

Title	タングラム・清少納言知恵の板で構成可能なポリアボロの全列挙
Author(s)	LI, HAO
Citation	
Issue Date	2026-03
Type	Thesis or Dissertation
Text version	author
URL	https://hdl.handle.net/10119/20515
Rights	
Description	Supervisor:上原 隆平, 先端科学技術研究科, 修士(情報科学)

修士論文

タングラム・清少納言知恵の板で構成可能なポリアボロの全列挙

LI HAO

主指導教員 上原 隆平

北陸先端科学技術大学院大学
先端科学技術専攻
(情報科学)

令和8年3月

Abstract

シルエットパズルは、複数の多角形（「ピース」と呼ばれる）を、互いに重ならないように配置し、指定されたターゲット形状（シルエット）を作り出すパズルである。その代表例として、中国発祥の「タングラム」と日本発祥の「清少納言知恵の板」が挙げられる。これらは共に正方形を7つのピースに分割して作られ、面積だけでなく、構成する辺の長さの総和までもが完全に一致するという幾何学的共通点を持つ。しかし、その構成能力（表現力）の差異については、これまで凸多角形という限定された範囲でしか議論されてこなかった。WangとHsiung（1942）はタングラムが13種類の凸多角形を構成可能であることを示し、Fox-Epsteinら（2016）は清少納言知恵の板が16種類を構成可能であることを示した。本研究では、この比較を一般的な幾何形状である「ポリアボロ（Polyabolo）」へと拡張し、計算機を用いた全列挙によって両者の幾何学的表現力を定量的に解明することを目的とした。

本研究では、ポリアボロを「 n 個の合同な直角二等辺三角形を辺対辺で接続した図形」と定義し、タングラム等のパズルピースを単位三角形の集合（ $n = 16$ ）としてモデル化した。形状の列挙にあたっては、回転・反転して重なるものは同一とみなし、非凸多角形や多重連結図形（穴あき形状）も対象とした。効率的な全列挙を実現するため、2次元整数グリッド上での「4ビットマスク符号化」手法を提案した。これは、グリッドセルの各辺（左上右下）における直角二等辺三角形の配置状態を4ビットの整数値で管理する手法である。これにより、ピース同士の衝突判定や結合処理を高速なビット演算（論理和や排他的論理和など）に置き換えることが可能となった。

探索アルゴリズムには、単純な全配置探索ではなく、連結性に基づく逐次構成法を採用した。これは任意の初期ピースから出発し、既存の形状に対して新しいピースを隣接ルールに従って追加していく手法である。 $n = 16$ という大規模な探索空間に対処するため、本研究では以下の主要な最適化戦略を導入した。第一に、初期ピースを固定することで探索の冗長性を排除した。第二に、生成された中間形状に対して正準化を行い、同型形状を早期に枝刈りした。第三に、動的バウンディングボックスを用いて、新規ピースの配置候補点を局所的な範囲に限定し、無駄な探索を削減した。第四に、最終ピース（第7ピース）の追加段階において、中間リストを介さず直接結果セットと照合する「直接挿入戦略」を導入し、メモリ消費を抑制した。

実装したアルゴリズムの正当性は、パズル制約のない16個の単位三角形を用いた全列挙実験によって検証された。その結果、生成された $n = 16$ のポリアボロの総数は28,616,815種類となり、オンライン整数列大辞典（OEIS）の数列A006074と完全に一致した。これにより、提案手法の正確性が実証された。

この検証済みのフレームワークを用いて、タングラムと清少納言知恵の板の全列挙を行った結果、タングラムが構成可能なポリアボロは5,583,516種類、清少納言知恵の板は10,889,227種類であることが判明した。この結果は、清少納言知恵

の板がタングラムの約 1.95 倍の形状を生成可能であり、高い幾何学的表現力を持つことを定量的に裏付けるものである。

さらに、この差異が生じる要因を特定するため、Fox-Epstein らが提案した 4 種の変種パズルおよび本研究で独自に設計した 3 種の変種パズル（計 7 種）についても同様の全列挙を行い、比較分析を行った。その結果、パズルの表現力を決定づける 3 つの幾何学的要因が明らかになった。第一に「大面積ピースの制約」である。面積の大きなピース（大三角形など）を含むパズルセットは、微細な隙間を埋める能力が低く、生成数が減少する傾向にある。第二に「ピース形状の多様性」である。構成ピースの形状が多様であるほど、対称性による重複が減少し、組み合わせの自由度が増大する。第三に「非対称性の有効性」である。正方形のような対称性の高いピースよりも、平行四辺形のような非対称なピースを含む方が、回転・反転による状態数が増え、より多くの形状を構成できることが示された。本研究は、これまで未解明であったシルエットパズルの表現力の差を、数千万規模のデータ生成を通じて定量的に明らかにし、その幾何学的根拠を提示したものである。

Abstract

Silhouette puzzles are geometric puzzles in which a set of polygons (called “pieces”) must be arranged to form a specific target shape (silhouette) without overlapping. Representative examples include “Tangram,” which originated in China, and “Sei Shonagon Chie no Ita,” which originated in Japan. Both puzzles consist of a square dissected into seven pieces. They share remarkable geometric similarities: not only is their total area identical, but the total length of their constituent edges is also exactly the same. However, the difference in their combinatorial expressiveness has primarily been discussed only within the limited scope of convex polygons. Wang and Hsiung (1942) proved that Tangram can form 13 types of convex polygons, while Fox-Epstein et al. (2016) showed that Sei Shonagon Chie no Ita can form 16 types. The objective of this study is to extend this comparison to “polyabolos”—general geometric shapes formed by connecting isosceles right triangles—and to quantitatively elucidate the geometric expressiveness of both puzzles through exhaustive computational enumeration.

In this study, a polyabolo is defined as a geometric figure formed by connecting n congruent isosceles right triangles edge-to-edge. We modeled the pieces of Tangram and other puzzles as sets of unit triangles ($n = 16$). In the enumeration process, shapes that coincide after rotation or reflection are considered identical, and the scope includes not only convex polygons but also non-convex and multiply connected (holey) shapes. To achieve efficient exhaustive enumeration, we proposed a “4-bit mask encoding” method on a 2D integer grid. This method manages the placement state of isosceles right triangles at each edge (left, top, right, bottom) of a grid cell using a 4-bit integer. This representation allows geometric operations, such as collision detection and piece integration, to be replaced by high-speed bitwise operations (e.g., OR, XOR).

For the search algorithm, rather than a simple brute-force placement approach, we adopted a “connectivity-based sequential construction” method. This approach starts with an arbitrary initial piece and sequentially adds remaining pieces to the existing shape according to edge-adjacency rules. To handle the massive search space of $n = 16$, we introduced the following key optimization strategies. First, we fixed the initial piece to eliminate search redundancy. Second, we applied canonicalization to generated intermediate shapes to prune isomorphic duplicates early. Third, we used a dynamic bounding box to restrict the candidate positions for new pieces to a local range, thereby reducing invalid searches. Fourth, we implemented a “direct insertion strategy” for the final step (adding the 7th piece), which checks generated signatures directly against the final result set without storing them in an intermediate list, significantly suppressing memory consumption.

The correctness of the implemented algorithm was verified through an exhaus-

tive enumeration experiment using 16 unconstrained unit triangles. The result showed that the total number of unique polyabolos for $n = 16$ is 28,616,815, which perfectly matches sequence A006074 in the On-Line Encyclopedia of Integer Sequences (OEIS). This result demonstrates the accuracy of the proposed framework.

Using this verified framework, we exhaustively enumerated the polyabolos constructible by Tangram and Sei Shonagon Chie no Ita. The results revealed that Tangram can form 5,583,516 distinct polyabolos, whereas Sei Shonagon Chie no Ita can form 10,889,227. This quantitative evidence proves that Sei Shonagon Chie no Ita has approximately 1.95 times the generative capacity of Tangram, indicating a significantly higher level of geometric expressiveness.

