On the Hardness of Framed Edge-Matching Puzzles

Dasyel Willems
10172548

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dhr. drs. Daan van den Berg

Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 24th, 2016

Abstract

This research focusses on the hardness of two-set Framed Generic Edge-Matching Puzzles by setting the amount of colours in both sets against the median iterations needed to solve the puzzles with multiple differing CSP-solver algorithms. The research is heavily based on research done by Ansótegui et al. and tries to either verify or falsify their results. The final results hint towards a connectivity based property of Generic Edge-Matching Puzzles as a main influence on the hardness of those puzzles, while the dissimilarities between the final results and those by Ansótegui et al raise questions about their methodology and conclusions.

1 Introduction

The interest for Generic edge-matching puzzles (GEMP) research spiked due to the introduction of the Eternity II puzzle and its accompanying reward of two-million dollars for finding a solution. The solution to the puzzle has not yet been found, at the time of writing, which means that GEMP are computationally very hard puzzles.
GEMP, similar to jigsaw puzzles and polyomino packing puzzles, are NP-Complete problems as proven by Demaine et al.[4], meaning that GEMP belong to both NP and NP-hard complexity classes, making it possible to check a solution for validity in polynomial time, while finding a solution is mostly not achievable in polynomial time unless P=NP. In addition, NP-Complete problems are, by definition, reducible to other NP-Complete problems.
This reducibility property has made GEMP a notable subject for research.

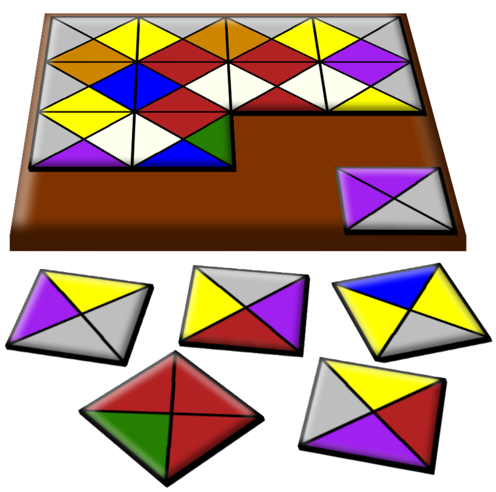


Figure 1: An example of a partially solved 4x4 Two-Set GEMP-F with 3 outer-colours and 4 inner-colours

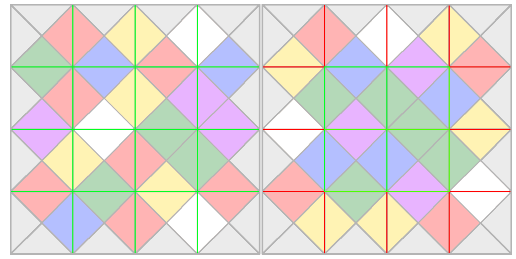


Figure 2: Examples of a One-Set GEMP-F (left) and a Two-Set GEMP-F (right) accentuating the division between the color sets.

1.1 One-set and Two-set GEMP-F

GEMP are puzzles of size $N \times M$, which consist of a set of $N \times M$ square tiles. All four sides of a tile have a colour. Each of the tiles is then placed on an $N \times M$ grid, such that no two neighbouring tiles have different colours on their matching sides. Framed Generic Edge-Matching Puzzles (GEMP-F) have an extra constraint of forcing one colour for the outermost sides of the $N \times M$ grid. The frame colour may not be assigned to any of the other edges within a GEMP-F.
In addition to these classifications and constraints for GEMP an extra distinction is made between one-set and two-set GEMP problems. One-set GEMP, as the name implies, use one set of colours for all edges. While two-set GEMP use two non-overlapping sets of colours where one set is applied to the touching sides of each set of neighbouring tiles on the edge of the grid, and the other set is used for the inner edges (see figure 2).

1.2 CSP-solvers vs. SAT-solvers

GEMP are Constraint-Satisfaction Problems (CSP), meaning that a solution for a CSP has been found when all imposed constraints are satisfied within the current state. A way to solve any CSP is by using a CSP-solver, which are algorithms specifically designed to simulate the problem or puzzle. The states of the problem are then changed by the solver until all constraints are satisfied.

Another approach to solving CSP is by using SAT-solvers, which solve for a satisfaction of a logical formula as shown by Niklas Eén et al.[6][5]. These formulas are mostly provided to the solver in the Conjunctive Normal-Form (CNF) or the Disjunctive Normal-Form (DNF). The crux with using SAT-solvers is the conversion from a CSP to either a CNF- or DNF formula. For GEMP-F there is currently no known best conversion and different methods result in differing SAT-solver performances.

1.3 Approximate Solutions

Edge-matching puzzles can either be solved for an exact solution or for a partial solution. The Eternity II puzzle, which is a still unsolved 16x16 two-set GEMP-F, has been a resourceful subject for researching approximation techniques for solving GEMP-F. Wauters et al.[7] have given an extensive report on a set of these approximation heuristics, which focus on optimizing a partial solution for the Eternity II puzzle and which have been awarded for being the most efficient and best achieving approximation heuristics.
In addition to this report, Antoniadis et al.[2] have shown that finding an approximate solution is APX-Complete, which means that finding an approximate solution is both in the APX and the APX-Hard complexity classes. This, in turn, means that finding an approximate solution with an error of at most a given constant c, cannot be done in polynomial time for every value for c higher than zero.

1.4 Exact Solutions

As opposed to looking at approximations, Ansótegui et al.[1] have researched the hardness of finding exact solutions for framed edge-matching puzzles by looking at both one-set and two-set square GEMP-F problems (sized 7×7 and 8×8). The research was conducted by varying the amount of inner- and outer-colours of generated GEMP-F and solving them with both SAT- and CSP-solvers. Results of the experiments imply that the hardest median puzzles for an $N \times N$ one-set GEMP-F lies around $N - 1$ colours, while the hardest median puzzles for a 7×7 two-set puzzle lies around 7 inner-colours and 3 outer-colours.

