON^{EDA} の、個体集団サイズ、サンプリングサイズはそれぞれ、50, 25 に設定した。また、ELTHON^{EDA} には、単純な EDA である PBIL [65, 66] (2.4.3.3(a) 項参照) を用いる。ELTHON^{GA} は、一様交差、ビット反転による突然変異、MGG 世代交代モデル [64] を用いた。停止条件、個体集団サイズ、交叉率、突然変異率は、50,000, 50, 0.5, 0.03 とした。図 6.2 に評価実験に用いた目標図形を示す。また、目標図形の頂点数 N は 180 点とする。

6.3.2 実験結果

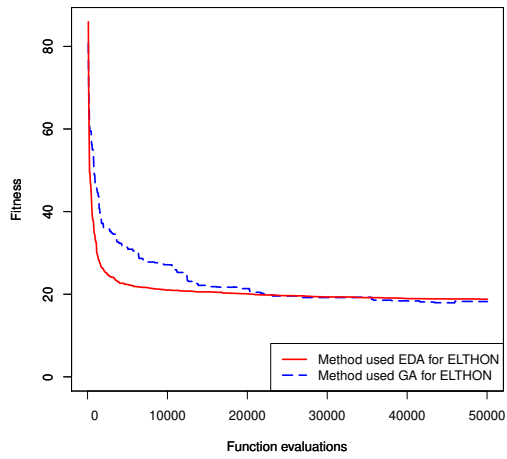
図 6.3 から図 6.6 は，個体集団中でもっとも優れた個体の適応度の遷移について比較した結果である．横軸は評価回数，縦軸は適応度を示す．各グラフは，各目標図形に対して 5 試行を行い，適応度の平均値をプロットした結果である．

図 6.3 から図 6.6 より， $\text{ELTHON}^{\text{EDA}}$ は $\text{ELTHON}^{\text{GA}}$ に比べて計算時間が短縮していることがわかる．最良解の適応度が発見された世代数の平均値を比較すると， $\text{ELTHON}^{\text{EDA}}$ は $\text{ELTHON}^{\text{GA}}$ に比べて，0.85 倍の世代数で最良解を発見している．また全目標図形において， $\text{ELTHON}^{\text{EDA}}$ は $\text{ELTHON}^{\text{GA}}$ より，少ない評価回数で収束していることがわかる．しかし， $\text{ELTHON}^{\text{GA}}$ は収束していない結果や， $\text{ELTHON}^{\text{GA}}$ により探索した最良解の適応度が $\text{ELTHON}^{\text{EDA}}$ により探索した最良解の適応度より優れている結果が存在する．ただし，本章の目的は収束速度の向上のため，実験条件における収束速度に評価基準を置く．

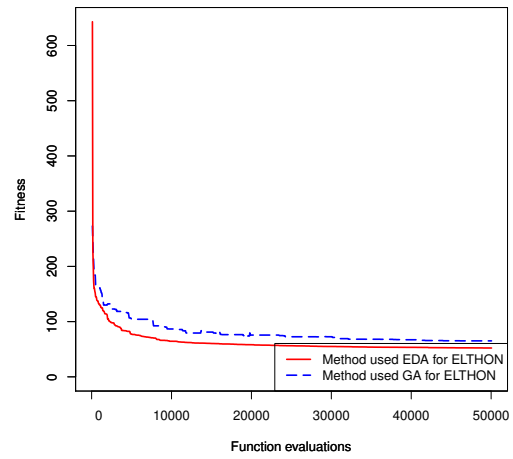
実験結果より，処理速度の高速化が行えたことが確認できた．単純 GA を用いた $\text{ELTHON}^{\text{GA}}$ と比較して，EDA を用いた $\text{ELTHON}^{\text{EDA}}$ は，収束に要する時間を 85% に短縮した． $\text{ELTHON}^{\text{EDA}}$ は，対話型評価を導入する際， $\text{ELTHON}^{\text{GA}}$ よりユーザの負担を軽減することが可能であると考ええる．

6.4 おわりに

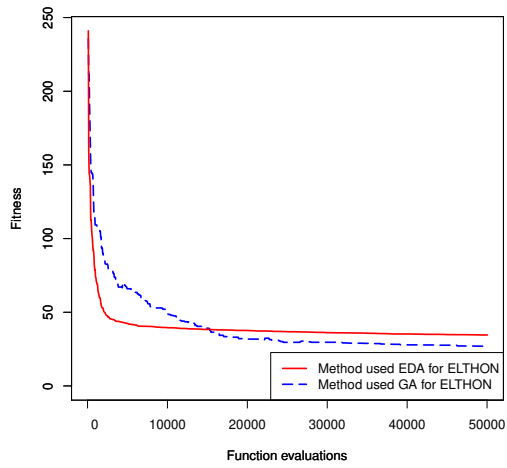
本章では，第 3 章にて提案した $\text{ELTHON}^{\text{GA}}$ を改良する手法 ($\text{ELTHON}^{\text{EDA}}$) を提案した． $\text{ELTHON}^{\text{EDA}}$ では，上位最適化層に用いる最適化手法を単純 GA から EDA に置き換えた．実験結果より， $\text{ELTHON}^{\text{EDA}}$ は $\text{ELTHON}^{\text{GA}}$ より収束速度が向上することが確認できた．



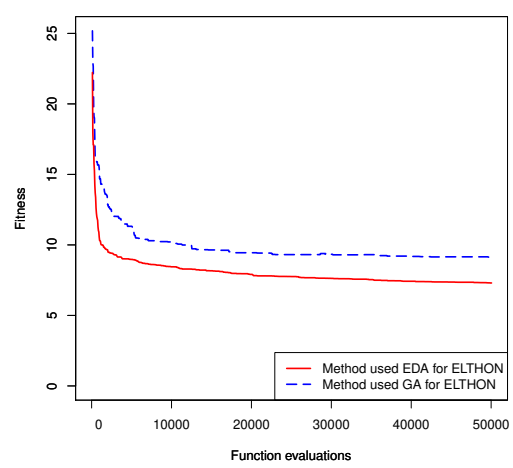
S_1



S_2

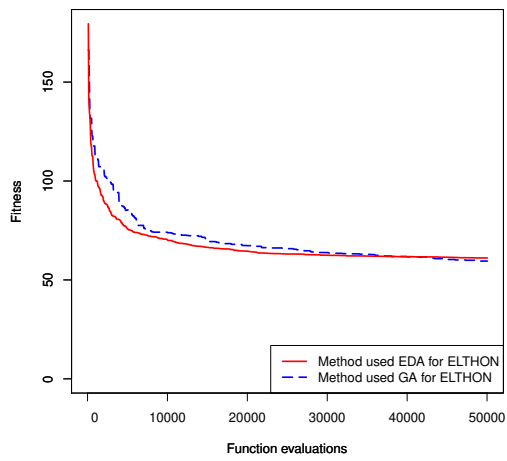


S_3

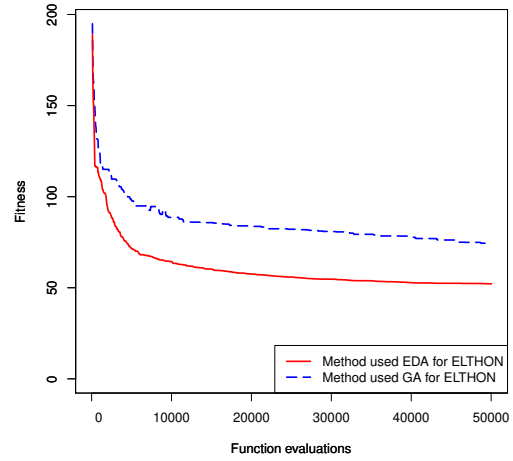


S_4

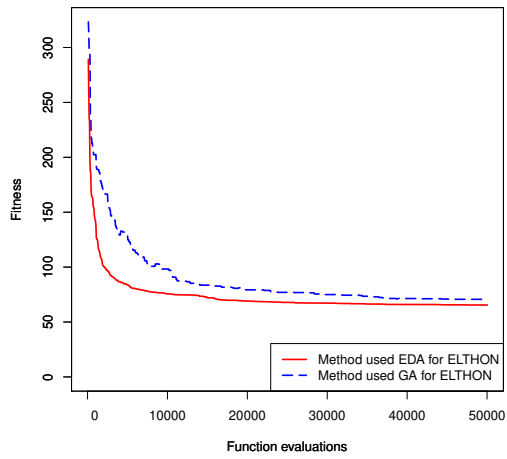
図 6.3. $ELTHON^{GA}$ と $ELTHON^{EDA}$ の収束速度の比較 ($S_1 - S_4$)



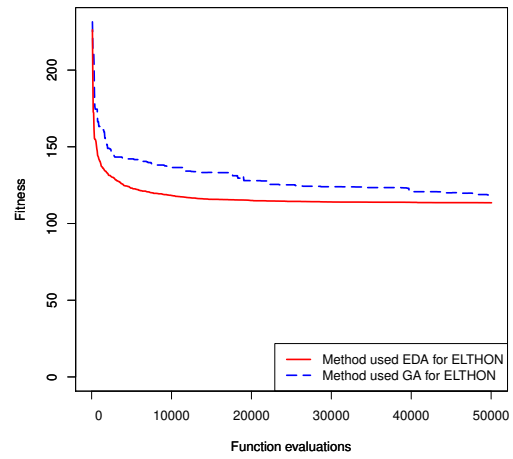
S_5



S_6

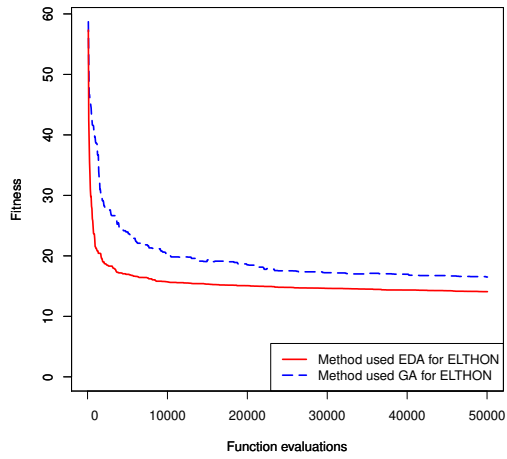


S_7

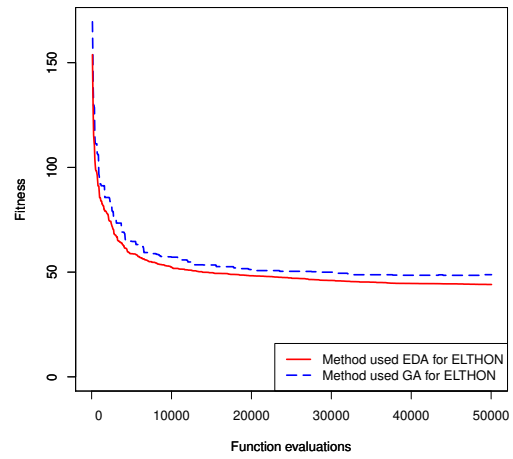


S_8

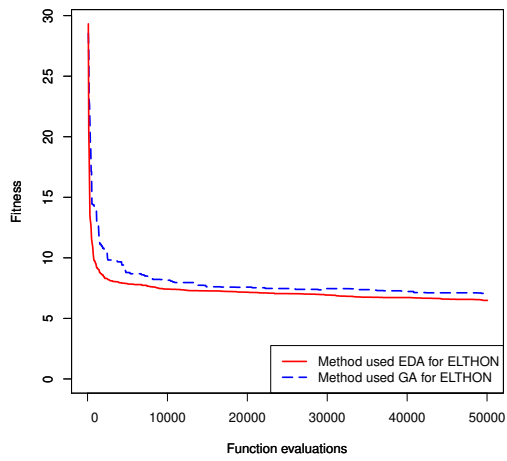
図 6.4. ELTHON^{GA} と ELTHON^{EDA} の収束速度の比較 ($S_5 - S_8$)



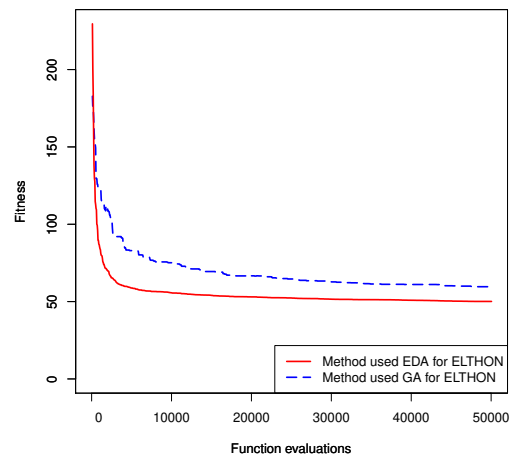
S_9



S_{10}



S_{11}



S_{12}

図 6.5. ELTHON^{GA} と ELTHON^{EDA} の収束速度の比較 ($S_9 - S_{12}$)