Furthermore, to identify the factors contributing to this difference, we conducted similar enumerations and comparative analyses for seven variant puzzles, including four variants proposed by Fox-Epstein et al. and three originally designed in this study. The analysis revealed three geometric factors that determine the expressiveness of a puzzle. The first factor is the “constraint of large-area pieces.” Puzzle sets containing large pieces (e.g., large triangles) tend to have a reduced number of configurations due to their limited ability to fill fine gaps. The second factor is “piece shape diversity.” Higher diversity in the shapes of constituent pieces reduces symmetry-induced duplicates and increases the degrees of freedom in combination. The third factor is the “effectiveness of asymmetry.” It was shown that including asymmetric pieces (like parallelograms) yields a greater number of configurations compared to highly symmetric pieces (like squares) because asymmetry allows for more distinct states through rotation and reflection. This study quantitatively clarifies the previously unexplored difference in expressiveness between silhouette puzzles through the generation of tens of millions of shapes and provides the geometric grounds for these differences.

目次

第1章 序論	1
1.1 研究背景	1
1.2 研究対象と本稿の貢献	2
第2章 提案手法	4
2.1 グリッド表現と4ビットマスク符号化	4
2.2 全列挙アルゴリズムの設計	4
2.2.1 探索戦略の選択	4
2.2.2 アルゴリズムの全体像	8
2.3 新規形状の生成アルゴリズム	9
2.3.1 変種の事前計算と配置走査	9
2.3.2 ルックアップテーブル (LUT) による高速な隣接判定	9
2.3.3 幾何学的補完による例外処理	10
2.3.4 生成処理のアルゴリズム記述	10
2.4 形状の正規化と同型判定	10
2.4.1 相対座標による正規化	10
2.4.2 最小署名の導出	13
第3章 実験結果	14
3.1 実験環境と実装詳細	14
3.2 形状全列挙の結果	14
3.2.1 アルゴリズム正当性の検証 ($n = 16$ ポリアボロ)	14
3.2.2 タングラムと清少納言知恵の板の比較	15
3.2.3 データセットおよび Web ビューアの公開	15
3.3 結果分析	16
3.4 結果分析：構成数を左右する主要な幾何学的要因	16
3.4.1 要因1：大面積ピースによる配置制約	16
3.4.2 要因2：ピース形状の多様性がもたらす組合せの増加	18
3.4.3 要因3：非対称な形状の有効性	18
第4章 結論と展望	20
4.1 結論	20

4.2 今後の展望	20
---------------------	----

目次

1.1	タングラムと清少納言知恵の板の構造比較	1
1.2	16分割構成	2
2.1	4ビットマスク符号化	5
2.2	ピース順序の重要性の証明	6
2.3	ピースの最も長く延伸する配置パターン1	12
2.4	ピースの最も長く延伸する配置パターン2	12
3.1	変種パズル (Sp.A – Sp.G) の形状一覧	17
4.1	ラッキーパズル	21

表 目 次

3.1	タングラムと清少納言知恵の板の構成可能形状数の比較	15
3.2	タングラム, 清少納言知恵の板および変種パズルのコンポーネント 構成と生成結果統計	16

第1章 序論

1.1 研究背景

シルエットパズルは幾何パズルの一種である。その基本ルールは、与えられた多角形ピースの集合 $P = \{p_1, p_2, \dots, p_k\}$ を用いて、指定された目標形状（シルエット）を構築することである。構築する場合、平面上で平行移動・回転・反転操作を行うことが許されるが、ピース同士を重ねて配置してはならない。

シルエットパズルの代表例として、中国発祥の「タングラム」と日本発祥の「清少納言知恵の板」が挙げられる（図 1.1）。タングラムの最古の文献は 1813 年の『七巧図合璧』である [1]。一方、清少納言知恵の板は、1742 年に刊行された書籍において確認されている [3]。このように起源や時代は異なるものの、両者は幾何学的構造において顕著な共通点を有している。

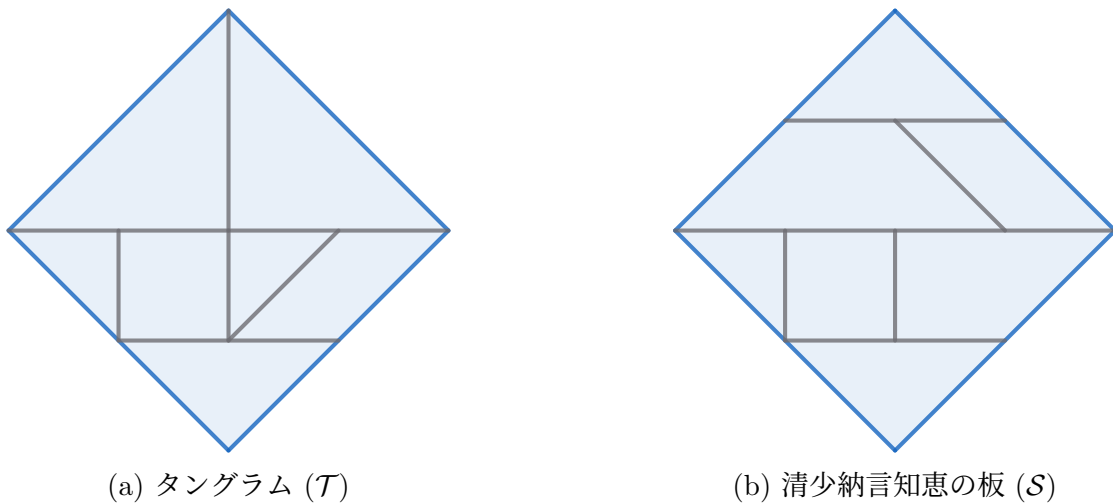


図 1.1: タングラムと清少納言知恵の板の構造比較

まず、両者は共に正方形を 7 つのピースに分割したパズルである。図 1.2 に示すように、両者のすべてのピースは、直角辺の長さが 1、斜辺の長さが $\sqrt{2}$ の「単位三角形」（記号 u ）16 個の集合として表現できる。さらに注目すべき点は、その分割パターンが異なるにもかかわらず、各ピースの辺長の総和までもが完全に一致していることである。具体的には、いずれも長さ 1 の辺 20 本と長さ $\sqrt{2}$ の辺 10 本から構成されている。すなわち両者は面積だけでなく、構成する辺の長さの総

和においても完全に一致する.

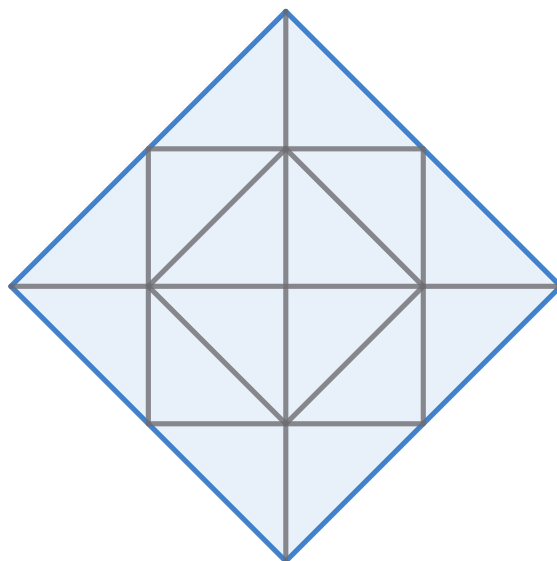


図 1.2: 16 分割構成

以上の共通点にもかかわらず，両者の構成能力の差異は未解明である．Wang と Hsiung は 1942 年，凸多角形に限定してタングラムの解析を行った．彼らは，16 個の単位三角形からなる凸多角形が全部で 20 種類存在することを示した上で，タングラムはそのうち 13 種類しか構成できないことを証明した [2]．これに対し，Fox-Epstein らは 2016 年に同様の手法を適用し，清少納言知恵の板が全 20 種類のうち 16 種類を構成可能であると証明した [3]．この結果は，凸多角形の範囲において，清少納言知恵の板がタングラムより高い表現力を持つことを定量的に示している．

