

裁ち合わせパズルの重複解なし全解探索システム

五十嵐 力 但馬 康宏 小谷 善行
東京農工大学

概要

ペントミノやヘキサミノのような箱詰めパズルの全解探索問題において、探索量の削減が大きな課題となっている。同一の解が回転や反転などによって複数現れてしまうと、出力される解の数が数倍になり、それに応じて探索量も倍加する。そこで本研究では、裁ち合わせパズルの全解を重複解が出ないように求める。裁ち合わせパズルとは、異なる二つの図形(図形1、図形2)に対し、図形1を任意数に分割し、分割したピースによって図形2を構成するパズルである。本稿では、図形1を重複なくすべての可能性で分割し、分割したピースを用いて図形2に重複なく箱詰めする手法を述べる。本システムにおいて、いくつかの裁ち合わせパズル問題に対して重複なく全解を求められることを示す。

A Total Solver of Dissection/Put-Together Shape Puzzles

Chikara IGARASHI Yasuhiro TAJIMA Yoshiyuki KOTANI
Tokyo University of Agriculture and Technology

Abstract

Decreasing amount of search is a big problem for a total solver of puzzle such as pentomino, hexomino and so on. The number of results that system output and the amount of search becomes several times when the system calculates overlapping solutions.

In this paper, we propose a method of a total solver of dissection/put-together shape puzzles. Dissection/put-together shape puzzle is a puzzle that composes another figure by dividing an arbitrary figure. We show that this system doesn't output overlapping solutions and all solutions are outputs for any dissection/put-together shape puzzle problems.

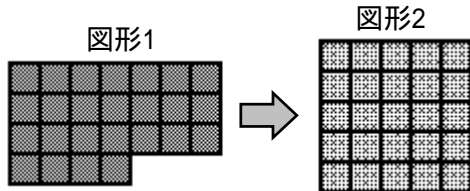
1. はじめに

ペントミノやヘキサミノなどの箱詰めパズルの全解を求めるような問題における解探索の効率化として、解の重複を排除することが挙げられる。6×10ペントミノの場合、解の回転、反転による重複を計算すると全解数が8倍にもなり、それに伴って探索量も増大する。これを排除するために、局面表を用いて一度計算した

状態を保持しておき、同じ状態が再び現れたときに再度計算しない手法もあるが、同じ状態が何度も現れるような計算をしないようにすることが望ましい。そこで本研究では、裁ち合わせパズルの全解を局面表などを用いることなく解の重複が現れないように求めることを目的とする。

2. 裁ち合わせパズル

裁ち合わせパズル[1]は、ある図形をN分割し、分割したピースを結合して別の図形にするパズルである。本稿では以後、分割する図形を図形1、分割ピースを結合して作成する図形を図形2と呼称する。図2.1に裁ち合わせパズルの問題例、図2.2にその解例を示す。



分割数 = 3

図2.1 裁ち合わせパズル問題例[1]

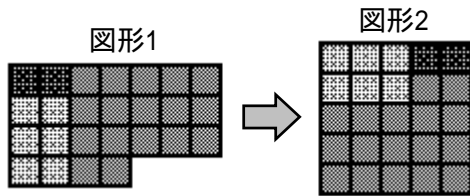


図2.2 裁ち合わせパズル解例

なお、本稿で扱う裁ち合わせパズルは[1]に倣い以下のルールに従って全解を求める。

- (1) ピースはマスに沿って分割する
- (2) 分割したピースを裏返さない
- (3) 図形2のピース配置が同じでも、図形1での分割が異なれば別解とする
- (4) 図形1の分割が同じでも、図形2のピース配置が異なれば別解とする
- (5) 図形を回転して同じ形なら重複解とする
- (6) 図形を反転して同じ形なら別解とする

図2.3、図2.4、図2.5にルールごとの解例を示す。

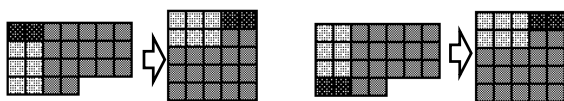


図2.3 ルール3の例(別解)