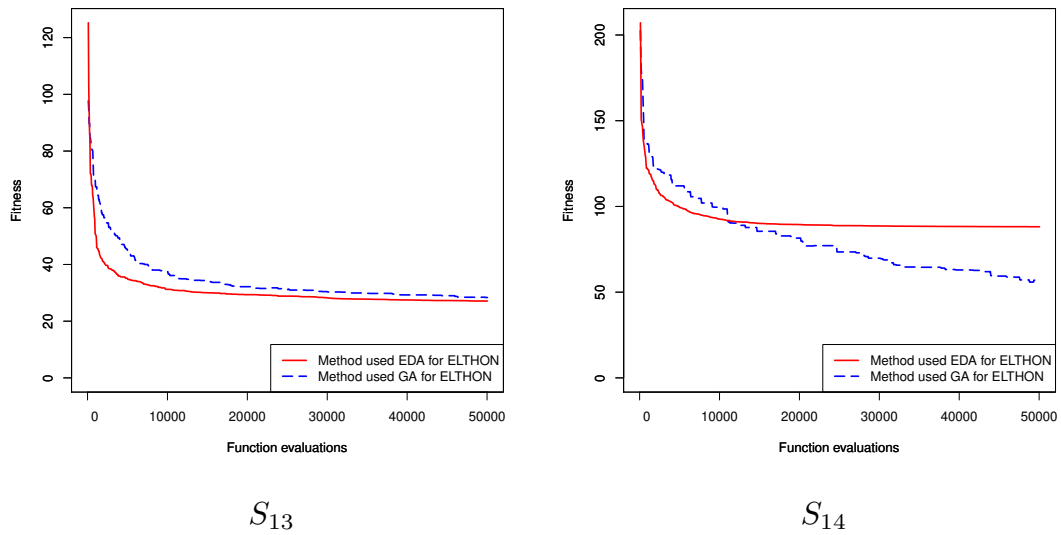


図 6.6. ELTHON^{GA} と ELTHON^{EDA} の収束速度の比較 ($S_{13} - S_{14}$)

第7章

結論

7.1 研究の結論

本論文は、求解を困難にする複数の要因を含む最適化問題において、要因を分けるように対象問題を階層的に表現し、分割した部分問題に対して異なる解法を適用する階層型最適化方式を提案した。本方式は、各部分問題を解く方式として、問題の特性に依存しない大域的最適化を行うメタヒューリスティクス、問題依存の効率的な局所探索アルゴリズムや、機械学習に基づく生成モデルなど、多様なアルゴリズムを利用できる点に特徴がある。

本論文では、上記のような複数の難しさを含む最適化問題の例として、タイリング（平面充填）が可能な図形（タイル）の生成問題 [16, 17] に着目した。本問題は、与えられた目標図形をタイルに変形する難しさ、図形間の類似度を定量化する難しさ、さらに、目標図形自体を用意することの難しさなどを含んでいる。それらの難しさを最適化の観点から捉えると、それぞれ、制約条件の強さ、目的関数の曖昧さ、および、探索空間の膨大さに相当する。このような求解を困難にする要因を分解し、タイル生成問題を2つまたは3つの部分問題に分割することで、与えられた目標図形に類似するタイルを生成する手法（ELTHON）、および、映像をもとに対象オブジェクトの目標図形およびタイルを生成する手法（TIMHON）の2つの手法を提案した。実験結果より、ELTHONは先行研究 [17–26] と同じ問題設定において、より安定的に良好なタイルを生成できることを確認した。また、TIMHONは、先行研究 [17–26] とは異なり目標図形を明示的に与えない場

第7章 結論

合であっても、映像からタイルに適した目標図形を補間的に生成できることを確認した。

第1章では、本論文の対象問題について説明し、その一例であるエッシャー化問題 [16, 17] を解決する階層型最適化方式を提案した。また、本方式と最適化問題における一般的なハイブリッド方式 [13–15] との相違点について述べた。

第2章では、関連研究について述べた。まず、対象問題の定義について述べ、先行研究、および、本研究で用いる最適化および機械学習の手法について説明した。先行研究では、エッシャー化問題を解析的に解く手法 [18–24]、および、メタヒューリスティクスを用いて解決する手法 [17, 25, 26] について説明した。

第3章では、階層型最適化方式を具体化し、エッシャー化問題に適用した手法である、階層型最適化方式に基づくエッシャー風タイルを生成する手法 (ELTHON) を提案した。ELTHON は、先行研究 [17–26] と同様の定式化のもとでエッシャー化問題を解く試みである。ELTHON はエッシャー化問題を二つの階層で表現し、上位最適化層にはメタヒューリスティクス、下位最適化層にはタイルの対応する2辺の形状が一致する制約条件を満たす図形を解析的に導出する先行研究 (解析的最適化手法) [18, 19] を用いた。上位最適化層は下位最適化層への適切な入力を探査し、下位最適化層は固有値問題を解くことでタイルを生成する。すなわち、上位最適化層は解析的最適化手法で必要とされる試行錯誤処理を代替するものである。実験結果より、ELTHON で生成したタイルは、解析的最適化手法の手作業を省きつつ、与えられた目標図形形状に近似することを確認した。また、3つの目的関数の性能を比較検討し、その有効性は与えられた目標図形形状に依存することを確認した。

第4章では、先行研究 [17–26] とは異なる問題設定とし、タイリングに適した目標図形を生成する問題を解決した。まず、タイルを生成する問題において、入力される目標図形は人間により描画されることを前提としていたが、本問題は、タイリングが可能な図形に変形が困難な図形、および、解は存在するが最適な解を発見することが難しい図形が存在することを議論した。このため、第3章で提案した ELTHON や先行研究 [17–26] では、入力する目標図形を試行錯誤的に描画するなど、タイリングに適した目標図形を用意することが困難であるという問題が存在する。このため、本章では、映像をもとにタイリングに適した対象オブジェクトの画像を生成し、タイルに変形する手法 (TIMHON) を提案した。TIMHON は三階層で構成され、上位最適化層において、対象オブジェクトの画像

をパラメトリックに生成するモデルを構築し、潜在変数の最適化によりタイリングに適した目標図形を生成した。中間・下位最適化層には第3章で提案した方式を用いた。具体的には、ConvVAE [72] を用いてキャラクターの特徴量空間を学習し、タイリングに適したキャラクターのポーズを探索して、映像に含まれていない新たなポーズを生成することができる。これにより、目標図形の描画の試行錯誤が不要となることが期待できる。実験結果より、TIMHON は、タイリングに適したさまざまなポーズのキャラクター画像を探索することに成功し、訓練データに含まれないポーズも生成することを確認した。

第5章では、提案手法が多様な産業に応用可能であることを示す一つの事例として、製菓への応用を試みた。本章では、第3章で提案した ELTHON に製菓特有の制約条件を導入した手法を提案した。本手法は、ELTHON の利点である目的関数や制約条件の柔軟性が高いことを活かし、上位最適化層に製菓特有の新たな制約条件を追加することで、タイルに狭くて脆い部分を生成しないようにした。実験結果より、制約条件を導入した手法は、狭い部分であっても一定の幅を維持したままタイルを生成することを確認した。

第6章では、ELTHON の処理速度の高速化についての検討を行った。第3章、第4章で提案した手法は、アルゴリズムの挙動の解釈の容易性、および、生成される解の多様性を考慮して単純なメタヒューリスティクスを使用している。このため、より優れた収束特性を持つメタヒューリスティクスを利用した際の高速化の度合いを検証する。提案した手法では、上位最適化層に用いる最適化手法を単純 GA [48] から EDA [65,66] に置き換えた。実験結果より、提案した手法は ELTHON より収束速度が向上することが確認できた。

7.2 今後の展望

本研究では、提案する階層型最適化方式の有効性をタイル生成問題 [16,17] においてのみ検証を行っており、他の問題における有効性を検証することは、今後の重要な課題である。種類の異なる複数の難しさを含む最適化問題として、タイル生成問題 [16,17] 以外に、不可能図形生成問題 [87] やスモッキングの展開図生成問題 [88] があり、これらの問題に階層型最適化方式を適用する。多様な問題に階層型最適化方式を応用することにより、階層型最適化方式の適用範囲の広さを示すことは重要である。

第7章 結論

また、タイル生成問題を対象とした階層型最適化方式における今後の課題は、生成されるタイルの評価方法の改善が必要であり、目的関数の改良、および、対話評価の導入の2つのアプローチが考えられる。前者に関して、本論文で用いた目的関数のみでは、人間の主観を完全には表現できないことから、対象オブジェクトに応じて関数を適応的に選択するなどの技術の導入を検討する。一方で、図形間の類似性を正確に評価できる万能な関数の発見は期待できないことから、人間による対話評価の導入は重要である。

謝辞

本研究を進めるにあたり，日頃から熱心にご指導いただいた小野智司教授に，深く感謝申し上げます．専門的な知識以外にも多くの知識や考え方を教えていただきました．心より感謝申し上げます．

拓殖大学の水野一徳教授には，定期的な研究に関するミーティングや毎年鹿児島までお越しいただき，多くのご指導をいただきました．また学会では，発表内容や発表姿勢など，詳しくご教授くださいました．心より感謝申し上げます．

本論文をまとめるにあたり，ご指導やご審査，および，適切なお助言をいただきました大塚作一教授と淵田孝康准教授に心から感謝いたします．

小野研究室の先輩，同輩，後輩の皆様には，さまざまな場面においてご支援いただきました．日々の他愛無い会話から，研究の議論など，皆様の存在なしにはこのように充実した研究生活を送れませんでした．ありがとうございました．

最後に，両親に深く深く感謝申し上げます．博士後期課程まで進学させていただき，多くの経験を通し，大きく成長することができました．本当にありがとうございます．

参考文献

- [1] 寒野善博. 最適化手法入門 データサイエンス入門シリーズ. 講談社, 2019.
- [2] 三宮信夫. 遺伝アルゴリズムと最適化. 朝倉書店, 1998.
- [3] Sigurd A Nelson, Matthew B Parkinson, and Panos Y Papalambros. Multicriteria optimization in product platform design. *J. Mech. Des.*, Vol. 123, No. 2, pp. 199–204, 2001.
- [4] Yash P Aneja and Kunhiraman PK Nair. Bicriteria transportation problem. *Management Science*, Vol. 25, No. 1, pp. 73–78, 1979.
- [5] Peter C Cheeseman, Bob Kanefsky, and William M Taylor. Where the really hard problems are. In *IJCAI*, Vol. 91, pp. 331–340, 1991.
- [6] Kazunori Mizuno and Seiichi Nishihara. Constructive generation of very hard 3-colorability instances. *Discrete Applied Mathematics*, Vol. 156, No. 2, pp. 218–229, 2008.
- [7] 英行高木, 達夫畝見, 隆雄寺野. 対話型進化計算法の研究動向 (j論文特集j対話型進化計算法). *人工知能学会誌*, Vol. 13, No. 5, pp. 692–703, sep 1998.
- [8] Yan Pei and Hideyuki Takagi. Research progress survey on interactive evolutionary computation. *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2018.
- [9] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, Vol. 27, No. 8, pp. 3998–4011, 2018.
- [10] 田中俊二. 組合せ最適化問題とスケジューリング. *システム/制御/情報*, Vol. 64, No. 6, pp. 200–206, 2020.
- [11] 棟朝雅晴. 遺伝的アルゴリズム: その理論と先端的手法. 森北出版, 2008.

- [12] Ankur A. Patel 著, 中田秀基訳. Python ではじめる教師なし学習——機械学習の可能性を広げるラベルなしデータの利用. O'Reilly Japan, Inc., 2020.
- [13] Pablo Moscato, Fernando Tinetti, Cetad Centro, and La Plata. Blending heuristics with a population-based approach: A "memetic" algorithm for the traveling salesman problem. *Universidad Nacional de La Plata*, pp. 92–12, 1992.
- [14] Nicholas J. Radcliffe and Patrick D. Surry. Formal memetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing*, pp. 1–16, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.
- [15] P Moscato. Memetic algorithms: a short introduction. *New ideas in optimization*, pp. 219–234, 1999.
- [16] Craig S Kaplan. Introductory tiling theory for computer graphics. *Synthesis Lectures on Computer Graphics and Animation*, Vol. 4, No. 1, pp. 1–113, 2009.
- [17] Craig S. Kaplan and David H. Salesin. Escherization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, p. 499–510. ACM Press/Addison-Wesley Publishing Co., 2000.
- [18] Hiroshi Koizumi and Kokichi Sugihara. Computer-aided design of escher-like tiling. *NICOGRAPH Paper Contest*, 2009.
- [19] Hiroshi Koizumi and Kokichi Sugihara. Maximum eigenvalue problem for escherization. *Graphs and Combinatorics*, Vol. 27, No. 3, p. 431, 2011.
- [20] S Imahori and S Sakai. A local-search based algorithm for the escherization problem. In *The IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 151–155, 2012.
- [21] Shinji Imahori, Shizuka Kawade, and Yoko Yamakata. *Escher-like Tilings with Weights*, pp. 132–142. Springer International Publishing, Cham, 2016.
- [22] 永田裕一, 今堀慎治. 一般化尺度を用いたエッシャー風タイリング問題の網羅的解法. 進化計算シンポジウム, 2017.
- [23] Yuichi Nagata. Escherization with a distance function focusing on the similarity of local structure. In *Parallel Problem Solving from Nature – PPSN XV*, pp. 108–120. Springer International Publishing, 2018.