しかし，先行研究の対象は凸多角形に限定されている．非凸多角形や多重連結（穴あき）を含む一般的な形状に対する，全列挙に基づく網羅的な比較研究は未だなされていない．

1.2 研究対象と本稿の貢献

タングラムと清少納言知恵の板の幾何学的表現力の差異を明らかにするため，本研究では分析の対象を従来の凸多角形から「ポリアボロ」（Polyabolo）へと拡張する．

定義 1 (ポリアボロ) : n 個の合同な直角二等辺三角形 u を「辺対辺」の形式で接続して形成される平面幾何図形．

この定義に基づく列挙においては，図形の同一性と範囲について以下の基準を適用する．第一に，回転または反転させて重なる図形は同一の形状とみなす．第

二に、対象となる形状は凸多角形のみならず、非凸多角形や多重連結図形も包含する。

図 1.2 に示すように、タングラムと清少納言知恵の板の各ピースは、単位三角形の集合として表現できる。したがって、辺対辺の規則に従って構成される形状は、必然的にサイズ $n = 16$ のポリアボロとなる。この事実は、ポリアボロという統一的な視点から、これら 2 種類のパズルの表現力を定量的に比較分析できることを意味する。

一方で、この分析には計算量的な困難が伴う。ポリアボロの形状数は n に応じて爆発的に増加するためである。実際、サイズ n のポリアボロの総数を表すオンライン整数列大辞典 (On-Line Encyclopedia of Integer Sequences; OEIS) の数列 **A006074** によれば、 $n = 16$ のときの理論総数は 28,616,815 通りに達する [5]。現状、学術研究の多くはこの数値の算出に留まり、具体的な形状データの整備には至っていない。

上記の問題に対し、本稿の主な貢献は以下の通りである。

- **効率的フレームワークの構築**：ポリアボロの幾何学的特性に特化したグリッド表現、および同型判定を含む高速な生成アルゴリズムを設計・実装した。
- **全ポリアボロ集合のデータ化**： $n = 16$ のポリアボロ全形状（約 2,860 万種）の生成と記録を完了し、これまで欠如していた幾何学的データセットを整備した。
- **表現力の定量的解明**：タングラム・清少納言知恵の板および変種パズルの構成可能なポリアボロを全列挙した。さらに各パズルの表現力の差異を、ピースの幾何学的特性（ピースのサイズ構成・非対称性など）に基づいて定量的に分析した。

第2章 提案手法

2.1 グリッド表現と4ビットマスク符号化

ポリアボロの構成単位である直角二等辺三角形を効率的に扱うため、2次元整数グリッド \mathbb{Z}^2 上でのビットマスク表現を採用した。各グリッドセルにおける三角形の配置状態は、図 2.1 に示す4ビットの整数値 $(b_3b_2b_1b_0)$ で管理する。各ビットはセルの4辺（左、上、右、下）に対応し、その辺に直角三角形の直角辺が存在する場合にフラグを立てる。

具体的には、各ビットはセルの対応する境界辺に三角形の直角辺が存在する場合に1、そうでない場合に0をとるよう定義される。例えば、直角頂点が左上に位置する三角形は 1100_2 (12)、右下に位置するものは 0011_2 (3) と符号化される。この表現形式は、ビット演算による効率的な幾何処理を可能にする。実際、同一のグリッドセル内互いに補完的な2つの三角形の統合操作はできる。例えば論理和を用いて $1100_2 \oplus 0011_2 = 1111_2$ (15) (本稿では、ビットマスクの合成操作を表すために記号 \oplus を用いる。) と定式化され、完全な正方形が形成される様子を簡潔かつ代数的に記述できる。

また、計算時のメモリ消費量を最小限に抑えるため、グリッドの最大サイズは必要最小限に設定すべきである。その値を決定するには、図形が取り得る最大の空間的広がりを考慮する必要がある。 $n = 16$ のポリアボロにおいて、すべての単位三角形が一直線状に連結された場合、そのスパン（幅）は最大となる。幾何学的な計算によれば、この時の最大長は9単位である。グリッドの一辺の長さを L とする。したがって、すべての可能な構成を配列の境界外参照なしに格納するためには、グリッドの一辺を少なくとも9（すなわち $L \geq 9$ ）に設定する必要がある。

2.2 全列挙アルゴリズムの設計

2.2.1 探索戦略の選択

ポリアボロの全列挙を実現するには、主に以下の2つのアプローチが考えられる。

1. **全配置探索法**：固定されたグリッド内（サイズ $L \times L$ ）において、すべてのピースの配置を試行する方法である。ピース数を k 、各ピースの可能な向き数を d とすると、この手法の探索空間の計算量は $O((L^2 \cdot d)^k)$ となる。

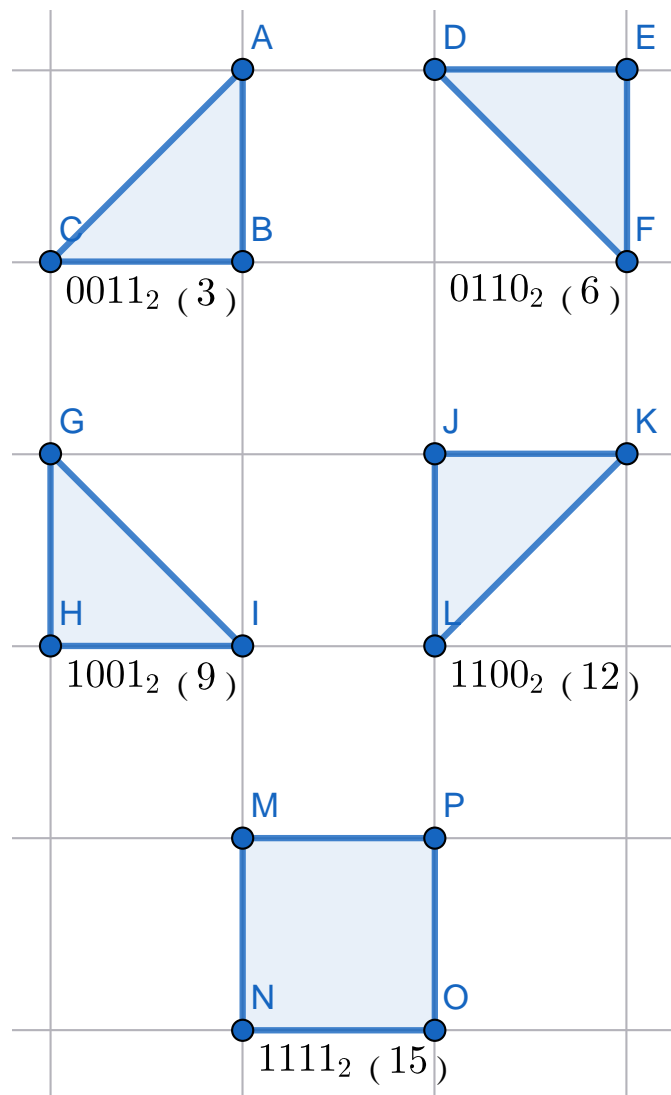


図 2.1: 4ビットマスク符号化

タングラムを例に、その理論的上限を見積もる。各ピースがグリッド内で約 9×9 通りの位置と 4 通りの向きを持つと仮定すると、状態総数は以下のようになる。

$$(9 \times 9 \times 4)^7 \approx 3.75 \times 10^{17}$$

実際には境界制約により有効な配置数はこれより少なくなるが、このオーダーは依然として既存の計算リソースで扱うには計算量が大きく、計算量的に現実的ではない。

2. **連結性に基づく逐次構成法**：任意の初期ピースを起点とし、既存の図形の辺に対して残りのピースを「逐次追加」していく方法である。各ステップにおいて、辺対辺の接続ルールを満たし、かつ重なりが生じない配置のみを展開する。