The focus of this research paper is to verify or falsify the findings of Ansótegui et al. by running multiple differing CSP-solver algorithms on sets of square GEMP-F ranging from 5×5 to and including 7×7. The results will then be compared between different algorithms to find general conclusions.

2 Method

The aim of this research is to reproduce the results on two-set GEMP-F found by Ansótegui et al. This research is split into two major steps. To start off, a set of two-set GEMP-F is generated for each of the sizes: 5x5, 6x6 and 7x7. Afterwards these sets of puzzles are solved with multiple CSP-solvers.

2.1 GEMP-F Generator

A generator for GEMP-F puzzles has been implemented according to the algorithm given by Ansótegui et al.[1]. Each set of two-set GEMP-F generated exists of 20 puzzles per possible inner-colour- and outer-colour amount combination. The amount of edges in an NxN puzzle take on the following values:

$$E_{total} = 2N^2 - 2N \quad (1)$$

$$E_{outer} = 4(N - 1) \quad (2)$$

$$E_{inner} = E_{total} - E_{outer} \quad (3)$$

Where E_{total} is the number of total edges in the puzzle excluding the edges assigned to the frame, E_{outer} is the number of outer edges and E_{inner} is the number of inner edges of the puzzle. The outer-colours can range from 1 up to E_{outer}, with the upper bound resulting in a different colour for each outer vertex. The same applies to the inner-colours with the E_{inner} as the upper bound.

No uniqueness and rotation-invariance for pieces within puzzles was taken into account. Also no inter-puzzle uniqueness was checked for the eventually generated sets. Both measures were omitted due to Ansótegui et al. not mentioning using these measures.

2.2 GEMP-F CSP-solvers

Some different algorithms and heuristics were implemented to solve the same sets of GEMP-F for eventual comparison purposes and testing the robustness of the results by Ansótegui et al.

Simple Depth-First

A basic depth-first algorithm was implemented, which solves the puzzle in a single, static traversal-order, namely left-to-right and top-to-bottom. The solver checks for available pieces which fit in the next index in the puzzle and places each of them in no particular order every time the algorithm backtracks to that index. When no pieces are available for a given index, the algorithm backtracks to the previous index. Once a solution is found, the algorithm stops and returns the solution. Even though the order of placing possible pieces for each index is not ordered, the algorithm does filter for duplicate pieces to prevent extraneous results due to the repeat of previously checked paths in the search-space.

Connectivity Based Heuristics

Cheeseman et al.[3] have researched the hardness of finding a Hamilton-Circuit in graphs using depth-first search extended with the heuristic to start at the node with the highest connectivity and to keep advancing to the next available node with the highest connectivity. This connectivity of nodes is defined as the possible nodes the path can traverse to from the current node. As GEMP-F and finding Hamilton-Circuits are both NP-Complete problems, the heuristic to choose the highest connectivity first has a chance of proving beneficial for finding solutions to GEMP-F. An algorithm was implemented, which still follows the same direction as the simple depth-first algorithm, enhanced with the heuristic to sort the available pieces for the next index in the puzzle by their connectivity. In this heuristic the connectivity of a piece is measured by the amount of available neighbour piece edges which can be placed against the right side of the piece. Notice that this definition for piece connectivity is rather naive, as it does not take into account the possible connections on the top, bottom and left side of a piece. This naive definition was, however, chosen to fit best with the Simple Depth-First traversal order.

The puzzles are also solved with the inverse of this heuristic, which prefers pieces with the lowest connectivity instead, as this method will intuitively eliminate possible pieces for indices faster.

Smallest Number of Possible Pieces First (SNOPF)

An algorithm close to the PLA heuristic used by Ansótegui et al. was implemented. The implemented algorithm keeps track of the possible pieces for each index in the puzzle. Each time a piece is added to the puzzle, the neighbouring indices are updated, eliminating all pieces no longer possible for that specific index. Pieces are eliminated, based on the side-colours of neighbouring matching pieces. The last placed piece is effectively eliminated from all indices in the puzzle as it is no longer available. The order the algorithm goes through the indices is based on the number of available pieces for each index, which is measured each time after the elimination step. The index with the smallest number of possible pieces is then chosen as the next index to place a piece on. The possible pieces for each index in the puzzle is not ordered by any measures.

3 Results

The use of the CSP-solver algorithms on the GEMP-F sets resulted in the graphs found in Appendix A. Due to minimal differences between both Cheeseman solvers and the simple depth-first solver, only the simple depth-first and the SNOPF algorithm results are shown in this section (see figures 3 and 4). Each graph shows the median iterations needed per combination of inner- and outer-colours. The median is based on the needed iterations to solve 20 puzzles for each data-point.

Table 1 shows where the exact peaks are located in the plots together with the median iterations measured at that peak. Note that the iteration limit for the 7x7 GEMP-F sets are lower than the limits for the 5x5 and 6x6 sets, due to time constraints after the last limit increase. The time to solve the full set of 7x7 two-set GEMP-F with a limit of two-million iterations using the Simple Depth-First algorithm increased the total solve time to 136,456.79 seconds as opposed to 53,208.03 seconds with a limit of half a million, while still no single peak location could be pinpointed. The results for the 7x7 puzzles are, however, included in appendix A as the graphs still show meaningful data.