-
- [24] Yuichi Nagata and Shinji Imahori. An efficient exhaustive search algorithm for the escherization problem. *Algorithmica*, pp. 1–33, 2020.
- [25] Satoshi Ono, Megumi Kisanuki, Hirofumi Machii, and Kazunori Mizuno. Creation support for escher-like tiling patterns by interactive genetic algorithms. In *SIGGRAPH Asia 2014 Posters, SA ' 14*. Association for Computing Machinery, 2014.
- [26] S. Ono, M. Kisanuki, H. Machii, and K. Mizuno. Figure pattern creation support for escher-like tiling by interactive genetic algorithms. Vol. 1 of *Proceedings in Adaptation, Learning and Optimization, 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 421–432, 2014.
- [27] Max Welling Diederik P. Kingma. Auto-encoding variational bayes. *ICLR*, 2014.
- [28] Hideyuki Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, Vol. 89, No. 9, pp. 1275–1296, 2001.
- [29] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, Vol. 295, pp. 407–428, 2015.
- [30] Kunihiro Hoki and Tomoyuki Kaneko. Large-scale optimization for evaluation functions with minimax search. *Journal of Artificial Intelligence Research*, Vol. 49, pp. 527–568, 2014.
- [31] L Darrell Whitley. Fundamental principles of deception in genetic search. In *Foundations of genetic algorithms*, Vol. 1, pp. 221–241. Elsevier, 1991.
- [32] Yew S Ong, Prasanth B Nair, and Andrew J Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA journal*, Vol. 41, No. 4, pp. 687–696, 2003.
- [33] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and multidisciplinary optimization*, Vol. 41, No. 2, pp. 219–241, 2010.

- [34] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34. Springer, 2011.
- [35] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, Vol. 5. Springer, 2007.
- [36] 佐藤寛之, 石渕久生. 進化型多数目的最適化の現状と課題. オペレーションズ・リサーチ: 経営の科学, Vol. 62, No. 3, pp. 156–163, 2017.
- [37] Brenda Cheang, Haibing Li, Andrew Lim, and Brian Rodrigues. Nurse rostering problems—a bibliographic survey. *European journal of operational research*, Vol. 151, No. 3, pp. 447–460, 2003.
- [38] 田村直之, 丹生智也, 番原睦則. 制約最適化問題と sat 符号化 (i 特集j 最近の sat 技術の発展). 人工知能, Vol. 25, No. 1, pp. 77–85, 2010.
- [39] 永井保夫ほか. 制約充足問題への近似解法の適用検討-n-クイーン問題を中心として. 東京情報大学研究論集, Vol. 8, No. 2, pp. 59–66, 2005.
- [40] 大山聖, 大林茂, 中橋和博, 中村孝. 遺伝的アルゴリズムを用いた三次元翼の空力最適化. 日本航空宇宙学会誌, Vol. 46, No. 539, pp. 682–686, 1998.
- [41] 青木研, 高木英行. 対話型 ga による 3 次元 cg ライティングデザイン支援. 電子情報通信学会論文誌 D, Vol. 81, No. 7, pp. 1601–1608, 1998.
- [42] Driss Halhal, Godfrey A Walters, Driss Ouazar, and DA Savic. Water network rehabilitation with structured messy genetic algorithm. *Journal of water resources planning and management*, Vol. 123, No. 3, pp. 137–146, 1997.
- [43] Leandro Vitor Pavão, Caliane Bastos Borba Costa, Mauro Antonio da Silva Sá Ravagnani, and Laureano Jiménez. Large-scale heat exchanger networks synthesis using simulated annealing and the novel rocket fireworks optimization. *AIChE Journal*, Vol. 63, No. 5, pp. 1582–1601, 2017.
- [44] 坂上啓. N700 系新幹線電車とその省エネルギー効果について. 精密工学会誌, Vol. 76, No. 1, pp. 41–45, 2010.
- [45] 巖原田. 新幹線車両の開発設計: 空気力学と cfd(i フォーラム小特集j 応用数理の最前線). 応用数理, Vol. 13, No. 3, pp. 220–225, 2003.

- [46] 規雄枇々木. ポートフォリオ最適化入門 (特集 はじめよう金融工学). オペレーションズ・リサーチ, Vol. 61, No. 6, pp. 335–340, jun 2016.
- [47] 若井亮介, 新里隆, 嶋崎善章. ランダム行列を用いたポートフォリオ最適化解析. 日本経営工学会論文誌, Vol. 65, No. 1, pp. 17–28, 2014.
- [48] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989.
- [49] Mauritz Cornelis Escher. 無限を求めて—エッシャー、自作を語る. 朝日選書, 1994.
- [50] Chi-Han Peng, Helmut Pottmann, and Peter Wonka. Designing patterns using triangle-quad hybrid meshes. *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, pp. 1–14, 2018.
- [51] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman, New York, 1987.
- [52] Teruhisa Sugimoto and Yoshiaki Araki. Convex pentagon tilings and heptiamonds, i. *arXiv preprint arXiv:1704.03997*, 2017.
- [53] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, Vol. 220, No. 4598, pp. 671–680, 1983.
- [54] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of combinatorial optimization*, pp. 2093–2229. Springer, 1998.
- [55] Danushka Bollegala 伊庭 齐志. 東京大学工学教程 知識情報処理. 丸善出版, 2016.
- [56] Janez Brest, Sao Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 6, pp. 646–657, 2006.
- [57] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, Vol. 11, No. 4, pp. 341–359, 1997.
- [58] Shota Eguchi, Yuki Matsugano, Hirokazu Sakaguchi, Satoshi Ono, Hisato Fukuda, Ryo Furukawa, and Hiroshi Kawasaki. A comparative study on self-adaptive differential evolution algorithms for test functions and a real-world prob-

- lem. In *International Conference on Learning and Intelligent Optimization*, pp. 131–136. Springer, 2015.
- [59] Jingqiao Zhang and A.C. Sanderson. Jade: Adaptive differential evolution with optional external archive. *Evolutionary Computation, IEEE Transactions on*, Vol. 13, No. 5, pp. 945–958, Oct 2009.
- [60] R. Tanabe and A. Fukunaga. Success-history based parameter adaptation for differential evolution. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 71–78, June 2013.
- [61] R. Tanabe and A. Fukunaga. Evaluating the performance of shade on cec 2013 benchmark problems. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 1952–1959, June 2013.
- [62] Nikolaus Hansen. The cma evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pp. 75–102. Springer, 2006.
- [63] 佐藤浩, 小野功, 小林重信. 遺伝的アルゴリズムにおける世代交代モデルの提案と評価. *人工知能*, Vol. 12, No. 5, pp. 734–744, 1997.
- [64] Hiroshi Satoh. Minimal generation gap model for gas considering both exploration and exploitation. In *Proc. of the 4th International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, pp. 494–497, 1996.
- [65] Martin Pelikan, David E Goldberg, and Fernando G Lobo. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, Vol. 21, No. 1, pp. 5–20, 2002.
- [66] Janusz Kacprzyk and Witold Pedrycz. *Springer handbook of computational intelligence*. Springer, 2015.
- [67] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, pp. 1942–1948. IEEE, 1995.
- [68] Andrzej Jaskiewicz. Genetic local search for multi-objective combinatorial optimization. *European journal of operational research*, Vol. 137, No. 1, pp. 50–71, 2002.

-
- [69] Samir W Mahfoud and David E Goldberg. Parallel recombinative simulated annealing: a genetic algorithm. *Parallel computing*, Vol. 21, No. 1, pp. 1–28, 1995.
- [70] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, Vol. 153, No. 1, pp. 235–256, 2007.
- [71] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Evolutionary bilevel optimization: An introduction and recent advances. In *Recent advances in evolutionary multi-objective optimization*, pp. 71–103. Springer, 2017.
- [72] Aditya Deshpande Jason Rock, Threerasit Issararin and David Forsyh. Authoring image decompositions with generative models. *Computing Research Repository*, 2016.
- [73] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*, 2017.
- [74] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. p. 184–192, 1995.
- [75] A. Hopgood, A. A. and Mierzejewska. Transform ranking: a new method of fitness scaling in genetic algorithms. In Max Bramer, Miltos Petridis, and Frans Coenen, editors, *Research and Development in Intelligent Systems XXV*, pp. 349–354, London, 2009. Springer London.
- [76] Vladik Kreinovich, Chris Quintana, and Olac Fuentes. Genetic algorithms: What fitness scaling is optimal? Vol. 24, pp. 9–26, 04 1993.
- [77] Shuihua Wang, Zeyuan Lu, Ling Wei, Genlin Ji, and Jiquan Yang. Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem. *SIMULATION*, Vol. 92, No. 7, pp. 601–616, 2016.
- [78] Esther M. Arkin, L. Paul Chew, Daniel P. Huttenlocher, and Joseph S. B. Mitchell Klara Kedem and. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 209–216, 1991.

- [79] Sven Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, Vol. 31, No. 8, pp. 983 – 1001, 1998.
- [80] B. N. Delaunay. Sur la sphère vide. *Bull. Acad. Sci. URSS*, Vol. 1934, No. 6, pp. 793–800, 1934.
- [81] Aksoy Yagiz, Oh Tae-Hyun, Paris Sylvain, Pollefeys Marc, and Matusik Wojciech. Semantic soft segmentation. *SIGGRAPH*, 2018.
- [82] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martinez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 3, pp. 314–317. IEEE, 2000.
- [83] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, Vol. 46, No. 5, pp. 1415 – 1432, 2013.
- [84] クラシエ. カラフルパズルチョコ (ミルクチョコ味 + いちごチョコ味) . https://www.kracie.co.jp/products/foods/10132773_21123.html(参照日: 2020 年 10 月 15 日).
- [85] バンダイキャンディ. キャラパキ. https://www.bandai.co.jp/candy/search/brand/index.html?brand_id=146(参照日: 2020 年 10 月 15 日).
- [86] Jean Serra. *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.
- [87] 樽井健人, 董芳艶, 畠山豊, 廣田薫. トーラス型不可能図形生成アルゴリズムに基づくだまし絵描画システム. 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集第 23 回ファジィシステムシンポジウム, pp. 317–317. 日本知能情報ファジィ学会, 2007.
- [88] 藤崎千晶, 吉田哲也. 折り紙の数理に基づくスモッキングのデザイン支援. 人工知能学会全国大会論文集 第 31 回全国大会 (2017), pp. 2I11–2I11. 一般社団法人 人工知能学会, 2017.

研究業績

論文誌

1. Asuka Hisatomi, Tomofumi Matsuyama, Takahiro Kinoshita, Kazunori Mizuno, Satoshi Ono, “Escher-like Tiling Design from Video Images Using Convolutional Variational Autoencoder”, Special Issue on 2019 TAAI of Journal of Information Science and Engineering, Vol.37 No.3, 18 pages, May 2021 ^{*1}.

国際会議（査読付き）

1. Asuka Hisatomi, Hitomi Koba, Makoto Kamizono, Kazunori Mizuno, Satoshi Ono, “Escher-like Tiling Design Using Hierarchical Optimization”, Proceedings of the Genetic and Evolutionary Computation Conference Companion(GECCO '17), pp.89-90, July 2017.
2. Asuka Hisatomi, Hitomi Koba, Kazunori Mizuno, Satoshi Ono, “Application of Escher-like Tiling Design to Confectionery Shape Design”, 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), A2-02, 6 pages, November 2019.

国際会議（査読なし）

1. Asuka Hisatomi, Hitomi Koba, Kazunori Mizuno, Satoshi Ono, “Escher-like Tiling Design Using Estimation of distribution algorithm”, AROB 24th 2019,

^{*1} 採録済み, 2021 年 5 月発行予定

GS10-2, 4 pages, January 2019.