本研究では、無効な非連結状態を生成しない「連結性に基づく逐次構成法」を採用した。

本手法で網羅性を保証するにはピースの順序を考える必要がある。すなわち、追加順序は生成可能な形状の集合に本質的な影響を及ぼす、全列挙するため全順序探索が必要である。証明するため以下の例をあげる。図 2.2 に示すように、ピースの組 $\{A, B, C\}$ があり、目標形状が A と B が C を介して接続される構造（すなわち A と C が直接接しない $A-B-C$ 構造）であると仮定する。もし $A \rightarrow C \rightarrow B$ の追加順序を採用した場合、第 2 ステップで C を配置する際、辺対辺の接続ルールに従うと、 C は A と直接接続しなければならない。その結果、 A と C が離れた中間状態を生成できず、最終的に $A-B-C$ という目標構造を形成することは不可能となる。これは、異なる追加順序が異なる幾何トポロジー構造を導くことを示している。したがって、全列挙を実現するには、ピースのすべての順序を網羅する必要がある。

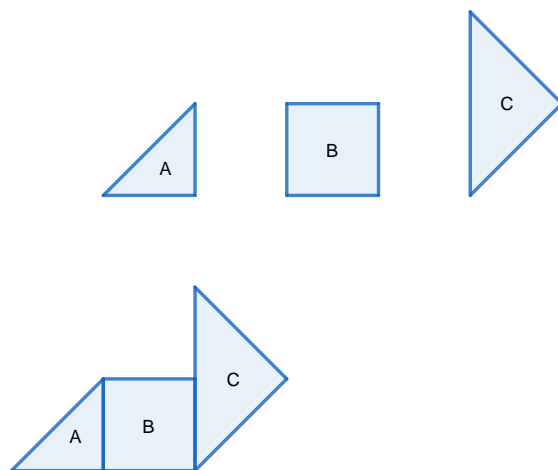


図 2.2: ピース順序の重要性の証明

単純な全順序探索は計算コストが高い。そこで、探索空間を効果的に抑制するために、以下の 2 つの基本戦略を適用した。

第一に、**初期ピースの固定**である。ポリアボロの連結性に基づけば、任意のピースを起点としても全形状を構成可能であることがグラフ理論的に保証される。証明は以下に示す。

- **命題**：辺隣接ルールによって構成される任意の連結パズル形状 G は、その内部の任意のピース v_{start} を初期ピースとし、隣接するピースを順次追加していくことで構成可能である。
- **証明（背理法）**：ある正当な連結パズル形状 G が存在と仮定する。その G のあるピース v_{start} を起点とした場合、隣接拡張によって G 内のすべてのピースを網羅できない。これは、 G 内に少なくとも 1 つのピース v_{target} が

存在し、 v_{start} から v_{target} へ至る隣接経路が存在しない（すなわち G が非連結である）ことを意味する。

しかし、ポリアボロの連結性の定義により、 G の双対グラフ（ピースを頂点とし、辺共有をエッジとするグラフ）は必然的に連結グラフとなる。連結グラフの性質上、任意の2頂点間には必ず経路が存在する。したがって、 v_{start} は有限ステップの隣接関係を通じて G 内の任意のピース v_{target} に到達可能である。これは仮定と矛盾する。したがって、仮定は誤りであり、命題は真である。

よって初期ピースを1つ固定し、残りの順列のみを探索対象とすることで、計算量を大幅に圧縮（ $7! \rightarrow 6!$ ）した。

第二に、署名による枝刈りである。各ステップで形状を正規化し、一意な符号化データ（署名）を用いて重複を厳密に排除することで、探索木の無駄な分岐を効果的に抑制した。

しかし、 $n = 16$ の場合、これらの基本戦略のみでは依然として探索空間が膨大である。そこで本研究では、計算効率とメモリ管理をさらに改善するため、以下の2つの実装レベルの最適化手法を導入した。

1. **動的バウンディングボックスによる探索空間の枝刈り**：単純な実装では、新規ピースの配置候補としてグリッド全体 ($O(L^2)$) を走査する必要がある。しかし、新規ピースは必ず既存形状と接するという「辺接続」の制約を考慮すれば、探索範囲は大幅に限定できる。具体的には、現在の図形の最大座標（以下 x_{max}, y_{max} とする）を動的に更新し、アンカー点の走査範囲 \mathcal{R} を以下のように制限する。

$$\mathcal{R} = [0, x_{max} + 1] \times [0, y_{max} + 1]$$

現在の図形のバウンディングボックスの幅と高さを、それぞれ w, h とする。この戦略により、広範囲スキャンは $O(wh)$ の局所探索へと変換され、無効な計算回数が大幅に削減される。

2. **最終段階における直接挿入戦略**：通常の探索ステップでは、生成された候補形状を一旦中間リストに格納し、ステップ完了後にマージを行う。しかし、最終ピース（第7ピース）を追加する段階では、生成される形状数が数百万オーダーに達するため、中間リストの構築だけで膨大なメモリを消費する。そこで最終ラウンドに限り、中間リストを介さず、生成された署名を直ちに最終結果セット S と照合する戦略を採用した。

- 署名が S に存在しない場合：即座に追加する。
- 既に存在する場合：即座に破棄する。

これにより、一時保存に伴うメモリ消費とオーバーヘッドを排除し、大規模データの効率的な処理を実現した。

2.2.2 アルゴリズムの全体像

以上の分析に基づき、本研究で構築した全列挙の処理手順をアルゴリズム 1 に示す。

アルゴリズム 1 ポリアボロ全列挙アルゴリズム (Main Enumeration)

入力: ピース集合 $P = \{p_1, \dots, p_6\}$, 初期ピース P_{init}

出力: 全ポリアボロ形状集合 $\mathcal{S}_{\text{final}}$

```
1:  $\mathcal{S}_{\text{final}} \leftarrow \emptyset$ 
2: for all permutation  $\pi$  of  $P$  do ▷ すべての順列に対して
3:    $\mathcal{L}_{\text{cur}} \leftarrow \{P_{\text{init}}\}$ 
4:   for  $step \leftarrow 0$  to 5 do
5:      $p_{\text{next}} \leftarrow \pi[step]$ 
6:      $\mathcal{L}_{\text{next}} \leftarrow \emptyset$ 
7:     for all  $S \in \mathcal{L}_{\text{cur}}$  do
8:        $Candidates \leftarrow \text{GENERATENEWSHAPES}(S, p_{\text{next}})$ 
9:       for all  $C \in Candidates$  do
10:         $\sigma \leftarrow \text{GETSIGNATURE}(C)$ 
11:        if  $step < 5$  then ▷ 通常ステップ: 中間リストを使用
12:          if  $\sigma \notin \mathcal{L}_{\text{next}}$  then
13:             $\mathcal{L}_{\text{next}} \leftarrow \mathcal{L}_{\text{next}} \cup \{\sigma\}$ 
14:          end if
15:        else ▷ 最終ステップ: 直接挿入戦略 (Optimization)
16:          if  $\sigma \notin \mathcal{S}_{\text{final}}$  then ▷ 直接大域リストと照合
17:             $\mathcal{S}_{\text{final}} \leftarrow \mathcal{S}_{\text{final}} \cup \{\sigma\}$ 
18:          end if
19:        end if
20:      end for
21:    end for
22:     $\mathcal{L}_{\text{cur}} \leftarrow \mathcal{L}_{\text{next}}$ 
23:  end for
24: end for return  $\mathcal{S}_{\text{final}}$ 
```