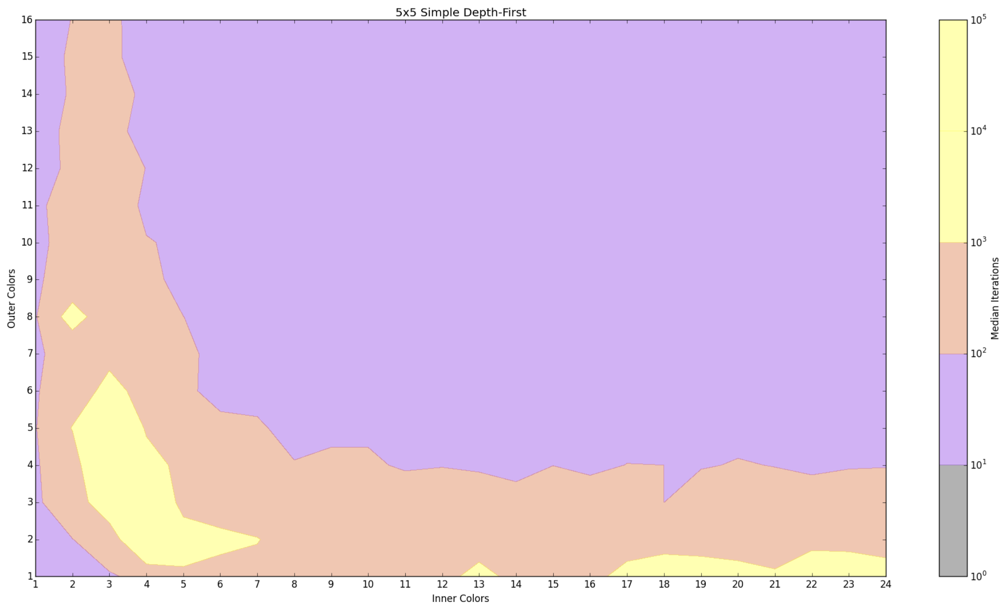
Table 1 shows that the Lowest Connectivity First algorithm performs the best on both the 5x5 and 6x6 sets as its highest peak in the median values is significantly lower than the peaks of the other algorithms and the total time to solve all puzzles in the set is lower too.

There is no apparent connection between the peak-locations in table 1, as the locations differ greatly between the different algorithms. The median iteration values around the peaks do, however, have overlapping areas.

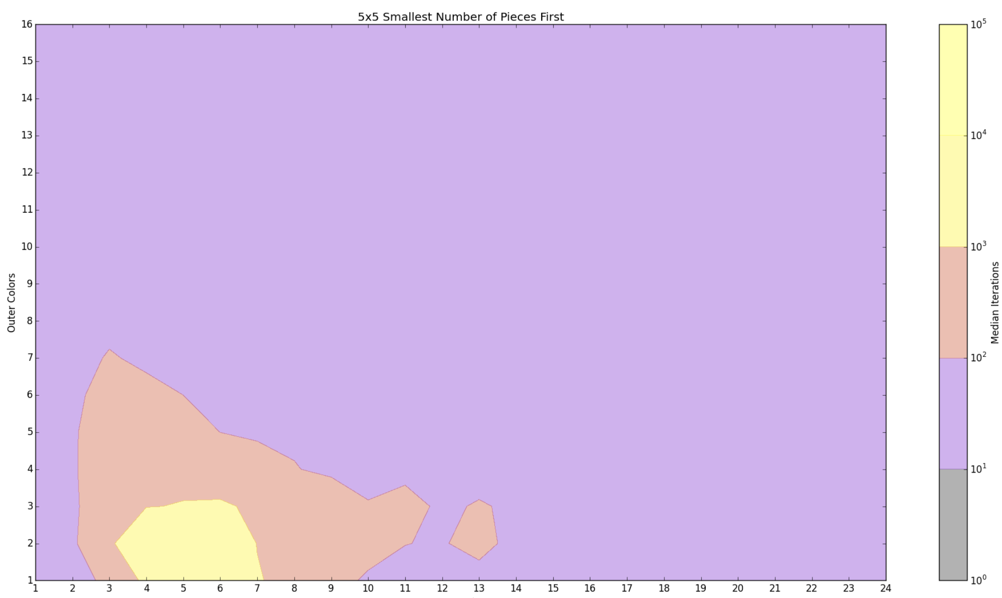
4 Conclusion and Discussion

The main focus of this research was to verify or falsify the results on two-set GEMP-F, found by Ansótegui et al. in their research on GEMP-F. This was done by submitting randomly generated solvable two-set GEMP-F to a set of different CSP solvers. The differences and similarities between the performances of the CSP-solvers should imply intrinsic properties of the two-set GEMP-F puzzles.

The results by Ansótegui et al. showed that 7x7 two-set GEMP-F have a single symmetrical and distinguished peak of measured median