国内会議

1. 久富あすか, 木場仁美, 神菌誠, 水野一徳, 小野智司, 「エッシャー風タイリング図形の自動設計における目的関数の検討」, 『進化計算シンポジウム 2016』, P2-12, 12月 2016 年
2. 久富あすか, 木場仁美, 神菌誠, 水野一徳, 小野智司, 「進化型多目的最適化を用いたエッシャー風タイリング図形の設計に関する基礎研究」, 『火の国情報シンポジウム 2017』, C1-3, 3月 2017 年
3. 久富あすか, 木場仁美, 神菌誠, 水野一徳, 小野智司, 「多目的最適化を用いたエッシャー風タイリング図形の生成に関する研究」, 『2017 年度 人工知能学会全国大会 (第 31 回)』, 2I1-2in2, 5月 2017 年

受賞歴

1. 久富あすか, 火の国情報シンポジウム 2017 情報処理学会九州支部奨励賞 受賞
2. 久富あすか, 2017 年度人工知能学会全国大会 (第 31 回) 全国大会優秀賞 受賞
3. Asuka Hisatomi, The 24th International Conference on Technologies and Applications of Artificial Intelligence Excellent Paper Award

その他

1. 中島 有貴, 松岡 淳一, 久富あすか, 小野 智司, 「協調型共進化の性能評価を目的としたベンチマーク問題と問題の再分割が性能に与える影響の検証」, 『2018 年度人工知能学会全国大会 (第 32 回)』, 3Pin1-02, 6月 2018 年
2. Yuki Nakashima, Jun-ichi Matsuoka, Asuka Hisatomi, Satoshi Ono, “A Preliminary Study on Collaboration Model of Sparsely-Synchronized Heterogeneous Coevolution”, 2018 JPNSEC International Workshop on Evolutionary Computation, 3-4, September 2018.
3. 鞍津輪一希, 上鶴晃平, 久富あすか, 川崎洋, 小野智司, 「白黒 2 階調の補助線を用いた幾何歪みに頑健な 2 次元コードとその復号方式に関する研究」, 情報処理学会論

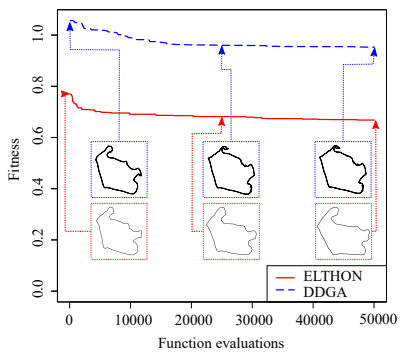
文誌:数理モデル化と応用 (TOM), Vol.12, No.2, pp.69-81, 7月 2019年

4. Asuka Hisatomi, Hitomi Koba, Kazunori Mizuno, Satoshi Ono, “ELTHON: An Escher-like Tile Design Method Using Hierarchical Optimization”, Applied Soft Computing, under review *2.

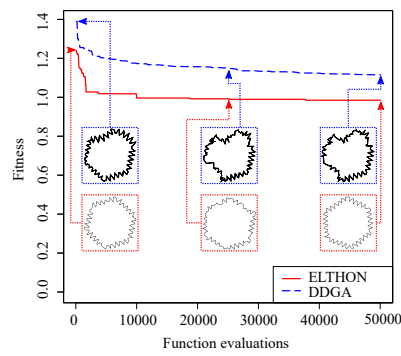
*2 修正中, 2021年1月25日論文査読結果 (Major revision) 受け取り, 2021年3月25日再提出予定

付録 A

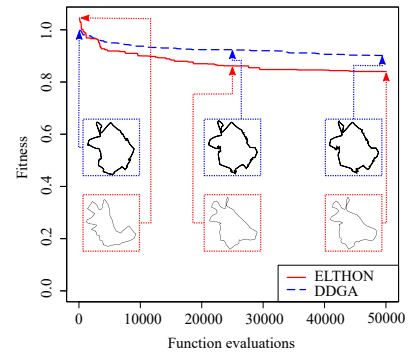
付録 1 : 第 3 章の実験 1 の全結果



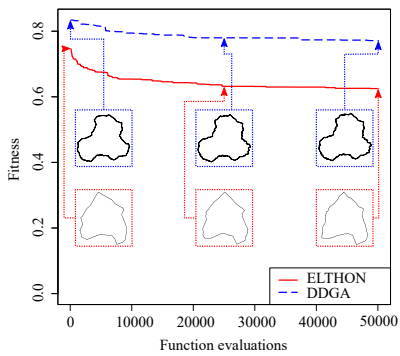
(a) S_1



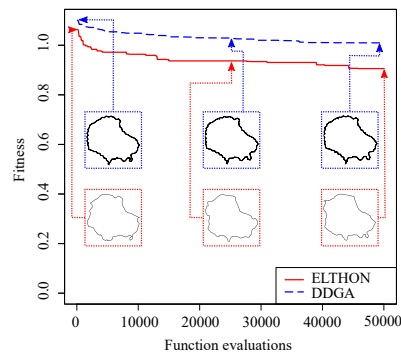
(b) S_2



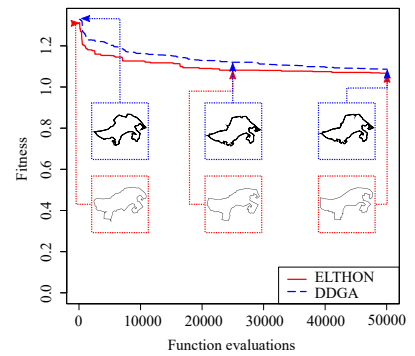
(c) S_3



(d) S_4

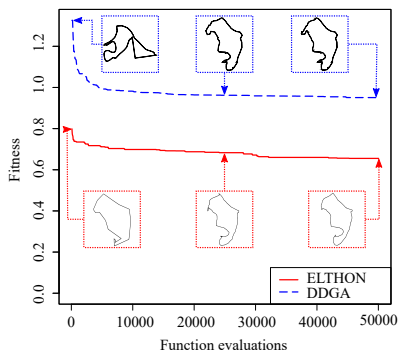


(e) S_5

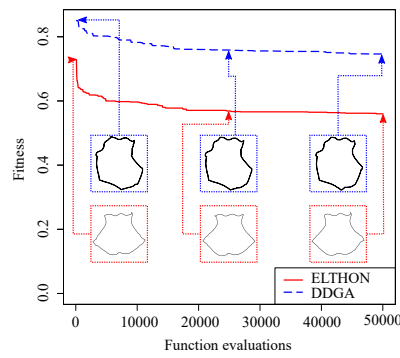


(f) S_6

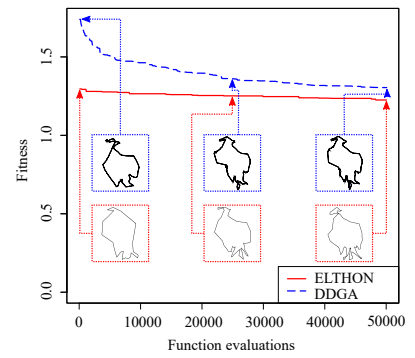
図 A.1. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_1 - S_6$)



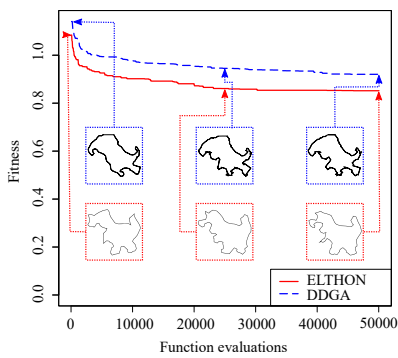
(g) S_7



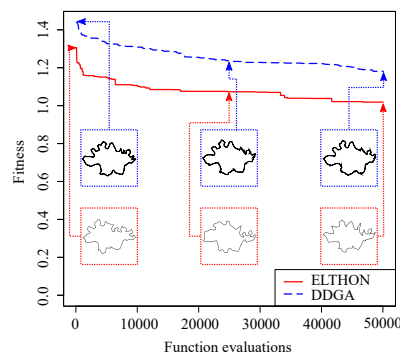
(h) S_8



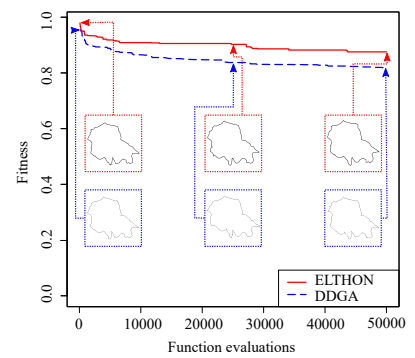
(i) S_9



(j) S_{10}

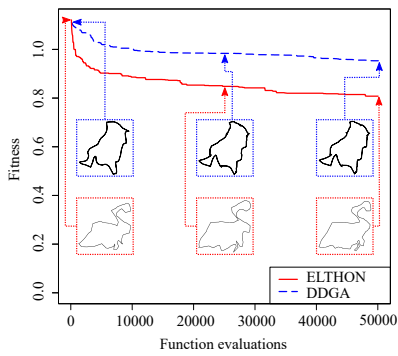


(k) S_{11}

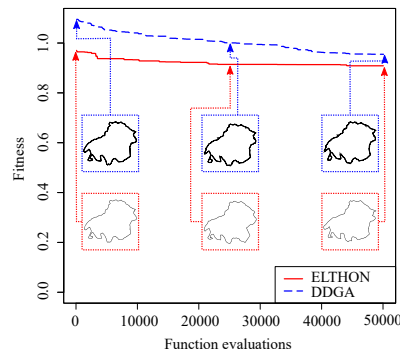


(l) S_{12}

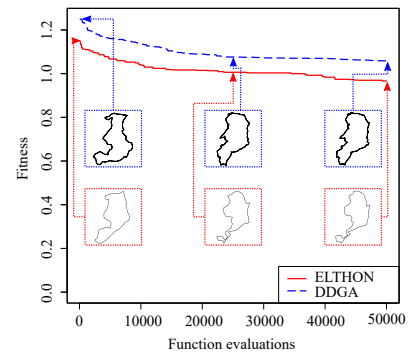
図 A.2. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_7 - S_{12}$)



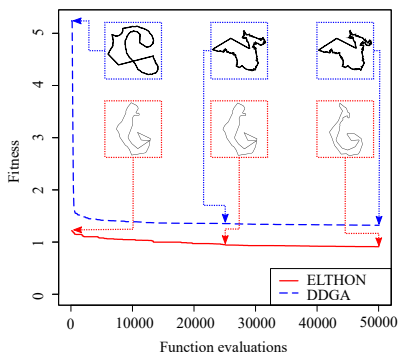
(m) S_{13}



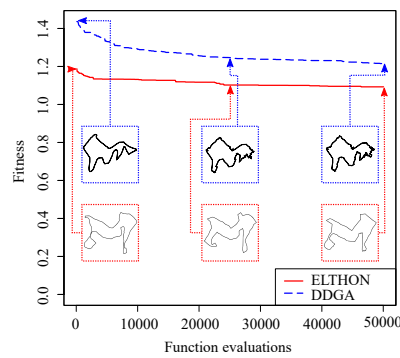
(n) S_{14}



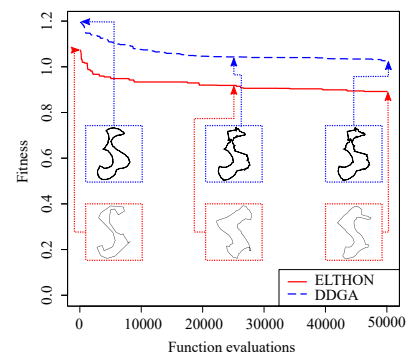
(o) S_{15}



(p) S_{16}

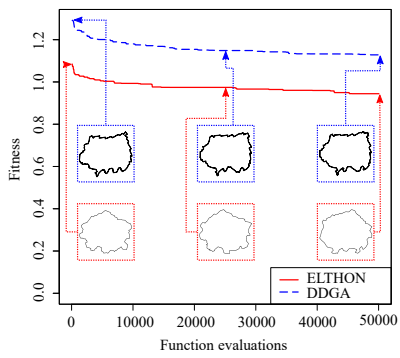


(q) S_{17}

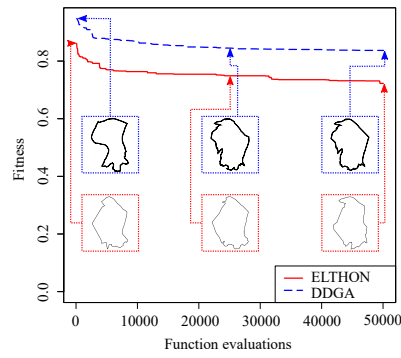


(r) S_{18}

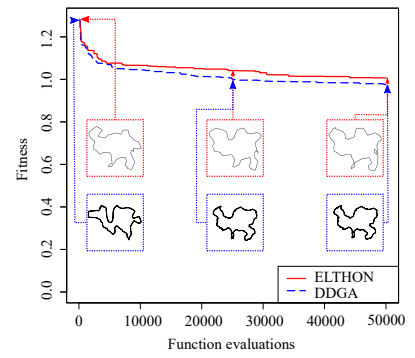
図 A.3. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_{13} - S_{18}$)



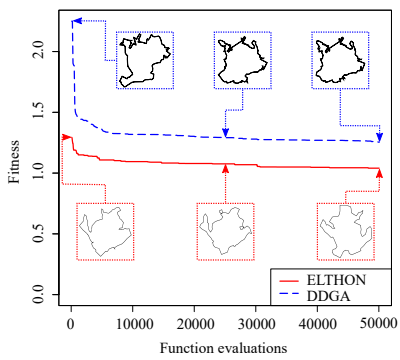
(s) S_{19}



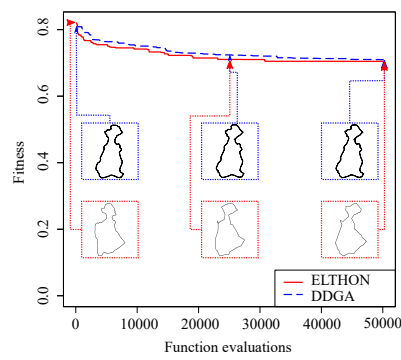
(t) S_{20}



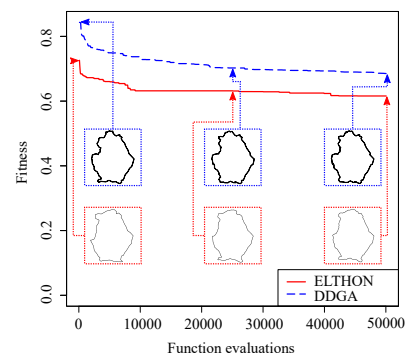
(u) S_{21}



(v) S_{22}



(w) S_{23}



(x) S_{24}

図 A.4. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_{19} - S_{24}$)