2.3 新規形状の生成アルゴリズム

本節では、既存の形状 C に対し、新たなピース p を正当に追加するための具体的な手順を説明する。

2.3.1 変種の事前計算と配置走査

網羅的な全列挙を行うには、追加ピース p のあらゆる向き（回転・反転）を考慮する必要がある。グリッドモデル上では、各ピースは最大8通り（回転4通り × 反転2通り）の幾何学的変種を持つ。探索の都度これらを計算するのは非効率であるため、本手法ではシステム初期化時にすべての変種 $V(p)$ を一括生成し、ルックアップテーブルとして保持する戦略を採用した。

実際の生成フェーズでは、ベース形状 C を構成するグリッド座標をアンカー点として走査し、各アンカー点に対してテーブルから参照した全変種 $v \in V(p)$ の配置を試行する。

2.3.2 ルックアップテーブル (LUT) による高速な隣接判定

ある変種 v がアンカー点 (x, y) に正当に配置できるか判定するには、以下の2つの幾何学的制約を同時に満たす必要がある。

1. **非衝突性**： v が占有するすべてのセルは、 C 内の既存のセルと（幾何学的補完関係にある場合を除き）重なってはならない。
2. **連結性**： v を構成するブロックのうち、少なくとも1つが C の既存境界と辺対辺で接合していなければならない。

本稿では、判定プロセス（特に連結性判定）を加速するために、効率的な**隣接ルックアップテーブル**を設計した。このLUTは3次元ブール配列 $LUT[T_{target}][D][T_{neighbor}]$ として定義され、次元は $16 \times 4 \times 16$ である。各次元の意味は以下の通りである。

- $T_{target} \in [0, 15]$ ：ターゲット（配置済み）グリッド内の図形タイプ符号。
- $D \in [0, 3]$ ：隣接方向（0: 上, 1: 下, 2: 左, 3: 右）。
- $T_{neighbor} \in [0, 15]$ ：隣接（新規配置）グリッド内の図形タイプ符号。

このテーブルはシステム初期化段階で事前に計算される。これにより、任意の2種類の単位三角形タイプが特定の方向で接触した際、それが有効な辺接続を構成するか否かを $O(1)$ で照会可能となる。

2.3.3 幾何学的補完による例外処理

衝突検出においては、幾何学的補完性に基づく例外処理を導入する。通常、 C のあるグリッド座標 (x_i, y_i) が既に占有されている（すなわち $Type \neq 0$ ）場合、配置は衝突とみなされる。しかし、その位置における p の三角形タイプと C の既存タイプが、互いに補完関係にある場合（例：左上 1100_2 と右下 0011_2 ）、両者は衝突せず、統合されて完全な単位正方形（ 1111_2 ）を形成するとみなす。この配置は正当（Valid）であり、かつ両者は内部で斜辺を共有して接合するため、連結性の要件も自動的に満たされる。

2.3.4 生成処理のアルゴリズム記述

前述の議論に基づく新規形状生成の具体的な手順を、アルゴリズム 2 に示す。

2.4 形状の正規化と同型判定

生成された形状が幾何学的に同一であることを判定するため、平行移動・回転・反転の自由度を排除した一意な表現形式を定義する。

2.4.1 相対座標による正規化

平行移動不変性を確保するため、形状 S を構成するブロック群の座標を、そのバウンディングボックスの左上を原点とする相対座標系へ変換する。形状 S の最小座標を (x_{\min}, y_{\min}) とするとき、各ブロックの座標 (x_i, y_i) は以下のように変換される。

$$x'_i = x_i - x_{\min} + \delta, \quad y'_i = y_i - y_{\min} + \delta \quad (2.1)$$

ここで δ は、実装上の配列表現においてインデックスの非負性を保証するためのオフセット定数である。本実験では $\delta = 2$ とした。

ここで選択したオフセット定数 $\delta = 2$ の正当性は、本研究の条件下において、1つのピースを追加した際に生じる座標延伸の最大値に基づいている。その幾何学的根拠は以下の通りである。

条件 1：グリッド上の延伸特性

固定された面積を持つピースが、任意の図形に最も長く延伸する配置には、2つのパターンが存在する。第一は、単位三角形を一行に並べる配置（図 2.3 参照）である。この場合、ピースの面積が 2つの単位三角形ごとに長さが 1 延伸する特性を持つ。

アルゴリズム 2 新規形状生成アルゴリズム (GenerateNewShapes)

入力: ベース形状 C , 追加ピース p

出力: 合法的な新形状集合 $\mathcal{S}_{\text{valid}}$

```
1:  $\mathcal{S}_{\text{valid}} \leftarrow \emptyset$ 
2:  $\mathcal{V} \leftarrow \text{GETVARIANTS}(p)$ 
3: for all  $V \in \mathcal{V}$  do
4:   for all anchor point  $(x, y)$  in  $C$ 's bounds do
5:      $IsValid \leftarrow \text{true}$ 
6:      $IsConnected \leftarrow \text{false}$ 
7:     for all block  $b \in V$  do
8:        $(x', y') \leftarrow (x + b_x, y + b_y)$  ▷ 絶対座標を計算
9:        $t_{\text{exist}} \leftarrow C[x'][y']$ 
10:       $t_{\text{block}} \leftarrow b_{\text{type}}$ 
11:      Check 1: 衝突判定
12:      if  $t_{\text{exist}} \neq 0 \wedge t_{\text{exist}} \neq \text{COMPLEMENTARY}(t_{\text{block}})$  then
13:         $IsValid \leftarrow \text{false}$ 
14:        break
15:      end if
16:      Check 2: 連結性判定
17:      if  $t_{\text{exist}} = \text{COMPLEMENTARY}(t_{\text{block}})$  then
18:         $IsConnected \leftarrow \text{true}$ 
19:      else if  $\text{CHECKNEIGHBORSLUT}(C, x', y', t_{\text{block}})$  then
20:         $IsConnected \leftarrow \text{true}$ 
21:      end if
22:    end for
23:    if  $IsValid \wedge IsConnected$  then
24:       $S_{\text{new}} \leftarrow \text{COMBINE}(C, V, x, y)$ 
25:       $S_{\text{norm}} \leftarrow \text{NORMALIZE}(S_{\text{new}})$ 
26:      if  $S_{\text{norm}} \notin \mathcal{S}_{\text{valid}}$  then
27:         $\mathcal{S}_{\text{valid}} \leftarrow \mathcal{S}_{\text{valid}} \cup \{S_{\text{norm}}\}$ 
28:      end if
29:    end if
30:  end for
31: end for return  $\mathcal{S}_{\text{valid}}$ 
```