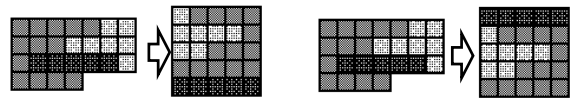


図2.4 ルール4の例(別解)

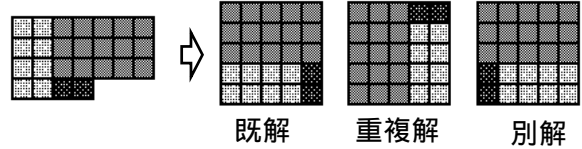


図2.5 ルール5, 6の例

3. 裁ち合わせパズル解探索システム

図3.1に裁ち合わせパズル解探索システムの処理の流れを示す。

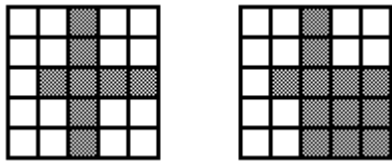
1. 図形1、図形2、分割数Nを入力
2. 図形1と図形2に入りうるピースを列挙
3. 列挙したピースに未使用のフラグを立てる
4. すべてのメインピースを大きい順に図形1に全通り入れる
 - 4.1 図形1の残りをN-1分割
 - 4.2 N分割されたピースにおいて図形2に箱詰め全解探索 解を出力する
 - 4.3 全解探索終了後4に戻る
5. メインピースに使用済フラグを立てる
6. メインピースがなくなれば終了

図3.1 裁ち合わせパズル解探索システムの処理の流れ

3.1 ピースの列挙

本システムでは、あらかじめ図形1と2に入りうるピースを列挙しておく。まず図形1と図形2を任意の位置で重ね合わせたときにできる重複部分のひとかたまりをピースとして登録する。このときできる最大サイズのピースが図形1と図形2を構成する最大サイズSのピースとなる。次に、最大サイズSのピースから1マスを取り除き、サイズS-1のピースを作る。このようにしてサイズを小さくしていき、サイズ1のピースができるまで繰り返す。サイズ1のピースまで登録し終わったら、登録したすべてのピースについて図形1および図形2に入れたときに分割数を超えないかどうかをチェックする。例えば図3.2左のようなピースの場合、どの場所に入れても分割数が4以上になってしまう

う。どの場所に入れても分割数を超えるようならば解として用いられないことがないのは自明であるため、この時点で排除する。

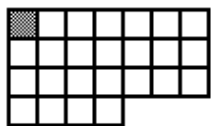


分割数3にならない 分割数3になる

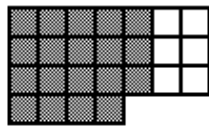
図3.2 ピース列挙における分割数考慮

3.2 メインピースのマッチング

図形1の分割において、サイズが最大となるピースをメインピースと呼び、まずメインピースを図形1に入れる。これは、最初に大きなピースを入れることで残りのピースサイズが小さくなり、分割の仕方の数が少なくなるためである(図3.3)。



サイズ1が最初



サイズ19が最初

サイズ24を2分割

サイズ6を2分割

図3.3 メインピースによる分割処理の違い

メインピースは、分割したピースの中で最大でなければならない。このため、メインピースとなりうるピースは図形1のサイズ÷分割数(小数切り上げ)以上のサイズになる。図2.1の問題ではサイズ9が最小のメインピースである。

3.3 残り図形の分割

メインピースを図形1に入れた後、図形1の残りの部分を分割数 $N-1$ に分割する。これは、図3.5に示すように、最初の n 個のピースの図形の集合体を、 $n-1$ 個のピースの集合体と1個の確定ピースとに分ける。 $n-1$ 個のピースの集合体が複数図形であれば、それぞれの図形に対して分割数を付与し、それぞれの図形に対して同様の分割を行っていく。 $n-1$ 個のピースの集合体が単一図形であれば、ピースを一つ入れて確定し、