付録 A 付録 1: 第 3 章の実験 1 の全結果

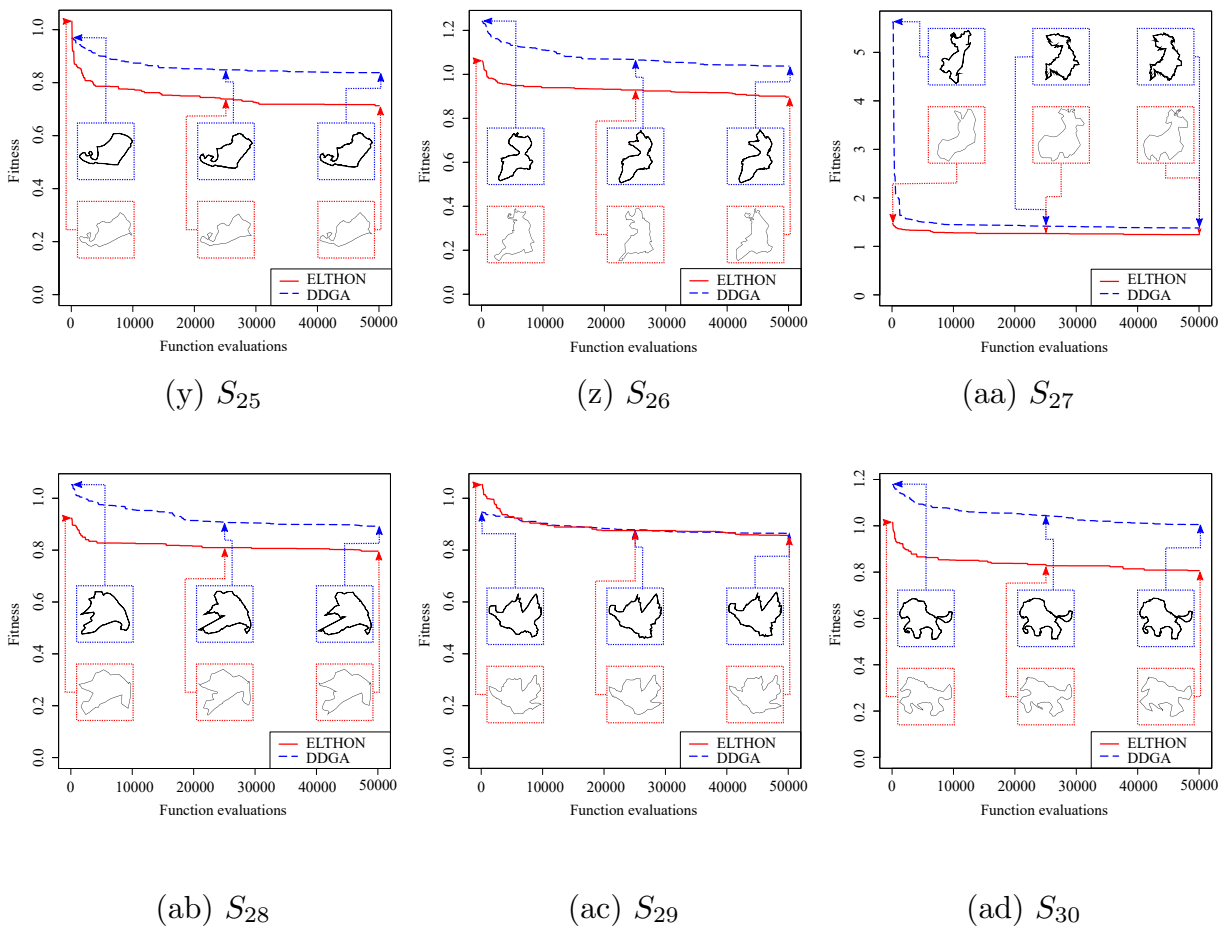


図 A.5. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_{25} - S_{30}$)

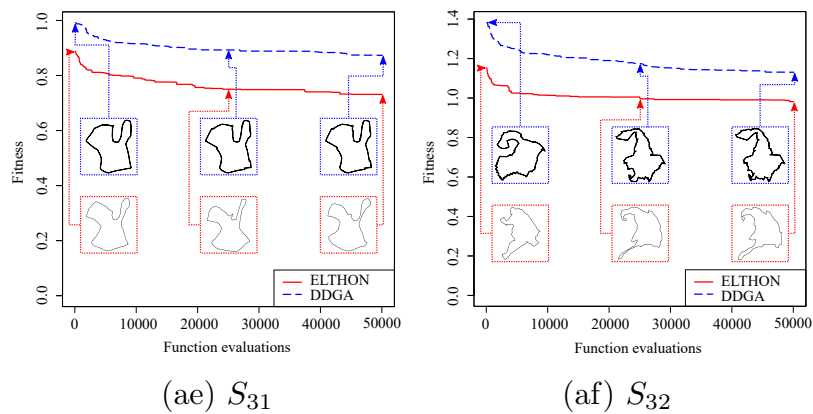


図 A.6. 実験 1 の結果: 最良解の適応度の遷移の比較 ($S_{31} - S_{32}$)

付録 B

付録 2 : 第 3 章の実験 2-1 の全結果


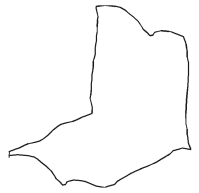
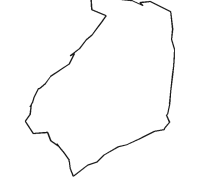
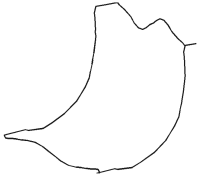
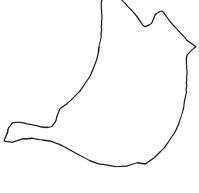
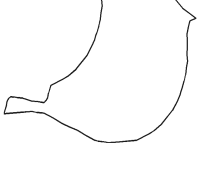
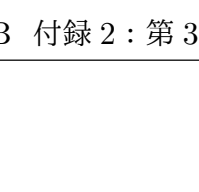

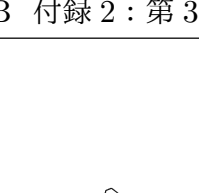
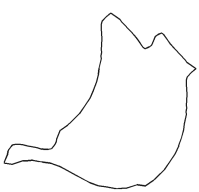
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON								
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}						
															
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.					
F_{TF}	45.7	198	(±105)	94.2	215	(±119)	12.1	16.8	(±3.13)	24	83.5	(±71.8)	19.4	21.6	(±2.1)
F_{BM+TF}	0.93	3	(±0.806)	0.67	0.846	(±0.122)	0.896	0.935	(±0.0454)	0.57	0.581	(±0.00996)	0.6	0.614	(±0.0114)
p	222	766	(±228)	249	318	(±206)	184	194	(±10.9)	134	194	(±71.4)	116	138	(±15)
Vertices	IH04	IH41	IH07	IH07	IH07	IH47	IH23	IH23	IH23	IH23	IH23	IH23	IH23	IH23	IH23
downsampling way	120	120	(±53.7)	60	60	(±0.00)	84	89.8	(±9.5)	100	97	(±9.38)	100	90.6	(±5.64)
	Σ^{TM}														

図 B.1. 実験 2-1 の結果 (S_1)

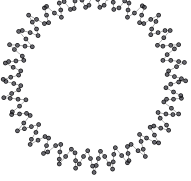
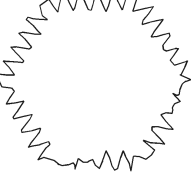
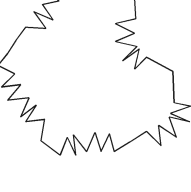
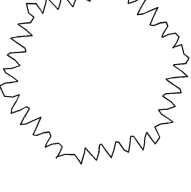
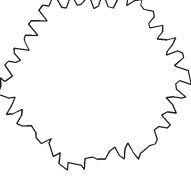
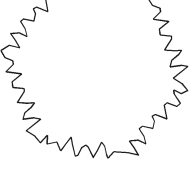



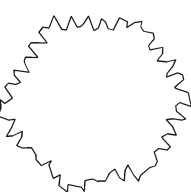
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON								
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}						
															
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.					
F_{TF}	21.1	172	(±195)	84.7	262	(±158)	19.5	22.2	(±2.59)	39.6	173	(±177)	26.3	34.1	(±4.72)
F_{BM+TF}	1.4	1.44	(±0.116)	1.53	1.61	(±0.0929)	1.09	1.41	(±0.199)	1.14	1.17	(±0.0277)	1.07	1.13	(±0.0392)
p	175	330	(±198)	272	439	(±155)	141	177	(±23.3)	165	301	(±177)	139	159	(±19.4)
Vertices	IH07	IH07	IH07	IH07	IH07	IH01	IH01	IH01	IH07	IH07	IH07	IH07	IH01	IH07	IH07
downsampling way	180	120	(±53.7)	60	60	(±0.00)	168	148	(±19.7)	120	119	(±12.5)	126	114	(±12.7)
	Σ^{TM}														

図 B.2. 実験 2-1 の結果 (S_2)







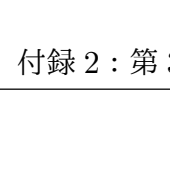
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
F_{BM}	Best Ave. 249	S.D. (± 198)	Best Ave. 119	S.D. (± 113)	Best Ave. 43.8	S.D. (± 4.09)	Best Ave. 60.4	S.D. (± 13.7)	Best Ave. 44.3	S.D. (± 3)
F_{TF}	1.08	(± 1.88)	0.959	(± 0.132)	0.792	(± 0.0645)	0.628	(± 0.0111)	0.606	(± 0.0231)
F_{BM+TF}	284	(± 476)	298	(± 98.5)	123	(± 19.2)	151	(± 12.8)	116	(± 14.8)
p	IH55	IH55	IH28	IH04	IH04	IH04	IH04	IH04	IH04	IH04
Vertices	180	(± 53.7)	60	(± 0.00)	90	(± 8.14)	90	(± 9.1)	102	(± 3.29)
downsampling way	Σ_{EI}									

図 B.3. 実験 2-1 の結果 (S_3)

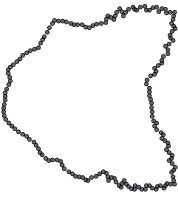
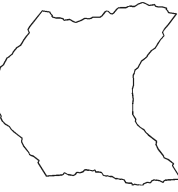
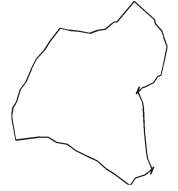




Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
F_{BM}	Best Ave. 28.2	S.D. (± 7.97)	Best Ave. 53.5	S.D. (± 114)	Best Ave. 20.8	S.D. (± 1.27)	Best Ave. 30.1	S.D. (± 7.84)	Best Ave. 26.1	S.D. (± 2.17)
F_{TF}	0.822	(± 0.151)	0.653	(± 0.0528)	0.686	(± 0.0812)	0.592	(± 0.00538)	0.603	(± 0.0165)
F_{BM+TF}	164	(± 21.1)	178	(± 112)	139	(± 12.5)	130	(± 7.56)	119	(± 11.5)
p	IH47	IH47	IH28	IH04	IH52	IH52	IH52	IH52	IH52	IH52
Vertices	180	(± 53.7)	60	(± 0.00)	96	(± 8.29)	96	(± 6.69)	104	(± 9.38)
downsampling way	Σ^{TM}									

図 B.4. 実験 2-1 の結果 (S_4)


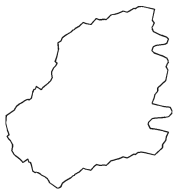
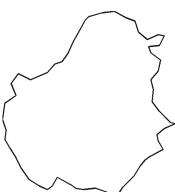
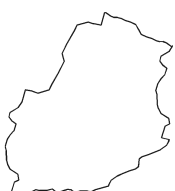
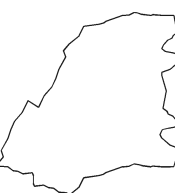

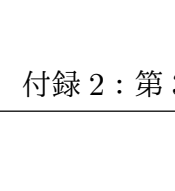
Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON	
	F_L	F_{TF}	F_{TF}	F_{TF}	F_{BM}	F_{BM+TF}
						
F_{BM} F_{TF} F_{BM+TF} p Vertices downsampl- ing way Σ^{TM}	Best Ave. 50.1 S.D. (± 117) 1.1 1.38 168 270 IH05 IH04 180 120 (± 53.7)	Best Ave. 36.2 S.D. (± 1.9) 0.83 0.916 127 137 IH07 IH07 60 60 (± 0.00)	Best Ave. 17.5 S.D. (± 0.558) 18.4 1.33 157 161 IH04 IH04 108 98.4 (± 9.1)	Best Ave. 50.6 S.D. (± 18.2) 64.2 0.917 155 166 IH05 IH04 90 95 (± 11.3)	Best Ave. 20.3 S.D. (± 3.33) 24.3 0.963 119 132 IH04 IH04 102 90 (± 7.35)	