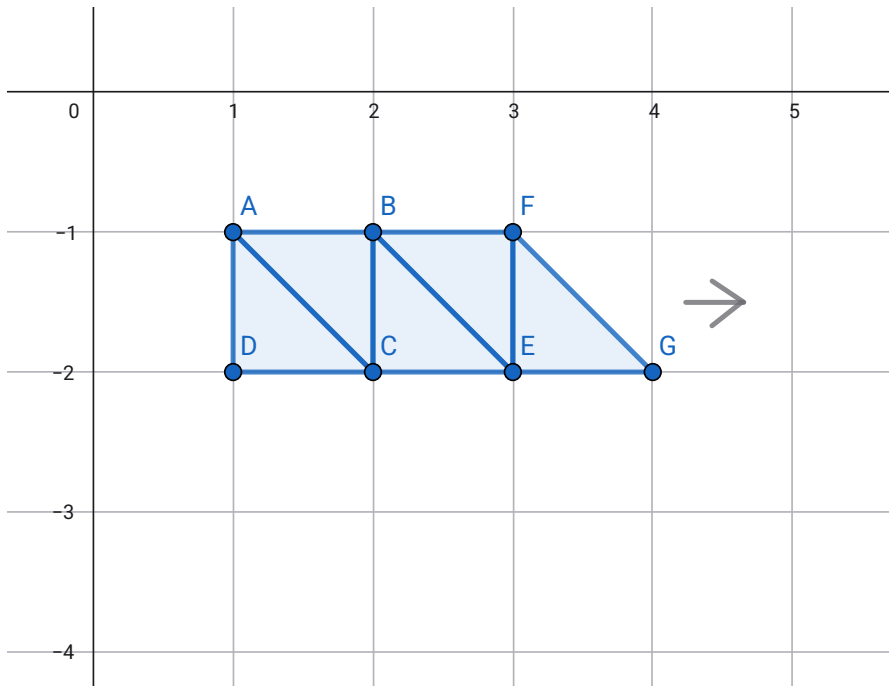


図 2.3: ピースの最も長く延伸する配置パターン 1

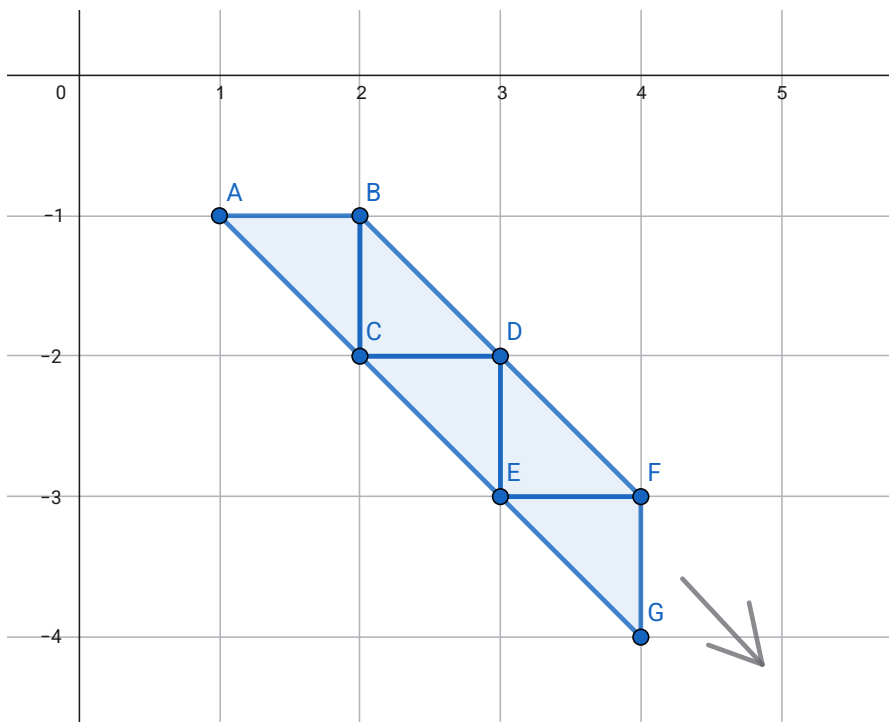


図 2.4: ピースの最も長く延伸する配置パターン 2

第二は、単位三角形を階段状（ジグザグ）に配置する方法（図 2.4 参照）である。この配置では、ピースの中で単位三角形が追加されるごとに、横方向および縦方向の長さが連鎖的に 1 単位を増加する。

条件 2：ピースの最大面積

本論文で対象とするパズルのピース集合において、最大のピース面積は単位三角形 4 個分である。

これら 2 つの条件を統合すると、対象となるピースがいかなる形状・配置をとったとしても、既存の形状に対する座標の延伸幅は、単一方向あるいは x, y 双方向において最大で 2 に留まることが確認できる。したがって、 $\delta = 2$ と設定することで、正規化後のいかなる形状追加においても、すべての座標が正の区間に収まることが保証される。

2.4.2 最小署名の導出

回転および反転による重複を排除するため、以下の手順で一意的な署名を生成する。

1. **変種の展開**：正規化された形状 S に対し、ユークリッド変換（回転 4 通り × 反転 2 通り）を適用し、8 つの幾何学的変種セット $\{V_1, \dots, V_8\}$ を生成する。
2. **ブロック列の整列**：各変種 V_k 内部において、構成ブロックの格納順序による不確定性を排除するため、ブロックの座標とタイプに基づきデータを昇順にソートする。
3. **最小列の選択**：整列された 8 つの変種シーケンスを辞書順で比較し、最小となるシーケンスを**最小署名**として採用する。

この手順により、同一形状は生成プロセスや向きに関わらず必ず同一の署名を持つことが保証される。なお、署名のハッシュ化および格納には、衝突率が低く計算コストの低い FNV-1a アルゴリズム [4] を用いた。

第3章 実験結果

3.1 実験環境と実装詳細

- CPU : Apple M2 Pro
- メモリ : 16 GB Unified Memory
- 実装言語 : C++ 14
- コンパイラ : Apple clang version 17.0.0 (clang-1700.6.3.2)
- コンパイルオプション : -O3 (最適化レベル 3)
- OS : macOS (Sequoia 15.7.3)
- 出力データ形式 : 実験結果はテキストファイル形式で保存した。各行には、1つの形状を構成するすべてのセルのグリッド座標 (x, y) および $Type$ 値をカンマ区切りで列挙している。この詳細な記録により、形状の完全な復元および可視化が可能である。

3.2 形状全列挙の結果

前述の高速化アルゴリズムに基づき、本研究では $n = 16$ のポリアボロ全集合、およびタングラム・清少納言知恵の板を含む変種パズルの構成可能形状の全列挙を実施した。

3.2.1 アルゴリズム正当性の検証 ($n = 16$ ポリアボロ)

まず、アルゴリズムの完備性と正確性を検証するため、16個の独立した単位三角形を入力とし、形状制約を設けず全列挙した。

計算の結果、 $n = 16$ におけるポリアボロの一意な形状総数は **28,616,815** となった。この数値は、OEIS の数列 **A006074** における $n = 16$ の値 [5] と完全に一致する。

ここで注目すべき点は、OEIS の **A006074** の既存データは主に数え上げ理論に基づく数値であり、具体的な幾何形状のデータセットとしては公開されていなかった。

表 3.1: タングラムと清少納言知恵の板の構成可能形状数の比較

パズル集合	構成可能形状総数	相対倍率
タングラム (Tangram)	5,583,516	1.00
清少納言知恵の板 (Sei Shonagon Chie-no-Ita)	10,889,227	1.95

た点である。本実験の結果は既存の数値と一致しており、その正しさを実証的に確認できた。この事実は、本稿で提案したグリッド生成と重複排除アルゴリズムが高い正確性で動作していることを強く裏付けるものである。

3.2.2 タングラムと清少納言知恵の板の比較

本研究はタングラム・清少納言知恵の板で構成可能なポリアボロを全列挙した。全列挙の結果は、その構成能力に顕著な差異があることを明らかにした。具体的な集計結果を表 3.1 に示す。

表 3.1 の結果より、清少納言知恵の板の構成可能形状数はタングラムの約 **1.95 倍** に達する。具体的には、 $n = 16$ のポリアボロ (28,616,815 種) のうち、タングラムの網羅率が約 19.5% に留まるのに対し、清少納言知恵の板は約 38.1% を構成可能である。両者は共に総面積 16 の同一な構成要素を持つにもかかわらず、生成数には倍近い開きがある。この事実は、清少納言知恵の板が幾何学的表現力および組合せ自由度において、タングラムを大きく上回っていることを定量的に裏付けるものである。