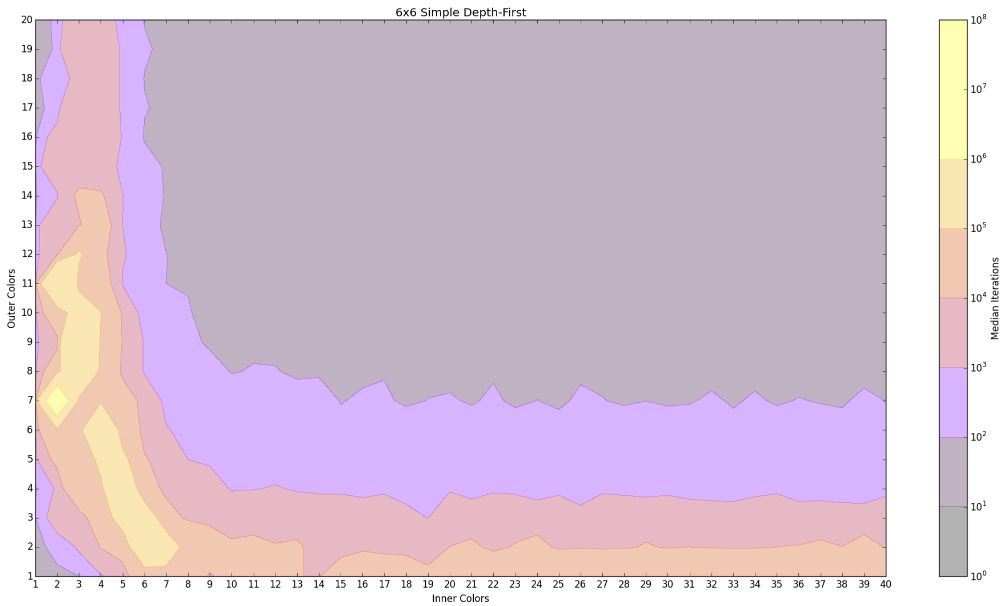


(a) Simple Depth-First

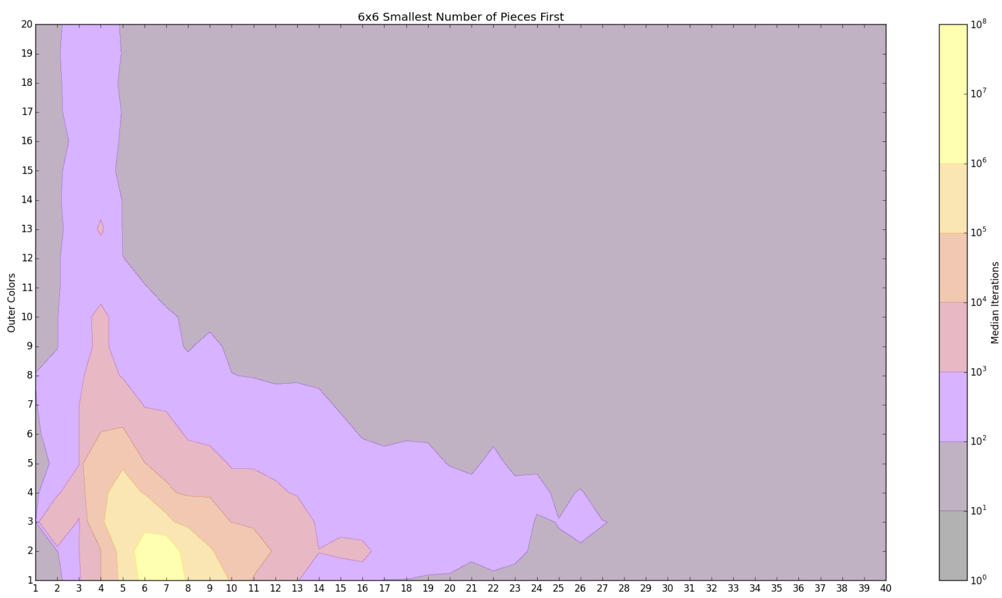


(b) Smallest Number of Pieces First

Figure 3: Results of solving the 5x5 two-set GEMP-F set with two distinguished algorithms.



(a) Simple Depth-First



(b) Smallest Number of Pieces First

Figure 4: Results of solving the 6x6 two-set GEMP-F set with two distinguished algorithms.

	Max Median Value	Inner-Colours	Outer-Colours	Iterations Limit	Total Solve-Time (seconds)
Simple Depth-First 5x5	2,829	5	3	2,000,000	68.90
SNOPF 5x5	6,185	5	1	2,000,000	75.30
Highest Connectivity First 5x5	4,619	4	3	2,000,000	86.26
Lowest Connectivity First 5x5	2,579	23	1	2,000,000	67.23
Simple Depth-First 6x6	2,000,000	7	2	2,000,000	11,876.86
SNOPF 6x6	2,000,000	2	7	2,000,000	18,454.05
Highest Connectivity First 6x6	913,158	3	9	2,000,000	15,957.48
Lowest Connectivity First 6x6	228,980	6	2	2,000,000	10,700.64
Simple Depth-First 7x7	500,000	multiple	multiple	500,000	53,208.03
SNOPF 7x7	500,000	multiple	multiple	500,000	63,718.49
Highest Connectivity First 7x7	500,000	multiple	multiple	500,000	63,715.66
Lowest Connectivity First 7x7	500,000	multiple	multiple	500,000	56,642.75

Table 1: Median iterations peak locations and heights

solve-time, whereas the results of this research show that the peak of 5x5 and 6x6 two-set GEMP-F is not symmetrically shaped. The 7x7 puzzle-set results also indicate an irregular peak neighbourhood, even though the limit on the amount of iterations was too small to pinpoint the exact peak. A second notable difference between the results of this research in comparison to the results by Ansótegui et al., is the hardness of puzzles with outer-colours=1. Where they have found that those puzzles are fairly easy in the neighbourhood of the median solve-time peak, while this research has found a consistent hardness on those puzzles around the peaks.

These differences in results might be caused by the different approaches the CSP solvers take to solve GEMP-F problems. The similarities between the Simple Depth-First algorithm graphs and the graphs of the connectivity based algorithms support this claim, as these algorithms are similar in traversal order, while the SNOPF algorithm, which shows a more symmetric peak region, is closer to the PLA algorithm Ansótegui et al. used. Another reason for the differing results may be the fact that Ansótegui et al. measured their solve-time in seconds while this research used the more stable measure of iterations which does not depend on the efficiency of implemented algorithms or cpu fluctuations.

While the peaks for the 5x5 and 6x6 puzzle median solve-times differ between the different solving algorithms, the immediate surrounding area of the peaks, with values close to the peak value, seem to overlap on the area denoted by the peak found by Ansótegui et al. This may indicate that two-set GEMP-F do have a single hardest combination of inner- and outer-colours, while the generated puzzle sets in this research where either too small or not sampled well enough to undermine outliers.