図 B.5. 実験 2-1 の結果 (S_5)








Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON	
	F_L	F_{TF}	F_{TF}	F_{TF}	F_{BM}	F_{BM+TF}
						
F_{BM} F_{TF} F_{BM+TF} p Vertices downsampl- ing way Σ^{TM}	Best Ave. 126 S.D. (± 317) 1.45 8.89 254 1480 IH53 IH53 60 120 (± 53.7)	Best Ave. 131 S.D. (± 119) 1.18 1.24 235 411 IH28 IH28 60 60 (± 0.00)	Best Ave. 52 S.D. (± 9.61) 64.6 1.21 1.27 158 176 IH04 IH04 96 87.6 (± 11.5)	Best Ave. 176 S.D. (± 103) 246 0.969 0.991 262 333 IH46 IH46 88 93.6 (± 7.8)	Best Ave. 50.3 S.D. (± 15.6) 69 1.09 1.16 148 170 IH04 IH04 90 86.4 (± 6.84)	

図 B.6. 実験 2-1 の結果 (S_6)


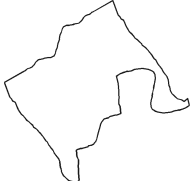
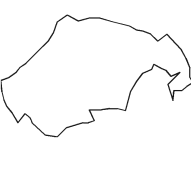
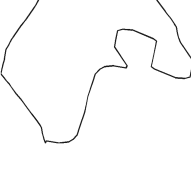
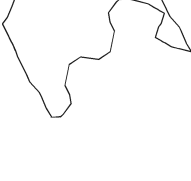
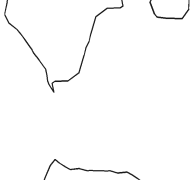
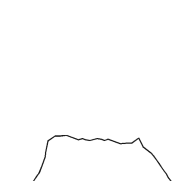
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{BM}	103	147 (± 40.4)	146	199 (± 96.9)	24.3	25.7 (± 2.25)	71.9	82.9 (± 11.8)	26	35.3 (± 7.05)
F_{TF}	0.786	0.678 (± 0.104)	0.739	0.847 (± 0.142)	0.875	0.905 (± 0.0347)	0.556	0.576 (± 0.0172)	0.526	0.608 (± 0.0527)
F_{BM+TF}	201	233 (± 40.4)	239	306 (± 92.6)	135	139 (± 3.75)	142	155 (± 13.3)	99	111 (± 11.2)
p	IH23	IH23	IH07	IH07	IH04	IH04	IH04	IH23	IH04	IH04
Vertices	180	120 (± 53.7)	60	60 (± 0.00)	96	98.4 (± 5.37)	90	95 (± 6.12)	90	94.8 (± 5.02)
downsampling way	Σ^{TM}									

図 B.7. 実験 2-1 の結果 (S_7)


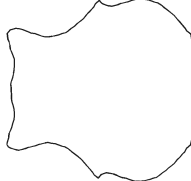
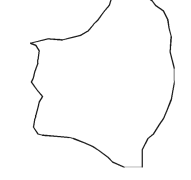
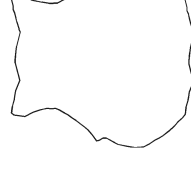
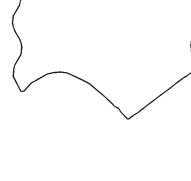
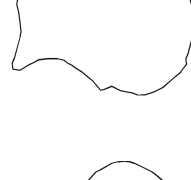
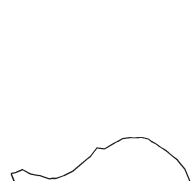
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{BM}	14	20 (± 4.93)	51.8	81 (± 48.8)	9.05	9.21 (± 0.101)	14.1	24.9 (± 7.72)	10.7	13.8 (± 3.03)
F_{TF}	0.727	0.661 (± 0.122)	0.56	0.888 (± 0.229)	0.643	0.733 (± 0.0613)	0.531	0.544 (± 0.0137)	0.523	0.535 (± 0.0135)
F_{BM+TF}	143	137 (± 19.4)	154	238 (± 70.9)	123	139 (± 10.9)	111	121 (± 9.07)	91.1	109 (± 11.2)
p	IH01	IH01	IH28	IH28	IH01	IH01	IH04	IH04	IH01	IH01
Vertices	180	120 (± 53.7)	60	60 (± 0.00)	108	106 (± 3.29)	108	92.4 (± 10.9)	96	98.4 (± 3.29)
downsampling way	Σ^{EI}									

図 B.8. 実験 2-1 の結果 (S_8)





Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}	
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{BM}	175	1260 (± 1560)	400	709 (± 286)	43.9	55.8 (± 8.93)	122	217 (± 89.7)	41.8	52.2 (± 10.5)
F_{TF}	0.974	19.4 (± 24.7)	1.19	1.32 (± 0.119)	0.967	1.06 (± 0.119)	0.925	0.939 (± 0.0112)	0.974	1.02 (± 0.0445)
F_{BM+TF}	263	3000 (± 3780)	509	828 (± 292)	133	151 (± 17.6)	208	301 (± 88.7)	134	144 (± 11.3)
p	IH23	IH46	IH07	IH28	IH04	IH04	IH04	IH02	IH04	IH04
Vertices	60	120 (± 53.7)	60	60 (± 0.00)	102	93.6 (± 9.1)	96	93.6 (± 3.29)	96	91.2 (± 7.82)
downsampling way	Σ^{TM}									

図 B.9. 実験 2-1 の結果 (S_9)





Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}	
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{BM}	113	286 (± 321)	133	184 (± 30)	57.1	59.6 (± 1.96)	116	146 (± 44.3)	62.3	66.1 (± 2.92)
F_{TF}	1.11	1.71 (± 1.69)	0.871	1 (± 0.103)	1.11	1.24 (± 0.109)	0.834	0.847 (± 0.0107)	1.11	1.15 (± 0.0415)
F_{BM+TF}	217	447 (± 478)	219	278 (± 34.4)	163	176 (± 11.8)	197	226 (± 44.6)	164	174 (± 7.39)
p	IH86	IH86	IH07	IH07	IH33	IH33	IH33	IH33	IH33	IH33
Vertices	120	120 (± 53.7)	60	60 (± 0.00)	104	101 (± 5.93)	100	91.6 (± 12.4)	96	93.6 (± 6.07)
downsampling way	Σ^{EI}									

図 B.10. 実験 2-1 の結果 (S_{10})







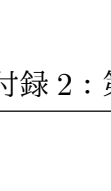
Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON											
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}										
																
F_{BM}	Best Ave. 118	S.D. (± 256)	Best Ave. 79.7	Best Ave. 56.4	Best Ave. 86.8	Best Ave. 75.7	Best Ave. 82.4	S.D. (± 5.17)								
F_{TF}	1	(± 2.22)	1.38	1.41	(± 0.0289)	1.51	1.6	(± 0.06)	0.958	1.02	(± 0.0629)	0.878	0.988	(± 0.0634)		
F_{BM+TF}	206	764	(± 422)	206	406	(± 144)	198	206	(± 6.45)	172	189	(± 19.9)	164	170	(± 9.83)	
p	IH04	IH23	IH07	IH33	IH04	IH04	IH04	IH04	IH04	IH04	IH04	IH04	IH04	IH04	IH04	
Vertices	60	120	(± 53.7)	60	60	(± 0.00)	104	94.8	(± 6.72)	108	94	(± 8.94)	96	93.6	(± 10)	
downsampling way	Σ^{TM}															

図 B.11. 実験 2-1 の結果 (S_{11})


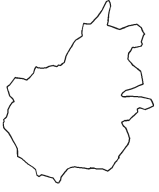
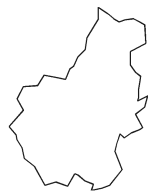




Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON													
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}												
																		
F_{BM}	Best Ave. 30.3	S.D. (± 10.6)	Best Ave. 38.8	Best Ave. 40.8	Best Ave. 23.7	Best Ave. 25.5	Best Ave. 32	Best Ave. 41	Best Ave. 32	Best Ave. 41	Best Ave. 32	Best Ave. 41	Best Ave. 32	Best Ave. 41	Best Ave. 32	Best Ave. 41	S.D. (± 1.82)	
F_{TF}	0.917	0.876	(± 0.166)	0.838	0.974	(± 0.145)	0.957	1.08	(± 0.107)	0.784	0.8	(± 0.0128)	0.761	0.772	(± 0.0107)	0.772	(± 0.0107)	
F_{BM+TF}	162	163	(± 21.3)	163	180	(± 20.4)	161	180	(± 14.9)	147	156	(± 12.2)	105	141	(± 31.3)	105	141	(± 31.3)
p	IH07	IH07	IH07	IH07	IH23	IH07	IH23	IH07	IH23	IH07	IH23	IH07	IH07	IH07	IH07	IH07	IH07	
Vertices	180	120	(± 53.7)	60	60	(± 0.00)	110	113	(± 12.9)	115	107	(± 5.39)	108	110	(± 5.37)	108	110	(± 5.37)
downsampling way	Σ^{TM}																	

図 B.12. 実験 2-1 の結果 (S_{12})



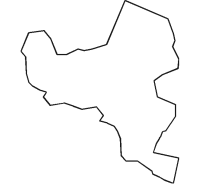
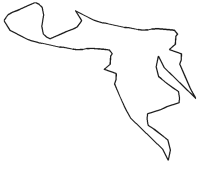
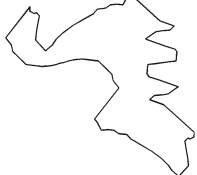

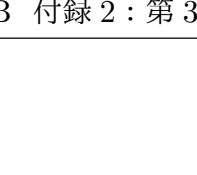
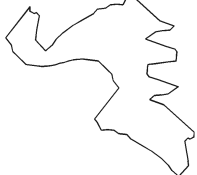

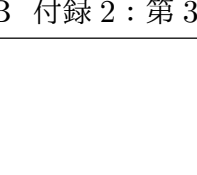
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON								
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}						
															
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.					
F_{TF}	374	472	(±120)	296	359	(±47.4)	101	110	(±7.36)	227	447	(±171)	103	109	(±5.62)
F_{BM+TF}	5.09	5.69	(±1.51)	0.923	1.04	(±0.164)	0.906	1.1	(±0.117)	0.765	0.797	(±0.0322)	0.837	0.973	(±0.104)
p	1190	1380	(±352)	445	525	(±52.8)	263	286	(±16)	355	574	(±169)	221	264	(±30.6)
Vertices	IH23	IH23		IH07	IH28		IH47	IH47		IH51	IH51		IH23	IH23	
downsampling way	180	120	(±53.7)	60	60	(±0.00)	96	90	(±6)	92	88	(±3.74)	80	84.2	(±4.92)
Σ^{TM}															

図 B.13. 実験 2-1 の結果 (S_{13})




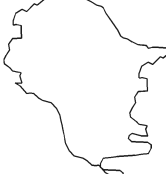

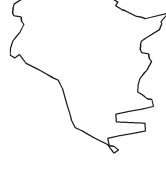
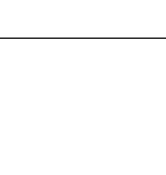

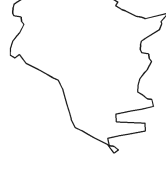
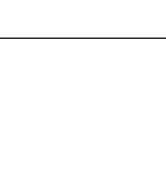
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON								
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{TF}	F_{BM+TF}						
															
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.					
F_{TF}	354	214	(±199)	97.6	243	(±151)	26.7	28.6	(±1.11)	66.2	128	(±98.3)	40.6	41.5	(±0.566)
F_{BM+TF}	1.22	2.78	(±2.4)	1.12	1.56	(±0.902)	1.23	1.29	(±0.0624)	0.848	0.865	(±0.0126)	0.945	0.958	(±0.0123)
p	478	496	(±436)	211	402	(±231)	154	160	(±5.93)	154	215	(±99)	126	139	(±14.4)
Vertices	IH23	IH23		IH07	IH07		IH23	IH23		IH07	IH07		IH23	IH04	
downsampling way	180	120	(±53.7)	60	60	(±0.00)	120	109	(±7.7)	102	103	(±10.8)	90	94.2	(±6.57)
Σ^{TM}															

図 B.14. 実験 2-1 の結果 (S_{14})


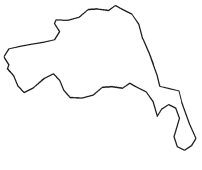
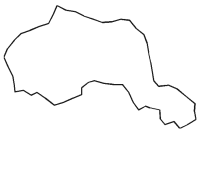
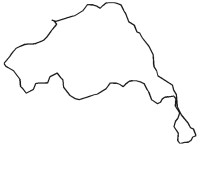
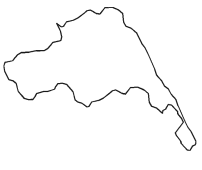
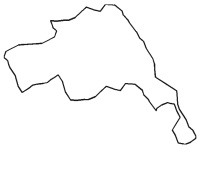
Goal figure W	AOM [18, 19] F_L	DDGA [25, 26] F_{TF}	F_{BM}	ELTHON F_{TF}	F_{BM+TF}
					