3.2.3 データセットおよび Web ビューアの公開

本実験で生成されたタングラムと清少納言知恵の板全列挙データセット、および $n = 16$ のポリアボロ全集合の図形を公開した。さらに、形状データを確認できるように、専用の Web ベース可視化ツールを開発した。以下の URL からアクセス可能である。

- タングラムと清少納言知恵の板全列挙データセット：

<https://leeono8.github.io/polyabolo-viewer/>

- $n = 16$ のポリアボロ全集合データセット：

<https://leeono8.github.io/n16polyaboloview1/>

<https://leeono8.github.io/n16polyaboloview2/>

<https://leeono8.github.io/n16polyaboloview3/>

表 3.2: タングラム, 清少納言知恵の板および変種パズルのコンポーネント構成と生成結果統計

ピースタイプ (面積)	T	S	Sp.A	Sp.B	Sp.C	Sp.D	Sp.E	Sp.F	Sp.G
1. 単位三角形 ($u, 1$)	2	1	3	2	2	1	2	2	2
2. 正方形 (2)	1	1	-	-	-	-	-	-	-
3. 中三角形 (2)	1	2	1	2	2	2	1	1	1
4. 平行四辺形 (2)	1	1	-	1	-	2	2	-	-
5. 直角台形 (3)	-	1	1	-	2	1	-	4	3
6. 大三角形 (4)	2	-	-	-	-	-	2	-	-
7. 長平行四辺形 (4)	-	-	2	2	1	1	-	-	-
8. 等脚台形 (4)	-	1	-	-	-	-	-	-	-
9. 小等脚台形 (3)	-	-	-	-	-	-	-	-	1
計算時間 (秒)	1600.14	9879.57	1733.72	4241.49	7710.09	11252.9	1703.17	1754.82	9565.51
ポリアボロ生成総数	5.58M	10.89M	6.42M	5.12M	10.15M	14.37M	7.08M	9.99M	17.96M

3.3 結果分析

ピースの幾何学的構成がパズルの表現力に与える影響をより詳細に検証するため, 本研究ではタングラムと清少納言知恵の板に加え, 計7種類の変種パズル (Special Puzzle A-G) についても同様に全列挙を行った. 対象とした変種パズルは, Fox-Epstein ら [3] 発見した4種「19種類の凸多角形を構成可能な、4つの最適な7片のセット」(Sp.A–D) である. 加え比較のために本研究で独自に設計した3種 (Sp.E–G) パズル (図 3.1 参照).

これらの実験データを比較分析した結果, 各パズルのポリアボロ生成数には顕著な差異が認められた. この事実は, 構成可能な形状の総数が, 単なるピースの数や面積の合計ではなく, 各ピースの幾何学的特性 (分割構造や対称性など) に強く依存していることを示唆するものである.

各パズルのコンポーネント構成, およびポリアボロ生成数の集計結果を表 3.2 に示す.

3.4 結果分析：構成数を左右する主要な幾何学的要因

データ分析の結果, パズルコンポーネントの幾何学的性質が構成空間の容量に顕著な影響を与えることが示された. 本稿では, これらの影響を以下の3つの主要因に集約する.

3.4.1 要因1：大面積ピースによる配置制約

実験データは, 大面積のコンポーネントの存在と構成数の間に負の相関関係があるという傾向を示した.

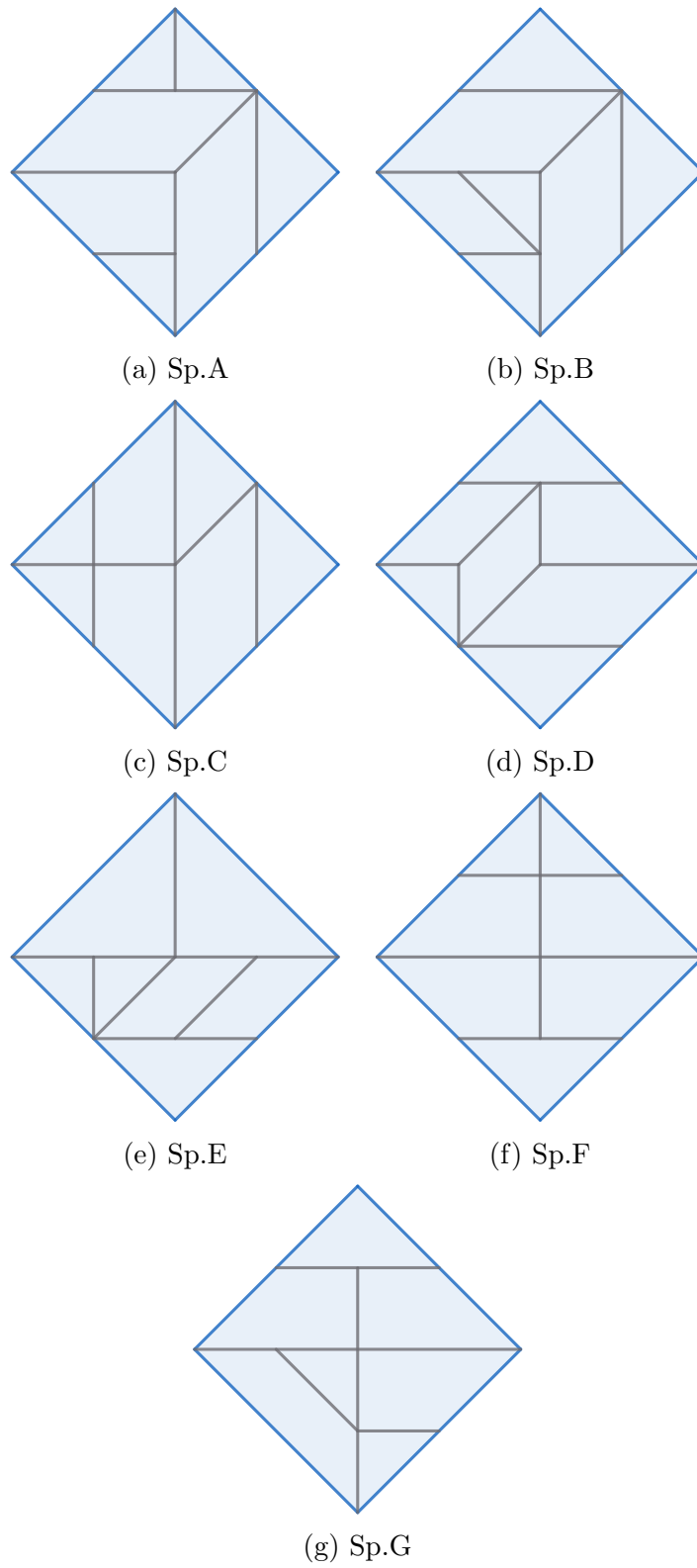


図 3.1: 変種パズル (Sp.A – Sp.G) の形状一覧

- **観察結果**：生成数が少ないグループ（タングラム, Special A, B, E など）は、いずれも面積 4 の大きなブロック（大三角形や長平行四辺形）を 2 つ含んでいる。これらのグループの生成数は 800 万件を超えない。対照的に、生成数が多いグループ（清少納言知恵の板, Special D, G など）は、大きなブロックを少なくとも 1 つ排除し、コンポーネントの小ピース化を実現している。特にすべての大ブロックを排除した **Special G** は、生成数が 1,796 万件に実現した。
- **分析と考察**：この結果は、最大コンポーネントの面積が充填プロセスにおける「隙間の埋めやすさ」を制限していることを示唆する。パズルシステム内に大面積のブロックが存在する場合、残りの空間に対して境界適合を行う際、より強い幾何学的制約を受けることになる。逆に、小ピース（面積 ≤ 3 ）コンポーネントは、複雑な局所境界に対して柔軟に適応できるため、正当な構成の生成数を著しく増大させる。