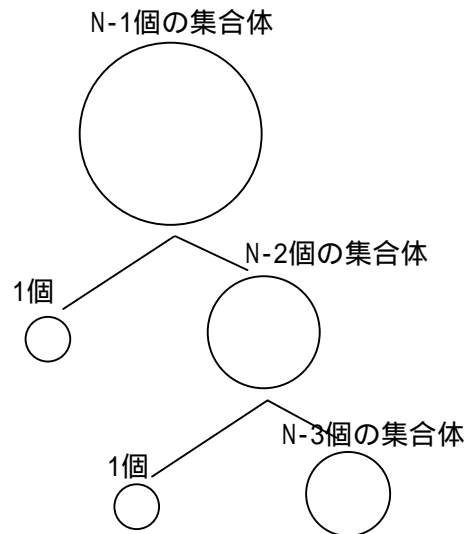


図3.5 残り図形の分割

残り図形に対して $n-2$ 分割する。これを再帰的に繰り返し、分割するものがなくなるまで、すなわち残り図形がなくなったときに分割が終了する。

3.3.1 複数図形の分割

分割した際に残りの図形が2個(図形a、図形b)あり、これを4個に分割するとする。このとき、図形aを1、図形bを3分割という分け方と、図形aを2、図形bを2分割という分け方、図形aを3、図形bを1分割という分け方の3通りが存在する(図3.6)。裁ち合わせパズルの全解を求めるには図形1のすべての切り分けを試みる必要があるため、このような複数ピースの分割の可能性をすべて網羅する必要がある。そこで本システムでは、 n_1 個の図形を n_2 個に分割するため、 n_1 個の図形それぞれに分割数を付与した構造体のスタックを作成する。そしてこのスタックが空になるまで図形を分割していくが、一つのピースを入れたときに残りが複数になった場合、この複数個の図形に対して再びすべての分割可能性を試し、スタックに保存する。

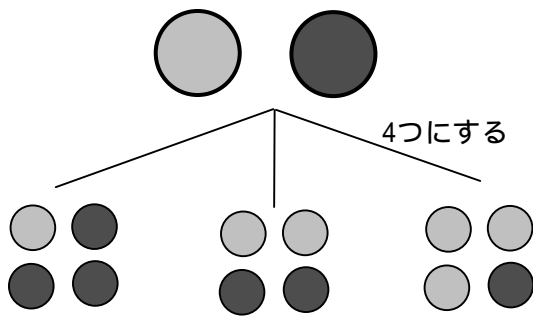


図3.6 複数ピース分割の可能性

3.3.2 単一図形の分割

一つの図形を n 分割するために、図形に確定ピースAを入れ、残りを未分図形Bとする。確定ピースAは最初に列挙したピースに含まれるものである。ここで、未分図形Bが一つの塊であれば、同様にして未分図形Bを $n-1$ 分割する。未分図形Bが複数に分かれてしまった場合は、3.3.1節で述べた複数図形の分割をすることになる。

ここで重複なくすべての分割をするために、ピースAは元図形の特定箇所(例えば最上段最左など)を必ず埋めるようにする。こうすることで、ピースの分割順序による重複を防ぐことができる。図3.7に箇所を特定しない分割による重複の例を示す。先に入れるピースをピースA、残りのピースをピースBとすると、ピースAをすべての形状、すべての位置で置いた場合、残り部分と同じ形状のピースを作成してしまい、同じ形状の分割が重複する。そこでピースAは図形の左上を含まなければいけないことにすれば、重複はなくなる。

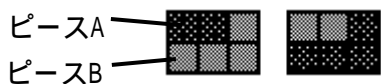


図3.7 分割の重複

また、ピースがすでにメインピースとして使われていた場合、メインピース以外の分割によって使ってしまうとメインピース同士の重複が起きる。このため最初に列挙したピースにはメインピースとして使用済みかどうかのフラグがつけられ、メインピースとして使用済みであるピースは分割ピースとして扱わない。