F_{BM}	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.
F_{TF}	134 1190 (± 1440)	106 385 (± 210)	78 79.5 (± 2.07)	88.9 118 (± 45)	81.3 85.5 (± 4.51)
F_{BM+TF}	0.749 17 (± 18.6)	0.772 0.93 (± 0.126)	0.946 1 (± 0.0602)	0.744 0.779 (± 0.03)	0.768 0.799 (± 0.0325)
p	218 3080 (± 3510)	210 488 (± 218)	183 191 (± 8.13)	173 205 (± 45.4)	154 175 (± 19.2)
Vertices	IH04 IH04	IH07 IH07	IH04 IH04	IH04 IH04	IH04 IH04
downsampling way	60 120 (± 53.7)	60 60 (± 0.00)	114 102 (± 9.49)	114 101 (± 16.6)	96 90 (± 9.49)
	Σ^{EI}				

図 B.15. 実験 2-1 の結果 (S_{15})

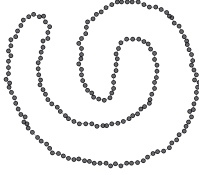


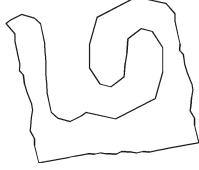

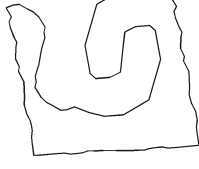
Goal figure W	AOM [18, 19] F_L	DDGA [25, 26] F_{TF}	F_{BM}	ELTHON F_{TF}	F_{BM+TF}
					
F_{BM}	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.	Best Ave. S.D.
F_{TF}	570 639 (± 65.4)	235 1010 (± 719)	129 170 (± 50.3)	295 325 (± 20.6)	120 135 (± 22.8)
F_{BM+TF}	5.34 5.79 (± 1.66)	0.969 1.18 (± 0.145)	0.938 1.1 (± 0.11)	0.714 0.728 (± 0.0101)	0.996 1.06 (± 0.0622)
p	1090 1200 (± 216)	329 1120 (± 727)	234 277 (± 42.3)	365 395 (± 20.4)	220 238 (± 20.6)
Vertices	IH21 IH21	IH07 IH07	IH23 IH23	IH23 IH23	IH23 IH23
downsampling way	120 120 (± 53.7)	60 60 (± 0.00)	85 89 (± 7.42)	95 92 (± 4.47)	100 91 (± 7.42)
	Σ^{TM}				

図 B.16. 実験 2-1 の結果 (S_{16})

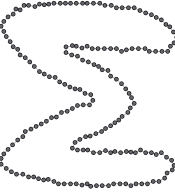
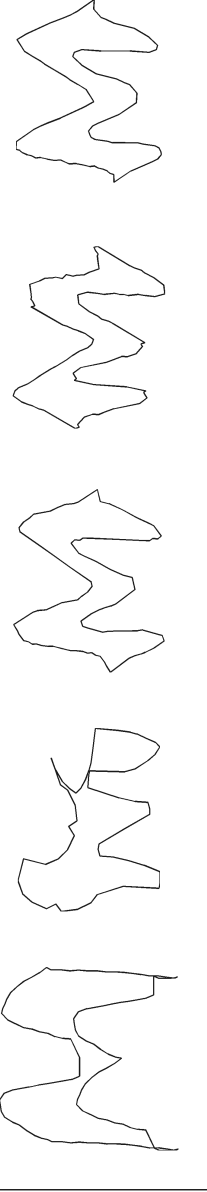
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON							
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}					
														
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{BM}	324	(±211)	183	(±49.7)	61.4	(±4.02)	173	198	(±20)	67.1	74.4	(±4.54)		
F_{TF}	3.08	(±4.25)	1.09	(±0.0418)	0.936	1.06	(±0.1)	0.751	0.763	(±0.0125)	0.949	0.97	(±0.0252)	
F_{BM+TF}	661	(±1410)	302	(±375)	(±52.9)	170	185	(±9.78)	255	281	(±21.1)	173	180	(±4.15)
p	IH01	IH01	IH28	IH28	IH46	IH46	IH46	IH47	IH46	IH47	IH46	IH46	IH46	
Vertices	120	120	60	60	(±0.00)	96	92	(±4)	100	98.4	(±4.56)	88	88	(±4)
downsampling way	Σ_{EI}													

図 B.17. 実験 2-1 の結果 (S_{17})

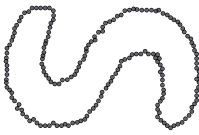
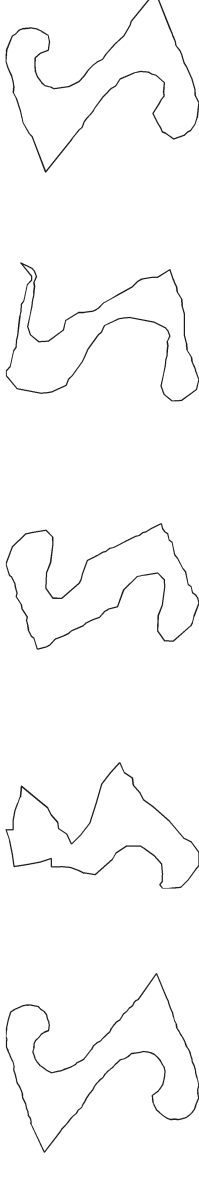
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON								
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}						
															
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.					
F_{BM}	116	(±51.6)	172	281	(±106)	78.1	91.6	(±19.6)	228	250	(±24)	114	128	(±12.2)	
F_{TF}	0.846	0.883	(±0.155)	0.93	1.37	(±0.786)	1.01	1.04	(±0.0352)	0.834	0.846	(±0.00946)	0.96	1	(±0.049)
F_{BM+TF}	217	271	(±48.9)	298	444	(±131)	198	215	(±21.5)	329	351	(±24.2)	238	248	(±9.74)
p	IH33	IH33	IH28	IH28	IH04	IH01	IH04	IH01	IH04	IH47	IH33	IH33	IH33	IH33	
Vertices	180	120	(±53.7)	60	60	(±0.00)	90	102	(±19.9)	96	97.6	(±6.69)	104	98.8	(±16.9)
downsampling way	Σ_{EI}														

図 B.18. 実験 2-1 の結果 (S_{18})

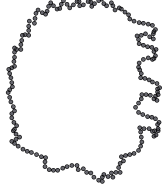
Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_L	F_{TF}	F_{BM}	F_{BM+TF}				
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{BM}	52.4	(±11.1)	45.9	102 (±85.4)	16.7	18.9 (±1.37)	47.5	148 (±120)	17.4	25.7 (±6.65)
F_{TF}	1.35	1.3 (±0.0842)	0.961	1.21 (±0.15)	1.26	1.31 (±0.0397)	0.867	0.921 (±0.0358)	0.923	0.955 (±0.03)
F_{BM+TF}	261	263 (±13.1)	230	289 (±67.4)	212	220 (±6.76)	191	290 (±122)	129	173 (±42.6)
p	IH23	IH23	IH07	IH07	IH04	IH01	IH04	IH04	IH04	IH04
Vertices	180	120 (±53.7)	60	60 (±0.00)	108	97.2 (±10.7)	108	94.8 (±10.7)	90	93.6 (±3.29)
downsampling way	Σ^{TM}									

図 B.19. 実験 2-1 の結果 (S_{19})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_L	F_{TF}	F_{BM}	F_{BM+TF}				
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{BM}	29.9	68.4 (±50.9)	105	170 (±84.2)	16	16.2 (±0.299)	32	43.7 (±8.13)	18.4	20.6 (±1.66)
F_{TF}	0.984	1 (±0.466)	0.932	1.36 (±0.918)	0.747	0.865 (±0.0826)	0.743	0.772 (±0.0199)	0.745	0.76 (±0.0155)
F_{BM+TF}	129	169 (±91.2)	201	306 (±151)	91	103 (±8.4)	109	121 (±7.65)	87.5	96.9 (±13.8)
p	IH04	IH04	IH07	IH07	IH04	IH04	IH04	IH04	IH04	IH04
Vertices	180	120 (±53.7)	60	60 (±0.00)	84	101 (±9.86)	90	92.4 (±3.29)	102	99.6 (±6.84)
downsampling way	Σ^{TM}									

図 B.20. 実験 2-1 の結果 (S_{20})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON	
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}
						
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{TF}	358	1610 (± 2450)	137	347 (± 175)	52.4	59 (± 5.94)
F_{BM+TF}	0.893	26.7 (± 46.7)	1.19	1.63 (± 0.863)	0.937	1.08 (± 0.127)
p	461	4660 (± 7790)	274	533 (± 252)	160	182 (± 18.9)
Vertices	IH88	IH23	IH07	IH07	IH23	IH23
downsampling way	60	120 (± 53.7)	60	60 (± 0.00)	100	97 (± 9.08)
Σ^{TM}					104	97.6 (± 9.63)
					75	90.4 (± 13.2)

図 B.21. 実験 2-1 の結果 (S_{21})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON	
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}
						
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{TF}	98.9	1350 (± 2700)	133	144 (± 15.2)	48	52 (± 4.42)
F_{BM+TF}	1.21	21.1 (± 45.9)	1.05	1.14 (± 0.111)	1.29	1.4 (± 0.0773)
p	201	3130 (± 6590)	228	240 (± 24)	159	170 (± 7.22)
Vertices	IH23	IH28	IH28	IH06	IH06	IH06
downsampling way	180	120 (± 53.7)	60	60 (± 0.00)	102	102 (± 6)
Σ^{TM}					92	91.2 (± 4.38)
					100	98.6 (± 5.81)

図 B.22. 実験 2-1 の結果 (S_{22})

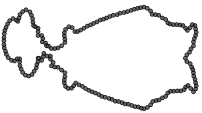
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON						
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}				
													
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	
F_{BM}	96.7	(±239)	74.8	(±92.1)	33.3	(±5.23)	71.9	(±8.03)	38.5	(±2.7)	41.6	(±2.7)	
F_{TF}	0.808	(±4.19)	0.732	(±0.137)	0.876	(±0.0538)	0.731	(±0.0151)	0.777	(±0.0138)	0.8	(±0.0138)	
F_{BM+TF}	204	(±791)	172	(±105)	157	(±10.1)	169	(±9.91)	123	(±17)	148	(±17)	
p	IH23	IH23	IH28	IH28	IH23	IH23	IH23	IH23	IH23	IH01	IH23	IH01	
Vertices	120	(±53.7)	60	(±0.00)	80	(±10.6)	90	(±5.48)	95	(±7.84)	92	(±7.84)	
downsampling way	Σ_{EI}												

図 B.23. 実験 2-1 の結果 (S_{23})


Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON					
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}			
												
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.
F_{BM}	7.12	(±39.1)	38.2	(±8.41)	6.87	(±0.113)	16.7	(±51.7)	8.48	(±2.06)	10.4	(±2.06)
F_{TF}	0.652	(±0.525)	0.67	(±0.0623)	0.578	(±0.0712)	0.552	(±0.00775)	0.587	(±0.00396)	0.592	(±0.00396)
F_{BM+TF}	92.7	(±107)	131	(±7.72)	82.9	(±9.35)	91.5	(±51.7)	68.5	(±22.7)	88	(±22.7)
p	IH01	IH01	IH07	IH07	IH01	IH01	IH03	IH03	IH03	IH03	IH03	IH03
Vertices	180	(±53.7)	60	(±0.00)	114	(±9.1)	132	(±16.1)	126	(±14.1)	108	(±14.1)
downsampling way	Σ_{EI}											

図 B.24. 実験 2-1 の結果 (S_{24})







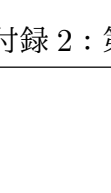
Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}				
										
F_{BM}	Best Ave. 489 (±204)	S.D. 104	Best Ave. 118 (±16.9)	S.D. 61.3	Best Ave. 62.5 (±0.766)	S.D. 178	Best Ave. 216 (±30)	S.D. 64.1	Best Ave. 68 (±2.24)	
F_{TF}	3.05	6.74 (±4.65)	0.995	1.1 (±0.124)	1.1	1.19 (±0.0538)	0.792	0.81 (±0.0197)	0.719	0.738 (±0.0191)
F_{BM+TF}	513	1120 (±621)	200	221 (±24.6)	165	173 (±4.89)	254	291 (±30.4)	131	137 (±5.6)
p	IH07	IH07	IH28	IH07	IH07	IH07	IH21	IH21	IH07	IH07
Vertices	60	120 (±53.7)	60	60 (±0.00)	102	97.2 (±5.02)	85	100 (±9.35)	96	90 (±4.24)
downsampling way	Σ_{EI}									

図 B.25. 実験 2-1 の結果 (S_{25})

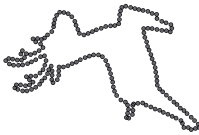
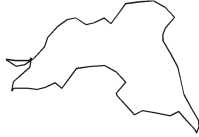





Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}				
										
F_{BM}	Best Ave. 303 (±92.4)	S.D. 161	Best Ave. 288 (±110)	S.D. 66.4	Best Ave. 73 (±5.35)	S.D. 205	Best Ave. 226 (±15.5)	S.D. 77.4	Best Ave. 81.8 (±3.83)	
F_{TF}	2.87	3.25 (±1.89)	0.95	1.02 (±0.0448)	1.13	1.18 (±0.0425)	0.856	0.876 (±0.0194)	0.897	0.984 (±0.0539)
F_{BM+TF}	644	738 (±189)	287	410 (±107)	201	214 (±10.2)	312	330 (±13.5)	186	199 (±12.9)
p	IH04	IH04	IH28	IH28	IH04	IH04	IH07	IH07	IH04	IH04
Vertices	60	120 (±53.7)	60	60 (±0.00)	102	99.6 (±9.1)	90	91.6 (±6.39)	96	96 (±4.24)
downsampling way	Σ_{EI}									

図 B.26. 実験 2-1 の結果 (S_{26})

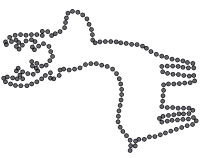
Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}				
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{BM}	237	1260 (± 2030)	184	311 (± 132)	74.1	76.8 (± 2.51)	153	176 (± 39.3)	67.4	72.8 (± 4.09)
F_{TF}	3.27	17.7 (± 29.9)	1.26	1.38 (± 0.123)	1.15	1.25 (± 0.062)	0.844	0.858 (± 0.0148)	1.03	1.09 (± 0.0386)
F_{BM+TF}	547	2940 (± 4870)	304	442 (± 142)	189	196 (± 4.45)	233	258 (± 40.6)	163	176 (± 11.7)
p	IH47	IH47	IH07	IH28	IH47	IH47	IH46	IH46	IH47	IH47
Vertices downsampling way	120	120 (± 53.7)	60	60 (± 0.00)	92	94.4 (± 3.58)	96	99.2 (± 6.57)	96	93.6 (± 6.07)
	Σ_{TM}									

図 B.27. 実験 2-1 の結果 (S_{27})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_{TF}	F_{BM}	F_{TF}	F_{BM+TF}				
										
	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{BM}	310	498 (± 402)	122	195 (± 92.9)	53.4	57.6 (± 3.08)	93.2	125 (± 39.9)	61.1	71 (± 7.81)
F_{TF}	5.09	6.53 (± 5.2)	0.878	1.02 (± 0.128)	0.702	0.762 (± 0.0815)	0.537	0.57 (± 0.0214)	0.557	0.614 (± 0.0603)
F_{BM+TF}	991	1370 (± 1090)	256	332 (± 98.2)	150	160 (± 10.5)	165	202 (± 41)	140	153 (± 13.3)
p	IH47	IH47	IH07	IH07	IH04	IH04	IH04	IH04	IH04	IH04
Vertices downsampling way	180	120 (± 53.7)	60	60 (± 0.00)	96	98.4 (± 5.37)	90	90.6 (± 8.59)	90	93.6 (± 5.37)
	Σ_{EI}									

図 B.28. 実験 2-1 の結果 (S_{28})



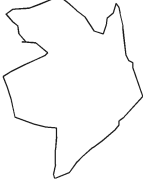



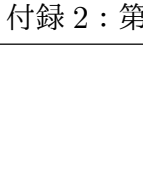


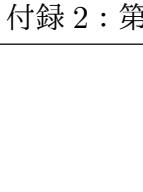
Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
F_{BM}	Best Ave. 1140	S.D. (± 2030)	Best Ave. 47.9	S.D. (± 101)	Best Ave. 29.3	S.D. (± 2.74)	Best Ave. 38.7	S.D. (± 95)	Best Ave. 29.3	S.D. (± 10.8)
F_{TF}	1.06	(± 29.1)	0.576	(± 0.144)	0.886	(± 0.0554)	0.767	(± 0.0436)	0.78	(± 0.0707)
F_{BM+TF}	185	(± 5300)	113	(± 104)	129	(± 6.98)	137	(± 96.5)	116	(± 24.5)
p	IH46	IH46	IH07	IH07	IH04	IH04	IH07	IH07	IH04	IH04
Vertices	120	(± 53.7)	60	(± 0.00)	96	(± 10.4)	102	(± 5.02)	90	(± 6.84)
downsampling way	Σ^{TM}									

図 B.29. 実験 2-1 の結果 (S_{29})











Goal figure W	AOM [18, 19]			DDGA [25, 26]			ELTHON			
	F_L	F_{TF}	F_{BM+TF}	F_L	F_{TF}	F_{BM+TF}	F_{TF}	F_{BM}	F_{BM+TF}	
										
F_{BM}	Best Ave. 522	S.D. (± 291)	Best Ave. 197	S.D. (± 47.8)	Best Ave. 94.7	S.D. (± 3.36)	Best Ave. 183	S.D. (± 17.6)	Best Ave. 93.6	S.D. (± 5.19)
F_{TF}	1.19	(± 3.12)	1.05	(± 0.0167)	1.33	(± 0.0875)	0.814	(± 0.0399)	1.07	(± 0.0704)
F_{BM+TF}	345	(± 596)	309	(± 46.6)	237	(± 9.8)	275	(± 15)	198	(± 21)
p	IH28	IH46	IH28	IH28	IH05	IH05	IH28	IH28	IH05	IH05
Vertices	180	(± 53.7)	60	(± 0.00)	102	(± 6.84)	105	(± 2.88)	96	(± 6.84)
downsampling way	Σ^{TM}									

図 B.30. 実験 2-1 の結果 (S_{30})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_L	F_{TF}	F_{TF}	F_{BM+TF}				
										
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{TF}	130	179 (± 73.8)	133	192 (± 97.9)	52.3	63.3 (± 14.1)	96.3	191 (± 88)	56.7	62.3 (± 4.86)
F_{BM+TF}	1.14	1.38 (± 0.878)	1.08	1.22 (± 0.0906)	0.97	1.06 (± 0.0541)	0.789	0.806 (± 0.0115)	0.726	0.805 (± 0.069)
p	291	375 (± 196)	314	366 (± 86)	203	214 (± 10.2)	210	306 (± 88.7)	164	177 (± 9.42)
Vertices	IH31	IH31	IH28	IH28	IH23	IH23	IH04	IH23	IH23	IH23
downsampling way	180	120 (± 53.7)	60	60 (± 0.00)	75	86 (± 7.78)	105	98 (± 7.58)	85	80.2 (± 4.76)
Σ^{TM}										

図 B.31. 実験 2-1 の結果 (S_{31})


Goal figure W	AOM [18, 19]		DDGA [25, 26]		ELTHON					
	F_L	F_{TF}	F_L	F_{TF}	F_{TF}	F_{BM+TF}				
										
F_{BM}	Best Ave.	S.D.	Best Ave.	S.D.	Best Ave.	S.D.				
F_{TF}	166	438 (± 216)	133	178 (± 58.2)	58.6	72 (± 8.94)	143	169 (± 31.1)	70.4	87.5 (± 16.1)
F_{BM+TF}	1.16	4.08 (± 3.73)	0.903	0.951 (± 0.0504)	1.06	1.13 (± 0.0677)	0.875	0.885 (± 0.00988)	0.939	0.967 (± 0.0402)
p	281	842 (± 579)	222	272 (± 59.2)	168	184 (± 13.3)	230	256 (± 31.9)	166	183 (± 12)
Vertices	IH21	IH21	IH28	IH28	IH04	IH04	IH46	IH46	IH04	IH23
downsampling way	180	120 (± 53.7)	60	60 (± 0.00)	90	94.8 (± 2.68)	92	92 (± 4.9)	84	94.6 (± 8.59)
Σ^{TM}										

図 B.32. 実験 2-1 の結果 (S_{32})

付録 C

付録 3 : 第 3 章の目標図形の複雑さと ELTHON の性能との関係

第 3 章で実験に用いた 32 個の目標図形は，図 C.1 に示すように，さまざまな形状の特徴を持っている．図 C.1 のグラフ軸は，それぞれ図形の形状の特徴を示している．横軸は， $(1 - \text{solidity})$ に対応しており， solidity は，図形の凸包の面積と対象図形の面積の比で表され，図形形状の複雑さを簡単に表現している．縦軸は，図形の変曲点の総数を表し，局所的に注目した図形形状の複雑さを表現している．

一方，図形形状の複雑さに着目すると，次のようなことがわかる．

Observation 9 (図形の複雑さと頂点数やタイリングパターンとの関係) 図形の複雑さと，選択した頂点数やタイリングパターンとの間には，明らかな関連性は見られなかった．

図 C.2 は，最適化に用いた 3 つの目的関数について，複雑さと選択した頂点数との関係を示している．グラフの各軸は，図 C.1 と同じである．つまり，軸は $(1 - \text{solidity})$ と変曲点の総数を示し，プロットされたマーカーの濃淡は，与えられた目標図形に対して得られた最良解で選択された頂点数を表している．

先行研究である AOM [18, 19] では，あらかじめ頂点数とタイリングパターンを決定しておく必要があるが，上記の結果から，予め決定しておくことが難しいことがわかる．一方，ELTHON は，最適化を行う中で頂点数やタイリングパターンを決定するため，有効であることがわかる．これにより，目標図形の頂点座標の試行錯誤を省くことができる．

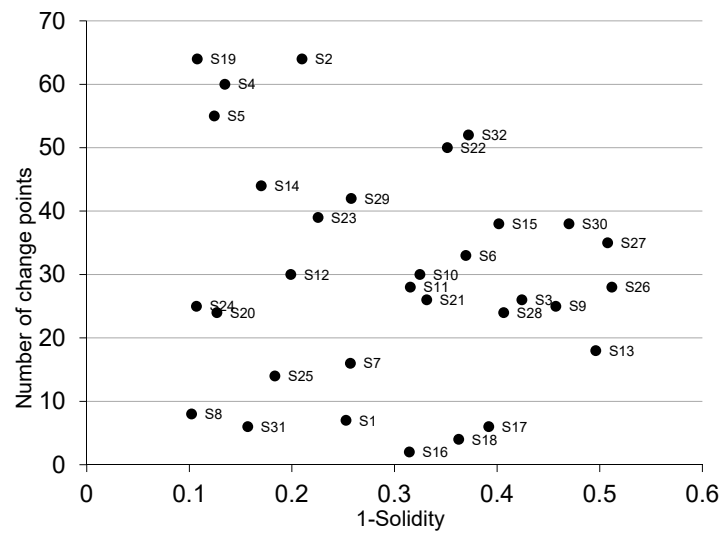


図 C.1. 目標図形の複雑さ

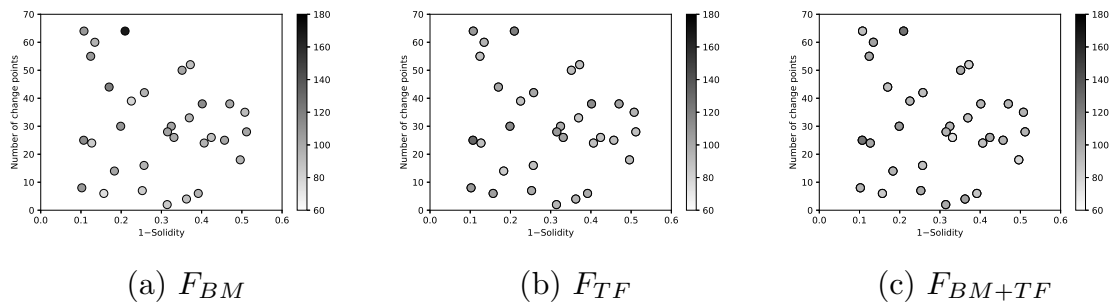


図 C.2. 頂点数と図形の複雑さの相関図

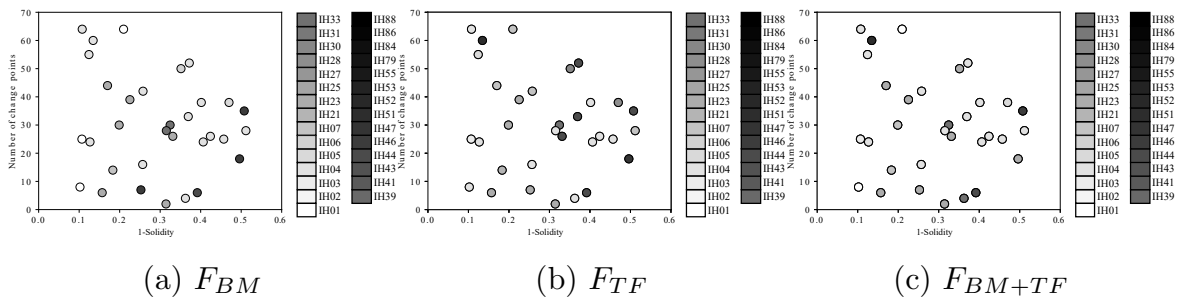


図 C.3. タイリングパターンと図形の複雑さの相関図