3.4.2 要因 2：ピース形状の多様性がもたらす組合せの増加

Special F と Special G のデータを比較すると、**ピース形状の多様性が構成空間の拡大に寄与することがわかる。**

- **観察結果**：Special F は 4 つの完全に同一な直角台形から構成され（高度に同質化）、生成数は約 998 万件である。一方、Special G は F 中の直角台形 1 つを、面積は等しいが形状が異なる等脚台形に置換しただけで（形状の多様性の導入）、生成数は 1,795 万件へと**増加した。**
- **分析と考察**：この現象は構成要素の形状の多様性が高まることで対称性が低下し、幾何学的に等価とみなされる重複配置が減少したことによって説明できる。同質化されたコンポーネントは、接続において**対称性による重複**を生じやすく、異なる配置であっても幾何学的に等価とみなされやすいため、有効解の数が制限される。一方、異なるタイプの形状の多様性を導入すると、この重複が解消され、より多様な幾何学的補完界面が提供されるため、潜在的な接続パターンが拡張される。

3.4.3 要因 3：非対称な形状の有効性

実験によれば、非対称な形状は、高度に対称な形状と比較して、より多くの構成を生み出す性質を持つことが確認された。

- **観察結果**：面積構成が同一の（共に大三角形 2 個を含む）対照群において、Special E（非対称な平行四辺形を使用）の生成数（707 万件）は、タングラム（全対称な正方形を使用、558 万件）よりも明らかに多い。

- **分析と考察**：正方形は高度な回転対称性を持ち，グリッド内では実質的に1種類の状態しかとり得ない．一方，平行四辺形は**非対称性**を持ち，反転操作（裏返し）によって異なる方向性を獲得できる．この方向の可変性は，幾何学的制約を緩和し，より多くの斜め方向や非対称な境界条件への適合を可能にする．その結果，正当な構成総数が増加する．

第4章 結論と展望

4.1 結論

本研究では、「タングラムと清少納言知恵の板の幾何学的表現力の差異」という古典的な問いに対し、 $n = 16$ のポリアポロに基づく全列挙計算フレームワークを構築し、定量的な評価を行った。具体的には、4ビットグリッド符号化モデルの設計、動的バウンディングボックスによる探索空間の枝刈り、および最終段階における直接挿入戦略の導入を通じて、単一の計算機環境下で数千万規模の幾何形状の生成と判定に成功した。

本研究の主な貢献と結論は以下の通りである。

1. **データの完全性**: 本研究において初めて、 $n = 16$ ポリアポロ全集合 (28,616,815 種) の形状生成を完了し、OEIS の **A006074** 数列の正当性を実証データに基づいて検証した。
2. **表現力の定量的解明**: 全列挙の結果、清少納言知恵の板の構成可能形状数 (10,889,227 種) は、タングラム (5,583,516 種) の約 **1.95 倍** に達することが確認された。このデータは、前者が複雑なシルエット (鳥獣や植物など) を表現する際になぜ高い自由度を持つのか、その幾何学的根拠を明確に与えるものである。
3. **幾何学的決定要因の特定**: 変種パズルを用いた対照実験を通じ、ピースのサイズ構成、多様性、形状の対称性がパズルの表現力を決定する3つの主要因であることを明らかにした。特に「大面積ブロックの排除」と「ピース形状の多様性の導入」は、構成空間を最大化するための有効な戦略であることが示された。

4.2 今後の展望

本稿では $n = 16$ の全列挙を実現したが、ポリアポロおよびシルエットパズルの数理的解析には、依然として広範な拡張の余地がある。

1. **形状のトポロジック特徴の分析**: 現在の研究は形状の「数量」の統計に留まっている。今後は形状の「質」、例えば**多重連結図形 (穴あき形状)** の出現頻

度の統計や、周長分布に基づく形状のコンパクトさの分析する余地がある。これはパズルの幾何学的な特性をより多角的に理解するために重要である。

2. **より高次 ($n > 16$) への拡張**： n の増加に伴い、ポリアボロの数は指数関数的に増大する。本研究では $n = 16$ のポリアボロを対象としたが、より高次の系における全列挙の可能性について議論する。具体例として、7片からなる「ラッキーパズル」(図 4.1) が挙げられる。一見すると、このパズルの総面積は単位三角形換算で $n = 20$ 相当に見える。しかし、構成片に含まれる微小な三角形(左下および右下のピースなど)を最小単位として厳密に定義すると、その総面積は $n = 40$ に達する。 $n = 40$ の全列挙は、計算量の観点から現状のリソースでは困難である。

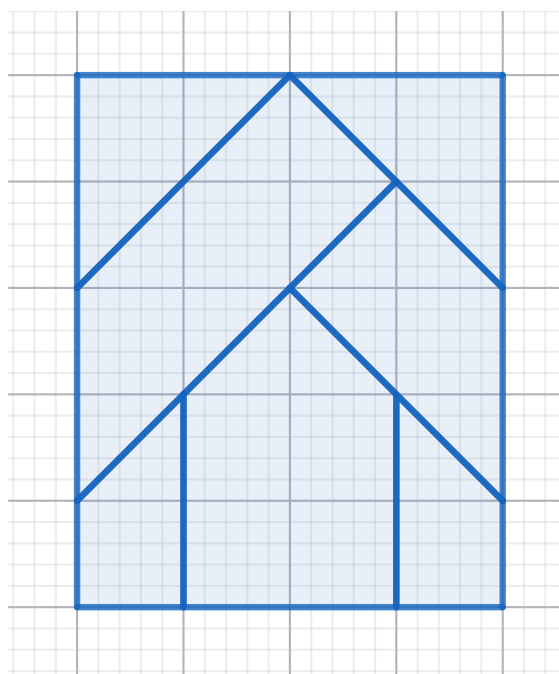


図 4.1: ラッキーパズル

ラッキーパズルに適したグリッドサイズおよび正規化座標へのパラメータ調整を行い、本プログラムによる試行計算を実施した。その結果、全列挙における最初の探索順序においてさえ、列挙を完了することができなかった。具体的には、6枚目のピースを配置した段階で中間状態が約 2,690 万種に達し、続く 7枚目の配置プロセスにおいてメモリおよび仮想メモリの不足によりプログラムが異常終了した。終了直前のメモリ使用量は 30GB を超えており、この探索順序だけでも少なくとも 3 億通り以上の組み合わせが存在すると推測される。

以上の結果から、現行のアルゴリズムのみでラッキーパズルの全列挙を完遂することは極めて困難であり、今後はさらなる計算効率の向上や新たなア

アプローチの導入が不可欠である。

3. **パズル設計の最適化（逆問題への応用）**：本稿で得られた「幾何学的要因」の知見に基づき、将来的に「パズルオプティマイザ」の構築が期待される。すなわち、目標とするシルエット集合（一般的な動物の形状など）が与えられたとき、それらを最も多くカバーできる「最適な7ピースの組み合わせ」を探索する。これは、パズル設計を一種の**逆問題**として定式化する試みである。

参考文献

- [1] J. Slocum. Tangram: the world's first puzzle craze, Published by Sterling (2003)
- [2] F. T. Wang and C. C. Hsiung. A theorem on the Tangram, *The American Mathematical Monthly*, vol. 49, no. 9, pp. 596–599 (1942)
- [3] E. Fox-Epstein, K. Katsumata, and R. Uehara. The convex configurations of “Sei Shonagon Chie no Ita,” tangram, and other silhouette puzzles with seven pieces, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 6, pp. 1084–1089 (2016)
- [4] Glenn Fowler, Landon Curt Noll, and Phong Vo. The FNV Hash, FNV-1a Algorithm Specification, <http://www.isthe.com/chongo/tech/comp/fnv/>, Accessed: 2025-11-29
- [5] OEIS Foundation Inc. Sequence A006074: Number of polyaboloes with n cells, <https://oeis.org/A006074>, Accessed: 2025-11-29