

箱入り娘パズル型スライディングパズルとその最適解の効率的探索

加藤 貴之[†] 山本 修身[†]

名城大学 理工学部 情報工学科[†]

1 はじめに

箱入り娘パズル [1] は、図 1 左に示す 4×5 の箱の中に 1×1 , 1×2 , 2×1 , 2×2 の 4 種類の形状の異なるコマを用いたスライディングパズルである。コマをスライドさせることによって、 2×2 の大きさの「娘」のコマを特定の位置に移動させることを目的としている。多くの場合の箱入り娘パズルは、娘のコマを下中央に移動させた状態を終了状態とする。このパズルと同じスライディングパズルに 15 パズルが知られている。15 パズルについては、コンピュータによる効率的な解法が知られているが、箱入り娘パズルは以下に述べるように、もともと解くための計算量がそれほど多くないため、その構造はあまり詳しく研究されていない。

図 1 左の箱入り娘の最適解は 81 手である。幅優先探索で探索を行った場合、初期状態から解を見つけるまでの探索ノード数は 55,758 であり、それまでにチェックする盤面の総数は 23,845 である。この総数はコンピュータで探索するにはそれほど大きいものではなく、Intel Xeon 3.3 GHz 上の Java 1.6.0 を用いて約 0.125 秒で解くことができる。このように短時間で解けるのは、多くの場合、ある状態から別の状態に遷移できる

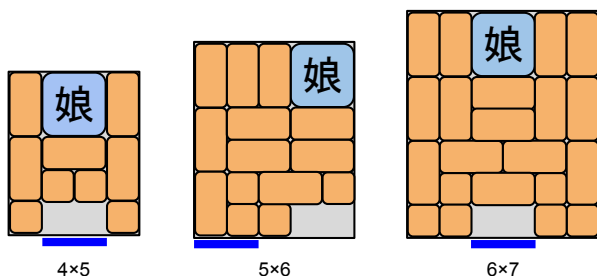


図 1: 4×5 , 5×6 , 6×7 の箱入り娘の初期状態。

Klotski type sliding puzzles and efficient search for their optimal solutions

[†]Takayuki Kato, Department of Information Engineering, Faculty of Science and Technology, Meijo University

[†]Osami Yamamoto, Department of Information Engineering, Faculty of Science and Technology, Meijo University

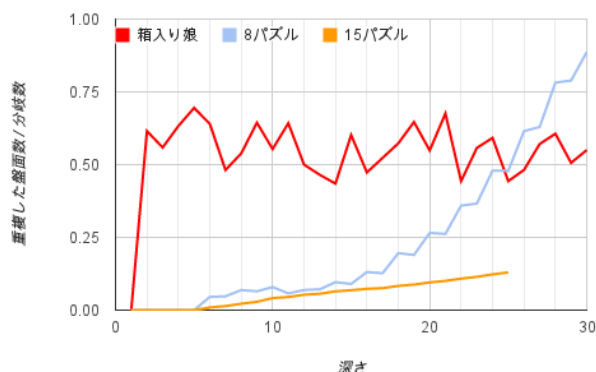


図 2: 違う手順で同じ盤面になる割合。

場合の数があまり大きくなりからである。また、初期状態から移動可能な状態数が 25,955 通りであり、15 パズルで用いられているような工夫なしでも解くことができる。

このように 4×5 の箱入り娘パズルを解くことは計算量的には困難さを伴わないが、依然として、このパズルの構造や効率的な探索アルゴリズムについては興味深いものがある。このような箱入り娘パズルの一般的な性質や解法を考えるためには、オリジナルの箱入り娘パズルを拡張して盤面を広げることにより、より計算量の大きな問題としてこれを研究対象とする。

2 効率的探索のために特徴を知る

箱入り娘パズルは、多くの点で 15 パズルに類似するが、(1) コマの形状が一樣ではないこと、(2) 終了状態が娘の位置により判定されるため、それ以外のコマはどのような配置でもよく、終了状態が複数存在するマルチゴールパズルであることが異なる。このため、15 パズルで一般的に用いられているヒューリスティックサーチの手法を適用しづらい。15 パズルでは、ゴールまでの最少手順数を与える評価関数を用いることで、効率的な IDA* アルゴリズム [2] を利用することができるが、このパズルでは上記の事情により手軽に意味のある評価関数を作ることも難しい。

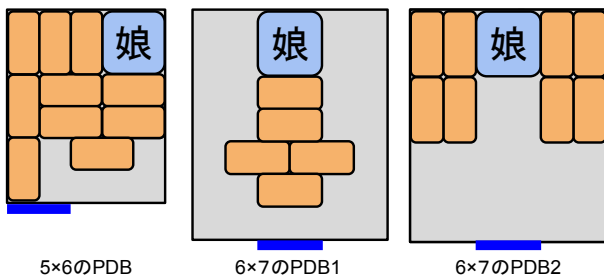


図 3: パターンデータベース作成に用いるコマの配置.