The performance results of the CSP solvers used in this research (Table 1) show that the Lowest Connectivity First algorithm is significantly faster iterations-wise at solving two-set GEMP-F. The median iterations graph of that algorithm (figure 12) compared to the SNOPF algorithm graph (figure 10) shows that the SNOPF algorithm performs better at inner- and outer-colour combinations further away

from the epicentre/peak of the graphs. As the SNOPF algorithm differs mostly in traversal order as compared to the other algorithms, a hybrid solver based on both connectivity and adaptable traversal-order might prove to be a more efficient solver than the ones presented in this research.
The efficiency of the Lowest Connectivity First algorithm might imply the influence of average inter-piece connectivity on the hardness of GEMP-F. The inner- and outer-colour values which define the hardest puzzles may actually have higher chances on generating the harder number of average connectivity. As correlation does not necessarily mean causation, changing the focus to average connectivity instead of inner- and outer-colours might prove interesting for future research.
Another point of critique, which may be useful in future research, is the fact that this research, just like the research by Ansótegui et al., did not take into account the occurrence of rotation invariant or duplicate pieces, as well as puzzle duplicates, in the puzzle-set generation process. These factors do affect the search-space of puzzle solutions significantly.

To summarize; Grasping the hardness of generic edge-matching puzzles proves to be a hard problem itself.

References

[1] Carlos Ansótegui, Ramón Béjar, César Fernàndez, and Carles Mateu. *Principles and Practice of Constraint Programming: 14th International Conference, CP 2008, Sydney, Australia, September 14-18, 2008. Proceedings*, chapter Edge Matching Puzzles as Hard SAT/CSP Benchmarks, pages 560–565. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[2] Antonios Antoniadis and Andrzej Lingas. *SOFSEM 2010: Theory and Practice of Computer Science: 36th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 23-29, 2010. Proceedings*, chapter Approximability of Edge Matching Puzzles, pages 153–164. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[3] Peter Cheeseman, Bob Kanefsky, and William M Taylor. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–340, 1991.

[4] Erik D. Demaine and Martin L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23(1):195–208.

[5] Niklas Eén and Armin Biere. *Theory and Applications of Satisfiability Testing: 8th International Conference, SAT 2005, St Andrews, UK, June 19-23, 2005. Proceedings*, chapter Effective Preprocessing in SAT Through Variable and Clause Elimination, pages 61–75. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[6] Niklas Eén and Niklas Sörensson. *Theory and Applications of Satisfiability Testing: 6th International Conference, SAT 2003, Santa Margherita Ligure, Italy, May 5-8, 2003, Selected Revised Papers*, chapter An Extensible SAT-solver, pages 502–518. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[7] Tony Wauters, Wim Vancroonenburg, and Greet Vanden Berghe. A guide-and-observe hyper-heuristic approach to the eternity ii puzzle. *Journal of Mathematical Modelling and Algorithms*, 11(3):217–233, 2012.

Appendix A: Result Graphs

The following pages show all median solve iterations graphs for the 5x5, 6x6 and 7x7 two-set GEMP-F sets. Note that the iteration limit for solving the 5x5 and 6x6 puzzle sets lies on two million iterations, while the limit for the 7x7 puzzle set lies on half a million iterations.