3.4 分割ピースの箱詰め探索

図形1を分割数 N のピースに分割し、そのすべてが図形2に個別に入ることを確認したら、図形2に N 個のピースをすべて入れる箱詰め探索を行う。図形2に N 個のピースすべてが入れば、その入れ方が問題の解となる。ここで、最少分岐選択法[2]の簡易版である初期ソート法[2]と同じような考えを用い、最初にピースをサイズ順に並べ、サイズの大きいピースから図形2に入れる(図3.8)。サイズの大きいピースは自然と入り方が少なくなるため、探索における分岐数が少ない。最初にサイズの大きいピースを入れると、探索木における根に近いノードの分岐数が少なくなり、全体の探索ノード数が少なくなる。

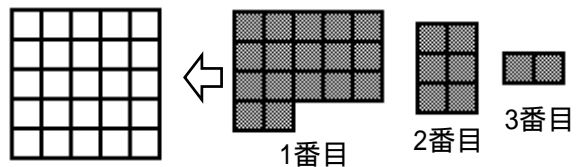


図3.8 ピースを入れる順序

3.4.1 回転による重複排除

箱詰め探索の際、裁ち合わせパズルのルール5による回転での解の重複を防ぐ必要がある。まず、図2.1の問題のように図形2が正方形である場合、90度回転させても全体として同じ形となる。このため解の重複を考えないと回転によって同じ解が4回現れる。これを防ぐために、箱詰め探索の際に最初に入れるピースは回転させない。また長方形のように180度回転させると同じ形になる場合は、最初に入れるピースを90度ずつ4回回転させると同じ解が2回現れる。このためこの場合は、ピースを回転させないものと90度回転させたものの2通りについて探索を行う。図形2が他の形状の場合は、ピース自体の回転が同じ形でない限りピースは90度ずつ4回回転させる(図3.9)。

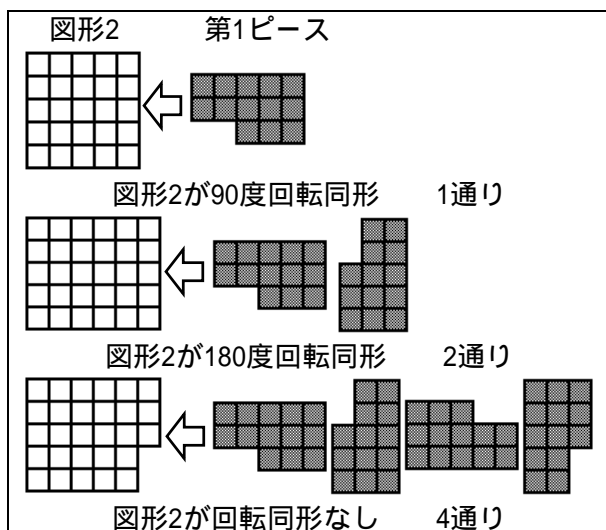


図3.9 図形2に対する第一ピースの回転数

さらに、図形2が180度回転して同形であり、かつ最初に入れるピースも180度回転して同形になる場合、入りうるすべての箇所にピースを入れると180度回転同形の解が現れてしまう。これを防ぐため、図形2および第1ピースの形状が180度回転同形である場合は、第1ピースを入れる位置は図形2の上半分とする。ただしピースが図形2の縦方向のちょうど真ん中に入る場合は、左半分に入れたものが180度回転させたときに右半分に入れた場合と同じになる。このためこのような場合は、さらに左側半分にしかなれないものとする(図3.10)。

