

# パターンデータベースを利用した 箱入り娘型スライディングパズルの最適解の探索

加藤 貴之<sup>†</sup> 山本 修身<sup>‡</sup>

名城大学 大学院 理工学研究科 情報工学専攻<sup>†</sup> 名城大学 理工学部 情報工学科<sup>‡</sup>

## 1 はじめに

箱入り娘パズル [1] は、 $4 \times 5$  の箱の中に  $1 \times 1$ ,  $1 \times 2$ ,  $2 \times 1$ ,  $2 \times 2$  の 4 種類の形状の異なるコマを用いたスライドパズルである (図 1 左). 特に  $2 \times 2$  のコマは「娘」と呼ばれ、パズルに一つだけ用いており、この娘のコマをそれぞれのコマのスライドによって、特定の位置へと移動させることを目的としている. 本稿では、この箱入り娘パズルの拡張版を定義し、その効率的な最適解の探索方法を考える.

拡張版の箱入り娘型パズルは、(1) オリジナルの箱入り娘パズルと同様に 4 種類のコマを用いる. (2) 盤面のサイズは  $n \times (n + 1)$  である. (3) コマの置かれていない空所が 2 箇所存在すると定義する.

$4 \times 5$  の箱入り娘パズルは比較的状态数が少ないことから、単純な幅優先探索で最適解を求めることができる. しかし、この問題は一般の盤面サイズに関して PSPACE-困難であることが知られている [2]. したがって、パズルのサイズが大きくなると飛躍的に解くことが難しくなる.

本稿では、箱入り娘型のパズルの最適解を効率的に解くための評価関数として、パターンデータベース [3] (PDB) を用いる方法を提案する.

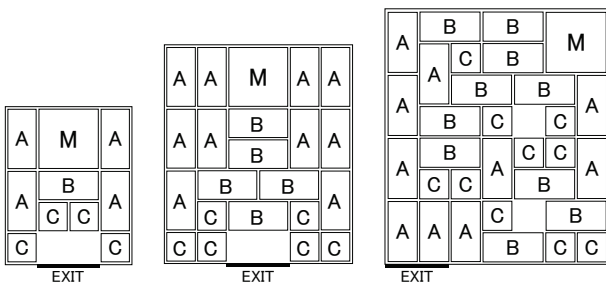


図 1: 左から順にここで用いるサイズ  $4 \times 5$ ,  $6 \times 7$ ,  $7 \times 8$  の箱入り娘型パズルの初期状態.

Finding optimal solutions of Hakoiri-Musume type sliding puzzles using a pattern database

<sup>†</sup>Takayuki Kato, Department of Information Engineering, Faculty of Science and Technology, Meijo University

<sup>‡</sup>Osami Yamamoto, Department of Information Engineering, Faculty of Science and Technology, Meijo University

表 1: 写像  $g$  に使用した 4 つの写像.

	部分問題から抽出した情報
full( $f$ )	PosM + PosB + PosS2
line( $l$ )	PosM + PosB + LineS2
cell( $c$ )	PosM + PosB + CellS2
blank( $b$ )	PosM + PosS2

PosM: 娘のコマの位置  
PosB: 空白 2 箇所の位置  
PosS2:  $1 \times 2$  と  $2 \times 1$  のコマの位置  
LineS2: 図 2 左と図 3 左参照  
CellS2: 図 2 右と図 3 右参照

## 2 箱入り娘型パズルのための PDB の構成

パターンデータベース (PDB) は、元の問題から部分問題を作成し、その部分問題の最短手数をデータベース化したものである. これを元の問題を解くための評価関数として利用する. このとき部分問題をどのように定義するかによって評価関数の性能が変わってくる. 本稿では、部分問題としていくつかの大きさ 2 のコマを  $1 \times 1$  のコマ 2 つへと分割した問題を部分問題とした. しかし、コマを分割すると元の問題では一手での移動だったものが、部分問題では二手での移動になってしまう場合がある. それは、隣接した 2 つの  $1 \times 1$  のコマが続けて同じ方向に同じ距離移動した場合である. この場合を部分問題では二手ではなく一手として数えることで、手数に整合性を取った.

また部分問題に、より多く大きさ 2 のコマを分割せずに残すことによって、PDB の性能を向上させること

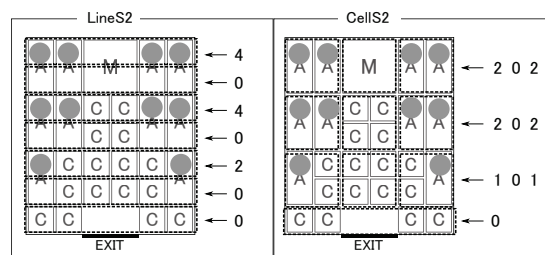


図 2:  $6 \times 7$  のパズルの LineS2 (左) と CellS2 (右) の写像を表した図. 行または  $2 \times 2$  に区切られた領域の中にある  $2 \times 1$  と  $1 \times 2$  のコマの個数により写像する.