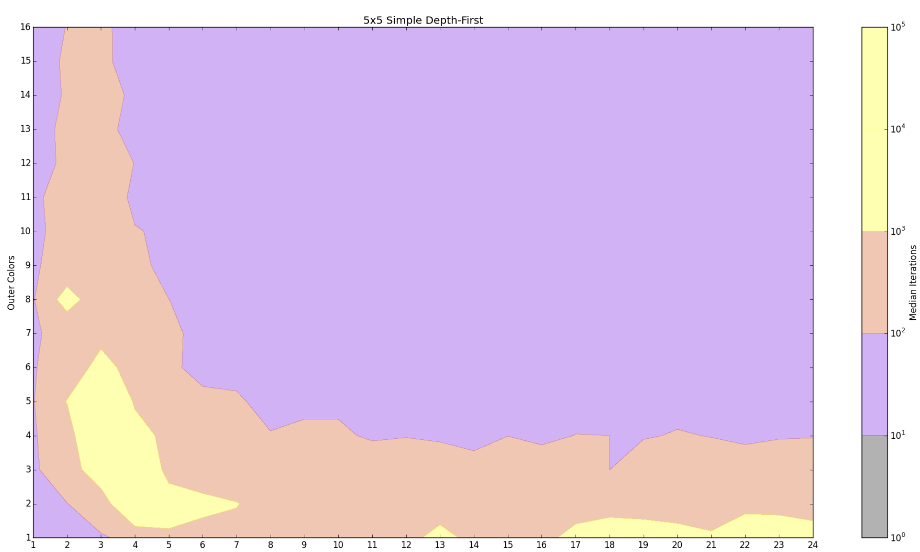


Figure 5: Median iterations for 5x5 Simple Depth-First

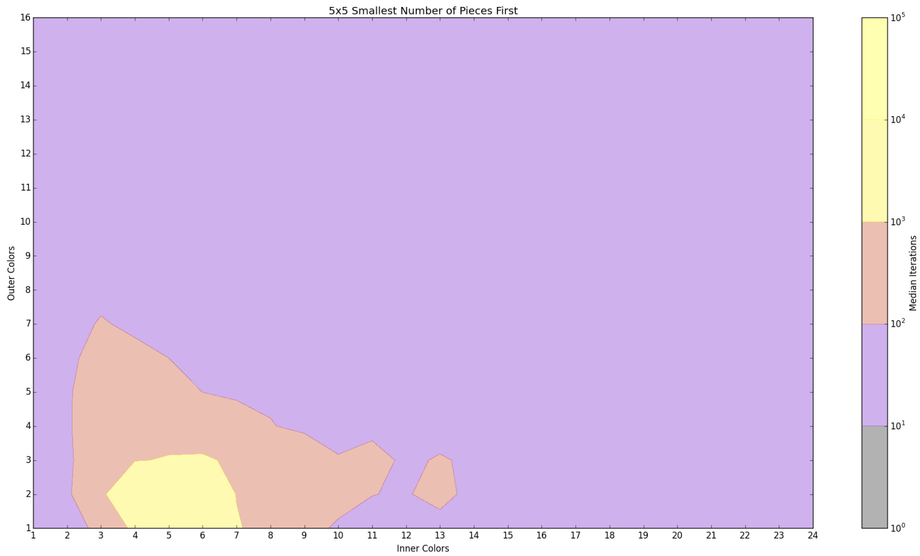


Figure 6: Median iterations for 5x5 Smallest Number of Pieces First

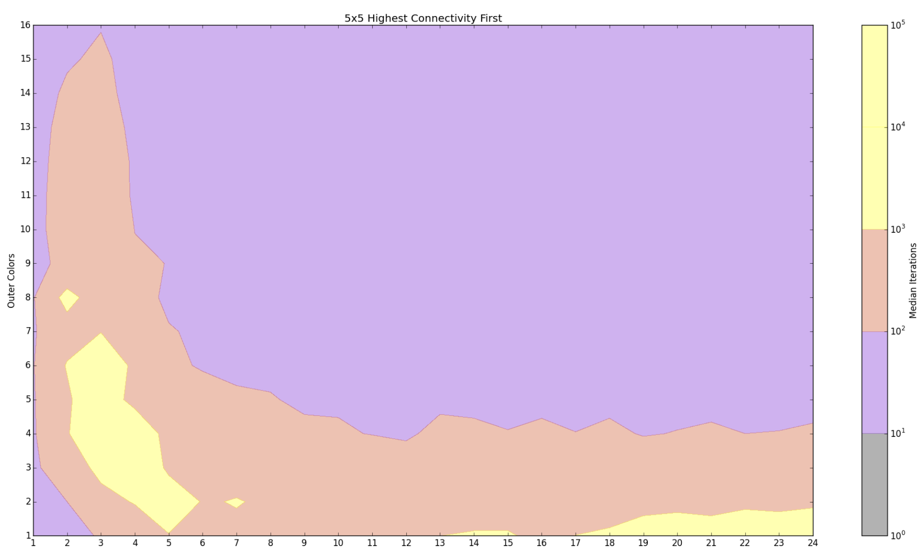


Figure 7: Median iterations for 5x5 Highest Connectivity First

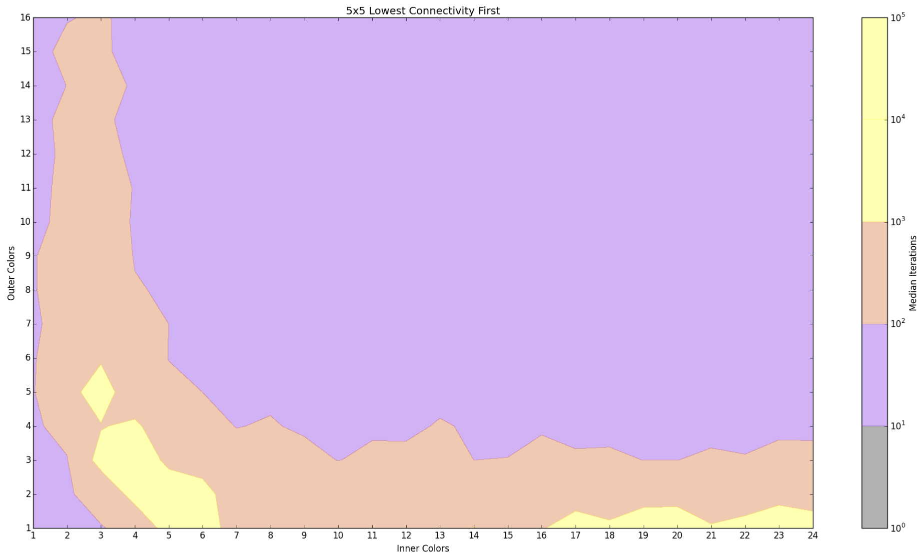


Figure 8: Median iterations for 5x5 Lowest Connectivity First