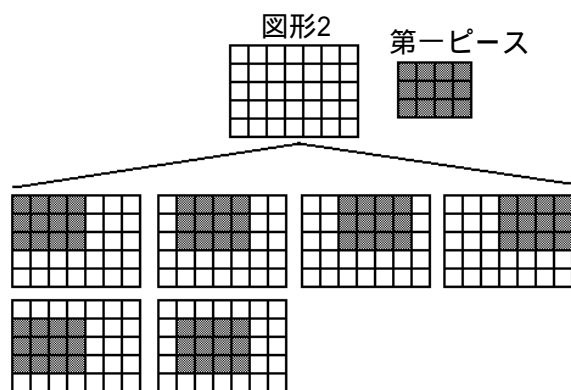


図3.10 180度回転同形図形へのピース配置

3.4.2 同一形状ピースによる重複解排除

図3.11に示すように分割ピースに同一形状のピースが複数存在する場合、それぞれを入替

えることによって同じ解が出力されてしまう。

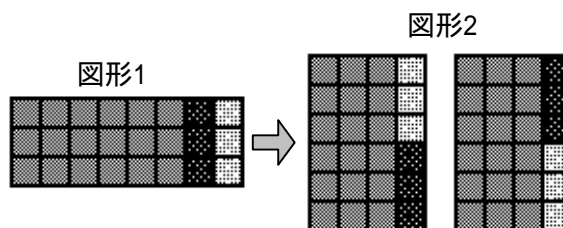


図3.11 同一形状ピースによる重複解

これを防ぐため、同一形状のピースが複数存在する場合はそれらを同時に入れる必要がある。そこで、複数の同一形状ピースにおいて、最初のピースは可能なすべての位置に置き、次のピースは最初のピースがまだ置かれていない場所に置き、3個めのピースは2個めのピースがまだ置かれていない場所に置き、という探索を行う(図3.12)。

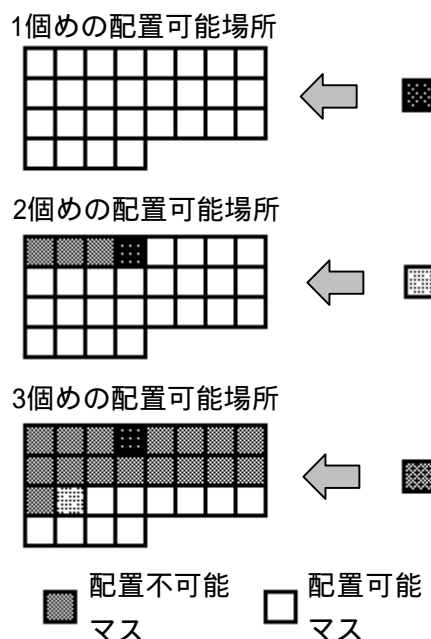


図3.12 同一形状ピースの同時配置

4. 裁ち合わせパズル解答実験

本システムにより、入力した裁ち合わせパズルを実際に解く実験を行う。今回は、図形1および2が8×8のサイズに収まるという制約の元で問題を選んだ。なお、実験環境は表4.1のとおりである。

表4.1 実験環境

コンパイラ	VC6
OS	WindowsXP
CPU	Athlon64 3200+
メモリ	1 GByte

いくつかの裁ち合わせパズル問題を解いた結果の例を表4.2、表4.3に示す。表におけるいくつかの要素を次に説明する。

・最少裁ち合わせ分割数

最少裁ち合わせ分割数とは解が一つ以上存在する最少の分割数のことである。例えば最少裁ち合わせ分割数が5の問題の場合、分割数を4にすると解が一つも出ない。これは少ない分割数から分割数を増やして解探索していき、解が存在したときの分割数を最少裁ち合わせ分割数としている。

・最大数ピース サイズ/個数

最初に列挙した使用可能ピースにおいて、最もピースの種類が多いサイズを最大数ピースサイズとし、その個数を最大数ピースの個数とする。この個数が多いほどピースの列挙に時間がかかる。図形1と図形2が似た形で、かつ隙間が少ないとこの個数が多くなる。