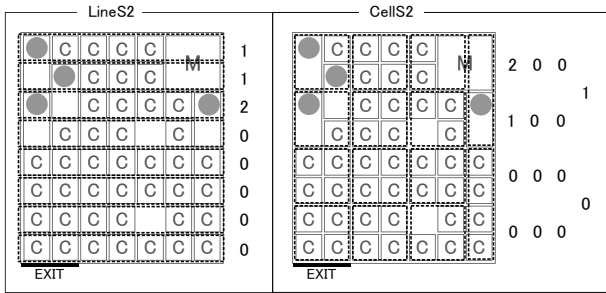


図 3: 7×8 のパズルの LineS2 (左) と CellS2 (右) の写像を表した図。行または 2×2, 1×4 に区切られた領域にある 2×1 と 1×2 のコマの個数により写像する。

ができるが、多く残すと PDB のサイズが大きくなりすぎてしまう。そのため、データベースに記録するときに部分問題からもう一段階写像 (写像  $g$ ) を行ってから PDB に記録した。こうすることで部分問題の探索空間の大きさを保ちつつ、PDB のサイズを抑えることができる。この写像  $g$  については、表 1 の 4 つのいずれかを用いた。また、PDB の表記は部分問題に使用した 1×1, 1×2, 2×1 のコマの個数を順に並べ、最後に写像  $g$  の方法を付加して表現した。

### 3 実験結果

前節で述べた PDB を用いて、A\* および IDA\* アルゴリズム [4][5] によって箱入り娘型パズルの最適解を探索した結果を表 2 と表 3 に示す。

図 1 中央の 6×7 のパズルの場合、IDA\* よりも A\* の方が探索時間が短くなった。PDB には、16-10-0- $l$  および 16-10-0- $b$  を用いた場合がもっとも良い結果となった。また、フロンティア探索 (幅優先探索) を用いて探索した場合、約  $3.4 \times 10^6$  ms の時間を要したことから、A\* により約 100 倍程度の高速化ができたことになる。

一方、図 1 右の 7×8 のパズルの場合は、PDB 42-4-0- $c$  を用いた場合を除き、6×7 のパズルとは異なり A\* よりも IDA\* の方が探索時間が短くなった。これは

表 2: 図 1 中央の 6×7 のパズルを A\* と IDA\* アルゴリズムを用いて解くために必要とした時間。

PDB	Time (ms)		PDB 情報	
	A*	IDA*	パターン数	作成時間 (ms)
36-0-0- $f$	$8.3 \times 10^4$	$5.0 \times 10^5$	$2.1 \times 10^4$	$2.7 \times 10^2$
30-0-3- $f$	$4.1 \times 10^4$	$3.0 \times 10^5$	$5.0 \times 10^7$	$2.5 \times 10^5$
30-3-0- $f$	$6.6 \times 10^4$	$4.6 \times 10^5$	$5.5 \times 10^7$	$2.6 \times 10^5$
16-10-0- $l$	$1.5 \times 10^4$	$1.2 \times 10^5$	$1.7 \times 10^7$	$3.6 \times 10^7$
16-10-0- $c$	$3.7 \times 10^4$	$2.4 \times 10^5$	$1.1 \times 10^7$	$3.6 \times 10^7$
16-10-0- $b$	$1.5 \times 10^4$	$9.3 \times 10^5$	$5.1 \times 10^7$	$3.6 \times 10^7$

サイズが大きくなったことにより空間計算量が増えたことが理由だと考えられる。ただし、PDB 42-4-0- $c$  を用いた場合は A\* のほうが IDA\* よりも高速になっており、PDB の中でもっとも高速になっている。

さらに、6×7 のランダムに作成した 100 個の問題について、PDB を用いて探索を行った。その結果、フロンティア探索 (幅優先探索) に比べて、A\* アルゴリズムでは約 1000 倍程度、IDA\* アルゴリズムでは約 100 倍程度高速化することができた。

### 4 今後の課題

本稿では、PDB の部分問題に用いる大きさ 2 のコマを増やすことで評価関数の性能を上げた。しかし、この方法ではパターン数が増加に対して、性能はそれほど良くなっていない。今後、ギャップ集合を用いて PDB の性能を引き上げるなど、別の方法を模索していく必要がある。

### 参考文献

- [1] 後藤英一, 川合慧, 佐藤充. 箱入り娘及び  $l^6$ : 解法と記述言語 (計算機によるゲームとパズルをめぐる諸問題研究会報告集). *RIMS Kokyuroku*, Vol. 98, pp. 144–149, sep 1970.
- [2] Robert A Hearn and Erik D Demaine. The nondeterministic constraint logic model of computation: Reductions and applications. In *Automata, Languages and Programming*, pp. 401–413. Springer, 2002.
- [3] Richard E Korf and Ariel Felner. Disjoint pattern database heuristics. *Artificial intelligence*, Vol. 134, No. 1, pp. 9–22, 2002.
- [4] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, Vol. 4, No. 2, pp. 100–107, 1968.
- [5] Alexander Reinefeld and T. Anthony Marsland. Enhanced iterative-deepening search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 16, No. 7, pp. 701–710, 1994.

表 3: 図 1 右の 7×8 のパズルを A\* と IDA\* アルゴリズムを用いて解くために必要とした時間。

PDB	Time (ms)		PDB 情報	
	A*	IDA*	パターン数	作成時間 (ms)
50-0-0- $f$	$7.8 \times 10^6$	$3.5 \times 10^6$	$5.5 \times 10^4$	$5.1 \times 10^2$
46-0-2- $f$	$6.5 \times 10^6$	$3.0 \times 10^6$	$4.1 \times 10^7$	$2.8 \times 10^5$
46-2-0- $f$	$1.3 \times 10^6$	$1.3 \times 10^6$	$4.2 \times 10^7$	$2.8 \times 10^5$
42-4-0- $l$	$8.3 \times 10^6$	$3.8 \times 10^6$	$1.2 \times 10^7$	$3.3 \times 10^7$
42-4-0- $c$	$5.7 \times 10^5$	$7.6 \times 10^5$	$9.1 \times 10^7$	$3.3 \times 10^7$
42-4-0- $b$	$2.3 \times 10^6$	$2.4 \times 10^6$	$4.3 \times 10^6$	$3.3 \times 10^7$