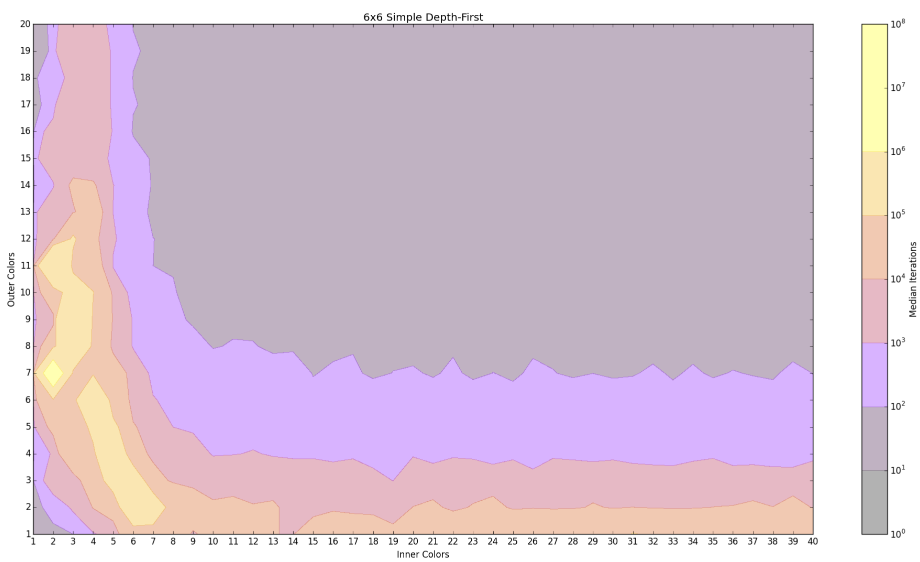


Figure 9: Median iterations for 6x6 Simple Depth-First

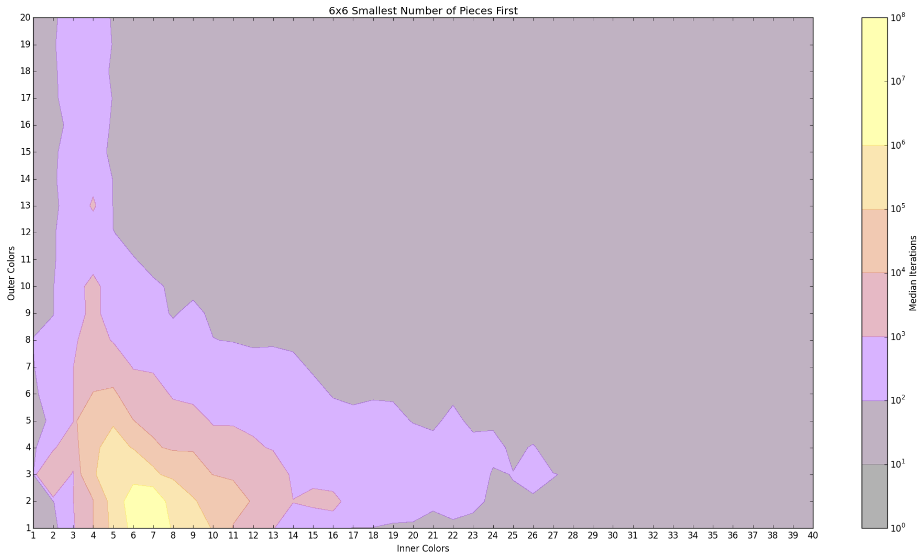


Figure 10: Median iterations for 6x6 Smallest Number of Pieces First

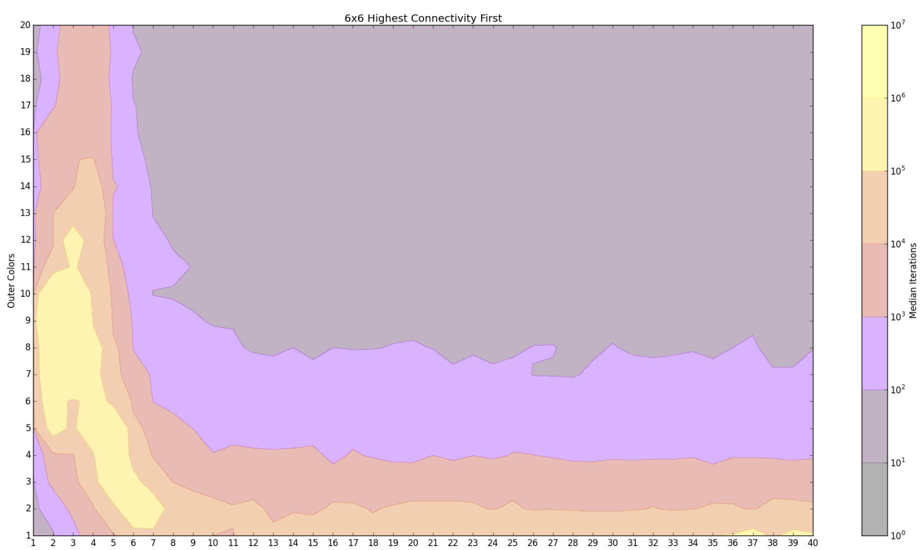


Figure 11: Median iterations for 6x6 Highest Connectivity First

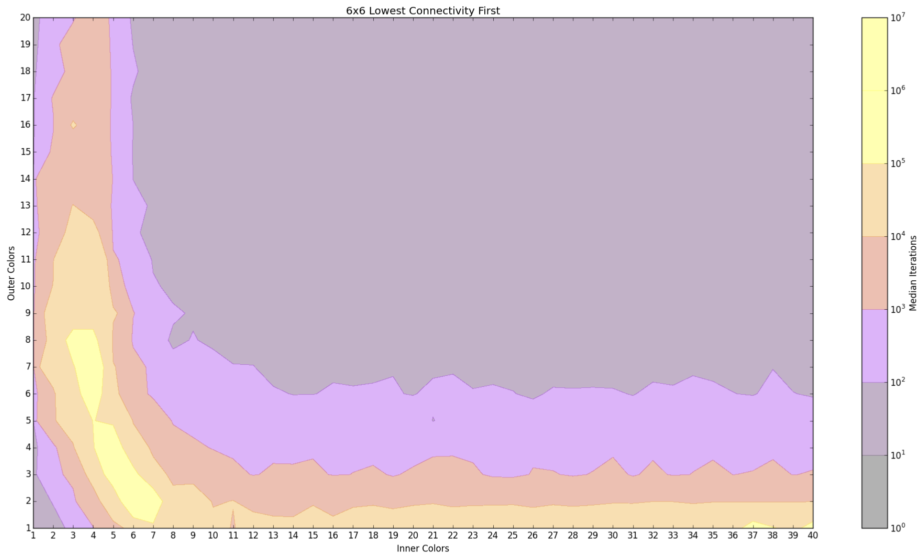


Figure 12: Median iterations for 6x6 Lowest Connectivity First