・分割可能性

図形1を指定の分割数で分割できる可能性の数を表したものが分割可能性である。プログラム上では、図形1を分割した後、図形2への箱詰めを行う前にカウンタを回している。

・分割ノード数

図形1を分割する際にピースを図形に位置をずらして当てはめるが、この当てはめようとした回数を分割ノード数とする。

・結合ノード数

分割されたピースを図形2に箱詰めする際、それぞれのピースについて図形2に位置をずらして当てはめる。この当てはめようとした回数を結合ノード数とする。

5. 裁ち合わせパズル解答システムの考察

裁ち合わせパズル解答システムでいくつかの問題を解いた結果、重複解は出なかった。ただ、今回扱った裁ち合わせパズルのルールにおいては、図形1および図形2が反転して同じ形になる場合を想定していない。ここでのルールは図形2を反転して同じ解ならば別解とするものであった。このため図形1、図形2共に反転させて同じ形になる問題では、図5.1のように両方を反転させた解が現れており、本来はこれも重複解とすべきである。これについては、図形1の分割の際に最初に入れるメインピースを反転させなければ排除できると考えられる。

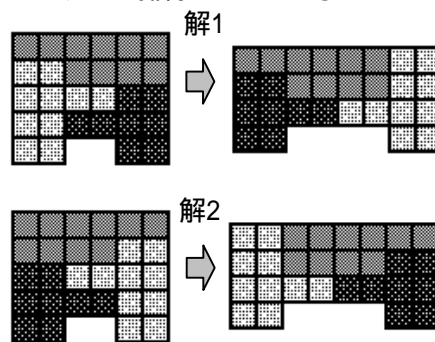


図5.1 両図形の反転による重複解例

6. おわりに

本稿では、任意分割数の裁ち合わせパズルの全解を解が重複することなく求めるシステムを提案し、開発した。

システムの実行の結果、ルールとして想定していた重複解については排除した全解を求めることができた。

参考文献

[1] GPCC2005問題 <http://hp.vector.co.jp/authors/VA003988/gpcc/gpcc05.htm>
 [2] Chikara Igarashi, Yasuhiro Tajima, Nobuo Inui, Yoshiyuki Kotani, A Logic Puzzle Solver by Selecting Smallest Branching Factor, IPSJ SIG Technical Reports,2005-GI-14,2005
 [3] Kohfuh's Lab, http://www.h6.dion.ne.jp/~k_lab/index.html

表4.2 裁ち合わせパズル解答結果例(1)

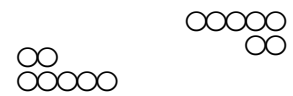




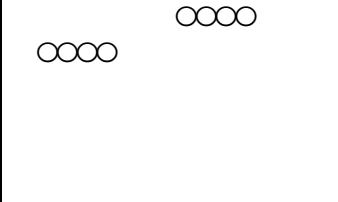

図形1	図形2	解答例	最少裁 ち合わ せ分割 数	全解 数	最大数 ピース サイズ / 個数	最大 ピース サイズ	分割可能 性	分割ノード数	結合ノード 数	ピース 作成時 間(s)	解探索 時間(s)	合計時 間(s)
			3	92	11 / 1387	19	35749	43106642	3008166	10.4	2.4	12.8
			3	92	11 / 1387	19	69450	133790519	5378648	10.4	6.0	16.4
			5	97	17 / 21536	28	12186478	430958895520	2710921286	1656.7	12428.4	14085.1
			4	1	17 / 245	26	90484	2820831934	7165611	0.2	124.5	124.6
			6	116	5,6 / 6	9	3186	630947	2192386	0.0	0.5	0.5

表4.3 裁ち合わせパズル解答結果例(2)

図形1	図形2	解答例	最少裁 ち合わ せ分割 数	全解 数	最大数 ピース サイズ / 個数	最大 ピース サイズ	分割可 能性	分割ノード 数	結合ノード 数	ピース 作成時 間(s)	解探索 時間(s)	合計時 間(s)
			5	32	8,9,10 / 6	16	34356	108350423	2280358	0.0	12.6	12.6
			5	1	10,11 / 63	19	9509635	14465702431	1415610936	0.1	811.9	812.0
			7	410	4 / 3	6	1178	23123	1222508	0.0	1.3	1.3
	[3]		6	16	5 / 5	7	823	35953	311204	0.0	0.1	0.1
			9	1455	8 / 32	13	281098	30905877	211388498	0.0	39.0	39.1