また、箱入り娘パズルは同じ形状のコマの区別をしないため、違う手順で同じ盤面になる場合が多数存在するという特徴を持っている(図2)。このため、反復深化探索を行うと同じ盤面を何度も探索をするという無駄な計算を多くしてしまい、手数が増えるにつれて探索ノード数が爆発的に増えてしまう。幅優先探索の場合、一度出てきた盤面を記憶するためこういった無駄な探索は行わない。そのため、箱入り娘パズルは反復深化探索よりも幅優先探索のほうが適している。

5×6の箱入り娘パズル(図1中央)は幅優先探索を用いて解くことができ、最少手数は72手であった。一方6×7の箱入り娘パズル(図1右)は対称性を利用してフロンティア探索を行った場合、67手で解くことができるとわかっている。

3 パターンデータベースの利用とその計算結果

パターンデータベース[3]は元の問題の部分問題の解をあらかじめ完全に計算し、その結果を元の問題を解くときの評価関数として利用する。今回、部分問題にはコマをいくつか取り除いたものを使用し、これをすべてのコマの配置について最適解をあらかじめ計算しておくことでパターンデータベースを作成する。5×6の箱入り娘パズルでは、1×1のコマを取り除いた部分問題(図3左)のパターンデータベースを作成した。6×7の箱入り娘パズルでは、1×2と1×1のコマを取り除いた部分問題(図3中央)と2×1と1×1のコマすべてと1×2のコマ2個だけを取り除いた部分問題(図3右)のパターンデータベースを作成し、これら2つの結果を足し合わせたものを評価関数として利用した。また、2つを足し合わせた結果が実際の終了状態までの手数を超えてしまうと最適性が得られないため、図3中央のパターンデータベースは娘のコマが動いた回数を数えていない。

表 1: パズルと探索方法別の探索時間(N/Tはメモリ不足でGCが発生し、停止しなかったことを表す)。

パズル	幅優先探索	対称性を利用したフロンティア探索	A*探索
5×6	1,700 ms	-	421 ms
6×7	N/T	29.4 min	N/T

表 2: パターンデータベースの作成時間とメモリ使用量。

PDB	メモリ使用量	作成時間
5×6(図3左)	84 MB	2.1 h
6×7(図3中央)	161 MB	3.9 h
6×7(図3右)	2,691 MB	38.6 h

表1にパターンデータベースを用いた探索とその他の探索の探索時間の結果を示す。5×6の箱入り娘パズルを単純な幅優先探索を行った場合、探索時間は1,700msであったのに対して、上記のパターンデータベースを用いてA*探索を行った場合、探索時間は421msとなり、探索時間を約4分の1に減らすことができた。それに対して、6×7の箱入り娘はA*探索を行った場合には10GBメモリを与えてもメモリが足りなくなり解けていない。これは元の6×7の箱入り娘からコマを多く取り除いてパターンデータベースを作成しているため、元の問題との類似性がほとんどなくなり、実際の手数とかけ離れた推定値を算出しているためである。これを改善するにはコマをあまり取り除かずにパターンデータベースを作成すればいいが、その場合パターンデータベースを作成する時間が増え、メモリの使用量も増えるという問題がある(表2)。また、コマ同士が邪魔をするために移動が制限されるという特徴を持つため、コマを取り除いてしまうと移動の制限が緩和され、その特徴が失われてしまうことも原因である。そのため、もっと効率的な評価関数を作るためには、上記の問題点を克服する方法を開発する必要がある。

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)」数理解析研究所講究録, 98巻: pp. 144-149 (1970)
- [2] Korf, R.E.: Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence*, Vol. 27, No. 1, pp. 97-109 (1985)
- [3] Felner, A., Korf, R.E., and Hannan, S.: Additive pattern database heuristics. *Journal of Artificial Intelligence Research*, Vol. 22, No. 1, pp. 279-318 (2004)