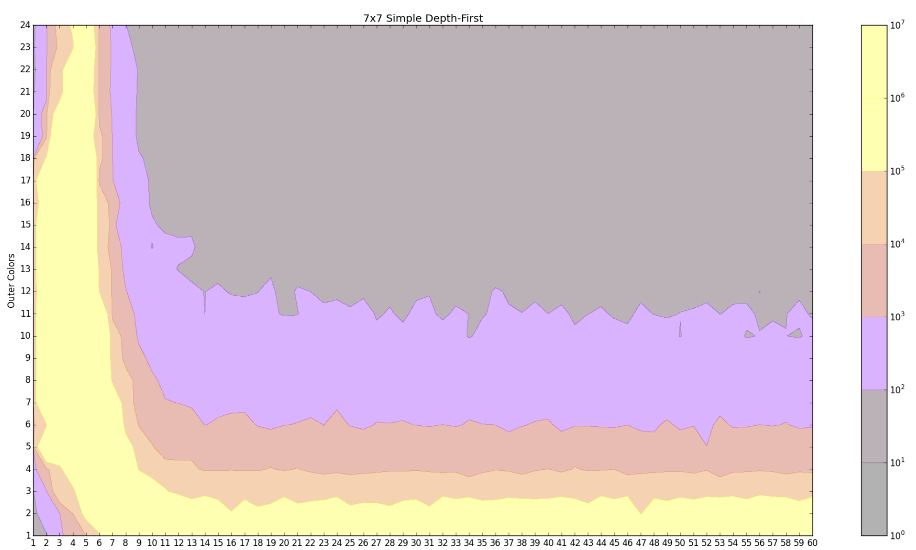


Figure 13: Median iterations for 7x7 Simple Depth-First

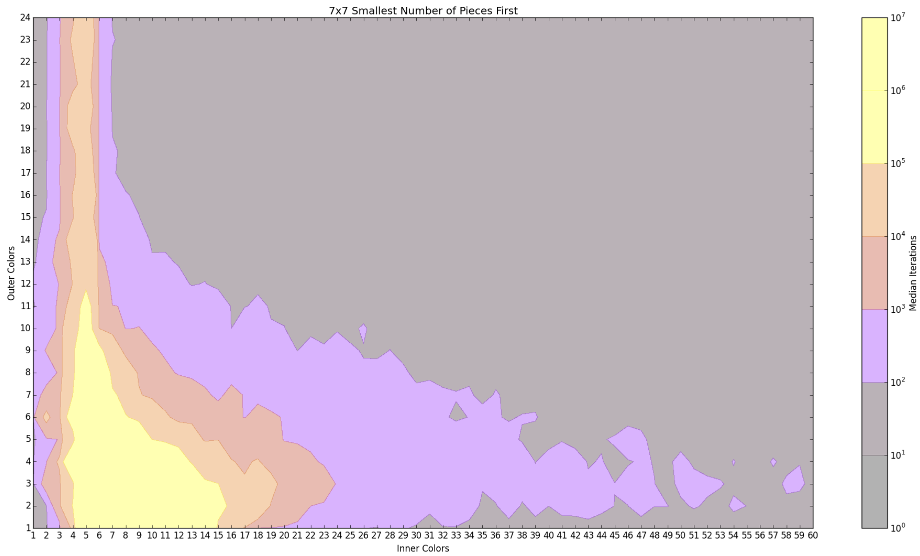


Figure 14: Median iterations for 7x7 Smallest Number of Pieces First

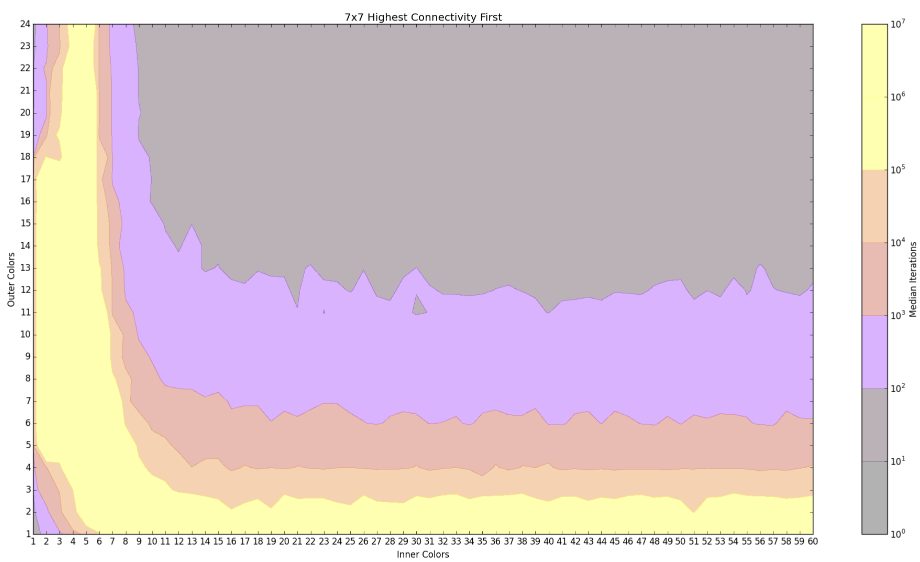


Figure 15: Median iterations for 7x7 Highest Connectivity First

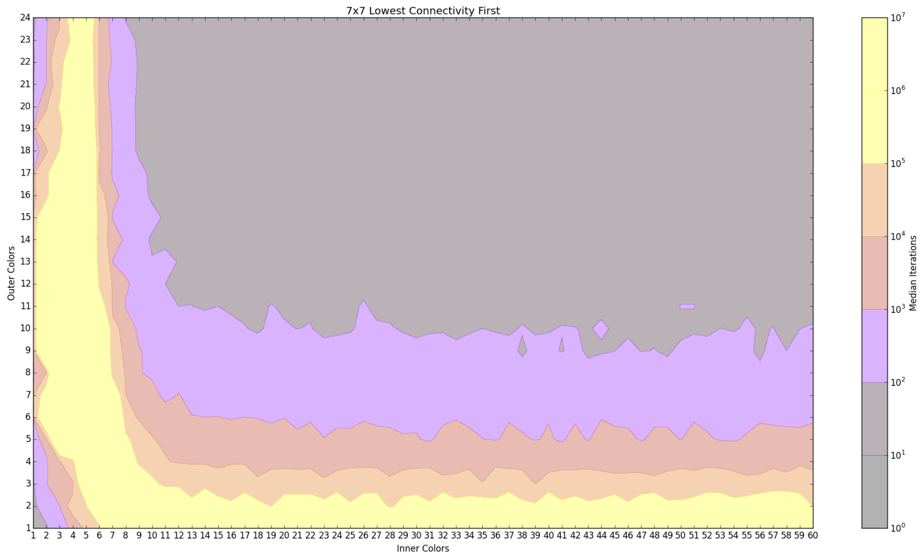


Figure 16: Median iterations for 7x7 Lowest Connectivity First